Model-based decision support for subjective preference blending

Thomas Johannes Kellermann

Dissertation presented for the degree of
Doctor of Philosophy
in the Faculty of Engineering at Stellenbosch University

Promoter: Prof JH van Vuuren

April 2019

Declaration

By submitting this dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: April, 2019

Abstract

In a *subjective preference blending problem* (SPBP), the objective is to select a most pleasing alternative from a set of alternatives, based on the relative performances of the alternatives in respect of a number of (possibly conflicting) selection criteria. What distinguishes this problem from other types of selection problems is that there is no universally accepted objective function for scoring the alternatives according to the selection criteria. Instead, every decision maker has his or her own relative preferences of the alternatives with respect to the various selection criteria — hence the prefix *subjective*.

Although various multi-criteria decision support frameworks or methodologies exist for estimating a particular decision maker's value function in respect of combined selection criteria in an SPBP instance, these processes are only effective if the decision space of the problem instance is not too large. This is because value function estimation procedures typically involve pairwise comparisons of the criteria, which has quadratic-order time complexity. Large decision spaces, in fact, often induce the phenomenon of *satisficing*, where a decision maker settles for an acceptable alternative rather than attempting to discover the *best* alternative — either because of time constraints or because of a lack of a full characterisation of the decision space.

The provision of automated decision support to a decision maker in respect of the exploitation and exploration of the decision space of an SPBP instance resides within the realm of *interactive preference learning*. The quest in preference learning is to predict the subjective preferences of a decision maker based on past observed preference information elicited from that decision maker or similarly minded decision makers. Preference learning becomes *interactive* if the decision maker is actively included in the (typically iterative) learning process, in which case such (subjective) learning is usually tailored to the specific decision maker involved.

An interactive preference learning decision support system is proposed in this dissertation for facilitating solution of SPBP instances. The working of the system is based on an iterative involvement of the DM in the form of the elicitation of pairwise comparisons of alternatives strategically selected by the system so as to achieve a trade-off between efficiently exploring the entire decision space and exploiting well-performing areas of the decision space according to the current value function estimation of the decision maker. The design objectives of the system specifically include workload reduction of the decision maker and the accommodation of low to moderate levels of decision maker inconsistency during pairwise comparisons of alternatives.

The effectiveness of the decision support system is assessed in view of its intrinsic ability to learn a decision maker's value function by applying methods from inferential statistics to the output recommendations of the system. The system is finally also validated by applying it to two SPBP instances involving real human decision makers.

Uittreksel

In 'n *subjektiewe voorkeur-vermengingsprobleem* (SVVP) is die doel om 'n mees verkose alternatief uit 'n versameling alternatiewe te kies, gebaseer op die relatiewe prestasies van die alternatiewe ten opsigte van 'n aantal (moontlik teenstrydige) seleksiekriteria. Wat hierdie probleem van ander tipes seleksieprobleme onderskei, is dat daar geen universeel-aanvaarde doelfunksie bestaan waarvolgens die alternatiewe met betrekking tot die seleksiekriteria gekwantifiseer kan word nie. In plaas daarvan het elke besluitnemer sy of haar eie relatiewe voorkeure van die alternatiewe met betrekking tot die verskillende seleksiekriteria — daarom die voorvoegsel *subjektiewe*.

Alhoewel daar verskeie multi-kriteria besluitsteunraamwerke of metodologieë vir die afskatting van 'n bepaalde besluitnemer se waardefunksie met betrekking tot gekombineerde seleksiekriteria in 'n SVVP-geval bestaan, is hierdie prosesse slegs doeltreffend as die beslissingsruimte van die probleemgeval nie te groot is nie. Die rede hiervoor is dat waardefunksie afskattingsprosedures tipies paarsgewyse vergelykings van die kriteria behels, wat kwadratiese-orde tydkompleksiteit het. Groot beslissingsruimtes veroorsaak trouens die verskynsel van *bevrediging*, waar 'n besluitnemer bloot 'n aanvaarbare alternatief selekteer eerder as om te poog om die *beste* alternatief te ontdek — óf as gevolg van tydsbeperkings óf weens die onbeskikbaarheid van 'n volle karakterisering van die beslissingsruimte.

Die verskaffing van ge-outomatiseerde besluitsteun aan 'n besluitnemer ten opsigte van die uitbuiting en verkenning van die beslissingsruimte van 'n SVVP-geval val binne die studieveld van *interaktiewe voorkeurleer*. Die doel in voorkeurleer is om die subjektiewe voorkeure van 'n besluitnemer te voorspel, gebaseer op vorige waargenome voorkeure van die besluitnemer of soortgelyke besluitnemers. Voorkeurleer word *interaktief* as die besluitnemer aktief betrokke is by die (tipies iteratiewe) leerproses. In sulke gevalle vind (subjektiewe) leer plaas wat spesifiek op die betrokke besluitnemer van toepassing is.

'n Interaktiewe voorkeurleer-besluitsteunstelsel word in hierdie proefskrif daargestel om die oplossing van SVVP-gevalle te fasiliteer. Die werking van die stelsel is gebaseer op 'n iteratiewe betrokkenheid van die DM in die vorm van die spesifikasie van paarsgewyse vergelykings van alternatiewe wat strategies deur die stelsel gekies is, ten einde 'n afruiling te bewerkstellig tussen die doel-treffende verkenning van die hele beslissingsruimte en die ontginning van hoë-kwaliteit areas van die beslissingsruimte volgens die besluitnemer se huidige waardefunksie-afskatting. Die ontwerp-doelwitte van die stelsel sluit spesifiek werksladingvermindering van die besluitnemer en die akkommodasie van lae tot matige vlakke van besluitnemer-inkonsekwentheid tydens paarsgewyse vergelyking van alternatiewe in.

Die doeltreffendheid van die besluitsteunstelsel ten opsigte van die intrinsieke vermoë daarvan om 'n besluitnemer se waardefunksie te leer, word beoordeel deur metodes uit inferensiële statistiek op die aanbevelingsafvoer van die stelsel toe te pas. Die stelsel word uiteindelik ook gevalideer deur dit op twee SVVP-gevalle toe te pas waarby menslike besluitnemers betrokke is.

Acknowledgements

The author wishes to acknowledge the following people and institutions for their various contributions towards the completion of this work:

- My family and friends for their encouragement, support and prayers, as well as always encouraging me to seek adventures in life. Without them this dissertation would not have been possible.
- Professor JH van Vuuren for his precision, guidance and unfaltering support. His incredible insight and passion for knowledge is inspiring.
- The SUnORE research group for financial support in the form of a post-graduate bursary and for their contributions towards improving my dissertation and overall learning experience over the last three years.

Table of Contents

List of Acronyms

2DCSP: Two-Dimensional Colour Selection Problem

3DCSP: Three-Dimensional Colour Selection Problem

AHP: Analytic Hierarchy Process

ANOVA: Analysis of Variance

CI: Consistency Index

CR: Consistency Ratio

CSP: Colour Selection Problem

DSS: Decision Support System

DM: Decision Maker

EM: Eigenvector Method

GLPMI: Global Learning Performance Measure Indicator

HMI: Human Machine Interface

LLPMI: Local Learning Performance Measure Indicator

LLSM: Log Least Squares Method

LSD: Least Significant Difference

MCDM: Multi-Criteria Decision Making

PHC: Parallel Hill Climbing

RI: Random Index

SA: Simulated Annealing

SPBP: Subjective Preference Blending Problem

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

Contents

1.1 Background

When facing the problems of everyday life, one is often prevented from carrying out an exhaustive consideration of all possible solutions to these problems. In such cases a *decision maker* (DM) may be forced merely to settle on a solution that is "acceptable" for a given decision problem instance, without considering the degree of desirability of that particular solution relative to other solutions that would also have been "acceptable" for the problem at hand. This phenomenon is known as *satisficing* (which stands in stark contrast to the notion of *optimising*). The Oxford Dictionary [161] defines *satisficing* as "deciding on and pursuing a course of action that will satisfy the minimum requirements necessary to achieve a particular goal."

There are numerous examples in modern-day life of *subjective preference blending problems* (SPBPs) in which DMs satisfice instead of making the highest quality decisions possible. This situation may be induced by strict time constraints as a result of the solution space being overwhelmingly large. Consider, for example, the SPBP instance of choosing a most pleasing colour for use in some context, henceforth referred to as the *colour selection problem* (CSP). Although the colour spectrum is theoretically infinite, a colour is typically represented on a computer using the well-known RGB coding scheme [199]. According to this coding scheme, a colour is represented by a triple (r, g, b), where r, g and b are each integers in the set $\{0, 1, 2, \ldots, 255\}$. These parameters represent respectively the amounts of red, green and blue present in a colour, as illustrated in Figure 1.1.

Since the red, green and blue values in an RGB colour specification can be chosen independently, there are $(256)^3 = 16\,777\,216$ distinct colours in the RGB scheme, and so the DM would have to consider $\binom{16\,777\,216}{2} = 140\,737\,479\,966\,720$ pairwise comparisons[1] of colours in order to be *sure*

[1] As will be described later in this dissertation, the problem of ordering alternatives in an SPBP instance is

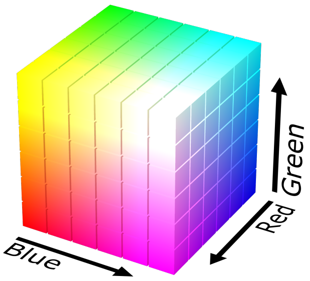


FIGURE 1.1: *Graphical representation of the RGB coding scheme used for colour representation on a computer.*

that a *most* pleasing colour is eventually chosen. This burden is so excessive that the DM will almost certainly satisfice instead of pursuing an "optimal" colour.

The reader may well object to the lack of practicality of the aforementioned example on the basis that many RGB-codes represent colours that are so slightly different as to be indistinguishable by the human eye. Indeed, in the practical problem of selecting a paint colour for the interior decoration of a room, for example, the DM is typically presented with a much smaller set of colour samples from which to choose. In such a case the DM may typically encounter a colour rack, such as the one shown in Figure 1.2, from which to make a selection.

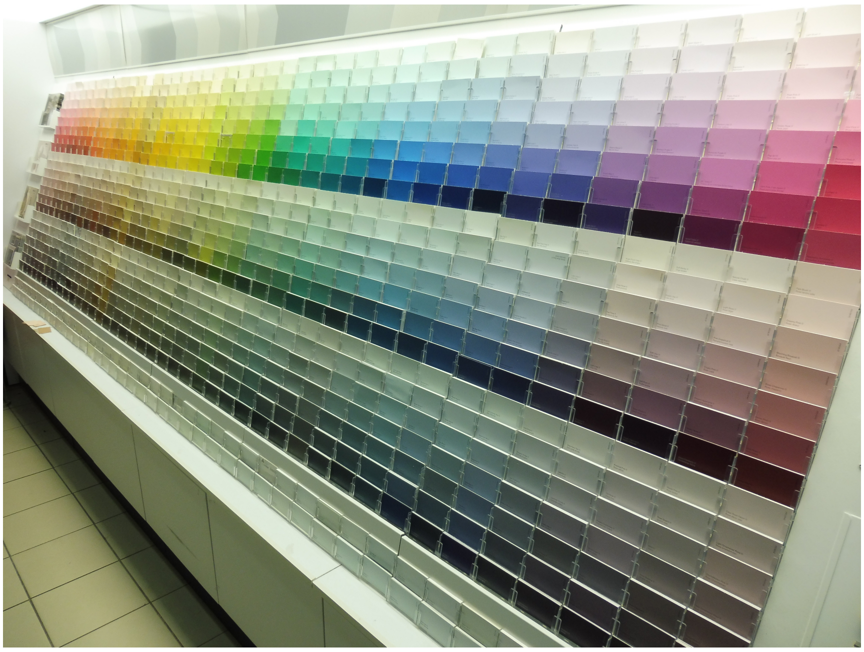


FIGURE 1.2: *A colour rack in a hardware store, containing colour sample cards for the selection of a pleasing paint colour.*

But even the colour rack in Figure 1.2 contains a total of $22 \times 42 = 924$ colours. If all pairs of these colours were to be compared in an attempt to discover the most pleasing colour, a total of $\binom{924}{2} = 426\,426$ pairwise comparisons would still be required, quite possibly causing the DM to satisfice yet again.

fundamentally reduced to a methodology involving at least some degree of pairwise comparison of alternatives by a DM in a bid to express his or her relative preferences of these alternatives.

Automated decision support facilitating *exploitation* of especially favourable regions of the decision space of an SPBP instance may therefore be particularly beneficial to a human DM, by potentially reducing his or her cognitive burden and thereby limiting the effects of satisficing due to time constraints.

Another reason for satisficing in the context of the SPBP may be that the DM might not be aware of all the available alternatives (*i.e.* might not fully be informed of all the regions of the entire decision space). If the DM is unable to characterise the full range of attributes embodied by the available alternatives, for instance, (s)he will likely be unaware of the best achievable alternative. In this case, (s)he may either aim for the unachievable when attempting to adopt an optimisation approach towards solving a particular SPBP, obtaining no solution as a result, or else (s)he might ask too little, obtaining a solution by satisficing that can be improved upon substantially.

Imagine, for an instant, that a (rather uninformed!) DM is unaware of the primary colour *red*. To the best knowledge of such a DM, the decision space of the aforementioned colour selection example would be restricted to colours characterised by RGB-codes of the form $(0, g, b)$ — *i.e.* colours within the rear green-blue slice of the RGB space in Figure 1.1. Presentation of a colour containing some degree of red to such a DM may well lead to an eye-opening discovery of an entire, previously unimaginable palette of colours that are *all* better than *any* alternative in the green-blue plane of the RGB cube.

Automated decision support facilitating *exploration* of the decision space of an SPBP instance may therefore also be beneficial to the DM, by reducing the limiting effects of satisficing as a result of the DM not being able to characterise the full range of attributes of alternatives in all parts of the decision space.

The problem of providing automated decision support to a human DM in respect of the exploitation and exploration of an SPBP decision space resides within the realm of *interactive preference learning*. Preference learning is a subfield that has relatively recently emerged from the larger field of *machine learning*, a research area in which the goal is to design computer algorithms that are able to evolve behaviours based on exposure to relevant empirical data [95]. More specifically, the prototype quest in preference learning is to predict the subjective preferences of a human DM based on past observed preference information elicited from that DM or similarly minded DMs. Learning of a human DM's behaviour is typically achieved in an iterative fashion, enabling better preference prediction as more and more iterations are carried out.

Classic examples of preference learning applications include the ability of an internet search engine to suggest search terms for a user based on information of past internet searches by that user or other users [205], the automated suggestion of merchandise that is likely to be interesting to a customer based on his or her history of past (online) shopping [103], or the recommendation of movies to an internet user within a video-sharing website, such as YouTube, based on videos the user has previously watched on that site [48].

Preference learning becomes *interactive* if a human DM is actively included in the iterative learning process, in which case such (subjective) learning is usually tailored to the specific DM involved. The interactive learning process usually involves exposing the DM to various decision alternatives in an iterative fashion and eliciting pairwise comparison judgements by the DM during each iteration. Although such a tailored approach to machine learning can produce very accurate preference predictions in the case of a consistent DM, the involvement of the DM in the learning cycle comes at the cost of burdening him or her with the cumbersome task of comparing each new alternative with all of those already encountered in terms of relative preference. Hence the effectiveness with which interactive preference learning takes place is often limited by the number of pairwise comparisons of decision alternatives the human DM can undertake within a reasonable timeframe or is willing to undertake.

1.2 Problem statement

The research hypothesis considered in this dissertation is that the number of pairwise comparisons required to estimate a relatively consistent DM's value function for a SPBP based on the consideration of n decision alternatives can be reduced to well below $\binom{n}{2}$ without sacrificing significantly on the quality of the function estimation. A generic interactive preference learning *decision support system* (DSS) is proposed in support of this hypothesis, and the working of the system is illustrated and tested in the context of the CSP described in §1.1. The degree to which DM inconsistency influences the quality of such value function estimation at a reduced cost in pairwise comparison effort, is also examined.

1.3 Dissertation objectives

The following six objectives are pursued in this dissertation:

I To *conduct* a survey of the academic literature on topics related to:

(a) value function construction based on the pairwise comparison of alternatives,

(b) preference modelling and computer-enhanced interactive preference learning,

(c) clustering methodologies relevant to discrete feature data,

(d) (single-objective) hypersurface optimisation methodologies,

(e) hypothesis testing within the broader field of statistical inference, and

(f) standard guidelines for the design, verification and validation of DSSs.

II To *establish*, based on the literature review of Objective I, a generic DSS for interactive preference learning in the context of the SPBP aimed at reducing the number of pairwise comparisons required.

III To *demonstrate* and *validate* the working of the interactive preference learning DSS of Objective II in the context of a hypothetical SPBP instance.

IV To *quantify*, in the context of the SPBP instance of Objective III, the ability of the interactive preference learning DSS of Objective II in respect of reducing the pairwise comparison burden on the part of a human DM when estimating his or her value function, and to *measure* the effect of DM inconsistency on this reduction quantification.

V To *evaluate* the effectiveness of the interactive preference learning DSS of Objective II when applying the system to a case study involving a realistic SPBP instance and a real human DM.

VI To *recommend* sensible follow-up future work related to the contributions of this dissertation.

1.4 Dissertation scope

The scope of this dissertation is limited to subjective preference modelling within the context of the SPBP, in which a number of alternatives are available, one of which should be selected based on an evaluation of the alternatives according to various selection criteria. Although there are admittedly subjective preference modelling contexts that do not conform to the structure of the SPBP, this archetypal subjective preference decision problem is nevertheless considered to be widely applicable.

It is furthermore assumed that all DM preferences related to decision alternatives and selection criteria can be expressed in one of the following forms:

Ratio-based comparisons. The DM is able to express the relative desirability of an alternative i over another alternative j as a ratio, by expressing this desirability in the form "alternative i is a_{ij} times more attractive than alternative j," where the desirability ratio $a_{ij} : 1$ is measured according to a suitable judgement scale.

Difference-based comparisons. The DM is able to express the relative severity of differences in attractiveness between pairs of alternatives according to some judgement scale.

Although a ratio-based preference modelling paradigm was historically one of the first to have been proposed and is intuitively easy to understand, it has been subjected to criticism in the literature [189]. For this reason, a difference-based preference modelling paradigm is also considered in this dissertation.

The interactive preference learning DSS put forward in this dissertation is illustrated, verified and validated within the context of a single instance of the SPBP — the CSP alluded to in §1.1. This SPBP instance is considered in two incarnations:

The *three-dimensional colour selection problem* (3DCSP). The problem of selecting a most desirable colour for use in some context from the full three-dimensional RGB-cube shown in Figure 1.1.

The *two-dimensional colour selection problem* (2DCSP). A special case of the 3DCSP in which the degree to which one of the three primary colours red, green or blue is present in a colour is fixed at some value and the most desirable colour with this property is sought for use in some context from the particular vertical or horizontal two-dimensional slice of the three-dimensional RGB-cube shown in Figure 1.1.

Although the interactive preference learning DSS put forward in this dissertation is designed to accommodate trade-offs between an arbitrary number of (partially) conflicting selection criteria, the system is validated in the form of case studies carried out in conjunction with real human DMs within the context of a single selection criterion only. Furthermore, no attempt is made to apply the DSS proposed in this dissertation to other instances of the SPBP. These two limitations of scope are simply the result of time and space constraints — it is nevertheless anticipated that the presentation of the DSS demonstration and validation will convince the reader that it may well find considerably wider application.

1.5 Dissertation organisation

Apart from this introductory chapter, the dissertation comprises a further nine chapters (organised in four parts), a bibliography and an appendix. Part I of the dissertation is a literature review, and consists of three chapters. The first chapter of Part I, Chapter 2, is devoted to a review of the academic literature on mathematical prerequisites pertaining to the topic of this dissertation. The chapter opens in §2.1 with a discussion on basic notions related to and central results on eigenvalues and eigenvectors of square matrices, since this forms one of the classical corner stones on which the elicitation of DM value function scores is based. The focus of the discussion then shifts to round-robin sports tournament modelling and the determination of winners in such tournaments by means of directed graphs in §2.2, since the theory of tournaments provides a very natural introduction to the ordering of SPBP alternatives according to a single selection criterion. Since clustering may profitably be applied to decision alternatives according to similarity based on DM value function score estimates for these alternatives, a brief overview of data clustering is presented in §2.3, singling out the k-means technique which is applied in this dissertation. A brief discussion also follows in §2.4 on exact and approximate techniques for solving single-objective optimisation problems. Two techniques are singled out for special attention during this discussion. These are the *parallel hill climbing algorithm* (PHC) and the method of *simulated annealing* (SA) — two methods that are employed in this dissertation within the interactive preference learning cycle. The DSS proposed in this dissertation for interactive preference learning involves stochastic elements, and so its validation necessarily has to be carried out within the realm of inferential statistics. A suite of well-known statistical tests is therefore reviewed in §2.5 for determining whether or not the means of a collection of approximately normally distributed data samples differ at a specified level of confidence. The chapter finally closes in §2.6 with a brief summary of the chapter content.

The second chapter of Part I, Chapter 3, follows the discussion on sports tournament results ranking by providing a review of very basic concepts in subjective preference modelling in §3.1, as well as a thorough review in §3.2 of the well-known *Analytic Hierarchy Process* (AHP), one of the early frameworks for solving multi-criteria instances of the SPBP. Another method of measuring subjective preference, is by *Measuring Attractiveness By a Categorical Based Evaluation TecHnique* (MACBETH), and is also discussed in §3.2. Thereafter, a review of various interactive preference learning models follows in §3.3. The chapter finally closes with a brief summary of the chapter content in §3.4.

The final chapter of Part I, Chapter 4, is a review of general guidelines in the literature for the design, verification and validation of DSSs. The first section of Chapter 4 contains a discussion on various types of DSSs available in the literature. Although these DSSs typically contain different elements, the three most important elements are discussed in §4.2. Three design DSS methodologies are then reviewed in §4.3. The penultimate section, §4.4, contains an overview of the various verification and validation methods available for DSSs. A brief summary of the chapter contents is finally provided in §4.5.

Part II of the dissertation consists of two chapters and focuses on the proposed DSS. Chapter 5 is the heart of the dissertation. In this chapter, the design of an interactive preference learning DSS is put forward. After providing a high-level overview of the iterative working of the proposed DSS in §5.1, an architecture is proffered for the framework in §5.2. After describing the two main components of the system (the *system configuration component* and the *data management component*) in detail in §5.3, the discussion turns to the proposed working of a central mechanism of the framework in §5.4 — the method of rescaling of DM value function scores as the system progresses through successive iterations of the preference learning cycle. The purpose of this mechanism is to achieve a trade-off between reducing the DM's workload associated with pairwise comparisons of decision alternatives, on the one hand, and accommodating small to moderate levels of DM inconsistency when carrying out these comparisons, on the other. The mechanism involves strategic use of a number of pivotal decision alternatives, called *anchors*. The presentation of the interactive preference learning DSS is concluded in §5.5 with a detailed process description, paying special attention to the order of events and the flows of data within the framework.

The second chapter of Part II, Chapter 6, contains two detailed worked examples in which the framework is applied to an instance of the 2DCSP, for a ratio-based approach (AHP) in §6.2 and for a difference-based approach (MACBETH) in §6.3 towards measuring subjective preference. The penultimate section, §6.4, contains a discussion of the results obtained in the worked examples. A brief summary of the chapter content follows in §6.5.

In the first part of Part III, Chapter 7, the interactive preference learning DSS proposed in Chapter 5 is subjected to a thorough validation in the context of the 2DCSP. After describing in §6.1 how the DM is modelled for system validation purposes and establishing two learning performance measure indicators according to which the framework effectiveness is evaluated, the intrinsic ability of the framework to learn is tested in §7.2 within the realm of inferential statistics at a 95% level of confidence for both a ratio-based and difference-based approach. The relative performances associated with adopting various judgement scales and value function aggregation methods are assessed for the ratio-based approach within the same inferential statistics paradigm in §7.3, after which the merits of exploration and exploitation of the 2DCSP decision space (in terms of preference learning effectiveness) are evaluated for both ratio- and difference-based approaches in §7.4, adopting the same inferential statistical approach. A classical sensitivity analysis is finally performed in §7.5 with respect to a variety of parameters that appear in the decision support framework, before the chapter closes in §7.6 with a brief summary of its content.

Whereas the entire validation process of Chapter 7 is carried out under the strong assumption of perfect DM consistency when eliciting pairwise comparisons of decision alternatives, Chapter 8 is devoted to a study of the effects of DM inconsistency on the learning facilitation effectiveness of the DSS of Chapter 5. The chapter opens in §8.1 with a description of how DM inconsistency is modelled for system evaluation purposes. The experimental setup of the aforementioned effectiveness evaluation analysis is also discussed. The first step in the analysis is to test in §8.2 how the the intrinsic ability of the DSS to learn DM preferences (for both ratio-based and difference-based approaches) is affected by the introduction of small to medium levels of DM inconsistency during the process of pairwise comparisons. This is followed in §8.3 by an assessment of the ability of the DSS to deal with DM inconsistency by increasing the number of anchors considered by the DM during each iteration. The chapter closes with a brief summary of the chapter content in §8.4.

Chapter 9 contains two realistic case studies based on the 2DCSP and involving real DMs. The case studies are conducted as further validation of the practical applicability of the DSS proposed in Chapter 5. After describing the experimental setup of the first case study, where a ratio-based approach is adopted, the progression of obtaining intermediate results from the DM when applying the system of Chapter 5 is presented in §9.1, culminating in a final decision recommendation to the DM and a discussion on the DM's response to this recommendation. The same format of reporting is followed in §9.2 for the second case study where a difference-based approach is incorporated, before the chapter closes with a short summary in §9.3.

The final part of the dissertation, Part VI, consists of two chapters, Chapters 10 and 11. Chapter 10 contains a summary of and reflection on the content of the dissertation. A summary is provided in §10.1 of the contents of the dissertation, after which an appraisal follows in §10.2 of the contributions of the dissertation. A number of ideas are finally provided in Chapter 11 with respect to possible future follow-up work building on the foundation laid in this dissertation.

Part I

Literature Review

CHAPTER 2

Mathematical prerequisites

Contents

This chapter contains a review of a number of mathematical prerequisites related to various aspects of the topic of this dissertation. This review is organised in such a manner as to provide the reader with the necessary mathematical and statistical background for developing an understanding of the material presented later in the dissertation.

The chapter opens in §2.1 with a brief discussion on the notions of eigenvalues and eigenvectors of square matrices. Basic properties of eigenvalues and eigenvectors are described, as well as how these quantities are computed. This is followed by a review of graph theoretic models of round-robin sports tournaments in §2.2 and how the players in such tournaments may be ranked. These models make extensive use of eigenvalues and eigenvectors. The review of §2.2 forms a basic framework for single-criterion SPBP decision support, as will be demonstrated in the following chapter.

The objective in data clustering is to group data points into groups so that the similarity between members of the same group is larger than that between members of different groups. It is often beneficial to be able to cluster alternatives in decision space together when making SPBP solution recommendations, because then the centroids of such clusters may be thought of as representing entire alternative clusters. In this manner, only these cluster representatives may be considered in the formulation of decision support to the DM (in a bid to avoid overwhelming him with a multitude of alternatives). For this reason, §2.3 is dedicated to a brief review of the large literature on data clustering. The two main traditional clustering paradigms, namely hierarchical clustering and partitional clustering, are briefly reviewed, after which the focus turns to other, more modern clustering algorithms outside of these paradigms. Attention is also afforded to the central questions of deciding *a priori* on the number of clusters into which the data should be partitioned, as well as the notion of cluster validation. The k-means clustering algorithm is singled out for a more detailed description, because it is employed later in this dissertation.

A central part of the validation process carried out later in this dissertation in respect of a newly proposed SPBP decision support framework hinges on an ability to perform (single-objective) optimisation. The focus of the chapter therefore shifts in §2.4 to a discussion on well-known optimisation techniques, distinguishing between the classes of approximation algorithms and approximate methods of optimisation. An emphasis is placed on two particular (approximate) optimisation methods employed later in this dissertation (the methods of PHC and SA).

Another set of tools employed later in this dissertation for decision support validation purposes is a suite of statistical inferential tests for distinguishing between the means of approximately normally distributed sample data. These (parametric) tests include the well-known *analysis of variance* (ANOVA) and Levene's test, as well as Fischer's *least significance difference* (LSD) *post hoc* test and the Games-Howell *post hoc test*. The purpose, working and underlying assumptions of each of these tests are described in §2.5.

2.1 Eigenvalues and eigenvectors

An *eigenvalue* λ of an $n \times n$ real matrix $\boldsymbol{A}$ is a real or complex number satisfying

$$\boldsymbol{Ax} = \lambda\boldsymbol{x} \tag{2.1}$$

for some nonzero real or complex vector $\boldsymbol{x}$, called the *eigenvector* of $\boldsymbol{A}$ associated with λ [90, 96]. These vectors play an important role in fields of study involving transformations of linear equations, such as genetics, quantum mechanics, economics and geometry [6, 97]. The value of a real eigenvalue λ of $\boldsymbol{A}$ determines whether the corresponding eigenvector $\boldsymbol{x}$ dilates, contracts or reverses direction upon multiplication by $\boldsymbol{A}$. Dilation occurs if $\lambda > 1$, contraction occurs if $0 < \lambda < 1$, and reversal of direction occurs if $\lambda < 0$, as illustrated in Figure 2.1 [7, 206].

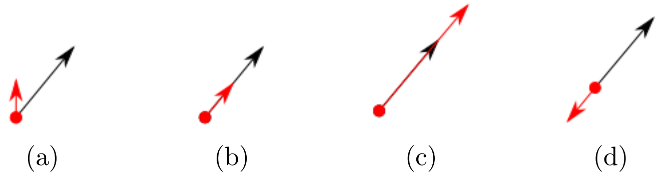


FIGURE 2.1: *Scalar multiple (red vector) of a vector $\boldsymbol{x}$ (black vector) by a matrix $\boldsymbol{A}$ in the cases where (a) $\boldsymbol{x}$ is not an eigenvector of $\boldsymbol{A}$, (b) $\boldsymbol{x}$ is an eigenvector of $\boldsymbol{A}$ and contraction occurs, (c) $\boldsymbol{x}$ is an eigenvector of $\boldsymbol{A}$ and dilation occurs, and (d) $\boldsymbol{x}$ is an eigenvector of $\boldsymbol{A}$ and reversal occurs.*

Equation (2.1) may be written as $\boldsymbol{Ax} = \lambda\boldsymbol{Ix}$ which may, in turn, be rewritten as

$$(\lambda\boldsymbol{I} - \boldsymbol{A})\boldsymbol{x} = \boldsymbol{0}, \tag{2.2}$$

where $\boldsymbol{I}$ denotes the $n \times n$ identity matrix. Equation (2.2) possesses a non-zero solution vector $\boldsymbol{x}$ if and only if the so-called *characteristic equation*

$$|\lambda\boldsymbol{I} - \boldsymbol{A}| = 0 \tag{2.3}$$

is satisfied [6, 24, 10, 162], where $|\bullet|$ denotes the determinant of a matrix argument $\bullet$. The left-hand side of this characteristic equation is an n-th order polynomial in view of the dimensions of $\boldsymbol{A}$, and hence admits n values of λ as solutions according to the following theorem.

Theorem 2.1 (Fundamental Theorem of Algebra, [71]) *Every polynomial equation of the form*

$$a_0\lambda^n + a_1\lambda^{n-1} + a_2\lambda^{n-2} + \cdots + a_{n-1}\lambda + a_n = 0 \tag{2.4}$$

in which the coefficients $a_0, a_1, \ldots, a_n$ are any complex numbers, whose degree n is at least one, and whose leading coefficient a_0 is not zero, possesses precisely n roots in the complex number system, provided that each multiple root of multiplicity m is counted as m roots.

Any scalar multiple of an eigenvector $\boldsymbol{x}$ associated with an eigenvalue λ of a matrix $\boldsymbol{A}$ is again an eigenvector of $\boldsymbol{A}$. To see why this is the case, notice that if $\boldsymbol{x}$ is a solution to (2.1), then so is $k\boldsymbol{x}$ for any constant $k \neq 0$, because then $\boldsymbol{A}(k\boldsymbol{x}) = \lambda(k\boldsymbol{x})$.

The *modulus* of an eigenvalue λ of a matrix is the absolute value of λ if λ is real or the value $\sqrt{a^2+b^2}$ if $\lambda = a + ib$ is complex (with $i = \sqrt{-1}$). This modulus is denoted by $|\lambda|$.

The computation of the eigenvalues and corresponding eigenvectors of a square matrix is illustrated in the following example.

Example 2.1 *Consider the 4×4 matrix*

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Because

$$\lambda\boldsymbol{I} - \boldsymbol{A} = \lambda \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \lambda & -1 & 0 & -1 \\ 0 & \lambda & -1 & -1 \\ -1 & 0 & \lambda & 0 \\ 0 & 0 & -1 & \lambda \end{bmatrix},$$

the characteristic equation (2.3) in this case becomes

$$|\lambda\boldsymbol{I} - \boldsymbol{A}| = \begin{vmatrix} \lambda & -1 & 0 & -1 \\ 0 & \lambda & -1 & -1 \\ -1 & 0 & \lambda & 0 \\ 0 & 0 & -1 & \lambda \end{vmatrix} = 0,$$

which simplifies to

$$\lambda^4 - 2\lambda - 1 = 0. \tag{2.5}$$

The solutions to (2.5) are $\lambda_1 = 1.39534$, $\lambda_2 = -0.46036 + 1.13632i$, $\lambda_3 = -0.46036 - 1.13632i$ and $\lambda_4 = -0.47463$. The eigenvector $\boldsymbol{x}_i$ of $\boldsymbol{A}$ associated with the eigenvalue λ_i may be obtained by substituting the value of λ_i into

$$\begin{bmatrix} \lambda_i & -1 & 0 & -1 \\ 0 & \lambda_i & -1 & -1 \\ -1 & 0 & \lambda_i & 0 \\ 0 & 0 & -1 & \lambda_i \end{bmatrix} \boldsymbol{x}_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.6}$$

and solving for $\boldsymbol{x}_i$ in (2.6). Substituting $\lambda_1 = 1.39534$ into (2.6), for example, yields the vector solution

$$\boldsymbol{x}_1 = \begin{bmatrix} 1.94691 \\ 1.71667 \\ 1.39534 \\ 1 \end{bmatrix}.$$

The same can be done for λ_2, λ_3 and λ_4, which have associated eigenvectors

$$\boldsymbol{x}_2 = \begin{bmatrix} -1.08612 - 1.04898i \\ 0.69512 - 0.75453i \\ -0.46035 + 1.13932i \\ 1 \end{bmatrix}, \quad \boldsymbol{x}_3 = \begin{bmatrix} -1.08612 + 1.04898i \\ 0.69512 + 0.75453i \\ -0.46035 - 1.13932i \\ 1 \end{bmatrix}, \quad \boldsymbol{x}_4 = \begin{bmatrix} 0.22527 \\ -1.10692 \\ -0.47463 \\ 1 \end{bmatrix},$$

respectively. In each of these cases, the eigenvectors corresponding to an eigenvalue has been scaled multiplicatively so that the entry with the largest modulus is 1. □

2.2 Tournaments

A tournament is a special type of directed graph which finds application in models of round-robin sports tournaments. After introducing certain basic concepts and notation related to tournament graphs in §2.2.1, a description follows in §2.2.2 of how such a tournament may be represented as a matrix. A natural method of ranking players in a tournament, known as the *first-order method*, is reviewed in §2.2.3, after which this method is generalised in §2.2.4 in order to be able to break ties. A refinement of this generalised ranking method, capable of taking into account the qualities of match victories, is finally described in §2.2.5.

2.2.1 Basic notions and terminology

A *directed graph* D, also known as a *digraph*, consists of a finite, non-empty set of elements, known as *vertices*, and an ordered set of distinct vertices, called *arcs* [16, 33, 35, 197]. The set of vertices of a digraph D is known as the *vertex set* of D and is denoted by $V(D)$. Similarly, the set of arcs of a digraph D is called its *arc set* and is denoted by $E(D)$. An arc of the form $e = (u, v)$ is said to *join* a vertex u to a vertex v. A direction from u to v is associated with such an arc. The cardinality of the vertex set of D is called the *order* of D, while the cardinality of its arc set is called the *size* of D. If a digraph D has vertex set V and arc set E, then D is sometimes written as $D = (V, E)$ [34].

In graphical representations of digraphs, the vertices are usually drawn as dots and the arcs as directed lines or curves. Figure 2.2 contains an example of a graphical representation of the digraph $D_1 = (V_1, E_1)$ with vertex set $V_1 = \{v_1, v_2, v_3, v_4, v_5\}$ and arc set $E_1 = \{(v_1, v_3), (v_2, v_1), (v_2, v_4), (v_3, v_4), (v_4, v_2), (v_4, v_5)\}$. The order of D_1 is 5, while its size is 6.

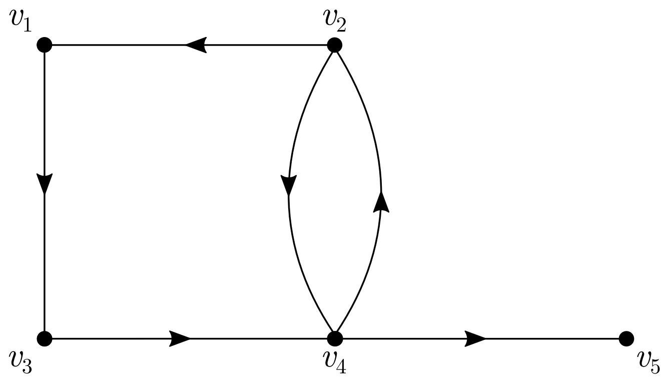


FIGURE 2.2: *A digraph D_1 with the vertex set $V(D_1) = \{v_1, v_2, v_3, v_4, v_5\}$ and arc set $E(D_1) = \{(v_1, v_3), (v_2, v_1), (v_2, v_4), (v_3, v_4), (v_4, v_2), (v_4, v_5)\}$.*

A *tournament* is a digraph in which, for every pair of vertices u and v, there is either an arc (u, v) from u to v, or else an arc (v, u) from v to u (but not both) [15, 16]. The name of this type of directed graph stems from the fact that such a digraph of order n may be used to model a round-robin tournament[1] for a sport in which individuals compete in pairs, where the vertices represent n competing players and an arc of the form (u, v) represents a match in which the player represented by vertex u beats the player represented by vertex v [34].

2.2.2 Matrix representation of tournaments

A well-known method of representing a digraph $D = (V, A)$ for computational purposes involves the use of an *adjacency matrix*. The adjacency matrix of a digraph D of order n is an $n \times n$ matrix denoted by $\boldsymbol{A}(D) = [a_{ij}]$, where a_{ij} is a binary parameter taking the value 1 if (i, j) is an arc of D, or the value 0 otherwise [88]. The adjacency matrix representation and the graphical representation of a round-robin tournament are demonstrated in the following example.

Example 2.2 *Consider a tournament T_1 consisting of four players v_1, v_2, v_3 and v_4, competing against one another in pairs. Suppose the adjacency matrix of T_1 is the matrix*

$$\boldsymbol{A}(T_1) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \tag{2.7}$$

of Example 2.1. Then T_1 may be represented graphically as a digraph $D_2 = (V_2, E_2)$ with vertex set $V_2 = \{v_1, v_2, v_3, v_4\}$ and arc set $E_2 = \{(v_1, v_2), (v_1, v_4), (v_2, v_3), (v_2, v_4), (v_3, v_1), (v_4, v_3)\}$, as shown in Figure 2.3. The digraph D_2, therefore, has order 4 and size 6. □

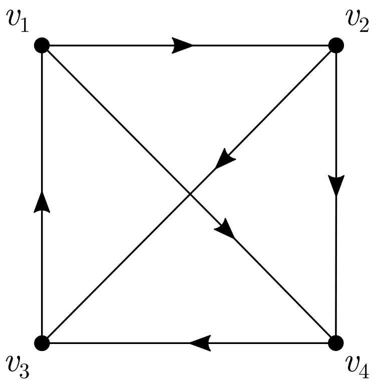


FIGURE 2.3: *Graphical representation of a round-robin tournament T of order 4, where an arc of the form $v_i \longrightarrow v_j$ indicates that player v_i beats player v_j.*

A v_i-v_j *walk* in a digraph is a distinct, alternating sequence of vertices and arcs that starts at vertex v_i, where each following vertex in the sequence is joined by an arc from the preceding vertex of the sequence to that vertex, and ends with vertex v_j [16, 35, 88]. The arcs of a walk are often omitted in the notation as they are obvious. This is demonstrated for the v_1-v_4 walk of length two, $v_1, (v_1, v_2), v_2, (v_2, v_4), v_4$ in the tournament T_1 of Example 2.2, which may simply be

[1] A tournament in which a number of participants compete against each other in all possible combinations of pairs.

expressed as v_1, v_2, v_4. The *length* of a v_i-v_j walk is the number of arcs in the walk. An example of a v_1-v_4 walk of length 1 in the tournament T_1 of Example 2.2 is v_1, v_4, while v_1, v_2, v_4 is an example of a v_1-v_4 walk of length two in the same tournament.

The distance from a vertex v_i to a vertex v_j in a digraph D is the length of a shortest v_i-v_j walk in D [16]. For example, the distance from v_1 to v_2 in the tournament T_1 of Example 2.2 is 1, while the distance from v_2 to v_4 is 3.

The following interesting theorem relates the number of v_i-v_j walks of length k in a tournament to the k-th power of the adjacency matrix of the tournament.

Theorem 2.2 *([16]) If $\boldsymbol{A}$ is the adjacency matrix of a tournament T, then the number of directed v_i-v_j walks of length k in T is the entry in row i and column j of the matrix power $\boldsymbol{A}^k$.*

The result of Theorem 2.2 is illustrated numerically in the following example.

Example 2.3 *The square of the adjacency matrix $A(T_1)$ of Example 2.2 is*

$$\boldsymbol{A}(T_1)^2 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \left[\begin{array}{cccc} 0 & 0 & 2 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{array}\right] \end{array}.$$

The 2 in row v_1 and column v_3 indicates, by Theorem 2.2, that there are two v_1-v_3 walks of length 2 in the tournament T_1 of Example 2.2. These two walks are v_1, v_2, v_3 and v_1, v_4, v_3. Similarly, the third power of the matrix $\boldsymbol{A}(T_1)$ is

$$\boldsymbol{A}(T_1)^3 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \left[\begin{array}{cccc} 2 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 2 & 1 \\ 0 & 1 & 0 & 1 \end{array}\right] \end{array}.$$

The 2 in row v_1 and column v_1 similarly indicates that there are two v_1-v_1 walks of length 3 in T_1. These walks are v_1, v_2, v_3, v_1 and v_1, v_4, v_3, v_1. □

2.2.3 Allocating first-order scores in tournaments

The number of arcs emanating from a player v in a tournament T is called the *outdegree* of v and represents the number of victories or the *score* of v. Chartrand [35] defines a winner w of a round-robin tournament as any player with the largest outdegree (score). It is therefore clear that a tournament T may have more than one winner. The following interesting result holds for tournaments.

Theorem 2.3 ([34]) *If v is a vertex of maximum outdegree in a tournament T, then the largest distance from v to any other player in T is 1 or 2.*

In a round-robin tournament, the implication of this theorem is that a winner w can only be defeated by players who themselves were beaten by players who were, in turn, defeated by w. The number of victories of each player in a round-robin tournament may be used to obtain a ranking of the players by ranking players with the most victories first, players with the second most victories next, and so on, until players with the fewest victories are ranked last [34]. This method of ranking is based on a score vector, denoted here by $\boldsymbol{t}^{(1)}$. The score vector for a tournament T of order n may be computed as

$$\boldsymbol{t}^{(1)} = \boldsymbol{A}(T)\boldsymbol{e}, \tag{2.8}$$

where $\boldsymbol{A}(T)$ is the adjacency matrix of T and $\boldsymbol{e}$ is a column vector of size n containing only ones. The score vector of the tournament in Examples 2.2 and 2.3 is $\boldsymbol{t}^{(1)} = [\ 2\ \ 2\ \ 1\ \ 1\ ]^T$, illustrating that players one and two are both ranked above players three and four. The ranking emanating from this score vector is denoted by $\{v_1, v_2\}$, $\{v_3, v_4\}$. No distinction is, however, made between players v_1 and v_2 in this ranking, and similarly for players 3 and 4.

2.2.4 Higher-order rankings in tournaments

Although a winner of a round-robin tournament T may be found by ranking players according to non-increasing outdegrees in T, as described in the previous section, it often occurs that two or more players receive the same ranking when this approach is adopted (as illustrated above). In some types of round-robin tournaments, quality grades may be associated with victories and losses in order to break such ties. By only counting the number of players that each player beats, as was done in the previous section, a so-called *first-order score vector* $\boldsymbol{t}^{(1)}$ is obtained. This method of scoring does not, however, take into account how good each player is who was beaten by a specific player. In order to measure how good the players are that were beaten by a particular player in a tournament, the number of players beaten by a particular player, who themselves have beaten other players may be counted. This gives rise to a *second-order score vector*, denoted by $\boldsymbol{t}^{(2)}$. A *third-order score vector* $\boldsymbol{t}^{(3)}$ may similarly be obtained by counting the number of players who have beaten other players, who themselves had beaten other players and the number of players these players had beaten. This gives rise to a sequence $\boldsymbol{t}^{(1)}$, $\boldsymbol{t}^{(2)}$, $\boldsymbol{t}^{(3)}$, ... of score vectors of increasing information complexity in the sense that by moving further down this sequence, more information is incorporated into the score vector $\boldsymbol{t}^{(k)}$ [88]. The question then arises as to how large the value of k should be in order to incorporate an appropriate amount of information in the score vector $\boldsymbol{t}^{(k)}$ so as to break all ranking ties. The limit

$$\theta = \lim_{k\to\infty} \boldsymbol{t}^{(k)} \tag{2.9}$$

clearly captures the maximum amount of information in such a score vector. It follows by Theorem 2.2 that

$$\boldsymbol{t}^{(k)} = \boldsymbol{A}(T)^k\boldsymbol{e}, \tag{2.10}$$

allowing the limit in (2.9) to be rewritten as

$$\theta = \lim_{k\to\infty} \boldsymbol{A}(T)^k\boldsymbol{e}. \tag{2.11}$$

In order to be able to evaluate the limit in (2.11) it is important to take cognisance of the concept of a *primitive* matrix. An $n \times n$ matrix $\boldsymbol{A}$ is *primitive* if there exists some exponent k such that all the entries of $\boldsymbol{A}^k$ are strictly positive [118]. The notion of matrix primitivity is demonstrated in the next example.

Example 2.4 *The adjacency matrix $\boldsymbol{A}(T_1)$ of the tournament T_1 in Examples 2.2 and 2.3 is primitive, since*

$$\boldsymbol{A}(T_1)^4 = \begin{bmatrix} 1 & 2 & 0 & 2 \\ 0 & 1 & 2 & 2 \\ 2 & 0 & 1 & 0 \\ 0 & 0 & 2 & 1 \end{bmatrix}, \qquad \boldsymbol{A}(T_1)^5 = \begin{bmatrix} 0 & 1 & 4 & 3 \\ 2 & 0 & 3 & 1 \\ 1 & 2 & 0 & 2 \\ 2 & 0 & 1 & 0 \end{bmatrix},$$

$$\boldsymbol{A}(T_1)^6 = \begin{bmatrix} 4 & 0 & 4 & 1 \\ 3 & 2 & 1 & 2 \\ 0 & 1 & 4 & 3 \\ 1 & 2 & 0 & 2 \end{bmatrix}, \qquad \boldsymbol{A}(T_1)^7 = \begin{bmatrix} 4 & 4 & 1 & 4 \\ 1 & 3 & 4 & 5 \\ 4 & 0 & 4 & 1 \\ 0 & 1 & 4 & 3 \end{bmatrix},$$

$$\boldsymbol{A}(T_1)^8 = \begin{bmatrix} 1 & 4 & 8 & 8 \\ 4 & 1 & 5 & 4 \\ 4 & 4 & 8 & 4 \\ 4 & 0 & 4 & 1 \end{bmatrix}, \qquad \boldsymbol{A}(T_1)^9 = \begin{bmatrix} 8 & 1 & 12 & 5 \\ 8 & 4 & 5 & 5 \\ 1 & 4 & 8 & 8 \\ 4 & 4 & 1 & 4 \end{bmatrix},$$

and so its ninth power is strictly positive. □

It is, in fact, known that the adjacency matrix of any tournament is primitive [88]. A well-known theorem that originated from the study of primitive matrices, known as the *Perron-Frobenius* theorem, may be used to evaluate the limit in (2.11).

Theorem 2.4 (Perron-Frobenius, [167]) *For a primitive, non-negative $n \times n$ matrix $\boldsymbol{A}$, the eigenvalue λ of $\boldsymbol{A}$ with largest modulus is real, positive and has multiplicity one. Furthermore, there exists an eigenvector $\boldsymbol{x}$ of $\boldsymbol{A}$ associated with λ, whose entries are all positive and which satisfies*

$$\lim_{k\to\infty} \frac{\boldsymbol{t}^{(k)}}{\lambda^k} = \lim_{k\to\infty} \frac{\boldsymbol{A}^k}{\lambda^k}\boldsymbol{e} = \boldsymbol{x}, \tag{2.12}$$

where $\boldsymbol{e}$ is an n-column vector containing only ones.

Application of the Perron-Frobenius theorem is demonstrated in the following example.

Example 2.5 *The first-order score vector of the tournament T_1 in Example 2.2 is given by the row-sum vector $\boldsymbol{t} = [\ 2 \ \ 2 \ \ 1 \ \ 1\ ]^T$ of the adjacency matrix $\boldsymbol{A}(T)$. As mentioned, no distinction can be made between players v_1 and v_2 based solely on this score vector, and similarly for players 3 and 4. The ranking emanating from this score vector is $\{v_1, v_2\}, \{v_3, v_4\}$. The solution to breaking these ties according to (2.11) is to raise the matrix $\boldsymbol{A}(T)$ to powers larger than one and to compute the corresponding row-sum vectors. The square of the adjacency matrix of T_1 was computed in Example 2.3, and yields the row-sum vector $\boldsymbol{t}^{(2)} = [\ 3 \ \ 2 \ \ 2 \ \ 1\ ]^T$ as second-order score vector. According to this score vector, a distinction can now be made between players 1 and 2 as well as between players 3 and 4, but not between players 2 and 3. This score vector results in the ranking $v_1, \{v_2, v_3\}, v_4$.*

The third power of the adjacency matrix of T_1 was also computed in Example 2.3 and yields the row-sum vector $\boldsymbol{t}^{(3)} = [\ 3 \ \ 3 \ \ 3 \ \ 2\ ]^T$ as a third-order score vector. This third-order score vector can only make a distinction between player 4 and the other players, while no distinction can be made between players 1, 2 and 3. This score vector results in the ranking $\{v_1, v_2, v_3\}, v_4$. Note the interesting phenomenon of rank reversal *that has occurred here: Whereas v_2 is ranked second according to the second-order score vector, v_2 is ranked first (albeit jointly) according to the third-order score vector.*

Rather than following the time–consuming process of iteratively increasing the value of the exponent k in the power matrix $\boldsymbol{A}(T_1)^k$ in (2.10), the Perron-Frobenius theorem may be invoked to estimate the limit in (2.12). As seen in Example 2.1, the eigenvalue of $\boldsymbol{A}(T_1)$ with largest modulus is λ_1=1.39534, confirming the statement of the Perron-Frobenius theorem that this eigenvalue is real and positive. A normalised associated eigenvector is

$$\boldsymbol{x_1} = \begin{bmatrix} 1.94691 \\ 1.71667 \\ 1.39534 \\ 1 \end{bmatrix}.$$

The entries of this vector are clearly all positive and represent the infinite-order score vector in (2.10) for the tournament T_1 in the limit as $k \to \infty$. This improved score vector now suggests v_1, v_2, v_3, v_4 as a full ranking, which effectively distinguishes between all four players. □

2.2.5 Recording the qualities of match victories in a tournament

In the discussion thus far, it was assumed that match victories in a tournament are recorded in a binary fashion. That is, the entry in row i and column j ($\neq i$) of a tournament's adjacency matrix is either a zero (recording the fact that player v_i beat v_j) or a one (recording the fact that v_j beat v_i). The margins of these victories (*i.e.* how convincingly players beat other players) are not, however, considered when recording victories in such a manner.

The methodology of §2.2.4 may nevertheless be extended to accommodate the situation where the margin of victory is recorded. Saaty [183] suggested using a nine-point linear scale which can be employed for this purpose when populating the adjacency matrix of a tournament instead of using a binary scale as described above. Paraphrased in terms of tournament match victories, this scale is presented in Table 2.1.

a_{ij}	Description
1	The match between players v_i and v_j resulted in a draw
3	Player v_i marginally beat player v_j
5	Player v_i easily beat player v_j
7	Player v_i convincingly beat player v_j
9	Player v_i crushed player v_j

Table 2.1: *Specification of the value a_{ij} in row i and column j of the adjacency matrix of a tournament according to a scale proposed by Saaty [183] (paraphrased in the context of tournaments). The values 2, 4, 6 and 8 may be used as intermediate judgement statements.*

The matrix of match outcome values a_{ij} for $i, j \in \{1, \ldots, n\}$ is called a *pairwise comparison matrix* of the players' performances in the tournament. The interpretation of the meanings of the entries in this matrix may be extrapolated as being indicative of the players' skill levels. In this sense the pairwise comparison matrix entries may be interpreted as ratios in terms of the players' relative abilities. More specifically, when populated according to the scale outlined in Table 2.1, an entry in row i and column j of the matrix may be interpreted as a ratio $a_{ij} : 1$ of player i's ability to that of player j (or, equivalently, representing the fact that player i is a_{ij} times as good as player j).

In order to promote internal consistency during the population of a tournament's pairwise comparison matrix, Saaty [183] suggested that if the margin of victory between players v_i and v_j is recorded in row i and column j as the value a_{ij} according to the judgement scale in Table 2.1, then the entry in row j and column i should be the multiplicative inverse $a_{ji} = \frac{1}{a_{ij}}$. Furthermore, the entries on the main diagonal of the tournament matrix representation should be recorded as a one (instead of a zero as before), representing the fact that player v_i is indistinguishable from him/herself — a situation reminiscent of a draw occurring between v_i and himself (for all $i = 1, \ldots, n$).

Since the aforementioned tournament representation is by definition primitive, it follows from the Perron-Frobenius theorem (Theorem 2.4) that the eigenvector methodology of §2.2.4 for infinite-order tournament scoring remains relevant in this generalised setting where the qualities of tournament match victories are recorded, as is illustrated in the following example.

Example 2.6 *Consider the refinement matrix representation*

$$\boldsymbol{A}'(T_1) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \left[\begin{array}{cccc} 1 & 7 & \frac{1}{5} & 3 \\ \frac{1}{7} & 1 & 5 & 3 \\ 5 & \frac{1}{5} & 1 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{3} & 2 & 1 \end{array}\right] \end{array} \qquad (2.13)$$

of the tournament T_1 of Example 2.2 in which graded qualities of the match victories are recorded according to the judgement scale of Table 2.1. Note that whereas the matrix entry $a_{12} = 1$ in (2.7) merely denoted the fact that player v_1 beat player v_2, the entry a'_{12} now records that player v_1, in fact, beat player v_2 convincingly. Similar interpretations hold for the remaining entries of $\boldsymbol{A}'(T)$. The eigenvalues of $\boldsymbol{A}'(T)$ are $\lambda'_1 = -0.139$, $\lambda'_2 = 7.618$, $\lambda'_3 = -1.739 - 4.904i$ and $\lambda'_4 = -1.739 + 4.904i$, and the corresponding eigenvectors are

$$\boldsymbol{x}'_1 = \begin{bmatrix} 3.006 \\ 2.344 \\ 2.417 \\ 1 \end{bmatrix}, \boldsymbol{x}'_2 = \begin{bmatrix} 0.031 \\ -0.419 \\ -0.505 \\ 1 \end{bmatrix}, \boldsymbol{x}'_3 = \begin{bmatrix} -2.434 - 3.037i \\ 2.692 - 0.604i \\ -1.412 + 3.059i \\ 1 \end{bmatrix} \text{ and } \boldsymbol{x}'_4 = \begin{bmatrix} -2.434 + 3.037i \\ 2.692 + 0.604i \\ -1.412 - 3.059i \\ 1 \end{bmatrix}.$$

As guaranteed by the Perron-Frobenius theorem, the eigenvalue with largest modulus (λ'_1) is associated with a real eigenvector ($\boldsymbol{x}'_1$), all of whose entries have the same sign. This eigenvector suggests the infinite-order ranking v_1, v_2, v_3, v_4 for the tournament. □

Scale types	Definition	Parameters
Linear (Saaty [182], 1977)	$c(x) = a \times x$	$a > 0$
Power (Harker and Vargas [85], 1987)	$c(x) = x^a$	$a > 1$
Geometric (Lootsma [141], 1989)	$c(x) = a^{x-1}$	$a > 1$
Logarithmic (Ishizaka *et al.* [102], 2010)	$c(x) = \log_a(x+1)$	$a > 1$
Root square (Harker and Vargas [85], 1987)	$c(x) = \sqrt[a]{x}$	$a > 1$
Balanced (Salo and Hämäläinen [188], 1997)	$c(w) = \frac{w}{1-w}$	$w = 0.5, 0.55, 0.6, \ldots, 0.9$

TABLE 2.2: *Judgement scales $c(x)$ for pairwise comparison matrix entries where $x \in \{1, 2, \ldots, 9\}$ [134].*

The linear judgement scale in Table 2.1 has been criticised by a number of authors for various reasons [91, 102]. Alternative judgement scales have subsequently been proposed to remedy these criticisms. A selection of these alternative scales is shown in Table 2.2. The scale in Table 2.1 is the linear scale in Table 2.2 corresponding to the parameter value $a = 1$ (rendering the judgement scale values $c(x) \in \{1, 2, \ldots, 9\}$). The power judgement scale with parameter $a = 2$ will, for example, result in the alternative pairwise comparison matrix specifications shown in Table 2.3.

a_{ij}	Description
1	The match between players v_i and v_j resulted in a draw
9	Player v_i marginally beat player v_j
25	Player v_i easily beat player v_j
49	Player v_i convincingly beat player v_j
81	Player v_i crushed player v_j

Table 2.3: *Specification of the value a_{ij} in row i and column j of the adjacency matrix of a tournament according to the power scale in Table 2.2 for the parameter $a = 2$ (paraphrased in the context of tournaments). The values 4, 16, 36 and 64 may be used as intermediate judgement statements.*

2.3 Data clustering

As mentioned in the introduction, it is sometimes useful to cluster the alternatives of an SPBP instance into groups of approximately similar DM preference levels. This kind of clustering has the potential to make for useful recommendations in terms of DM decision support (a representative of each cluster may, for example, be presented to the DM). A description of the well-known k-means clustering algorithm, which may be employed for this purpose, is therefore provided in §2.3.2. The section, however, opens in §2.3.1 with a brief overview of the landscape of available data clustering techniques as a means to providing general cluster analytic context.

2.3.1 A classification of data clustering techniques

The research area of data clustering is concerned with the process of grouping data objects into similarity classes. These similarity classes, referred to as *clusters*, should ideally capture the natural structure of the data set so as to ensure that any subsequent data clustering analysis is meaningful. The goal in data clustering is usually to group data objects so that the objects included in a particular cluster are both sufficiently similar to one another and sufficiently different from the objects included in other clusters [126]. A clustering exhibiting this characteristic is called *strong*. One of the advantages of applying clustering to data is that when the clustering is strong, one object (typically the centroid) of a cluster may represent the characteristics of that cluster as a whole. This approach toward data representation allows for considerable computational simplification when working with large data sets.

While data clustering may be thought of as a fundamental task in data mining classification, it is not limited to that particular field of study. Scientific disciplines which routinely benefit from data clustering include psychology, biology, genetics, applied mathematics and statistics [74]. The process of data clustering is, in general, an NP-hard[2] combinatorial optimisation problem. Numerous approaches towards data clustering have been proposed in the literature. A taxonomy of some of these approaches was put forward by Jain *et al.* [106], and an extension of this taxonomy is illustrated graphically in Figure 2.4.

The two major categories of clustering techniques are *hierarchical clustering methods* and *partitional clustering methods*. *Hierarchical clustering methods* typically involve the successive merging together or partitioning of data clusters. In contrast, *partitional clustering methods* take

[2] In complexity theory, a computational problem is *NP-hard* if its underlying (binary) decision problem is *NP-complete*. A decision problem is *NP-complete* if it is at least as hard to solve as any decision problem which can be answered "yes" in polynomial time, given additional information, called a *certificate*. Practically speaking, if a computational problem is NP-hard, it is anticipated that the best possible algorithm for solving it will run in exponential time [69].

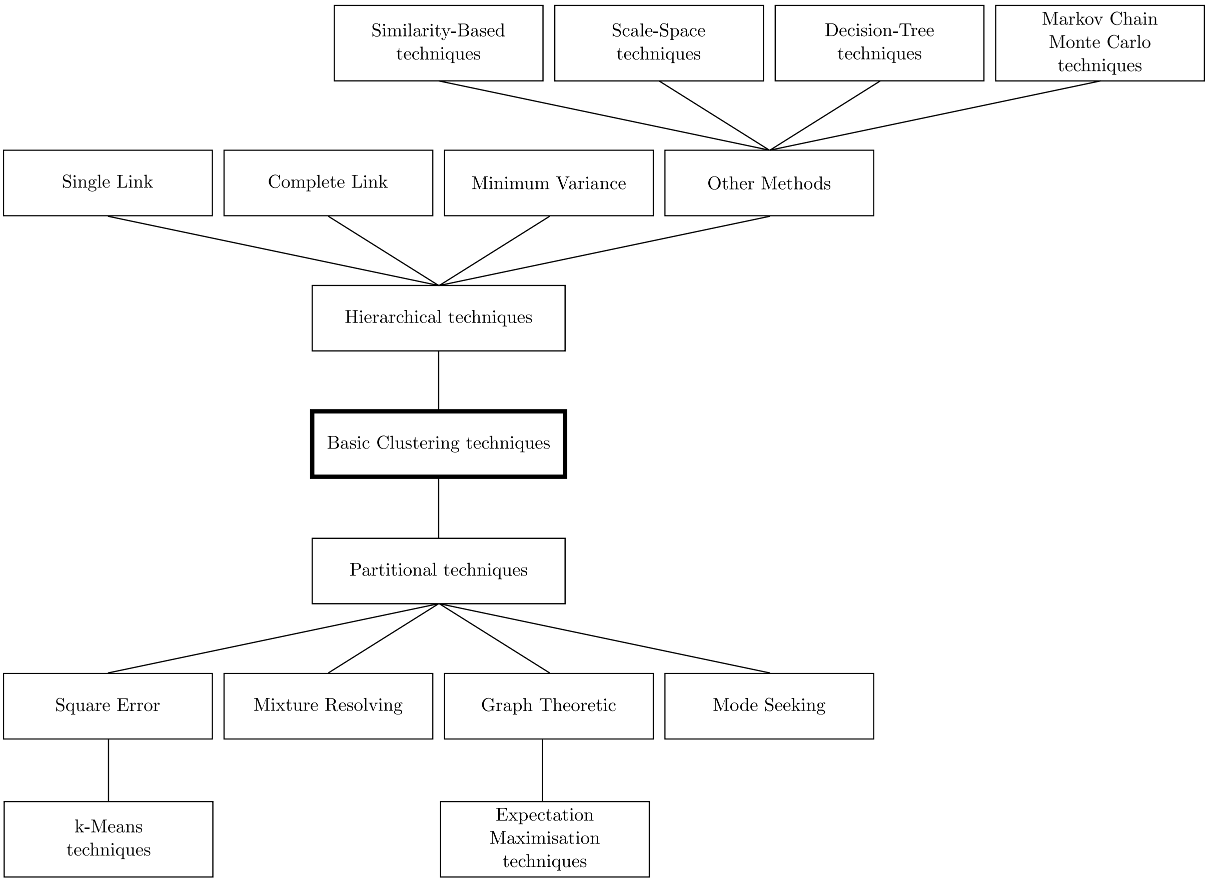


Figure 2.4: *A taxonomy of the most basic data clustering techniques (adapted from Jain* et al. *[106]).*

existing data clusterings and function by re-assigning data points to different clusters until some convergence criterion is met. Since the literature on data clustering techniques is very large, the discussion in this section on data clustering is by no means exhaustive. Neither is the goal to be even representative. Instead, a number of the most popular clustering techniques are discussed within each of the two basic categories mentioned above.

Hierarchical clustering techniques

There are two paradigms according to which hierarchical clustering may be performed: the agglomerative paradigm and the divisive paradigm. In the *agglomerative paradigm*, the process starts with singleton clusters (*i.e.* each cluster containing a single data point), and pairs of clusters are then iteratively merged together, thus obtaining successively larger and larger clusters, until some termination criterion is met. Clustering within this paradigm is therefore often said to occur in a *bottom-up* fashion. In the *divisive paradigm*, on the other hand, the process starts with a single grand cluster (containing all the data points), and a selected cluster is iteratively partitioned into two clusters, thus obtaining successively smaller and smaller clusters, until some termination criterion is met. Clustering within this paradigm may therefore be considered to occur in a *top-down* fashion.

Agglomerative hierarchical clustering algorithms are often variations on the well-known *single-link* [201], *complete-link* [117], or *minimum-variance* (Ward's method) [223] techniques. The distinguishing feature of these algorithms is the manner in which they measure the (dis)similarity between pairs of clusters. According to the single-link approach, the dissimilarity between two clusters is taken as the smallest distance between all pairs of data points drawn from the two clusters (one point from each cluster), as illustrated in Figure 2.5(a), while according to the complete-link approach, this dissimilarity is taken as the largest of all pairwise distances between two data points in different clusters, as illustrated in Figure 2.5(b). According to the minimum-variance approach, this dissimilarity is taken as the average pairwise distance between two data points in different clusters, as illustrated in Figure 2.5(c).

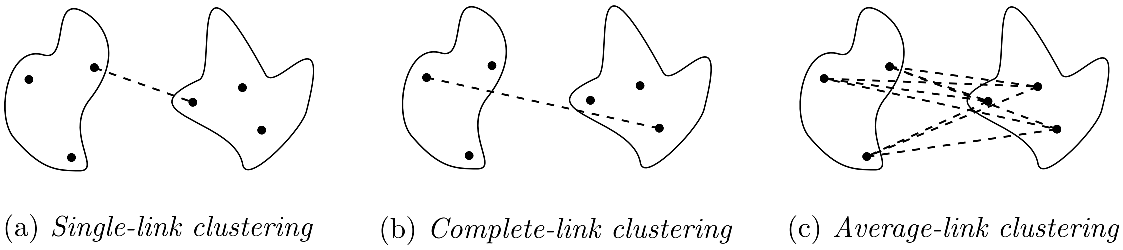
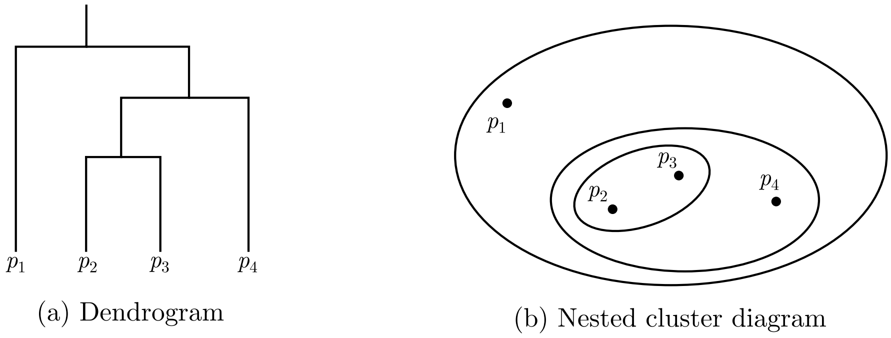

(a) *Single-link clustering* (b) *Complete-link clustering* (c) *Average-link clustering*

FIGURE 2.5: *Three similarity measures employed in agglomerative hierarchical clustering techniques.*

All three the aforementioned algorithms adopt a bottom-up, iterative clustering approach. During each iteration, two clusters of minimum dissimilarity are merged together to form a larger cluster. In this way, progressively larger clusters are gradually formed, until some termination criterion is met. A tree-representation of this iterative process, called a *dendrogram*, facilitates the exploration of the clustering hierarchy at different levels of granularity [20]. An example of such a dendrogram is shown in Figure 2.6(a). This example corresponds to the nested cluster diagram in Figure 2.6(b). The order in which clusters are merged together over time in Figure 2.6(b) is captured by the outward growing nested clusters (first the clusters $\{p_2\}$ and $\{p_3\}$ are merged together, thereafter the clusters $\{p_2, p_3\}$ and $\{p_4\}$ are merged together, and so on). This order of cluster mergers also emerges when sliding a virtual dynamic horizontal section of the dendrogram in Figure 2.6(a) upwards and noting the cluster intersections that occur over time.



(a) Dendrogram (b) Nested cluster diagram

FIGURE 2.6: *(a) A dendrogram capturing the process of agglomerative hierarchical clustering in (b).*

Baeza-Yates [12] showed that the complete-link algorithm usually produces tightly knit or compact clusters, while Nagy [157] showed that the single-link algorithm often suffers from the so-called *chaining effect* (a tendency to return clusters that are elongated in the data feature space). Hierarchical clustering algorithms are reportedly more versatile than their partitional counterparts, because they accommodate data sets that do not admit isotropic clusters[3] with ease [157]. Day [49] pointed out, however, that the time and space complexities of partitional algorithms are usually considerably more favourable than those of hierarchical algorithms.

[3] *Isotropic clusters* exhibit a similar nature for all data features considered.

An alternative agglomerative hierarchical clustering algorithm was proposed in 2000 by Leung *et al.* [135]. This algorithm was inspired by human visual research based on scale-space theory. Here clustering is interpreted as the blurring together of objects when one squints one's eyes. Each data point is analogously regarded as a point of light and a data cluster corresponds to a blurred blob. The algorithm progressively merges these blobs, initially starting out from many different light sources (corresponding to the data points themselves), until the entire image becomes one blurred light blob.

In 2002, Li and Biswas [137] proposed another hierarchical data clustering algorithm, called the *similarity-based agglomerative clustering algorithm*. The algorithm makes use of the so-called Goodall similarity measure[4]. The process is carried out by calculating the proximity of mixed data type observations based on less common matches of data features.

Castro *et al.* [31] introduced the *Markov chain Monte Carlo-based hierarchical clustering method* and the *hierarchical likelihood tree algorithm* in 2004. The working of both of these algorithms is based on a maximum likelihood principle. The algorithms are relatively robust with respect to errors in the cluster similarity matrix, because they employ a generative tree-structured model which represents relationships between the data objects as opposed to directly modelling properties of the data objects.

In 2005, Basak and Krishnapuram [18] proposed an unsupervised decision tree algorithm for divisive hierarchical data clustering. The distiguishing property of this decision tree is its facilitation of the interpretation of the clustering results by means of a set of interpretation rules later formalised by Quinlan [170]. The algorithm starts out at the root node of the decision tree and successively partitions the set of data points based on a particular feature identified by four distinct criteria from information theory [79]. This partitioning procedure is repeated until the number of data points within a clustering node is smaller than some pre-specified value.

Partitioning clustering techniques

Whereas hierarchical clustering starts out with singleton clusters and successively merges clusters (within the agglomerative paradigm) or starts out with one grand cluster containing all data points and successively partitions clusters (within the divisive paradigm), a partitional clustering algorithm takes as input a complete clustering and aims to optimise it by iteratively relocating data points between clusters until some (locally) optimal partitioning criterion is satisfied. The partitional clustering approach is advantageous in applications that involve very large data sets. In such applications, clustering according to a dendrogram is usually not feasible [227]. Even small-scale clustering problem instances can lead to excessively large clustering solution spaces [140].

Partitional clustering algorithms furthermore often provide better insight into major structures within the data, because larger clusters are typically generated during the early stages of the clustering process. This means that the algorithm is typically far less likely to suffer from accumulated erroneous clustering decisions (such as often occur in an agglomerative hierarchical approach and cannot be corrected during subsequent cluster merging decisions) [113]. A major concern associated with partitional clustering algorithms, however, is the selection of a suitable number of output data clusters. A rule-of-thumb recommendation as to a desirable number of clusters was given by Dubes [59], but it is ultimately safer to base this decision on an empirical

[4] The *Goodall similarity measure* is based on the probability of the likelihood of a random sample of two data points exhibiting similar values for some data feature under consideration. The cumulative probability for the observed pair is then computed by combining the separate probabilities. A similarity matrix is finally computed from the complements of these combined probabilities [76].

analysis of the particular data set. Exhaustive searches for an optimal number of clusters is usually infeasible in practical applications and this has led to the heuristic approach of executing a partitional clustering algorithm multiple times, each time choosing a different starting state, and then finally selecting the best resulting cluster configuration [107].

The most popular partitional clustering criterion is the so-called *squared error criterion.* The *sum of squares error value* associated with a clustering containing k clusters $C_1, \ldots, C_k$ is

$$E^2 = \sum_{i=1}^{k} \frac{1}{|C_i|} \sum_{j=1}^{|C_i|} ||\boldsymbol{x}_j^i - \boldsymbol{c}_i||^2, \tag{2.14}$$

where $\boldsymbol{x}_j^i$ denotes a vector representation of data point j of cluster C_i and $\boldsymbol{c}_i$ denotes the centroid of cluster C_i. This clustering criterion is typically more effective in accommodating isolated and compact clusters than many other criteria.

The most commonly employed partitional clustering algorithm is the *k-means method* [147]. According to this method, each data point is assigned to the cluster whose centre (the centroid of all the data points already contained within the cluster) is closest to it. The algorithm iteratively re-assigns data points to clusters until some pre-specified convergence criterion is met. Since the k-means algorithm is employed later in this dissertation, the method is singled out for a more detailed description in the following section.

A well-known *graph theoretic (partitional) clustering algorithm* involves constructing a shortest spanning tree for a complete graph in feature space on the data. The edge weights of the complete graph represent the distances between pairs of data points. The edges with the largest weights are iteratively deleted from this spanning tree so as to generate clusters of data points that are close to one another in feature space. The class of hierarchical clustering algorithms is also related to graph theoretic clustering, because the graphs associated with single-link clusters are components of a minimum spanning tree on the data [78], while those associated with complete-link clusters form maximal cliques (complete subgraphs) [14]. Hartuv and Shamir [86] interpreted clusters as highly-connected subgraphs and recursively applied a minimum-weight cut procedure to disconnect the graph by deleting the smallest number of edges in order to identify these highly connected subgraphs.

The underlying assumption in *mixture-resolving (partitional) clustering algorithms* is that the data points which have to be clustered are drawn from a known distribution, and the goal is to identify the parameters of this distribution. Most of the clustering algorithms in this class assume that the data points of the mixture considered follow a Gaussian distribution [107]. Traditional mixture-resolving algorithms iteratively obtain a maximum-likelihood estimate of the distribution parameters [105]. More recent mixture-resolving algorithms, however, seem to adopt expectation maximisation approaches toward estimating data distribution parameters [151]. There are also non-parametric techniques for density-based clustering, such as the algorithm proposed by Jain and Dubes [105].

The so-called *mean-shift algorithm* is perhaps the most popular member of the class of mode-seeking (partitional) clustering algorithms [61]. The algorithm was proposed by Fukunaga [67] in 1975, and was popularised by Cheng [36] and by Comaniciu and Meer [42].

Other classes of clustering algorithms

The clustering algorithms described in the previous two sections are basic, well-known algorithms. More sophisticated methods of clustering have also been proposed in the literature, but these algorithms are not discussed here in any detail. These alternative clustering methods reside within the following classes of algorithms:

- rearrangement clustering algorithms [20, 40],
- constrained clustering algorithms [23, 104, 213], and
- metaheuristic clustering algorithms [11, 26, 83, 207, 212].

Selecting the number of clusters

The problem of determining the ideal number of clusters into which a data set should be clustered is one of the major challenges associated with cluster analysis [104]. As mentioned above, a common approach is to repeat execution of a clustering algorithm multiple times with different numbers of clusters and then selecting the best number of clusters based on the results obtained. There are, however, also more sophisticated methods of determining a suitable number of clusters for a given data set. These methods include basing the decision on the number of clusters on:

- a combination of the so-called *minimum message length criterion* [219, 220] and a Gaussian mixture model [60],
- the so-called *principle of minimum description length* [84],
- Bayes' information criterion [194],
- the *Akaike information criterion* [187], and
- so-called *gap statistics* [211].

Cluster validation

Since clustering algorithms return clusters irrespective of whether or not there are natural clustering features present in the data, it is crucial to validate the clustering returned by a clustering algorithm. Clusterings may be validated based on three different kinds of criteria: internal criteria, relative criteria, or external criteria [104]. Validity indices based on *internal criteria* validate the fit between the structure imposed by a clustering and the actual features of the data, taking into account the features of the data only. Validity indices based on *relative criteria* usually involve a comparison of many clustering structures (generated by a variety of clustering algorithms) and a subsequent decision as to which algorithm produces the best clusters. Validity indices based on *external criteria* typically measure the quality of clusters by matching clustering structures to prior information.

Lange *et al.* [128] introduced the concept of *cluster stability* to validate data clusters. This concept is a measure of the variation inherent in a clustering over sub-samples drawn from the input data. A variety of measures of variation may be used to derive different incarnations of the notion of cluster stability. Shamir and Tisby [198], for example, suggested defining stability as the ability of a clustering algorithm to generalise a clustering, given more data.

2.3.2 The k-means clustering algorithm

The k-means algorithm is certainly the simplest and most popular partitional clustering method. The parameter k in the name of the algorithm is user-specified and represents the desired number of clusters. Suppose a set of $\mathcal{S}$ data points has to be clustered into k clusters, denoted by

$C_1, \dots, C_k$. These clusters are represented by their centroids $\boldsymbol{c}_1, \dots, \boldsymbol{c}_k$ in feature space, respectively. A high-level pseudo-code description of the standard k-means clustering method is given in Algorithm 2.1.

Algorithm 2.1: **Standard k-means clustering algorithm**

```
Input  : A data set $\mathcal{S}$ and the number $k$ of clusters required.
Output: $k$ Data clusters (the union of these clusters is the complete data set $\mathcal{S}$).
choose $k$ data points randomly from the set $\mathcal{S}$;
while the sum of squares error value in (2.14) of consecutive iterations are not equal do
    assign each data point to a cluster corresponding to the closest centroid;
    re-calculate the sum of squares error value in (2.14);
    update the centroids of the clusters by taking the mean (2.15) of all points in a cluster;
end
```

The first step is to select the initial set of centroids as k of the data points according to some heuristic procedure. The choice of this initial set of centroids can dramatically influence the quality of the final set of clusters returned. This risk may be mitigated to some extent by repeatedly executing the algorithm for different initial sets of centroids in the hope of identifying a high-quality final clustering from the resulting set of solutions.

The next step is to assign each remaining data point to the centroid nearest to it according to some preferred distance measure in feature space. The centroids are then updated. The new centroid of cluster C_i is

$$\boldsymbol{c}_i = \frac{1}{|C_i|} \sum_{j=1}^{|C_i|} \boldsymbol{x}_j^i, \tag{2.15}$$

where the meaning of the symbol $\boldsymbol{x}_j^i$ is as defined in §2.3.1. An iterative process of re-assigning data points to their closest centroids is then carried out until the sum of squares error value in (2.14) becomes so small that the process may be judged to have converged, at which point the algorithm terminates, returning the final clustering.

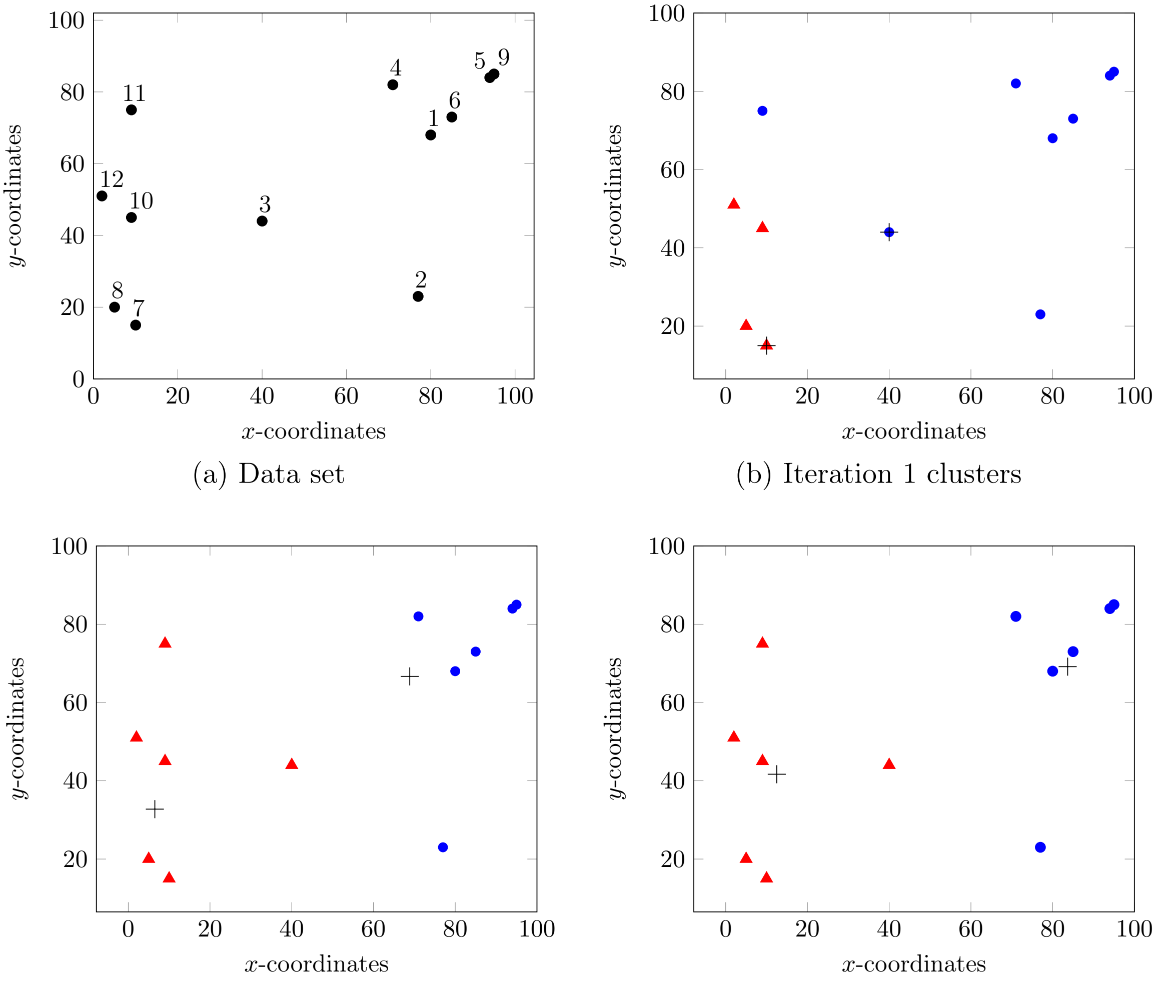
The working of the k-means algorithm is illustrated in the following example.

Example 2.7 *Consider the data set in Table 2.4 consisting of twelve data points, taken from Nel [160]. These data are shown graphically in Figure 2.7(a).*

Points	1	2	3	4	5	6	7	8	9	10	11	12
x	80	77	40	71	94	85	10	5	95	9	9	2
x	68	23	44	82	84	73	15	20	85	45	75	51

TABLE 2.4: *Illustrative data set used in numerical example of Example 2.7.*

Suppose the k-means algorithm is applied to these data in order to cluster the data into $k = 2$ clusters, selecting the data points 3 and 7 as initial cluster centres. Then the algorithm clusters the points 7, 8, 10 and 12 together in one cluster and all other data points in the other cluster during the first iteration, as shown in the first row of Table 2.5 and in Figure 2.7(b). The sum of squares error value associated with this clustering is $3\,127.88$. Thereafter, the cluster centroids are computed as $(68.87, 66.67)$ and $(6.50, 32.75)$. During the second iteration, the algorithm forms the clusters shown in the second row of Table 2.5 and in Figure 2.7(c), and the sum of squares error value associated with this clustering drops to $1\,437.97$. The cluster centroids are updated *as* (83.67, 69.16) *and* (12.50, 41.67)*, and the clusters of the previous iteration remain unchanged, as shown in the third row of Table 2.5 and in Figure 2.7(d). At this point a minimum sum of squares error value of* 1 097.83 *is reached, and the algorithm terminates.* □



(a) Data set

(b) Iteration 1 clusters

(c) Iteration 2 clusters

(d) Iteration 3 clusters

FIGURE 2.7: *The k-means algorithm applied to the data set in Table 2.4:* ● = *data clustered into cluster 1,* ▲ = *data clustered into cluster 2,* + = *cluster centroid.*

Iteration	Clusters
1	$\{1,2,3,4,5,6,9,11\}$, $\{7,8,10,12\}$
2	$\{1,2,4,5,6,9\}$, $\{3,7,8,10,11,12\}$
3	$\{1,2,4,5,6,9\}$, $\{3,7,8,10,11,12\}$

TABLE 2.5: *Clusters formed at each iteration for the data set in Table 2.4.*

Numerous improvements have been suggested for the standard k-means clustering algorithm described above. These improvements include the addition of a priority measure for cluster selection [69], the inclusion of a Minkowski distance metric in the clustering criterion [50], and the introduction of a number of initialisation procedures [4, 13, 32]. The popularity of the k-means algorithm may be ascribed to its ease of implementation as well as its linear time complexity. A major drawback of the algorithm, however, is that the quality of the solutions returned by the algorithm depends sensitively on the initial partition, as mentioned above. There seems to be no universally applicable, efficient method for computing initial partitions that consistently results in high-quality clustering solutions [227].

2.4 Single-objective optimisation

This section contains descriptions of two popular single-objective optimisation techniques applied later in this dissertation. The section, however, opens in §2.4.1 with a brief overview of the landscape of available solution techniques for single-objective optimisation problems, including exact, heuristic and metaheuristic techniques. This is followed by discussions on the method of PHC in §2.4.2 and the method of SA in §2.4.3.

2.4.1 A classification of single-objective optimisation techniques

The field of single-objective optimisation is a large and active research field. The aim in this section is merely to provide the reader with a very high-level overview of the different methods and approaches that are available within this field.

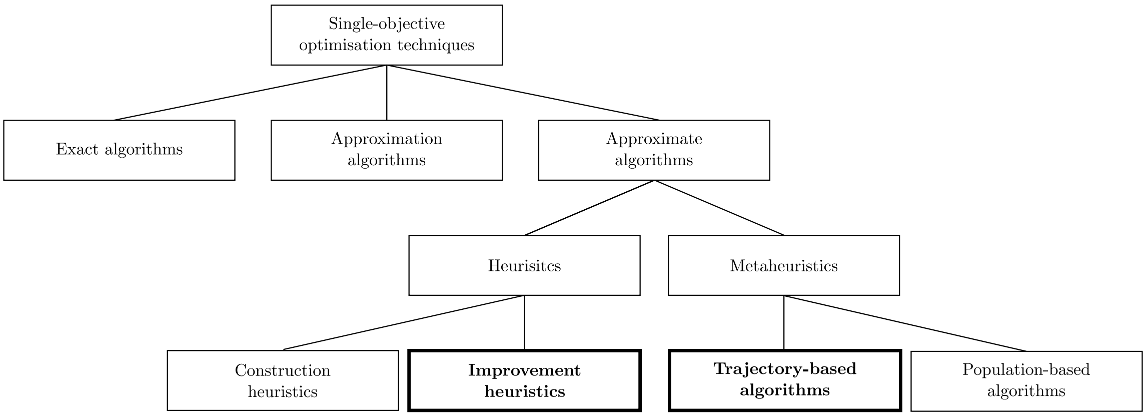


FIGURE 2.8: *A classification of single-objective optimisation techniques.*

Single-objective optimisation techniques may broadly be classified into two classes, namely *exact* methods and *approximate* methods, as illustrated in Figure 2.8. Whereas exact methods yield globally optimal solutions to optimisation problem instances, they are often associated with considerable computational burdens, especially in the case of solving large problem instances. Furthermore, there are not many classes of optimisation problems for which exact solution techniques are available. These techniques are typically applicable to problems of a very specific structure only. For example, the celebrated simplex algorithm [47] and the well-known branch-and-bound method [127] are two exact optimisation algorithms, but are only applicable to linear programming problems and linear integer programming problems, respectively. While the simplex algorithm has a polynomial average-case time complexity, its worst-case time complexity is exponential [175]. The worst-case time complexity of the branch-and-bound method is similarly exponential [224]. Exact algorithms are, in fact, usually only available for problems in which both the objective function and the constraints are linear, or, if they are nonlinear, satisfy very strict regularity constraints (such as being quadratic, or convex and twice differentiable, for example). An example of an exact method that is applicable to optimisation problems with equality constraints in which both the objective function and the constraint functions are continuous and differentiable is the method of Lagrange multipliers.

In cases where the objective function and/or constraint functions are continuous but are not differentiable, or are discrete, the only recourse is often to resort to non-exact solution techniques. As the name suggests, such techniques usually do not yield globally optimal solutions, but instead deliver solutions that are hopefully close to optimal, and do so in a fraction of the computation time required by exact algorithms. When a theoretical guarantee is available for an approximate technique in terms of the closeness of the results that it yields relative to the optimal solution, then the method is called an *approximation algorithm* [163]. The guarantee is often specified in the form of returning a result that is no worse than some multiple ϵ of the optimal objective function value (in the case of a minimisation problem), in which case the algorithm is known as an *ϵ-approximation algorithm*. Christofides [39], for example, designed an $\frac{3}{2}$-approximation algorithm for the celebrated *travelling salesperson problem*[5].

If no guarantee of closeness to optimality is associated with a non-exact optimisation technique, then the technique is referred to as an approximate algorithm [221]. The class of approximate algorithms consists of the subclasses of *heuristics*[6] and *metaheuristics*[7]. According to Rardin [174], there are two fundamental types of heuristics: *construction heuristics* and *improvement heuristics*. Adopting some rule of thumb, the former type of heuristic iteratively builds up a candidate solution to an optimisation problem, starting from an empty solution structure. At any point during execution of the algorithm, a partial solution is available to the optimisation problem at hand — only upon termination of the algorithm does a complete candidate solution emerge. The nature of a construction heuristic is therefore dictated by the fundamental characteristics of the optimisation problem under consideration. The well-known *nearest neighbour heuristic* and *cheapest insertion heuristic* are examples of construction heuristics for the travelling salesperson problem [224]. An improvement heuristic, on the other hand, is initialised with a full candidate solution to the optimisation problem at hand, and the algorithm then iteratively attempts to improve the quality of the solution as quantified by the objective function [100]. An example of an iterative heuristic is the well-known PHC algorithm [181].

A metaheuristic is a flexible, high-level procedure for finding, generating, or selecting suitable heuristics (partial search algorithms) that may provide solutions of sufficient quality to optimisation problems. Metaheuristics often sample from a set of solutions that is too large to enumerate exhaustively and typically employ some form of stochasticity. Metaheuristics are usually based on very few underlying assumptions about the optimisation problem being considered, which typically results in their applicability to a large variety of problem types [94]. Like heuristics, metaheuristics also do not necessarily yield globally optimal solutions to optimisation problems[8]. Metaheuristics are traditionally classified as being either *trajectory-based* or *population-based*. A trajectory-based metaheuristic maintains a single current candidate solution throughout its (typically iterative) execution. This class of metaheuristics includes the method of *tabu search* [73] and the method of SA [120]. Population-based metaheuristics, on the other hand, maintain an entire current population of candidate solutions throughout their iterative execution, and this class of metaheuristics includes *genetic algorithms* [75], *ant colony optimisation* [56] and *particle swarm optimisation* [115].

Two (single-objective) optimisation techniques are employed later in this dissertation. These are a parallelisation of the hill climbing algorithm (which is a member of the class of improvement

[5] The *travelling salesperson problem* is the problem of finding a shortest (closed) tour visiting each of a number of cities.

[6] Derived from the Greek word *heuristikein*, which means “to find” (although the phrase “to search” would perhaps have been a more appropriate choice).

[7] The prefix “meta” is the Greek for *over*, *more than* or *beyond*. The word *metaheuristic* therefore means *beyond finding* if translated literally.

[8] The exception is the method of SA, for which a theoretical guarantee is available [177].

heuristics) and the method of SA (which is a member of the class of trajectory-based metaheuristics). The remainder of this section is therefore devoted to detailed discussions on these two techniques.

2.4.2 The method of hill climbing

Hill climbing, also sometimes referred to as *improving local search* [174], is a heuristic which belongs to the family of *local search* optimisation techniques (all of which are improvement heuristics). It is an iterative search procedure that starts with an arbitrary selected feasible solution to a (typically hard) single-objective optimisation problem, and then repeatedly attempts to find a better solution to the problem by incrementally perturbing a single element or component of the solution. If the perturbation produces a higher-quality solution, it is accepted as the next solution in a sequence of iteratively visited solutions, and this process is repeated until no further improvements can be made, at which point the search terminates, returning the last solution considered. A pseudo-code description of the process is given in Algorithm 2.2.

Algorithm 2.2: Hill climbing (for a maximisation problem)

```
Input  : An objective function $f$ and an initial solution $\boldsymbol{x}^{(0)}$.
Output: A local maximum of $f$.
$stop \leftarrow false$
$\boldsymbol{x} \leftarrow \boldsymbol{x}^{(0)}$
while $stop = false$ do
    $L(\boldsymbol{x}) \leftarrow$ {neighbours of $\boldsymbol{x}$}
    $best \leftarrow -\infty$
    for each $\boldsymbol{x}' \in L(\boldsymbol{x})$ do
        if $f(\boldsymbol{x}) > best$ then
            $\boldsymbol{x} \leftarrow \boldsymbol{x}'$
            $best \leftarrow f(\boldsymbol{x}')$
    if $best \leq f(\boldsymbol{x})$ then
        stop $\leftarrow$ true
output $\boldsymbol{x}$
```

The method of perturbation employed depends on the characteristics of the optimisation problem at hand and a judicious choice of this perturbation operator is crucial in the successful implementation of a hill climbing algorithm. The method of perturbation is usually defined in terms of a specific type of neighbourhood associated with each candidate solution to the optimisation problem considered. In *simple hill climbing*, the closest neighbouring solution to the current solution is selected as the new current solution during each iteration according to a suitable distance measure, and ties are broken by selecting the smallest numbered closest solution (according to an ordering decided upon *a priori*) in the case of the existence of multiple closest neighbouring solutions.

In *steepest ascent hill climbing*, on the other hand, all closest neighbouring solutions of the current solution are evaluated and the neighbouring solution that performs best in terms of the objective function is selected as the new current solution. In problem settings where the neighbourhood of a current solution is very large, it may not be desirable or feasible to examine *all* the neighbours of a current solution. This gives rise to another variation on the basic idea of hill climbing, called *stochastic hill climbing*, in which a neighbouring solution is selected randomly after which a decision is made as to whether or not to accept the neighbour as the new current solution. This decision is usually based on the degree of improvement that the neighbouring solution brings about in terms of the objective function relative to the objective function value of the current solution. If the neighbouring solution is not accepted as the new current solution, then the random selection procedure is repeated, either until a neighbouring solution is selected or until a maximum number trials have taken place, at which point the algorithm terminates.

Ridges and plateaus in the objective function may potentially pose significant challenges to hill climbing algorithms. Because a hill climbing search usually only involves adjustment of a single current solution element in vector representations of candidate solutions during each iteration of the search, each step typically involves moving in an axis-aligned direction within the solution space. If the objective function exhibits a narrow ridge that is not aligned with any axis, then the search can only ascend the ridge in a zig-zagging fashion. If the sides of the ridge are steep, then the algorithm may be forced to take many very tiny steps during each iteration as it zig-zags to ever-better candidate solutions. It may therefore take unreasonably long to ascend the ridge fully.

Another problem that sometimes hampers efficient hill climbing progression is the presence of a plateau in the objective function (*i.e.* a flat or relatively flat portion of the objective function hypersurface). In such an area of the solution space, the objective function value associated with the current solution may be numerically indistinguishable from the value associated with its neighbouring solutions. This situation may result in a general inability of the hill climbing algorithm to determine in which direction it should proceed, causing it to wander in directions that do not lead to significant improvement.

If carried out for sufficiently many iterations, the method of hill climbing returns an optimal solution in the case of a convex optimisation problem — that is, a problem in which the objective function is concave[9] (in the case of maximisation) or convex[10] (in the case of minimisation) over the entire feasible region, which itself is required to be a convex set[11]. A prime example of a hill climbing algorithm that returns an exact solution is the celebrated simplex algorithm in linear programming [47]. If applied in the context of non-convex optimisation problems, however, hill climbing is not guaranteed to return a globally optimal solution, instead often becoming trapped at a local optimum [174].

A number of mechanisms have been proposed to allow a hill climbing algorithm to escape from local optima when applied to non-convex optimisation problems. The simplest of these is to launch a number of independent hill climbing searches — from different, randomly selected initial solutions. If this procedure is carried out sequentially, the approach is called *multi-start hill climbing*, whereas if the various hill climbing search instances are carried out in parallel, then the approach is known as PHC [174]. PHC is adopted later in this dissertation for the purpose of maximisation of a DM's value function.

The working of the PHC algorithm is illustrated in the following example.

Example 2.8 *Consider the function*

$$f(x,y) = \exp\left[-\tfrac{5^2}{2}\left(\left(\tfrac{x-63}{255}\right)^2 + \left(\tfrac{y-63}{255}\right)^2\right)\right] + \tfrac{2}{3}\exp\left[-\tfrac{20^2}{18}\left(\left(\tfrac{x-191}{255}\right)^2 + \left(\tfrac{y-191}{255}\right)^2\right)\right] \qquad (2.16)$$

[9] A function is *concave* if the straight line joining any two points on the function hypersurface never exceeds the hypersurface.

[10] A function is *convex* if the function hypersurface never exceeds the straight line joining any two points on the function hypersurface.

[11] A set is *convex* if the straight line between two arbitrary elements of the set remains entirely within the set.

on the square domain $(x, y) \in [0, 255] \times [0, 255]$*, depicted in Figure 2.9. This function achieves a global maximum value of* 1 *at* $(x, y) = (63, 63)$ *and a local maximum of* $\frac{2}{3}$ *at* $(x, y) = (191, 191)$*. A contour plot of the same function is shown in Figure 2.10. A regular mesh has been superimposed on the function domain in the latter figure.*

Suppose the PHC were to be employed to find the maximum of this function, limiting candidate solutions considered during the search process to mesh points. Suppose, furthermore, that a neighbourhood of four adjacent grid point candidate solutions is defined for any such candidate solution, as illustrated for the point × *in Figure 2.10. The trajectories followed by a double-threaded PCH, starting from the points* A *and* C*, respectively, are indicated in Figure 2.10. The search starting at* A *returns the approximate solution at* B*, while that starting at* C *returns the approximate solution at* D*. Clearly, the search starting at* A *yields the superior approximation of the true global maximum* $(63, 63)$*.* □

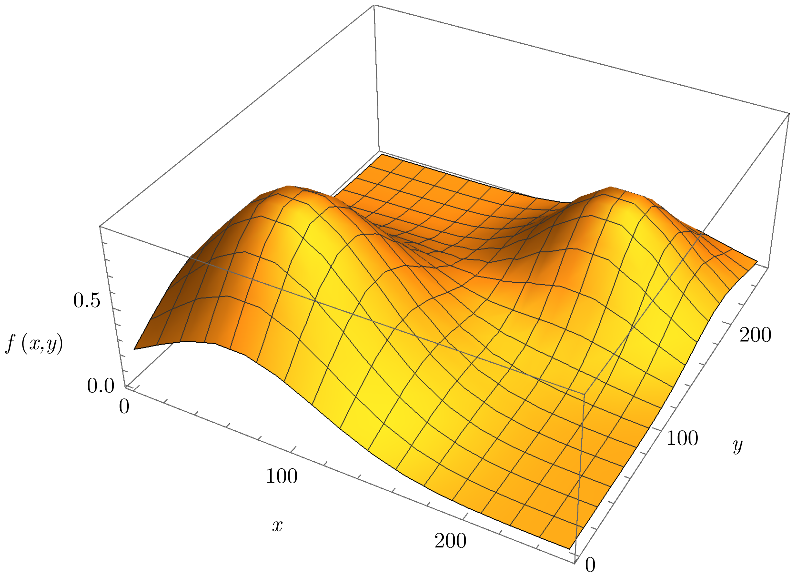


FIGURE 2.9: *The function* $f(x, y)$ *in (2.16), exhibiting a global maximum value of 1 at* $(x, y) = (63, 63)$ *and a local maximum value of* $\frac{2}{3}$ *at* $(x, y) = (191, 191)$*.*

2.4.3 The method of simulated annealing

SA is a metaheuristic optimisation technique proposed in 1983 by Kirkpatrick *et al.* [119] which mimics the physical annealing process by which solids are strengthened in metallurgy. This process involves applying heat treatment to a metal with the objective of altering its molecular structure. The metal is heated above its recrystallisation temperature, after which it is allowed to cool down slowly. This heating process causes the atoms of the material to become excited, vibrating stochastically into higher energy states [21]. As the metal is then left to cool down slowly, its atoms vibrate less vigorously until they settle down into a low-energy state. If the cooling process is managed appropriately, the atoms settle into lower energy states than their initial energy states with high probability.

When designed properly, an SA algorithm is capable of controlling the cycling phenomenon which typically results from the acceptance of non-improving moves in the local search paradigm of hill climbing according to probabilities tested by pseudo-random numbers. The method is also trajectory-based and a very basic pseudo-code description of the core process of SA is given in Algorithm 2.3 (for a minimisation problem).

Algorithm 2.3: Simulated annealing (for a minimisation problem)

```
    Input  : An initial candidate solution $\boldsymbol{x}^{(0)}$, a maximum allowable number of poor epochs
             $poorepochs$, a maximum number $maxattempt$ of move attempts per epoch, an
             initial temperature $q$, a cooling schedule, and a reheating schedule.
    Output : An approximately optimal solution $\hat{\boldsymbol{x}}$.
 1  $t \leftarrow 0$, $epoch \leftarrow 1$, $\hat{\boldsymbol{x}} \leftarrow \boldsymbol{x}^{(0)}$
 2  $poorepochs \leftarrow 0$, $attempts \leftarrow 0$, $accept \leftarrow 0$
 3  if $poorepochs > maxepochs$ then
 4      output $\hat{\boldsymbol{x}}$ and stop
 5  if $attempts > maxattempts$ then
 6      raise temperature $q$ according to the reheating schedule
 7      if $accepts = 0$ then
 8          $poorepochs \leftarrow poorepochs + 1$
 9      $epoch \leftarrow epoch + 1$, $attempts \leftarrow 0$, $accepts \leftarrow 0$
10      return to Step 3
11  if $accepts > maxattempts$ then
12      lower temperature $q$ according to cooling schedule
13      $epoch \leftarrow epoch + 1$, $attempts \leftarrow 0$, $accepts \leftarrow 0$
14  Generate a neighbouring solution $\boldsymbol{x}'$ of $\boldsymbol{x}^{(t)}$
15  if $rand(0,1) > \min\{1, \exp(-\Delta obj/q)\}$ then
16      $t \leftarrow t + 1$, $attempts \leftarrow attempts + 1$
17      return to Step 4
18  $\boldsymbol{x}(t+1) \leftarrow \boldsymbol{x}'$, $accepts \leftarrow accepts + 1$
19  if $\boldsymbol{x}'$ is superior to $\hat{\boldsymbol{x}}$ then
20      $\hat{\boldsymbol{x}} \leftarrow \boldsymbol{x}'$
21  $t \leftarrow t + 1$
22  return to Step 9
```

SA requires as input an initial candidate solution $\boldsymbol{x}^{(0)}$, an upper bound on the number of so-called poor epochs that may be carried out during the search process, a maximum number of move attempts allowed per epoch, an initial temperature q, a cooling schedule and a reheating schedule. The initial solution is stored as the incumbent solution at initialisation. A neighbouring solution $\boldsymbol{x}^{(t+1)}$ is generated within the neighbourhood of the current solution $\boldsymbol{x}^{(t)}$ during iteration t of the algorithm. This perturbation is the result of applying a so-called *neighbourhood move operator*. This move operator usually involves changing a single aspect, element or component of $\boldsymbol{x}^{(t)}$ according to a set of possible moves. The move set is usually specified *a priori* and is problem-specific.

A move that results in an improvement of the objective function value associated with $\boldsymbol{x}^{(t)}$ is always accepted, while a move that is unable to achieve such an improvement is accepted according to the so-called *Metropolis rule* [29], *i.e.* with probability

$$\exp(-\Delta_{obj}/q_i),$$

where Δ_{obj} denotes the change in the objective function value when moving from the current solution $\boldsymbol{x}^{(t)}$ to the randomly selected neighbouring solution and where q_i denotes the temperature during stage or epoch i of the SA search. This temperature controls the randomness of the search. In the case where the neighbouring solution is accepted, it becomes the new current solution $\boldsymbol{x}^{(t+1)}$ during the next iteration of the search. If, however, the neighbouring solution is rejected, the move operator is re-applied to $\boldsymbol{x}^{(t)}$, repeating this process until acceptance occurs. If the objective function value associated with $\boldsymbol{x}^{(t+1)}$ is superior to that of the current incumbent $\hat{\boldsymbol{x}}$, then $\boldsymbol{x}^{(t+1)}$ is taken as the new incumbent.

For large values of the temperature q_i, the majority of randomly generated neighbouring solutions of current solutions are accepted — this provides a mechanism allowing the search to escape when it becomes trapped at a local minimum. When the temperature q_i is small, on the other hand, only new worsening solutions that result in small degradations in objective function value are typically accepted. The SA search process is therefore normally initialised with a large value of q_0 in order to allow for sufficient exploration of the solution space during the early stages of the search.

According to Busetti [27], the initial temperature q_0 should be chosen so that approximately 80% of all non-improving moves are accepted early on during the search. One way of selecting such an initial temperature is to conduct a trial search during which all non-improving moves are accepted. An appropriate estimate of the initial temperature is then $q_0 = \Delta^+/\ln 0.8$, where Δ^+ denotes the mean change in objective function value resulting from accepting non-improving moves during the trial search.

The temperature q_t is typically kept constant during a number of consecutive search iterations. A collection of successive iterations during which the temperature remains constant is known as a search *epoch*. The length of such an epoch is usually not fixed — it rather has a maximum number of iterations associated with it. The lengths of the various search epochs may hence differ from one another. The length of search epoch i is typically modelled as a Markov chain of length L_i. According to Busetti [27], this value should ideally be customised to the optimisation problem under consideration rather than being viewed as a function of i. This may be achieved by requiring a pre-specified minimum number of move acceptances during any epoch before lowering the temperature and initiating the following epoch. Let $A_{\min}$ denote this pre-specified number of move acceptances. As the temperature q_i decreases towards zero as i increases, non-improving moves are accepted with decreasing probability, resulting in the number of trials expected before accepting $A_{\min}$ moves growing unbounded as the search progresses (irrespective of the value of $A_{\min}$). In order to remedy this situation, the epoch may be terminated once L moves have been attempted (by increasing the temperature) or once $A_{\min}$ moves have been accepted (by lowering the temperature), where $L > A_{\min}$. According to Dreo *et al.* [57], a sensible rule of thumb is to set $L = 100$ and $A_{\min} = 12N$, where N is a measure of the number of degrees of freedom related to the optimisation problem at hand.

Cooling and/or reheating may occur multiple times during the search according to the scheme outlined above. While cooling is aimed at making it harder to accept worsening solutions (in a bid to promote exploitation of promising areas of the solution space), reheating is aimed at making it easier to accept worsening solutions (in a bid to promote exploration of hitherto unknown areas of the solution space by facilitating escape from local optima). When the algorithm terminates, the incumbent solution $\hat{\boldsymbol{x}}$ is returned as output.

In order to enhance the search efficiency of the SA process, a tailor-made set of input parameters (including the initial temperature, the termination criterion, the cooling schedule, the heating schedule and the epoch termination criterion) is required for every optimisation problem instance.

Typical cooling schedules include the well-known geometric schedule, a linear schedule and various adaptive schedules [1, 21, 101, 217]. Of these cooling schedules, the geometric schedule is undoubtedly the most commonly employed. According to the geometric cooling schedule, each epoch temperature is reduced by a constant factor α, called the *cooling parameter* (which ranges between 0 and 1). The temperature during epoch $i+1$ is therefore

$$q_{i+1} = \alpha q_i, \qquad i = 0, 1, 2, \ldots \tag{2.17}$$

Vigeh [217] claimed that the best results are generally found when the cooling parameter α is chosen between 0.8 and 0.99.

Reheating takes place in a similar fashion, except that the temperature is increased by some factor when reheating. According to a geometric reheating schedule, the temperature at epoch $i+1$ is

$$q_{i+1} = \beta q_i, \qquad i = 0, 1, 2, \ldots, \tag{2.18}$$

where $\beta > 1$ is known as the *reheating parameter.*

The working of the method of SA is illustrated in the following example.

Example 2.9 *Consider again the function*

$$f(x,y) = \exp\left[-\tfrac{5^2}{2}\left(\left(\tfrac{x-63}{255}\right)^2 + \left(\tfrac{y-63}{255}\right)^2\right)\right] + \tfrac{2}{3}\exp\left[-\tfrac{20^2}{18}\left(\left(\tfrac{x-191}{255}\right)^2 + \left(\tfrac{y-191}{255}\right)^2\right)\right]$$

of Example 2.8, depicted in Figure 2.9, which exhibits a global maximum value $f(63, 63) = 1$ *and a local maximum function value* $f(191, 191) = \frac{2}{3}$. *Suppose the method of SA has to be used to maximise* $f(x, y)$ *on the square grid domain* $\mathcal{D} = \{0, 1, \ldots, 255\} \times \{0, 1, \ldots, 255\}$ *of* 65 536 *integer coordinates. Suppose the neighbourhood of a candidate solution is defined as the* 35×35 *sub-grid of points in* $\mathcal{D}$ *centred around the candidate solution.*

Moreover, suppose the cooling parameter in (2.17) is taken as $\alpha = 0.95$ *and that the reheating parameter in (2.18) is taken as* $\beta = 1.25$. *Upon implementing the method proposed by Busetti [27], the initial temperature* $q_0 = 0.208$ *was obtained from a trial search of 100 moves.*

Suppose furthermore that a search epoch is terminated upon executing $L = 7$ *attempted, but unaccepted, moves (after which reheating occurs) or upon accepting* $A_{min} = 1$ *moves (after which cooling occurs). These values were obtained by adopting the suggestion of Dreo et al. [57], as described above, taking the measure of the number of degrees of freedom as* $N = 65\,536/10^6$.

Suppose finally that the initial solution is chosen as $(196, 233)$ *and that the termination criterion is to stop the search when reheating is required for a second time. Adopting these parameter values, the SA search progressed as shown in Figure 2.11, terminating after 83 iterations. The temperature profile and the values of the function* $f(x, y)$ *in (2.9) are shown in Figure 2.12 as the search progressed. One instance of reheating occurred — at Iteration 67 — and when a second reheat was required, the algorithm terminated at Iteration 83. The temperature at Iteration 67 was already so small (*0.0067*) that the reheat (to a temperature of* 0.0084*) is not visible to the human eye in Figure 2.12. Notice how the algorithm deftly sidesteps the local maximum at* $(191, 191)$ *by accepting worsening solutions while the temperature is still relatively high, but then zooms in on the global maximum at* $(63, 63)$ *as the temperature approaches a frozen state. The incumbent returned by the SA algorithm is the global maximum* $(x, y) = (63, 63)$. □

2.5 Statistical tests for distinguishing between sample means

Consider N samples of real numbers, each containing M observations, and suppose the following question must be answered: *Which pairs of sample means may be considered to be statistically different at a significance level of α?* A methodology from the realm of inferential statistics for

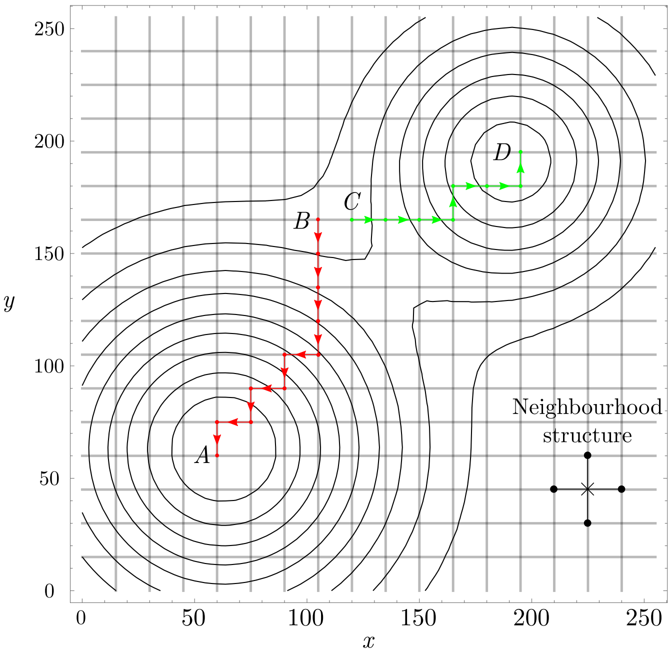


FIGURE 2.10: *A double-threaded PHC search for the maximum of the function $f(x,y)$ in (2.16), starting at A (indicated in red) and at C (indicated in green), respectively, superimposed on a contour plot of $f(x,y)$.*

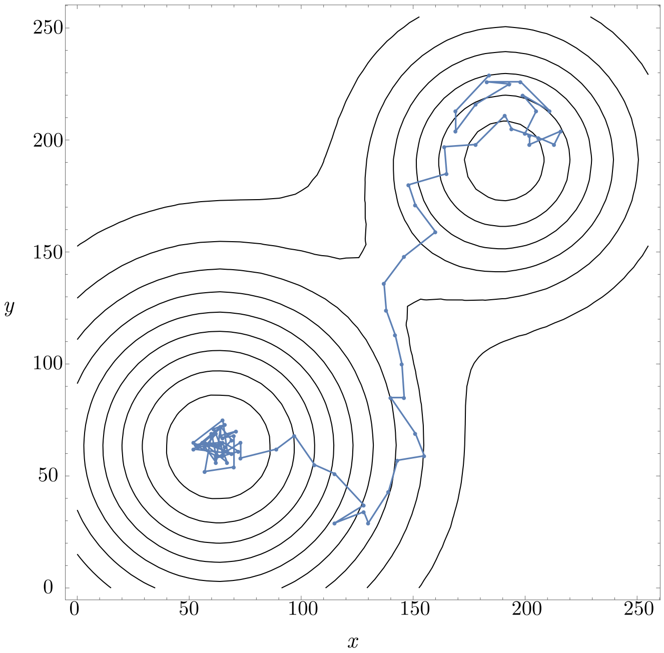


FIGURE 2.11: *Progression of the SA search described in Example 2.9 for the maximum of the function $f(x,y)$ in (2.16), starting at $(x,y) = (196,233)$, superimposed on a contour plot of $f(x,y)$.*

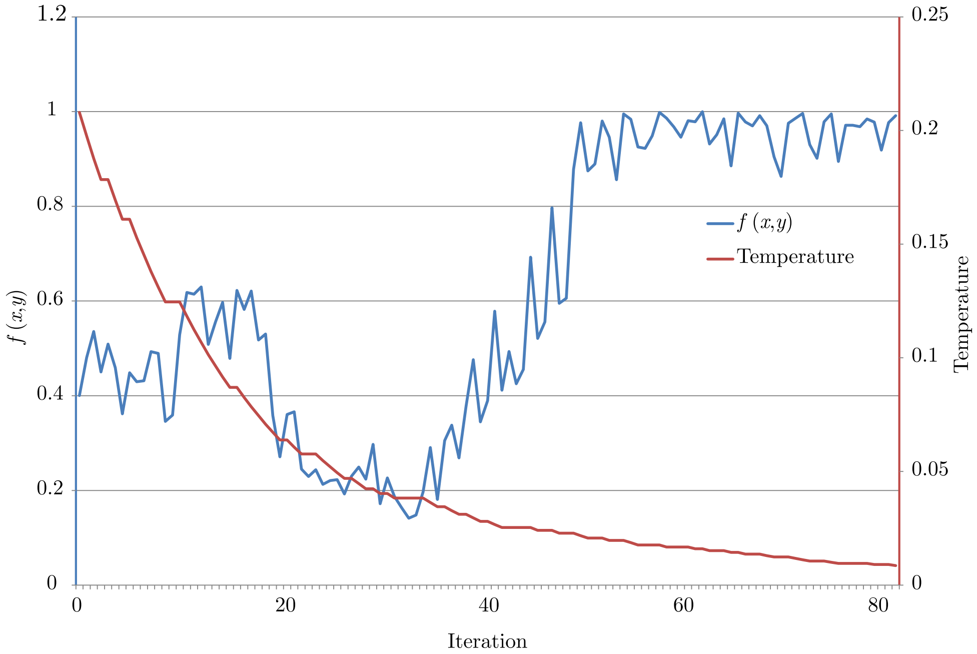


Figure 2.12: *Temperature profile as the SA search of Example 2.9 progressed, together with the values of the function $f(x, y)$ in (2.16) at each iteration.*

answering this question is described in this section for the case where the samples all contain the same number of observations and each sample is drawn from a normal distribution.

Denote the i-th sample by $x_1^{(i)}, \ldots, x_M^{(i)}$ for all $i \in \{1, \ldots, N\}$. Furthermore, let $\bar{x}^{(i)}$ be the mean of the i-th sample and let s_i be its standard deviation. Finally, let $\bar{x}$ denote the mean of all the sample means $\bar{x}^{(1)}, \ldots, \bar{x}^{(N)}$.

2.5.1 The ANOVA test

An ANOVA may be carried out to determine whether there are statistical differences between the sample means at a significance level of α [80]. In this case, the null-hypothesis H_0 is taken as *there are no significant differences between the means of any of the samples*, while the alternative hypothesis H_1 is that *there are significant differences between at least two of the sample means.* An ANOVA utilises both the sum of squares of observations between samples (denoted by S_b) and the sum of squares of observations within samples (denoted by S_w) to test the null-hypothesis. The test statistic is the ratio S_b/S_w, where

$$S_b = \frac{M}{N-1}\sum_{i=1}^{N}\left(\bar{x}^{(i)} - \bar{x}\right)^2$$

and

$$S_w = \frac{1}{MN-N}\sum_{i=1}^{N}\sum_{j=1}^{M}\left(x_j^{(i)} - \bar{x}^{(i)}\right)^2. \tag{2.19}$$

The test statistic is compared with a critical value $F(d_1, d_2, \alpha)$ of the F-distribution, where $d_1 = N - 1$ denotes the number of degrees of freedom for the numerator, $d_2 = MN - N$ denotes the number of degrees of freedom for the denominator, and α denotes the level of statistical significance as before. The value of $F(N-1, MN-N, \alpha)$ may, for example, be found in [80, Appendix F].

If $S_b/S_w > F(N-1, MN-N, \alpha)$, then the null-hypothesis H_0 may be rejected in favour of the alternative H_1 at an α level of significance (*i.e.* it is concluded that there are significant differences between at least two of the sample means at a $(1-\alpha)$ level of confidence). Otherwise, if H_0 is not rejected in favour of H_1, it is concluded that there are no significant differences between the sample means at a $(1-\alpha)$ level of confidence.

If the ANOVA reveals that there are indeed significant differences between at least two of the sample means, a *post hoc* test is required to determine between which pairs of samples the differences in mean values actually occur. Care should, however, be taken when selecting a suitable *post hoc* test for this purpose, since many *post hoc* tests rely on the assumption of homoscedasticity of the samples (*i.e.* the underlying assumption of homogeneity of the sample variances), which may not, in fact, be the case. Levene's test [136] may be used to test for homoscedasticity of the samples.

2.5.2 The Levene test

In order to carry out Levene's test, the sample data must be transformed by computing the values

$$y_j^{(i)} = \left| x_j^{(i)} - \bar{x}^{(i)} \right|$$

for all $i = 1, \ldots, N$ and all $j = 1, \ldots, M$. Furthermore, let $\bar{y}^{(i)}$ be the mean of the i-th sample of transformed values $y_1^{(i)}, \ldots, y_M^{(i)}$ for all $i = 1, \ldots, N$ and let $\bar{y}$ denote the mean of all the transformed sample means $\bar{y}^{(1)}, \ldots, \bar{y}^{(N)}$.

Then Levene's test is simply an ANOVA carried out in respect of the transformed sample means. The null-hypothesis H_0 now becomes *there are no significant differences between the variances of any of the original samples* (that is, *there are no significant differences between the means of any of the transformed samples* or, in other words, the assumption of homoscedasticity of the samples is valid), while the alternative hypothesis H_1 is that *there are significant differences between at least two of the original sample variances* (that is, *there are significant differences between at least two of the transformed sample means* or the assumption of homoscedasticity of the samples is violated). The test statistic is the ratio S_b'/S_w', where

$$S_b' = \frac{M}{N-1} \sum_{i=1}^{N} \left(\bar{y}^{(i)} - \bar{y} \right)^2$$

and

$$S_w' = \frac{1}{MN-N} \sum_{i=1}^{N} \sum_{j=1}^{M} \left(y_j^{(i)} - \bar{y}^{(i)} \right)^2,$$

and this statistic is again compared with the critical value $F(N-1, MN-N, \alpha)$ of the F-distribution, as described in §2.5.1.

2.5.3 The Fisher's LSD *post hoc test*

Fisher's LSD *post hoc* test is a powerful parametric test for comparing two sample means under the assumption of homoscedasticity of the samples. The test has, however, been criticised due to concerns that it does not sufficiently protect from inflated Type I error rates[12], although this only seems to be the case when the number of samples being compared is more than three [87].

[12] A Type I error is the incorrect rejection of a true null-hypothesis.

According to Kidd [116], however, the test is nevertheless appropriate in large experimental designs involving many *post hoc* comparisons, as long as both practical[13] and statistical significance are taken into account.

For Fisher's LSD test, the null-hypothesis H_0 is that *there is no significant difference between the means $\bar{x}^{(k)}$ and $\bar{x}^{(\ell)}$ of two samples.* At a level of significance α, the statistic of the LSD test is $\left|\bar{x}^{(k)} - \bar{x}^{(\ell)}\right|$, while its critical value is given by $L_\alpha = t_{\alpha/2,d_2}\sqrt{2S_w/M}$, where $t_{\alpha/2,d_2}$ denotes the entry in the two-sided t-distribution table [80, Appendix D] at a level of significance α with $d_2 = MN - N$ degrees of freedom and where S_w is the ANOVA sum of squares within samples value in (2.19).

If $\left|\bar{x}^{(k)} - \bar{x}^{(\ell)}\right| > L_\alpha$, then the null-hypothesis H_0 may be rejected at a significance level of α (*i.e.* there is a significant difference between the two means $\bar{x}^{(k)}$ and $\bar{x}^{(\ell)}$ at a $(1-\alpha)$ level of confidence). Otherwise, the means cannot be considered different at an α level of significance.

2.5.4 The Games-Howell *post hoc test*

The Games-Howell *post hoc* test [99, 98] is a non-parametric test for detecting statistical differences between sample means that is applicable in the case of a violation of the assumption of homoscedasticity of the samples. This test has been described as one of the most robust modern methods of *post hoc* testing, being more conservative than the majority of the other *post hoc* tests available in the literature. The null-hypothesis H_0 is again that *there is no significant difference between the means $\bar{x}^{(k)}$ and $\bar{x}^{(\ell)}$ of two samples*, and the test statistic is again $\left|\bar{x}^{(k)} - \bar{x}^{(\ell)}\right|$. The critical value is taken from studentised range distribution [80, Appendix G], denoted by $q_{\sigma(k,\ell),d(k,\ell),\alpha}$, where

$$\sigma(k,l) = \sqrt{\frac{s_k^2 + s_\ell^2}{2M}}$$

is the standard error, $d(k,\ell)$ denotes the degrees of freedom, calculated here as

$$d(k,\ell) = \frac{M-1}{(s_k^2/M)^2 + (s_\ell^2/M)^2}\left(\frac{s_k^2 + s_\ell^2}{M}\right)^2$$

using Welch's correction, and α is again the level of statistical significance.

If $\left|\bar{x}^{(k)} - \bar{x}^{(\ell)}\right| > q_{\sigma(k,\ell),d(k,\ell),\alpha}$, then the null-hypothesis H_0 may be rejected at a significance level of α (*i.e.* there is a significant difference between the two means $\bar{x}^{(k)}$ and $\bar{x}^{(\ell)}$ at a $(1-\alpha)$ level of confidence). Otherwise, the means cannot be considered different at an α level of significance.

2.5.5 p-Values

The method of *fixed significance level* testing is a common approach to reporting on the results of an hypothesis test, by demonstrating whether or not a null-hypothesis H_0 should be rejected at a declared level of significance (α). The so-called p-value denotes the probability, when employed in fixed significance level testing, that the test statistic will take on a value at least as extreme as the observed value in the case that the null-hypothesis is true. This means that the p-value is the smallest level of significance which would lead to rejection of the null-hypothesis [154]. To demonstrate the calculations involved in determining a p-value, consider the two-sided hypothesis in the Fisher LSD test

[13] Practical significance here involves evaluating whether statistically significant differences are large enough to be of any value in a practical sense within the context of some application.

$$H_0 : \left|\bar{x}^{(k)} - \bar{x}^{(\ell)}\right| = 0, \qquad H_1 : \left|\bar{x}^{(k)} - \bar{x}^{(\ell)}\right| \neq 0. \tag{2.20}$$

The p-value is then calculated as

$$1 - P\left(-\frac{\left|\bar{x}^{(k)} - \bar{x}^{(\ell)}\right|}{\sqrt{2S_w/M}} < t_{\alpha/2,d_2} < \frac{\left|\bar{x}^{(k)} - \bar{x}^{(\ell)}\right|}{\sqrt{2S_w/M}}\right). \tag{2.21}$$

After the p-value has been calculated, it may be compared with the predefined significance level α from which a decision can be made. If the p-value is smaller than α, the null hypothesis H_0 may be rejected, while H_0 may not be rejected in the case where the p-value exceeds α. The p-value therefore provides a measure of risk that an incorrect decision regarding the null-hypothesis has been made, because the p-value indicates the probability that the null-hypothesis is wrongly rejected (*i.e.* it represents the probability of making a so-called Type I error) [154]. p-Values may similarly be computed for the ANOVA, Levene and Games-Howell tests, instead of using the appropriate probability distributions mentioned in (2.21), in each case.

2.6 Chapter summary

This chapter contained a review of the literature of a number of disparate mathematical prerequisites aimed at facilitating a thorough understanding of the remaining material presented in this dissertation. A total of six areas of research were touched upon in this review. The first was a brief background in §2.1 on the notions of eigenvalues and eigenvectors of square matrices, and how these are computed. This background material was required in the discussion of the second topic of review, namely the graph theoretic modelling of sports tournaments presented in §2.2. While the quest of determining a winner in a round-robin sports tournament and the pursuit of providing decision support in respect of solving SPBP instances are seemingly very different from an application point of view, they are mathematically quite similar. This similarity was exploited, thereby laying the foundation for a discussion in the following chapter on the problem of DM value function construction.

Since the notion of data clustering can be very beneficial during the process of alternative reduction when solving SPBP instances, the main paradigms of data clustering, namely hierarchical clustering and partitional clustering, were reviewed in §2.3. Various algorithms in these two methodological classes were highlighted, as were algorithms in other, more modern clustering paradigms. Attention was also afforded to the important questions of deciding *a priori* on the number of clusters into which a data set should be partitioned, as well as the notion of cluster validation. The k-means clustering algorithm was singled out for a more detailed description, because it is employed later in this dissertation.

The fourth topic of review represented a very large area of research activity, namely the field of (single-objective) optimisation. It was described in §2.4 how algorithms for solving optimisation problems each resides in one of three classes, namely exact algorithms, approximation algorithms and approximate algorithms. Since it is cumbersome to solve the kind of optimisation problems considered later in this dissertation exactly with an approximation guarantee, the emphasis of review was placed on the class of approximate algorithms. The two prevailing subclasses of this class of algorithms, namely heuristics and metaheuristics, were described. The two main types of heuristics (construction heuristics and improvement heuristics) and the two main classes of metaheuristics (trajectory-based algorithms and population-based algorithms) were touched upon. Two particular optimisation methods which are employed in this dissertation were finally singled out for more detailed descriptions. These methods were the method of PHC (an improvement heuristic) and the method of SA (a trajectory-based metaheuristic).

In the final section of the chapter, §2.5, a number of statistical tests for distinguishing between the means of samples drawn from a normal distribution were described. These tests were the ANOVA (for detecting differences between a group of sample means), the Levene test (for homoscedasticity), Fischer's LSD *post hoc* test (for detecting differences between the means of a pair of homoscedastic samples), and the Games-Howell *post hoc* test (for detecting differences between the means of a pair of samples of differing levels of variance). These tests are used in later chapters during the validation process of the DSS put forward in this dissertation.

CHAPTER 3

Preference modelling

Contents

This chapter contains a brief overview of the literature on those aspects of the general research field of preference learning that are relevant to the discourse in this dissertation. The chapter opens in §3.1 with a brief introduction to various basic notions in the area of preference modelling. These notions include an overview and classification of the wide variety of different types of *multi-criteria decision making* (MCDM) problems in the literature, the standard practice of DM preference articulation in the form of the expression of binary relations in respect of decision alternatives, and the concept of a value function as a numerical representation of DM preference. The focus then shifts in §3.2 to detailed descriptions of two of the best-known methods in the literature for value function construction — the AHP, a ratio-based method proposed by Saaty [183] in 1980, and the *Measuring Attractiveness by a Categorical-Based Evaluation Technique* (MACBETH), a difference-based method proposed by Bana e Costa and Vansnick [62] in 1994. A review of modern developments in the area of interactive preference learning then follows in §3.3 before the learning approach adopted in this dissertation is elucidated in §3.3.5. The chapter finally closes with a brief summary of its contents in §3.4.

3.1 Introduction

Classifications of the broad class of *multi-criteria decision making* (MCDM) problems and methods available for solving instances within this class are recounted briefly in §3.1.1 so as to provide a context for the literature review in this chapter and also for the later contributions of this dissertation. Since the theory of binary representations on sets may be employed succinctly to articulate DM preferences, various basic notions within this theory are briefly reviewed in §3.1.2. The notion of a *value function* as a numerical representation of a binary preference relation is finally introduced formally in §3.1.3.

3.1.1 A classification of MCDM problems

There are two major subclasses of MCDM problems in general. The distinction between these two classes is based on whether the solutions to problem instances are defined explicitly or implicitly:

Multiple-criteria evaluation problems. Problems in this subclass admit a finite or a countably infinite number of alternatives which are known explicitly and defined or specified *a priori*. In this case, each alternative is typically represented by its performance in terms of multiple criteria. The problem requires recommending a best alternative or a set of good alternatives to the DM. It is sometimes also required to sort or classify the alternatives in some way. *Sorting* here refers to assigning each alternative to some set within a collection of preference-ordered classes (such as assigning credit-ratings to countries), while *classifying* involves assigning the alternatives to some set within a collection of unordered sets (such as diagnosing patients based on their observed symptoms). The reader is referred to Triantaphyllou [63] for an authoritative discussion of and comparison between problems within this class (as well as methods for solving them).

Multiple-criteria design problems. Also known as the subclass of *multi-objective optimisation problems*, members of this class exhibit the distinguishing feature that the alternatives are not explicitly known *a priori*. An alternative (or solution) is found by solving an appropriate mathematical model. The number of alternatives admitted by problems in this class is either infinite and not countable (in the case of continuous variables) or usually very large if countable (in the case of discrete variables). The reader is referred to Deb [51] for an excellent treatment of this subclass of MCDM problems.

For problems in both of the aforementioned subclasses, preference information is required from the DM in order to be able to distinguish between the relative qualities of solutions. Methods for solving MCDM problems are usually classified according to the timing of obtaining preference information from the DM:

Methods requiring DM preference information at the start. Methods in this class require prior articulation of preferences by the DM and effectively transform the problem into a single-criterion problem. Examples of solution methods for multiple-criteria evaluation problems in this class include methods based on estimating a so-called value function or employing outranking relations, the AHP, MACBETH and certain decision rule-based methods. There are also methods in this class for solving multiple-criteria design problems that are based on the articulation of preferences by means of value function construction, such as goal programming [133]. Once the value function has been constructed, the resulting single-objective optimisation problem is solved for a preferred solution.

Methods requiring DM preference information throughout. Methods in this class are often referred to as interactive methods or methods that require progressive articulation of preferences. There are methods in this class for solving both multiple-criteria evaluation problems [70, 124] and multiple-criteria design problems [204].

The SPBP considered in this dissertation resides within the class of multiple-criteria evaluation problems. Furthermore, while the working of the DSS put forward in this dissertation is based on ratio-based or difference-based DM preference articulation, which is decidedly a member of the subclass of methods that require DM preference elicitation at the start of the solution process, the iterative and interactive nature of the proposed DSS of this dissertation places the system squarely within the class of methods requiring DM preference information specification throughout their working.

3.1.2 Preference articulation in the form of binary relations

There are three prevailing perspectives on preference modelling [22] in multiple-criteria evaluation problems. According to the *normative perspective*, a link is sought between the output of a preference model and the rational behaviour of a DM, whereas preference models aim to be compatible with experimental results according to the *descriptive perspective*. Finally, the quest in preference modelling according to the *prescriptive perspective* is to aid a DM in the process of structuring a difficult or vague decision problem formalisation. The prescriptive perspective is adopted in this dissertation.

The typical setup in discrete preference modelling under the prescriptive perspective involves consideration of a finite or countably infinite set $\mathbb{X}$ of decision alternatives that has has been identified *a priori*, from which one and only one element will eventually be chosen or implemented by any particular DM. The goal is to help the DM to select his or her most preferred alternative in a structured manner [179]. The following standard notational convention is usually adopted in preference articulation:

- The DM preference statement "*alternative* $\boldsymbol{x} \in \mathbb{X}$ *is at least as attractive as alternative* $\boldsymbol{y} \in \mathbb{X}$" is denoted in binary relation form by $\boldsymbol{x} \succeq \boldsymbol{y}$.
- The DM preference statement "*alternative* $\boldsymbol{x} \in \mathbb{X}$ *is strictly preferred over alternative* $\boldsymbol{y} \in \mathbb{X}$" is denoted in binary relation form by $\boldsymbol{x} \succ \boldsymbol{y}$.
- The DM preference statement "*alternatives* $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{X}$ *are indistinguishable from a preference point of view*" is denoted in binary relation form by $\boldsymbol{x} \sim \boldsymbol{y}$.

If $\boldsymbol{x} \succeq \boldsymbol{y}$ or $\boldsymbol{y} \succeq \boldsymbol{x}$ for each pair $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{X}$, then the binary relation $\succeq$ is *complete*. Furthermore, if $\boldsymbol{x} \succeq \boldsymbol{y}$ and $\boldsymbol{y} \succeq \boldsymbol{z}$ together imply that $\boldsymbol{x} \succeq \boldsymbol{z}$ for each triple $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{X}$, then the binary relation $\succeq$ is *transitive*. A binary relation $\succeq$ which is both complete and transitive defines a *weak order*[1] (of preference) on the set $\mathbb{X}$. This desirable property of the relation $\succeq$ does not hold if any one of the conditions of completeness or transitivity is violated [66].

Note that the binary relation $\succeq$ may be thought of as being composed of the two binary relations $\succ$ and $\sim$ in a disjunctive fashion. The strict preference part of the relation $\succeq$, denoted by $\succ$, is *asymmetric* (which means that if $\boldsymbol{x} \succ \boldsymbol{y}$ then $\boldsymbol{y} \not\succ \boldsymbol{x}$ for any pair $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{X}$), *transitive* (which means that if $\boldsymbol{x} \succ \boldsymbol{y}$ and $\boldsymbol{y} \succ \boldsymbol{z}$, then $\boldsymbol{x} \succ \boldsymbol{z}$ for any triple $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{X}$) and also *negatively transitive* (which means that if $\boldsymbol{x} \not\succ \boldsymbol{y}$ and $\boldsymbol{y} \not\succ \boldsymbol{z}$, then $\boldsymbol{x} \not\succ \boldsymbol{z}$ for any triple $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{X}$).

The indifference part of $\succeq$, denoted by $\sim$, furthermore induces so-called equivalence classes in $\mathbb{X}$. An *equivalence class* of $\mathbb{X}$ is a maximal subset $\tilde{\mathbb{X}} \subseteq \mathbb{X}$ such that $\boldsymbol{y} \sim \boldsymbol{x}$ for any pair of elements $\boldsymbol{x}, \boldsymbol{y} \in \tilde{\mathbb{X}}$. The following important result links the aforementioned three binary relations.

Theorem 3.1 ([22]) *If the binary relation $\succeq$ is both complete and transitive, then the binary relation $\succ$ induces a total order*[2] *(of preference) over the equivalence classes of $\mathbb{X}$ brought about by the binary relation $\sim$.*

[1] This means that although any two elements are comparable under the binary relation $\succeq$, it is not necessarily possible to distinguish *in terms of strict preference* between any two elements of $\mathbb{X}$ (*i.e.* certain elements of $\mathbb{X}$ may be indistinguishable in terms of preference).

[2] That is, it is possible to distinguish, in terms of *strict* preference, between any two elements of different equivalence classes of $\mathbb{X}$.

The (desirable) implications of Theorem 3.1 are that:

- The binary relations $\sim$ and $\succeq$ are *exhaustive*. That is, for each pair $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{X}$, at least one of the following holds: $\boldsymbol{x} \sim \boldsymbol{y}$, $\boldsymbol{x} \succeq \boldsymbol{y}$, $\boldsymbol{y} \succeq \boldsymbol{x}$.
- The binary relations $\sim$ and $\succ$ are *exclusive*. That is, for each pair $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{X}$, at most one of the following holds: $\boldsymbol{x} \sim \boldsymbol{y}$, $\boldsymbol{x} \succ \boldsymbol{y}$, $\boldsymbol{y} \succ \boldsymbol{x}$.
- There are no incomparable elements in $\mathbb{X}$.

In particular, the truth table in Table 3.1 holds for the binary relations $\succeq$, $\succ$ and $\sim$.

And	$\boldsymbol{y} \succeq \boldsymbol{x}$	$\boldsymbol{y} \not\succeq \boldsymbol{x}$
$\boldsymbol{x} \succeq \boldsymbol{y}$	$\boldsymbol{x} \sim \boldsymbol{y}$	$\boldsymbol{x} \succ \boldsymbol{y}$
$\boldsymbol{x} \not\succeq \boldsymbol{y}$	$\boldsymbol{y} \succ \boldsymbol{x}$	$\emptyset$

TABLE 3.1: *Truth table for the binary relations $\succeq$, $\succ$ and $\sim$ according to Theorem 3.1.*

3.1.3 The notion of a value function

The standard approach in prescriptive preference modelling is to assume that the DM is able to articulate preferences in terms of the binary relations $\succeq$, $\succ$ and $\sim$ discussed in the previous section and to attempt to construct a function $u : \mathbb{X} \mapsto \mathbb{R}$, called a *value function* which captures the DM's preferences with respect to the alternatives in $\mathbb{X}$ without having to consider all pairs of alternatives explicitly. More specifically, the objective is to construct the value function u so that $u(\boldsymbol{x}) \geq u(\boldsymbol{y})$ if and only if $\boldsymbol{x} \succeq \boldsymbol{y}$ [125]. The existence of such a value function under the assumptions of completeness and transitivity of the binary relation $\succeq$ is guaranteed by the following classical result due to Cantor [30].

Theorem 3.2 ([30]) *Let $\mathbb{X}$ be a finite or countably infinite set and let $\succeq$ be a binary relation on $\mathbb{X}$. Then there exists a real-valued function u on $\mathbb{X}$ such that $u(\boldsymbol{x}) \geq u(\boldsymbol{y})$ whenever $\boldsymbol{x} \succeq \boldsymbol{y}$ for all pairs $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{X}$ if and only if $\succeq$ induces a weak order (of preference) on $\mathbb{X}$.*

The value function u whose existence is guaranteed by Theorem 3.2 in the case of a complete and transitive binary relation $\succeq$ on $\mathbb{X}$ as a result of Theorem 3.1 is called a *numerical representation* of $\succeq$ on $\mathbb{X}$ [125]. Such a numerical representation is, of course, not unique, because if u is a numerical representation of $\succeq$ on $\mathbb{X}$, then so is the transformation $f(u)$ for any strictly increasing function $f : \mathbb{R} \mapsto \mathbb{R}$.

It is often costly[3] to compute a numerical representation of a binary relation $\succeq$ on *all* the alternatives in the set $\mathbb{X}$ if $\mathbb{X}$ is large, or impossible if $\mathbb{X}$ is countably infinite. For this reason, analysts typically settle for an estimation of a numerical representation of a binary relation $\succeq$ on $\mathbb{X}$ based on the consideration of a small (often strategically chosen) subset of alternatives in $\mathbb{X}$ only [224]. These numerical representations of the binary relations on $\mathbb{X}$, according to the different decision criteria, are then combined by means of an appropriate aggregation method to arrive at a selection of sample points on the DM's value function. Thereafter, the regions between these sample points are filled in by some form of interpolation or regression to construct a continuous hypersurface estimate of the DM's value function. A final decision recommendation can then be made to the DM based on a maximisation of this value function estimate.

[3] If, for example, there are n elements in the alternative set $\mathbb{X}$, then $\binom{n}{2}$ pairwise preference articulations are required by the DM in order to define the binary relation $\succeq$ on $\mathbb{X}$, as described in §3.2.

3.2 Value function construction

A number of methodologies have been proposed in the literature for value function construction. Two of these methods are described in some detail in this section. The first method is the AHP, discussed in §3.2.1, which is a ratio-based value function construction method dating from 1980. The second method is MACBETH, described in §3.2.2, which is a difference-based value function construction method dating from 1994. The working of these methods is described in general in the form of a methodological overview, after which actual application of the methods is demonstrated by means of detailed numerical examples. Both of these methods are applied in later chapters during the validation of the DSS proposed in this dissertation.

3.2.1 The AHP — A ratio-based approach

The so-called *eigenvector method* (EM) of determining tournament winners described in §2.2 may, in fact, be viewed as a framework for facilitating single-criterion decision support. The decision in this case is deciding a winner for the tournament, and the single decision criterion is merit or sporting skill. Saaty [183] went on to generalise this framework so as to incorporate multiple decision criteria, arriving at what is now known as the AHP. This section is devoted to a discussion on this framework for multi-criteria decision support, which may be incorporated into the DSS put forward later in this dissertation. The working of the AHP is described in detail within the context of the SPBP, after which the notion of DM inconsistency and how this may be measured, is touched on. A detailed numerical example of applying the AHP in a multi-criteria SPBP context is thereafter provided, before focussing the reader's attention on a number of advantages and disadvantages associated with use of the AHP. The section finally closes with the establishment of a terminology convention for the single-criterion special case of the AHP adopted in this dissertation.

Working of the AHP

The AHP may be thought of as a decision support framework involving three fundamental elements, namely a *goal*, a number of selection *criteria* and a variety of decision *alternatives*. The working of the AHP is described in this section within the context of the SPBP in which n alternatives are available for selection and in which the relative suitability of each alternative is measured in terms of m distinct selection criteria, as illustrated graphically in Figure 3.1.

Broadly speaking, application of the AHP consists of the following four steps which are carried out sequentially:

1. **Setting up of problem instance.** In this step, the particular SPBP instance is examined from a practical point of view and the decision elements in Figure 3.1 are established by the DM. This includes formalising the overarching goal of the selection problem, eliciting the selection criteria and establishing the available decision alternatives.

2. **Pairwise comparison of decision elements.** Next, the DM is required to carry out comparisons of the relative attractiveness of the decision elements in pairs. The results of these pairwise comparisons are captured in the form of matrices, much like the pairwise comparison matrices formed in the method of tournament winner selection described in §2.2.5.

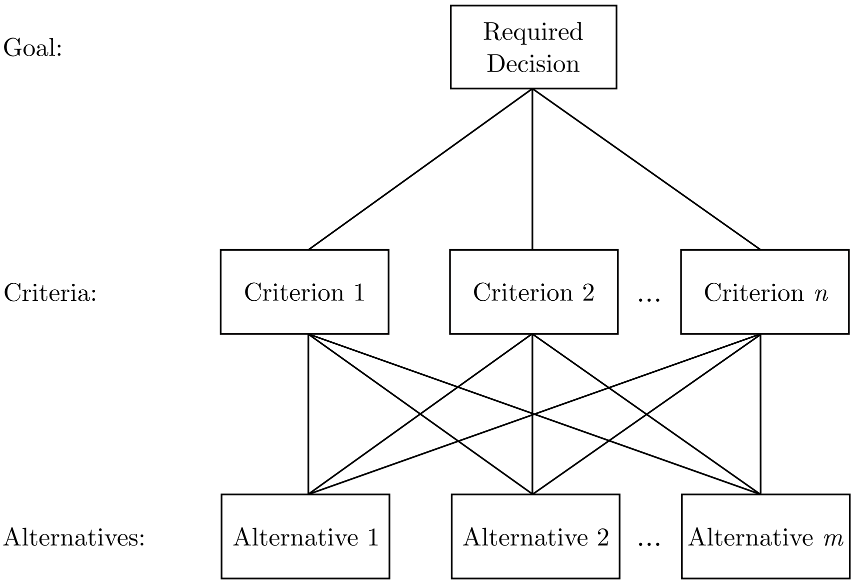


Figure 3.1: *Graphical representation of the three types of elements present in the AHP [185].*

This step is carried out in two parts:

(a) First, the n alternatives are compared and rated relative to one another in pairwise fashion with respect to each criterion separately, according to some judgement scale, such as those in Table 2.2.

(b) Thereafter, the m criteria are compared and rated relative to one another in pairwise fashion, again according to a preferred judgement scale.

3. **Estimation of decision element scores or weights.** The pairwise comparison matrices of Step 2 above are analysed in order to estimate scores for the decision alternatives with respect to each selection criterion and weights for the selection criteria.

4. **Aggregation of overall alternative scores.** The scores associated with the decision alternatives (with respect to each selection criterion *separately*) and the weights associated with the selection criteria in Step 3 above are combined into a single overall score for each decision alternative (with respect to all the selection criteria *combined*).

Whereas Step 1 above is intuitively self-evident, the remaining steps of the AHP are rather technical. The remainder of this section is therefore devoted to a detailed description of how Steps 2–4 above are carried out.

The second step of the AHP involves eliciting pairwise comparisons of the decision elements by the DM. During the pairwise comparison of decision alternatives i and k in Step 2(a) of the AHP, for example, the DM is required to specify that he finds alternative i $a_{ik}^{(j)}$ times as attractive as alternative k in respect of selection criterion j. The value $a_{ik}^{(j)}$ is specified according to a preferred judgement scale, examples of which may be found in Table 2.2. By comparing all pairs of alternatives $i, k \in \{1, \ldots, n\}$ in this fashion, an $n \times n$ pairwise comparison matrix $\boldsymbol{A}^{(j)}$ associated with the n decision alternatives in respect of selection criterion j may be populated as

$$\boldsymbol{A}^{(j)} = \begin{bmatrix} a_{11}^{(j)} & a_{12}^{(j)} & \cdots & a_{1n}^{(j)} \\ a_{21}^{(j)} & a_{22}^{(j)} & \cdots & a_{2n}^{(j)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}^{(j)} & a_{n2}^{(j)} & \cdots & a_{nn}^{(j)} \end{bmatrix}. \tag{3.1}$$

In order to satisfy the most basic consistency requirements, the entries on the main diagonal of $\boldsymbol{A}^{(j)}$ are required to be ones whilst the entries in mirror image positions relative to the main diagonal are required to be reciprocals, *i.e.*

$$a_{ik}^{(j)} = 1 \Big/ a_{ki}^{(j)}, \quad i, k = 1, \ldots, n, \quad j = 1, \ldots, m.$$

Because the entries of the pairwise comparison matrix (3.1) may be interpreted as ratios (the relative attractiveness of alternatives i and k in terms of selection criterion j may, for example, be thought of as occurring in the ratio $a_{ik}^{(j)} : 1$), the AHP is classified as a *ratio-based method.*

During the pairwise comparison of selection criteria j and k in Step 2(b) of the AHP, the DM is similarly required to specify that he finds criterion j b_{jk} times as important as criterion k. By comparing all pairs of criteria $j, k \in \{1, \ldots, m\}$ in this fashion, an $m \times m$ pairwise comparison matrix $\boldsymbol{B}$ associated with the m selection criteria may be populated as

$$\boldsymbol{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & \vdots & & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mm} \end{bmatrix}. \tag{3.2}$$

It is again required for consistency reasons that the entries on the main diagonal of $\boldsymbol{B}$ should be ones whilst the entries in mirror image positions relative to its main diagonal should be reciprocals (*i.e.* $b_{jk} = 1/b_{kj}$, for all $j, k = 1, \ldots, m$). The values b_{jk} may again be specified by the DM according to any of the judgement scales in Table 2.2.

Inspired by the EM of determining tournament winners, as described in §2.2.5, relative scores of the n decision alternatives may be estimated according to selection criterion j during the third step of the AHP by solving the equation

$$\boldsymbol{A}^{(j)}\boldsymbol{s}^{(j)} = \lambda^{(j)}\boldsymbol{s}^{(j)} \tag{3.3}$$

for the eigenvalue $\lambda^{(j)}$ with largest modulus and corresponding (suitably normalised) eigenvector $\boldsymbol{s}^{(j)}$ of the $n \times n$ pairwise comparison matrix $\boldsymbol{A}^{(j)}$. Saaty [183], in fact, showed that (a multiplicative scaling of) the vector of relative scores of the n decision alternatives according to selection criterion j of a DM approaches the vector $\boldsymbol{s}^{(j)}$ in (3.3) as the consistency of the DM increases. Furthermore, for a perfectly consistent DM, $\lambda^{(j)} = n$. It has for this reason been suggested [224] that the level of consistency of a DM should be measured by the closeness of $\lambda^{(j)}$ to n.

While the eigenvalue $\lambda^{(j)}$ of $\boldsymbol{A}^{(j)}$ may, of course, be found by solving the characteristic equation of the matrix, as described in §2.1, substituting the resulting value of $\lambda^{(j)}$ into (3.3) and solving for the eigenvector $\boldsymbol{s}^{(j)}$, Saaty [183] presented an alternative method of approximating the eigenvector $\boldsymbol{s}^{(j)}$. This approximation method is so simple that it may be implemented as a recipe without any requirement of understanding what an eigenvector actually is. The accuracy of the approximation furthermore increases as the consistency of the DM increases (*i.e.* the closer $\lambda^{(j)}$ is to n). The approximation method hinges on normalising the pairwise comparison matrix $\boldsymbol{A}^{(j)}$

by dividing each column of the matrix by its sum to form the matrix

$$\overline{\boldsymbol{A}}^{(j)} = \begin{bmatrix} a_{11}^{(j)}/\sum_{k=1}^{n} a_{k1}^{(j)} & a_{12}^{(j)}/\sum_{k=1}^{n} a_{k2}^{(j)} & \cdots & a_{1n}^{(j)}/\sum_{k=1}^{n} a_{kn}^{(j)} \\ a_{21}^{(j)}/\sum_{k=1}^{n} a_{k1}^{(j)} & a_{22}^{(j)}/\sum_{k=1}^{n} a_{k2}^{(j)} & \cdots & a_{2n}^{(j)}/\sum_{k=1}^{n} a_{kn}^{(j)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}^{(j)}/\sum_{k=1}^{n} a_{k1}^{(j)} & a_{n2}^{(j)}/\sum_{k=1}^{n} a_{k2}^{(j)} & \cdots & a_{nn}^{(j)}/\sum_{k=1}^{n} a_{kn}^{(j)} \end{bmatrix}. \tag{3.4}$$

Denote the entry in row i and column ℓ of $\overline{\boldsymbol{A}}^{(j)}$ by $\overline{a}_{i\ell}^{(j)}$ (*i.e.*

$$\overline{a}_{i\ell}^{(j)} = a_{i\ell}^{(j)} \Bigg/ \sum_{k=1}^{n} a_{k\ell}^{(j)}, \quad i, \ell = 1, \ldots, n, \quad j = 1, \ldots, m).$$

Then $\boldsymbol{s}^{(j)}$ is approximately the vector of arithmetic means of the rows of $\overline{\boldsymbol{A}}^{(j)}$, that is the i-th entry of the vector $\boldsymbol{s}^{(j)}$ is given by

$$s_i^{(j)} \approx \frac{\sum_{\ell=1}^{n} \overline{a}_{i\ell}^{(j)}}{n}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, m. \tag{3.5}$$

The aforementioned method of estimating the i-th entry of the vector $\boldsymbol{s}^{(j)}$ is called the *eigenvector metod* (EM), because of the fact that the approximation in (3.5) approaches the entry of a true eigenvector of $\boldsymbol{A}^{(j)}$ as the DM's consistency increases, as mentioned above.

A popular alternative method of estimating the vector $\boldsymbol{s}^{(j)}$ is called the *log least square method* (LLSM). The LLSM returns a vector $\boldsymbol{s}^{(j)} = [s_1^{(j)}, \ldots, s_n^{(j)}]^T$ of relative score values of the n decision alternatives according to selection criterion j that minimises the quantity

$$\sum_{i=1}^{n} \sum_{k=1}^{n} \left(\log a_{ik}^{(j)} - \log s_i^{(j)}/s_k^{(j)} \right)^2, \tag{3.6}$$

from which the name of the method derives. The sum of log squares in (3.6) may be minimised by taking partial derivatives of the quantity with respect to each of the unknowns $s_1^{(j)}, \ldots, s_n^{(j)}$, setting each derivative equal to zero and solving the resulting system of equations. The i-th entry of this solution vector is given by

$$s_i^{(j)} = \frac{\left(\prod_{k=1}^{n} a_{ik}^{(j)} \right)^{\frac{1}{n}}}{\sum_{i=1}^{n} \left(\prod_{k=1}^{n} a_{ik}^{(j)} \right)^{\frac{1}{n}}}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, m. \tag{3.7}$$

Although certainly the most popular methods of estimation, the EM and the LLSM are not the only means by which the relative scores of decision alternatives can be estimated according to a particular selection criterion. Five other estimation methods are listed in Table 3.2. Descriptions of the working of these alternative estimation methods may be found in [19, 45, 64, 109, 112, 172, 183].

While the discussion on methods available for decision element scores or weight estimation has so far in this section focussed on the estimation of relative scores for the decision alternatives with respect to a single selection criterion, exactly the same methods may be used to estimate relative weights for the selection criteria themselves.

According to the EM, for example, the weight associated with selection criterion j is given by

$$w_j \approx \frac{\sum_{k=1}^{m} \overline{b}_{jk}}{m}, \quad j = 1, \ldots, m, \tag{3.8}$$

	Relative weight estimation methods
1	Eigenvector method (EM)
2	Arithmetic mean method
3	Geometric mean method
4	Log least squares method (LLSM)
5	Harmonic mean method
6	Renormalisation after the estimation of ratios
7	Mean transformation method

Table 3.2: *Methods available for the estimation of decision element scores or criterion weights.*

where $\overline{b}_{jk}$ denotes the entry in row j and column k of the normalised $m \times m$ pairwise comparison matrix

$$\overline{\boldsymbol{B}} = \begin{bmatrix} b_{11}/\sum_{k=1}^{m} b_{k1} & b_{12}/\sum_{k=1}^{m} b_{k2} & \cdots & b_{1m}/\sum_{k=1}^{m} b_{km} \\ b_{21}/\sum_{k=1}^{m} b_{k1} & b_{22}/\sum_{k=1}^{m} b_{k2} & \cdots & b_{2m}/\sum_{k=1}^{m} b_{km} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1}/\sum_{k=1}^{m} b_{k1} & b_{m2}/\sum_{k=1}^{m} b_{k2} & \cdots & b_{mm}/\sum_{k=1}^{m} b_{km} \end{bmatrix} \tag{3.9}$$

derived from the original $m \times m$ pairwise comparison matrix in (3.2).

According to the LLSM, on the other hand, it follows in a manner similar to the argument preceding the expression in (3.7) that the weight associated with selection criterion j is given by

$$w_j = \frac{\left(\prod_{k=1}^{m} b_{jk}\right)^{\frac{1}{m}}}{\sum_{j=1}^{m}\left(\prod_{k=1}^{m} b_{jk}\right)^{\frac{1}{m}}}, \quad j = 1, \ldots, m. \tag{3.10}$$

The other methods in Table 3.2 may, of course, also be used to estimate the weights associated with the selection criteria.

The final step of the AHP involves the aggregation of information related to the DM's pairwise preference information on decision alternatives with respect to each of the selection criteria and the criteria weights in order to associate a single score with each decision alternative. Two main methods of aggregation are commonly employed within the AHP, namely the *additive method* and the *multiplicative method*, with the former being the most popular [37]. The additive method is based on the notion of the weighted *arithmetic mean*, while the multiplicative method is based on the notion of the weighted *geometric mean*.

Recall that the relative weights of the m selection criteria are denoted by $w_1, \ldots, w_m$ (computed as in (3.8) and (3.10)) and that the relative score of decision alternative i with respect to selection criterion j is denoted by $s_i^{(j)}$ (computed as in (3.5) and 3.7). The final score of alternative i is given by

$$S_i = \sum_{j=1}^{m} s_i^{(j)} w_j, \quad i = 1, \ldots, n \tag{3.11}$$

according to the additive method of aggregation, whereas the final score of alternative i is given by

$$\tilde{S}_i = \prod_{j=1}^{m} \left(s_i^{(j)}\right)^{w_j}, \quad i = 1, \ldots, n \tag{3.12}$$

according to the multiplicative method of aggregation.

Measuring DM inconsistency

Brief references were made to very basic consistency requirements in respect of the population of the pairwise comparison matrices $\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(m)}$ and $\boldsymbol{B}$ above. These requirements were that the entries on the main diagonals of these pairwise comparison matrices should be ones and that the entries in mirror image positions relative to the main diagonal of each of these pairwise comparison matrices should be multiplicative reciprocals. There is, however, a more fundamental type of inconsistency that may be present in the population of these matrices, even if the above basic conditions are met. This type of intrinsic inconsistency will be present if the so-called *transitivity requirement*, described in §3.1.2, is not satisfied by all triples of matrix entries.

The transitivity requirement states that if the DM deems alternative i $a_{ik}^{(j)}$ times more attractive with respect to selection criterion j than alternative k, and also deems alternative k $a_{k\ell}^{(j)}$ times more attractive with respect to the same selection criterion than alternative ℓ, then a consistent DM should find alternative i $a_{ik}^{(j)} \times a_{k\ell}^{(j)}$ times more attractive with respect to selection criterion in question than alternative ℓ for each triple of alternatives i, k and ℓ. To expect a DM to meet this strict consistency requirement is, of course, unrealistic. Most human DMs will deviate from the transitivity requirement to some extent (*i.e.* be inconsistent to some degree). If the DM is too inconsistent, however, the AHP framework may yield results that are not trustworthy [224]. Hence there is a need to be able to measure in some way the level of inconsistency encapsulated in pairwise comparison matrix information gathered during Step 2 of the AHP as a result of failing to meet the transitivity requirement.

The most commonly adopted method of measuring DM transitivity-related consistency during the pairwise comparison process of Step 2 of the AHP involves use of a so-called *consistency index* (CI). For the $n \times n$ pairwise comparison matrix $\boldsymbol{A}^{(j)}$ of the n alternatives according to selection criterion j, this CI is given by

$$CI^{(j)} = \frac{\lambda^{(j)} - n}{n - 1},$$

where $\lambda^{(j)}$ denotes the eigenvalue with largest modulus of $\boldsymbol{A}^{(j)}$. This CI is used to form a so-called *consistency ratio* (CR) given by

$$CR = \frac{CI}{RI},$$

where RI denotes a *random index*, taken as the average CI-value resulting from a large number of randomly populated pairwise comparison matrices, using the same judgement scale that was used to populate $\boldsymbol{A}^{(j)}$. RI-values emanating from 500 randomly populated pairwise comparison matrices of dimensions 2×2 to 10×10 by means of the linear judgement scale of Table 2.2, for example, are given in Table 3.3.

n	2	3	4	5	6	7	8	9	10
RI	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.51

TABLE 3.3: *RI-values applicable to $n \times n$ pairwise comparison matrices populated by means of the linear judgement scale within the AHP [224].*

If the DM is perfectly consistent, then the CR will be zero, because then $\lambda^{(j)} = n$. Otherwise, the CR will be small for a slightly inconsistent DM. If the CR is less than 0.1, the matrix is usually considered sufficiently consistently populated with respect to the transitivity requirement [224].

The discussion above was aimed at measuring the inconsistency present in an $n \times n$ pairwise comparison matrix $\boldsymbol{A}^{(j)}$ of the n alternatives according to selection criterion j. Exactly the same method may, however, be applied to measure the transitivity-related inconsistency embedded in the $m \times m$ comparison matrix $\boldsymbol{B}$ for the selection criteria. In this case the CI should, of course, be defined as

$$CI = \frac{\lambda - m}{m-1},$$

where λ denotes the eigenvalue with largest modulus of $\boldsymbol{B}$.

A numerical example

The four steps of the AHP, the possible alternatives within each step, as well as a method for measuring consistency were reviewed above, and these steps are demonstrated by means of a worked example in this section. Table 3.4 combines all of the alternatives mentioned in each of the sections above. Taking into account only the alternatives mentioned in this section, there are $7 \times 7 \times 2 = 98$ different combinations in which the AHP can be implemented. Only the alternatives typeset in boldface in Table 3.4 are, however, considered in the example of this section. This provides a combination of twelve sets of results that can be compared.

Step 2	Step 3	Step 4
Linear scale	**Eigenvector method**	**Additive aggregation**
Power scale	Arithmetic mean method	**Multiplicative aggregation**
Geometric scale	Geometric mean method	
Logarithmic scale	**Log least squares method**	
Root square scale	Harmonic mean method	
Balanced scale	Renormalisation after the estimation of ratios method	
	Mean transformation method	

TABLE 3.4: *Alternatives for Steps 2–4 of the AHP, where the boldfaced alternatives are considered in the worked example of this section.*

Example 3.1 *Suppose a DM has to choose between four houses based on four selection criteria, namely the size of the house (SOH), the yard space (YS), the cost (C) and the neighbourhood (NH). The DM first has to set up the problem hierarchy. The aforementioned alternatives and criteria are represented graphically in Figure 3.2 according to the general hierarchy presented in Figure 3.1, where H_1 to H_4 represents four alternative houses under consideration.*

Suppose the DM returns the pairwise comparison matrix

$$\boldsymbol{B}_{Lin} = \begin{array}{c} \\ SOH \\ YS \\ C \\ NH \end{array} \begin{array}{c} \begin{array}{cccc} SOH & YS & C & NH \end{array} \\ \left[\begin{array}{cccc} 1 & 2 & 2 & 4 \\ \frac{1}{2} & 1 & 3 & 3 \\ \frac{1}{2} & \frac{1}{3} & 1 & 4 \\ \frac{1}{4} & \frac{1}{3} & \frac{1}{4} & 1 \end{array}\right] \end{array}$$

in respect of the relative importance of the four selection criteria according to the linear scale in Table 2.1 with parameter $a = 1$. The corresponding pairwise comparison matrix according to the

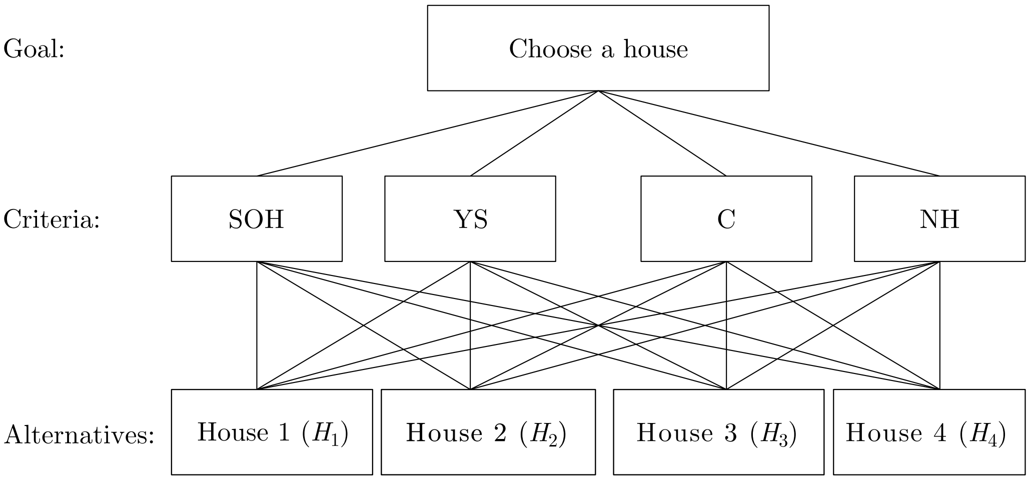


FIGURE 3.2: *Decision problem hierarchy for choosing a house based on four attributes.*

geometric scale in Table 2.2 with parameter $a = 2$ *is*

$$\boldsymbol{B}_{Geo} = \begin{array}{c} \\ SOH \\ YS \\ C \\ NH \end{array}\!\!\begin{array}{c} \begin{array}{cccc} SOH & YS & C & NH \end{array} \\ \left[\begin{array}{cccc} 1 & 2 & 2 & 8 \\ \frac{1}{2} & 1 & 4 & 4 \\ \frac{1}{2} & \frac{1}{4} & 1 & 8 \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} & 1 \end{array}\right] \end{array},$$

while that according to the logarithmic scale in Table 2.2 with parameter $a = 2$ *is*

$$\boldsymbol{B}_{Log} = \begin{array}{c|cccc} & SOH & YS & C & NH \\ \hline SOH & 1 & 1.584 & 1.584 & 2.321 \\ YS & 0.630 & 1 & 2 & 2 \\ C & 0.630 & 0.500 & 1 & 2.321 \\ NH & 0.430 & 0.500 & 0.430 & 1 \end{array}.$$

The pairwise comparison matrix $\boldsymbol{B}_{Lin}$ *may be normalised as in (3.9) to obtain*

$$\overline{\boldsymbol{B}}_{Lin} = \begin{array}{c|cccc} & SOH & YS & C & NH \\ \hline SOH & 0.444 & 0.545 & 0.320 & 0.333 \\ YS & 0.222 & 0.272 & 0.480 & 0.250 \\ C & 0.222 & 0.090 & 0.160 & 0.333 \\ NH & 0.111 & 0.090 & 0.040 & 0.083 \end{array}.$$

The criterion weight vector $\boldsymbol{w}^{EM}_{Lin}$ *may now be estimated according to the EM by averaging the values in the rows of* $\overline{\boldsymbol{B}}_{Lin}$ *to find*

$$w^{EM}_{Lin,1} = \frac{0.444 + 0.545 + 0.320 + 0.333}{4} = 0.4108,$$

$$\begin{aligned} w^{EM}_{Lin,2} &= \frac{0.222 + 0.272 + 0.480 + 0.250}{4} = 0.3062, \\ w^{EM}_{Lin,3} &= \frac{0.222 + 0.090 + 0.160 + 0.333}{4} = 0.2016, \\ w^{EM}_{Lin,4} &= \frac{0.111 + 0.090 + 0.040 + 0.083}{4} = 0.0813 \end{aligned}$$

according to (3.8), yielding the ranking vector $\boldsymbol{w}^{EM}_{Lin} = (0.410, 0.306, 0.201, 0.081)^T$. *In order to demonstrate the differences in the resulting ranking, the LLSM applied to the matrix* $\overline{\boldsymbol{B}}_{Lin}$ *produces the weights*

$$w^{LLSM}_{Lin,1} = \frac{(1+2+2+4)^{\frac{1}{4}}}{(1\times 2\times 2\times 4)^{\frac{1}{4}} + (\frac{1}{2}\times 1\times 3\times 3)^{\frac{1}{4}} + (\frac{1}{2}\times\frac{1}{3}\times 1\times 4)^{\frac{1}{4}} + (\frac{1}{4}\times\frac{1}{3}\times\frac{1}{4}\times 1)^{\frac{1}{4}}} = 0.3654,$$

$$w^{LLSM}_{Lin,2} = \frac{(\frac{1}{2}+1+3+3)^{\frac{1}{4}}}{(1\times 2\times 2\times 4)^{\frac{1}{4}} + (\frac{1}{2}\times 1\times 3\times 3)^{\frac{1}{4}} + (\frac{1}{2}\times\frac{1}{3}\times 1\times 4)^{\frac{1}{4}} + (\frac{1}{4}\times\frac{1}{3}\times\frac{1}{4}\times 1)^{\frac{1}{4}}} = 0.3491,$$

$$w^{LLSM}_{Lin,3} = \frac{(\frac{1}{2}+\frac{1}{3}+1+4)^{\frac{1}{4}}}{(1\times 2\times 2\times 4)^{\frac{1}{4}} + (\frac{1}{2}\times 1\times 3\times 3)^{\frac{1}{4}} + (\frac{1}{2}\times\frac{1}{3}\times 1\times 4)^{\frac{1}{4}} + (\frac{1}{4}\times\frac{1}{3}\times\frac{1}{4}\times 1)^{\frac{1}{4}}} = 0.3278,$$

$$w^{LLSM}_{Lin,4} = \frac{(\frac{1}{4}+\frac{1}{3}+\frac{1}{4}+1)^{\frac{1}{4}}}{(1\times 2\times 2\times 4)^{\frac{1}{4}} + (\frac{1}{2}\times 1\times 3\times 3)^{\frac{1}{4}} + (\frac{1}{2}\times\frac{1}{3}\times 1\times 4)^{\frac{1}{4}} + (\frac{1}{4}\times\frac{1}{3}\times\frac{1}{4}\times 1)^{\frac{1}{4}}} = 0.2454$$

according to (3.10), yielding the ranking vector $\boldsymbol{w}^{LLSM}_{Lin} = (0.283, 0.271, 0.254, 0.190)^T$.

Applying instead the EM and the LLSM to the matrices $\boldsymbol{B}_{Geo}$ *and* $\boldsymbol{B}_{Log}$ *similarly yields the weight vectors*

$$\begin{aligned} \boldsymbol{w}^{EM}_{Geo} &= (0.418, 0.340, 0.194, 0.046)^T, \\ \boldsymbol{w}^{LLSM}_{Geo} &= (0.277, 0.256, 0.258, 0.161)^T, \\ \boldsymbol{w}^{EM}_{Log} &= (0.339, 0.310, 0.230, 0.119)^T, \text{ and} \\ \boldsymbol{w}^{LLSM}_{Log} &= (0.288, 0.278, 0.262, 0.224)^T. \end{aligned}$$

Upon computing the vector

$$B_{Lin}(\boldsymbol{w}^{EM}_{Lin})^T = \begin{bmatrix} 1 & 2 & 2 & 4 \\ \frac{1}{2} & 1 & 3 & 3 \\ \frac{1}{2} & \frac{1}{3} & 1 & 4 \\ \frac{1}{4} & \frac{1}{3} & \frac{1}{4} & 1 \end{bmatrix} \begin{bmatrix} 0.4108 \\ 0.3062 \\ 0.2016 \\ 0.0813 \end{bmatrix} = \begin{bmatrix} 1.7518 \\ 1.3605 \\ 0.8344 \\ 0.3365 \end{bmatrix},$$

the principal eigenvalue may be estimated as

$$\begin{aligned} \lambda &= \frac{1}{4}\sum_{i=1}^{4} \frac{i\text{-}th\ entry\ in\ A\boldsymbol{w}^T}{i\text{-}th\ entry\ in\ \boldsymbol{w}^T} \\ &= \left(\frac{1}{4}\right)\left(\frac{1.7518}{0.4108} + \frac{1.3605}{0.3062} + \frac{0.8344}{0.2016} + \frac{0.3365}{0.0813}\right) \\ &= 4.2458. \end{aligned}$$

Therefore, the DM's CI with respect to house attributes may be computed as

$$CI = \frac{\lambda - n}{n-1} = \frac{4.2458 - 4}{3} = 0.0819$$

and compared with the corresponding RI in Table 3.3 for $n = 4$*. From the table, this RI-value is* 0.90 *and so* $\frac{CI}{RI} = \frac{0.0819}{0.90} = 0.091$*. The DM's judgements with respect to house attributes may be considered sufficiently consistent with respect to the transitivity requirement, because* $0.091 < 0.10$*.*

The final step of the AHP is to aggregate the scores of the alternatives using the corresponding weight vectors in order to produce final rankings. Since the alternatives need to be compared with respect to each criterion, the same judgement scale is used for each selection criterion's comparison matrix. The four alternatives H_1, H_2, H_3 *and* H_4 *are compared in respect of each criterion. Suppose the DM's linearly populated comparison matrices for size of house (SOH), yard space (YS), cost (C) and neighbourhood (NH) of the four houses are*

$$\boldsymbol{A}^{(SOH)}_{Lin} = \begin{array}{cc} & \begin{matrix} H_1 & H_2 & H_3 & H_4 \end{matrix} \\ \begin{matrix} H_1 \\ H_2 \\ H_3 \\ H_4 \end{matrix} & \begin{bmatrix} 1 & 3 & 4 & 2 \\ \frac{1}{3} & 1 & 3 & 5 \\ \frac{1}{4} & \frac{1}{3} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{5} & 2 & 1 \end{bmatrix} \end{array}, \qquad \boldsymbol{A}^{(C)}_{Lin} = \begin{array}{cc} & \begin{matrix} H_1 & H_2 & H_3 & H_4 \end{matrix} \\ \begin{matrix} H_1 \\ H_2 \\ H_3 \\ H_4 \end{matrix} & \begin{bmatrix} 1 & \frac{1}{4} & \frac{1}{4} & 2 \\ 4 & 1 & 4 & \frac{1}{2} \\ 4 & \frac{1}{4} & 1 & \frac{1}{2} \\ \frac{1}{2} & 2 & 2 & 1 \end{bmatrix} \end{array},$$

$$\boldsymbol{A}^{(YS)}_{Lin} = \begin{array}{cc} & \begin{matrix} H_1 & H_2 & H_3 & H_4 \end{matrix} \\ \begin{matrix} H_1 \\ H_2 \\ H_3 \\ H_4 \end{matrix} & \begin{bmatrix} 1 & \frac{1}{5} & \frac{1}{3} & \frac{1}{4} \\ 5 & 1 & 3 & \frac{1}{2} \\ 3 & \frac{1}{3} & 1 & \frac{1}{2} \\ 4 & 2 & 2 & 1 \end{bmatrix} \end{array}, \qquad \boldsymbol{A}^{(NH)}_{Lin} = \begin{array}{cc} & \begin{matrix} H_1 & H_2 & H_3 & H_4 \end{matrix} \\ \begin{matrix} H_1 \\ H_2 \\ H_3 \\ H_4 \end{matrix} & \begin{bmatrix} 1 & 2 & 6 & 4 \\ \frac{1}{2} & 1 & 3 & \frac{1}{2} \\ \frac{1}{6} & \frac{1}{3} & 1 & 4 \\ \frac{1}{4} & 2 & \frac{1}{4} & 1 \end{bmatrix} \end{array}.$$

The corresponding pairwise comparison matrices of the alternatives with respect to each selection criterion according to the geometric judgement scale is similarly given by

$$\boldsymbol{A}^{(SOH)}_{Geo} = \begin{array}{cc} & \begin{matrix} H_1 & H_2 & H_3 & H_4 \end{matrix} \\ \begin{matrix} H_1 \\ H_2 \\ H_3 \\ H_4 \end{matrix} & \begin{bmatrix} 1 & 4 & 8 & 2 \\ \frac{1}{4} & 1 & 4 & 16 \\ \frac{1}{8} & \frac{1}{4} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{16} & 2 & 1 \end{bmatrix} \end{array}, \qquad \boldsymbol{A}^{(C)}_{Geo} = \begin{array}{cc} & \begin{matrix} H_1 & H_2 & H_3 & H_4 \end{matrix} \\ \begin{matrix} H_1 \\ H_2 \\ H_3 \\ H_4 \end{matrix} & \begin{bmatrix} 1 & \frac{1}{8} & \frac{1}{8} & 2 \\ 8 & 1 & 8 & \frac{1}{2} \\ 8 & \frac{1}{8} & 1 & \frac{1}{2} \\ \frac{1}{2} & 2 & 2 & 1 \end{bmatrix} \end{array},$$

$$\boldsymbol{A}^{(SOH)}_{YS} = \begin{array}{cc} & \begin{matrix} H_1 & H_2 & H_3 & H_4 \end{matrix} \\ \begin{matrix} H_1 \\ H_2 \\ H_3 \\ H_4 \end{matrix} & \begin{bmatrix} 1 & \frac{1}{16} & \frac{1}{4} & \frac{1}{8} \\ 16 & 1 & 4 & \frac{1}{2} \\ 4 & \frac{1}{4} & 1 & \frac{1}{2} \\ 8 & 2 & 2 & 1 \end{bmatrix} \end{array}, \qquad \boldsymbol{A}^{(NH)}_{Geo} = \begin{array}{cc} & \begin{matrix} H_1 & H_2 & H_3 & H_4 \end{matrix} \\ \begin{matrix} H_1 \\ H_2 \\ H_3 \\ H_4 \end{matrix} & \begin{bmatrix} 1 & 2 & 32 & 8 \\ \frac{1}{2} & 1 & 4 & \frac{1}{2} \\ \frac{1}{32} & \frac{1}{4} & 1 & 8 \\ \frac{1}{8} & 2 & \frac{1}{8} & 1 \end{bmatrix} \end{array},$$

while those populated according to the logarithmic judgement scale are

$$\boldsymbol{A}^{(SOH)}_{Log} = \begin{array}{c|cccc} & H_1 & H_2 & H_3 & H_4 \\ \hline H_1 & 1 & 2.000 & 2.322 & 1.585 \\ H_2 & 0.500 & 1 & 2.000 & 2.585 \\ H_3 & 0.431 & 0.500 & 1 & 0.631 \\ H_4 & 0.631 & 0.387 & 1.585 & 1 \end{array}, \quad \boldsymbol{A}^{(C)}_{Log} = \begin{array}{c|cccc} & H_1 & H_2 & H_3 & H_4 \\ \hline H_1 & 1 & 0.431 & 0.431 & 1.585 \\ H_2 & 2.322 & 1 & 2.322 & 0.631 \\ H_3 & 2.322 & 0.431 & 1 & 0.631 \\ H_4 & 0.631 & 1.585 & 1.585 & 1 \end{array},$$

$$\boldsymbol{A}^{(YS)}_{Log} = \begin{array}{c|cccc} & H_1 & H_2 & H_3 & H_4 \\ \hline H_1 & 1 & 0.387 & 0.500 & 0.431 \\ H_2 & 2.585 & 1 & 2.000 & 0.631 \\ H_3 & 2.000 & 0.500 & 1 & 0.631 \\ H_4 & 2.322 & 1.585 & 1.585 & 1 \end{array}, \quad \boldsymbol{A}^{(NH)}_{Log} = \begin{array}{c|cccc} & H_1 & H_2 & H_3 & H_4 \\ \hline H_1 & 1 & 1.585 & 2.807 & 2.322 \\ H_2 & 0.631 & 1 & 2.000 & 0.631 \\ H_3 & 0.356 & 0.500 & 1 & 2.322 \\ H_4 & 0.431 & 1.585 & 0.431 & 1 \end{array}.$$

After normalisation of the above matrices, the values of each row within a matrix may be averaged to provide a weight for each alternative based on the criterion specified. The scores thus obtained for the four decision alternatives in respect of the four selection criteria in the case of the linear judgement scale are given by

$$\boldsymbol{s}^{(1)}_{Lin} = [0.444, 0.317, 0.088, 0.150],$$

$$\boldsymbol{s}^{(2)}_{Lin} = [0.178, 0.346, 0.189, 0.287],$$

$$\boldsymbol{s}^{(3)}_{Lin} = [0.074, 0.341, 0.176, 0.408], \text{ and}$$

$$\boldsymbol{s}^{(4)}_{Lin} = [0.476, 0.198, 0.167, 0.159].$$

Applying the additive method to the weights obtained by the EM ($\boldsymbol{w}^{EM}_{Lin}$) and the LLSM ($\boldsymbol{w}^{LLSM}_{Lin}$), overall scores for alternative one to four are calculated as

$$\boldsymbol{S}^{EM}_{Lin} = [0.4815, 0.3681, 0.1250, 0.0817] \text{ and}$$

$$\boldsymbol{S}^{LLSM}_{Lin} = [0.3317, 0.3258, 0.1575, 0.1909],$$

respectively. Similarly, applying the multiplicative aggregation method to the above results yields the alternative score vector

$$\tilde{\boldsymbol{S}}^{EM}_{Lin} = [0.3524, 0.2409, 0.3703, 0.0802] \text{ and}$$

$$\tilde{\boldsymbol{S}}^{LLSM}_{Lin} = [0.2277, 0.2080, 0.4274, 0.1883].$$

The geometric and logarithmic score weights may be aggregated in the same way to yield the score vectors in Table 3.5. □

Rankings of the alternatives in an SPBP instance may be represented in the same manner as players in a tournament were ranked in §2.2.1. From Table 3.5 it follows that the ranking of the alternatives is H_1, H_2, H_3, H_4 when employing the EM and the additive aggregation method, according to the linear, geometric and logarithmic judgement scales. In the case of the LLSM and the additive aggregation method, however, the ranking of alternatives becomes H_1, H_2, H_4, H_3

Judgement scale	Estimation method	Additive aggregation	Multiplicative aggregation
Linear	EM	$0.4815, 0.3681, 0.1250, 0.0817$	$0.3525, 0.2409, 0.3703, 0.0802$
	LLSM	$0.3317, 0.3258, 0.1575, 0.1909$	$0.2277, 0.2080, 0.4274, 0.1883$
Geometric	EM	$0.5203, 0.4493, 0.0996, 0.0423$	$0.3376, 0.2402, 0.4308, 0.0588$
	LLSM	$0.3448, 0.3383, 0.1324, 0.1481$	$0.2022, 0.1651, 0.4987, 0.1863$
Logarithmic	EM	$0.3695, 0.3553, 0.1730, 0.1203$	$0.3075, 0.2611, 0.3309, 0.1162$
	LLSM	$0.3139, 0.3187, 0.1971, 0.2264$	$0.2574, 0.2304, 0.3650, 0.2203$

TABLE 3.5: *Overall scores of alternatives H_1–H_4 in Figure 3.2 according to the four criteria SOH, YS, C and NH resulting from the use of three different judgement scales and two different aggregation methods within the AHP in the context of Example 3.1.*

according to both the linear and geometric judgement scales. This difference in ranking is again an example of the occurrence of *rank reversal* (where the fourth alternative moves from the last to the third position in the ranking order) merely by changing the weight estimation method. Note, however, that by employing multiplicative aggregation instead of additive aggregation, rank reversal again occurs, this time where the third alternative moves to the first position for four of the six judgement scale and weight estimation combinations, providing the ranking H_2, H_3, H_4, H_1 for the linear and logarithmic judgement scales combined with the LLSM, and for the geometric and logarithmic judgement scales in combination with the EM. Another case of rank reversal, different to the previously mentioned case, occurs for the linear judgement scale in conjunction with the EM when the multiplicative aggregation method is employed. The ranking H_1, H_3, H_2, H_4 in this case is the result of the third alternative moving from the third position to the second position.

When, however, the LLSM is applied in conjunction with the additive aggregation method, the ranking H_2, H_1, H_4, H_3 is obtained for the logarithmic judgement scales. The geometric scale provides the ranking H_3, H_1, H_4, H_2 when aggregation is again changed to the multiplicative method. These rankings differ from all other cases previously mentioned.

From the aforementioned results it is clear that none of the judgement scales returns a consistent ranking in Example 3.1, when the aggregation method changes. In fact, the only constant ranking is exhibited when the EM and additive aggregation method are employed and the judgement scale changes. The case of so-called *strong variability*[4] does not, however, occur in this example. The opposing term, known as *weak variability*, does indeed occur in this example when the weight estimation method changes for the linear or geometric judgement scales, for both the additive and multiplicative aggregation methods.

It is difficult to determine which combination of methods should be used or which ranking is the best. Saaty [186] argued (unconvincingly, in the opinion of the author) that the EM is the only valid method for derivation of priority from a pairwise comparison matrix, particularly for matrices populated by inconsistent DMs. Variability in ranks often occurs, in fact, even if the individual pairwise comparison matrices do not exhibit significant inconsistency, due to the multicriteria process itself.

[4] Two distinct rankings are *weakly variable* if they differ in exactly two entries. Otherwise, the rankings are *strongly variable* [186].

Advantages and disadvantages of using the AHP

There have been many concerns about applying the AHP and whether or not it should be used for deciding between alternatives. In this section, some criticisms are mentioned that have been levelled at each step of the AHP. Some general advantages of using the AHP are also reviewed [171, 186].

The following are considered some of the most important advantages of using the AHP [171]:

1. The method provides a systematic approach for the identification of objectives and preferences.

2. The method allows for the incorporation of quantifiable as well as non-quantifiable factors in the analysis.

3. Scientific judgement and personal opinions can be combined in the evaluation of alternatives within the AHP framework.

4. Pairwise comparison is a relatively simple and intuitive process for the elicitation of preferences for objectives.

5. The AHP is a relatively fast and low cost method from a computational point of view if there are not too many alternatives and/or selection criteria to consider.

The following disadvantages are, however, associated with the various steps of the AHP:

Setting up of problem instance. The composition of a decision hierarchy by the DM may be incomplete [193]. This may generate counter-intuitive composite weights for the decision elements. At the moment no theoretical framework exists for modelling decision problems in this hierarchy although such a contribution may provide valuable insight into the differences in informational values.

Aggregation of overall alternative scores. The collection of input data is assumed to occur on the basis that the DM is consistent to some extent, which is not always the case. Studies have suggested that a possible approach toward modelling input data is to consider these data as observations with random errors [184]. This suggestion is, however, subject to debate as the data collection method differs from that of the multi-attribute value estimation. Another disadvantage related to AHP input data is that of bounded input data (*e.g.* $1/9 - 9$ for the linear judgement scale in Table 2.2) as opposed to unbounded data and the consequences of selecting a judgement scale.

Estimation of decision element scores or weights. Estimating relative weights for the alternatives has attracted the most attention and controversy in the literature [166]. A large number of methods have been suggested for weight estimation (as may be seen in Table 3.2) and there is no consensus in the literature as to which method is superior. The EM is most often used as it has the advantage of having a long history of use in software products on the market [143]. Another disadvantage associated with the estimation of scores for alternatives and weights for criteria arises from the nature of the underlying pairwise comparison process. This can be a very tedious process for the DM if there are many alternatives and/or selection criteria to consider — a total of $m\binom{n}{2} + \binom{m}{2}$ such comparisons have to be performed if there are n decision alternatives and m selection criteria. This number grows very rapidly as m and n increase, as illustrated in Table 3.6.

$n \rightarrow$	10	20	30	40	50	60	70	80	90	100
$m = 1$	45	190	435	780	1 225	1 770	2 415	3 160	4 005	4 950
$m = 2$	91	381	871	1 561	2 451	3 541	4 831	6 321	8 011	9 901
$m = 3$	138	573	1 308	2 343	3 678	5 313	7 248	9 483	12 018	14 853
$m = 4$	186	766	1 746	3 126	4 906	7 086	9 666	12 646	16 026	19 806
$m = 5$	235	960	2 185	3 910	6 135	8 860	12 085	15 810	20 035	24 760

Table 3.6: *The number of pairwise comparisons required by the DM when applying the AHP to an SPBP instance involving n decision alternatives and m selection criteria.*

Pairwise comparison of decision elements. Disadvantages related to the final step of the AHP hinge on the outcome of step 3 (estimation of scores for alternatives and weights for criteria), since the composite weights computed in this step determine the final outcome of the AHP. This is mainly due to the statistical and rank-preserving properties inherent in estimating weights. Another disadvantage related to the final AHP step arises when data are collected for more than one evaluator (and in particular in combining judgements). Reaching a consensus in such a case is an important consideration.

The special case of the AHP where $m = 1$

In the special case where $m = 1$ (*i.e.* if there is only one selection criterion present) in the AHP process, it is clear that Step 4 of the process is superfluous. In this single-criterion subjective preferential selection case, the terminology convention is adopted in this dissertation that Steps (3) and (4) are together referred to as *aggregation* (as opposed to *estimation* followed by *aggregation*), because then the estimation step already produces aggregated score results for each of the alternatives.

3.2.2 MACBETH — A difference-based approach

In 1994, Bana e Costa and Vansnick [62] developed a difference-based approach towards value function construction, called MACBETH. This approach stands in contrast to the ratio-based approach of Saaty's AHP described in the previous section and was, in fact, proposed in response to the disadvantages of the AHP, as described in the previous section.

The working of the MACBETH procedure is described in general in this section for single-attribute preference selection problems and is illustrated by means of a simple example, after which a brief discussion follows on how the method may be extended so as to be applicable to multi-attribute preference selection.

Working of the MACBETH procedure

Suppose the alternatives to be compared by the DM are $\mathbb{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, numbered in order of decreasing[5] attractiveness as indicated by the DM *a priori* (that is, $\boldsymbol{x}_1 \succ \boldsymbol{x}_2 \succ \boldsymbol{x}_3 \succ \cdots \succ \boldsymbol{x}_{n-1} \succ \boldsymbol{x}_n$ in the notation of §3.1.2). Furthermore, let $\mathbb{X}^2$ be the set of all *ordered* pairs from $\mathbb{X}$

[5] It is therefore assumed that there are no equivalence classes in $\mathbb{X}$. If indeed there were such equivalence classes, then a representative of each of these classes could be included in $\mathbb{X}$ while discarding the remaining members of each class. After computing a value function based on DM preference information about the remaining alternatives, the discarded equivalence class members can then be reintroduced, ensuring that they receive the same value function values as their class representatives.

of the form $(\boldsymbol{x}_i, \boldsymbol{x}_j)$, where $\boldsymbol{x}_i \succ \boldsymbol{x}_j$. The upper-triangular part of a pairwise comparison matrix $\boldsymbol{A}$ is again populated by eliciting relative preference information from the DM in respect of these alternatives.

This preference information is essentially expressed by the DM during a process whereby he partitions the set of ordered pairs $\mathbb{X}^2$ into six so-called *difference categories* instead of employing a ratio-based judgement scale as in the AHP. The first three of these categories are called *fundamental difference categories* (and are denoted here by C_2, C_4 and C_6), while the other three categories are called *intermediate difference categories* (and are denoted here by C_1, C_3 and C_5). The definitions of these categories are provided in Table 3.7. The entry in row $i \in \{1, \ldots, n-1\}$ and column $j \in \{i+1, \ldots, n\}$ of the pairwise comparison matrix $\boldsymbol{A}$ is the category index $k \in \{1, 2, 3, 4, 5, 6\}$ for which $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in C_k$.

Category	Description
$(\boldsymbol{x}_i, \boldsymbol{x}_j) \in C_1$	if $\boldsymbol{x}_i \succ \boldsymbol{x}_j$, but the difference of attractiveness between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is negligible
$(\boldsymbol{x}_i, \boldsymbol{x}_j) \in C_2$	if $\boldsymbol{x}_i \succ \boldsymbol{x}_j$, but the difference of attractiveness between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is weak
$(\boldsymbol{x}_i, \boldsymbol{x}_j) \in C_3$	if $\boldsymbol{x}_i \succ \boldsymbol{x}_j$, and the difference of attractiveness between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is moderate
$(\boldsymbol{x}_i, \boldsymbol{x}_j) \in C_4$	if $\boldsymbol{x}_i \succ \boldsymbol{x}_j$, and the difference of attractiveness between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is strong
$(\boldsymbol{x}_i, \boldsymbol{x}_j) \in C_5$	if $\boldsymbol{x}_i \succ \boldsymbol{x}_j$, and the difference of attractiveness between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is very strong
$(\boldsymbol{x}_i, \boldsymbol{x}_j) \in C_6$	if $\boldsymbol{x}_i \succ \boldsymbol{x}_j$, and the difference of attractiveness between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is extreme

TABLE 3.7: *Interpretation of the meanings of the MACBETH difference categories.*

It follows from the category definitions in Table 3.7 that the DM's population of the pairwise comparison matrix $\boldsymbol{A}$ is consistent with his or her previous ranking of the alternatives in order of decreasing relative attractiveness if the rows of $\boldsymbol{A}$ are non-decreasing and its columns are non-increasing.

The process followed in the MACBETH procedure consists of two stages carried out consecutively. The purpose of the first stage is to determine the level of inconsistency of the DM embodied in the pairwise comparison matrix $\boldsymbol{A}$ and also to generate a first-order estimate of the DM's value function for the alternatives in $\mathbb{X}$, while the objective during the second phase is to refine the output of the first phase into a more equitable value function.

The first phase of the MACBETH procedure involves determining:

- Five positive real numbers σ_1, σ_2, σ_3, σ_4 and σ_5 satisfying the inequality chain
$$0 = \sigma_0 < \sigma_1 < \sigma_2 < \sigma_3 < \sigma_4 < \sigma_5 < \sigma_6 = \infty \tag{3.13}$$
and serving as the breakpoints of a partition of the set $\mathbb{R}^+$ of positive real numbers into six real intervals $(\sigma_0, \sigma_1]$, $(\sigma_1, \sigma_2]$, $(\sigma_2, \sigma_3]$, $(\sigma_3, \sigma_4]$, $(\sigma_4, \sigma_5]$ and (σ_5, σ_6).
- A value function $u : \mathbb{X} \mapsto \mathbb{R}$ with the property that
$$\sigma_{k-1} < u(\boldsymbol{x}_i) - u(\boldsymbol{x}_j) \leq \sigma_k \text{ if and only if } (\boldsymbol{x}_i, \boldsymbol{x}_j) \in C_k \text{ for some } k \in \{1, 2, 3, 4, 5, 6\}. \tag{3.14}$$

Because category C_1 in Table 3.7 contains pairs of alternatives in $\mathbb{X}^2$ for which differences in attractiveness are negligible, a DM response of the form

$$(\boldsymbol{x}_i, \boldsymbol{x}_j) \in C_k \text{ for some } k \in \{1, 2, 3, 4\}, \quad (\boldsymbol{x}_j, \boldsymbol{x}_k) \in C_1 \quad \text{and} \quad (\boldsymbol{x}_i, \boldsymbol{x}_k) \in C_\ell \text{ for some } \ell > k+1$$

is clearly incoherent for any triple $(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k)$ of alternatives. It is therefore required that the length of the difference interval $(\sigma_{k-1}, \sigma_k]$ should be at least as large as that of $(\sigma_0, \sigma_1]$ for any

$k \in \{2,3,4,5\}$. Mathematically, this may be enforced by requiring that

$$\sigma_k - \sigma_{k-1} \geq \sigma_1 - \sigma_0 \quad \text{for all } k \in \{2,3,4,5\}. \tag{3.15}$$

By letting $u(\boldsymbol{x}_i) = u_i$ for all $i \in \{1,\dots,n\}$, the first stage of the MACBETH procedure involves solving a linear programming problem in which the objective is to

$$\text{minimise} \;\; c \tag{3.16}$$

subject to the constraints

$$\begin{array}{rclll}
\sigma_0 = 0 \;\text{ and }\; \sigma_1 & = & 1, & & (3.17)\\
\sigma_k - \sigma_{k-1} & \geq & 1, & k \in \{2,3,4,5\}, & (3.18)\\
u_i - u_j & \geq & \theta, & (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathbb{X}^2, & (3.19)\\
u_n & = & 0, & & (3.20)\\
\sigma_{k-1} + \theta - c \;\leq\; u_i - u_j & \leq & \sigma_k + c, & (\boldsymbol{x}_i, \boldsymbol{x}_j) \in C_k, \quad k \in \{1,2,3,4,5\}, & (3.21)\\
\sigma_5 + \theta - c & \leq & u_i - u_j, & (\boldsymbol{x}_i, \boldsymbol{x}_j) \in C_6, & (3.22)\\
c & \geq & 0, & & (3.23)\\
\sigma_i & \geq & 0, & i \in \{1,2,3,4,5\}, & (3.24)\\
u_j & \geq & 0, & j \in \{1,\dots,n\}. & (3.25)
\end{array}$$

The variable c in (3.16) is called the *incoherency value* of the DM. The objective is to pursue the smallest value of this variable for which the responses of the DM may be incorporated into the pairwise comparison matrix $\boldsymbol{A}$ in a consistent manner. If it is found that $c = 0$ upon solving (3.16)–(3.25), then the DM may be considered perfectly consistent in terms of his specification of preference information. Otherwise, the DM exhibits some level of inconsistency measured by the value of c. Furthermore, the parameter θ in (3.19) and (3.21)–(3.22) is a small positive real number allowing for the enforcement of strict inequality constraints.

Fixing the first difference interval as the unit interval in constraint set (3.17) and fixing the magnitude of the value function $u(\boldsymbol{x}_n)$ at the least favourable alternative $\boldsymbol{x}_n \in \mathbb{X}$ as zero in constraint (3.20) are both anchoring choices without loss of generality. Constraint set (3.18) furthermore ensures that the coherency conditions in (3.15) are met, while constraint set (3.19) ensures that the value function values of the alternatives in $\mathbb{X}$ are suitably separated under the assumption that all alternatives are distinguishable in terms of DM preference. Constraint sets (3.21)–(3.22) together ensure that the conditions in (3.14) are satisfied in a coherent manner, while constraint sets (3.23)–(3.25) finally ensure that all variables are non-negative.

Chained with the linear programming problem (3.16)–(3.25) is another linear programming problem which is solved during the second phase of the MACBETH process. The purpose of this second phase is to adjust the difference interval boundaries found by solving (3.16)–(3.25) for a fixed value $c = c_{\min}$ (say), as computed during the first phase of the procedure, so as to obtain more equitable difference intervals.

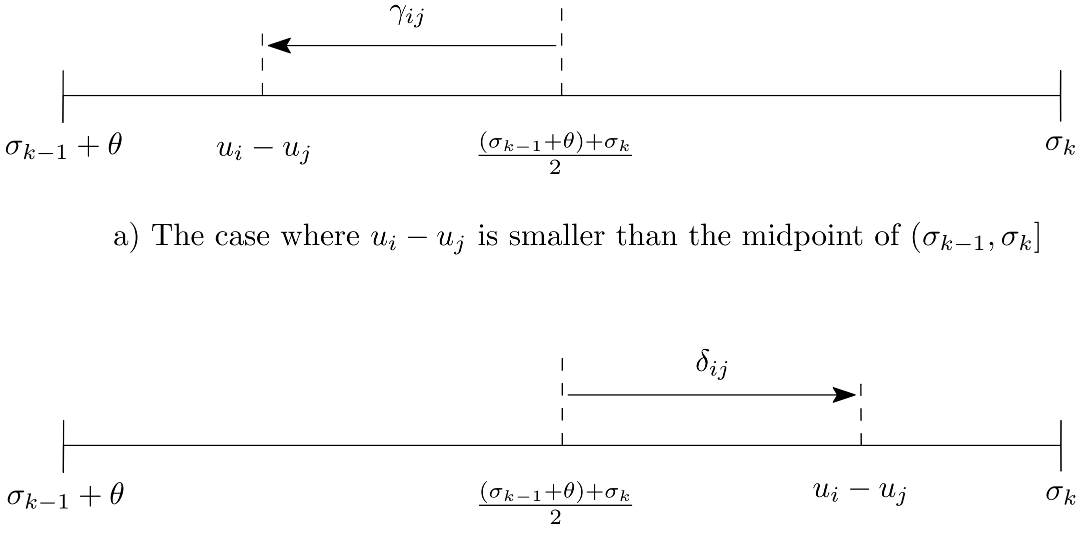
More precisely, the purpose of the second stage is to ensure that

- The difference in attractiveness $u_i - u_j$ for each pair of alternatives $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in C_k$ is as close as possible to the centre of the corresponding interval $(\sigma_{k-1}, \sigma]$, for all $k \in \{1,2,3,4,5\}$, and

- The difference in attractiveness $u_i - u_j$ for each pair of alternatives $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in C_6$ is as small as possible.

For this purpose, two new sets of auxiliary variables, denoted by γ_{ij} and δ_{ij}, are introduced.

These variables are essentially under-achievement and over-achievement quantities, respectively, with respect to the ability of placing the difference in attractiveness $u_i - u_j$ for some pair of alternatives $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in C_k$ at the midpoint of the corresponding interval $[\sigma_{k-1} + \theta, \sigma_k]$, for all $k \in \{1, 2, 3, 4, 5\}$. The meanings of these two variables are illustrated in Figure 3.3.



a) The case where $u_i - u_j$ is smaller than the midpoint of $(\sigma_{k-1}, \sigma_k]$

b) The case where $u_i - u_j$ is smaller than the midpoint of $(\sigma_{k-1}, \sigma_k]$

FIGURE 3.3: *Graphical illustration of the meanings of the auxiliary variables γ_{ij} and δ_{ij}*

The second stage of the MACBETH procedure is achieved by solving another linear programming problem in which the objective is to

$$\text{minimise} \sum_{\substack{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in C_k \\ k\neq 6}} (\gamma_{ij} + \delta_{ij}) \tag{3.26}$$

subject to the constraints

$$\begin{array}{rclll}
\sigma_0 = 0 \quad \text{and} \quad \sigma_1 & = & 1, & & (3.27)\\
\sigma_k - \sigma_{k-1} & \geq & 1, & k \in \{2,3,4,5\}, & (3.28)\\
u_i - u_j & \geq & \theta, & (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathbb{X}^2, & (3.29)\\
u_n & = & 0, & & (3.30)\\
\sigma_{k-1} + \theta - c_{\min} \;\leq\; u_i - u_j & \leq & \sigma_k + c_{\min}, & (\boldsymbol{x}_i, \boldsymbol{x}_j) \in C_k, \quad k \in \{1,2,3,4,5\}, & (3.31)\\
\sigma_5 + \theta - c_{\min} & \leq & u_i - u_j, & (\boldsymbol{x}_i, \boldsymbol{x}_j) \in C_6, & (3.32)\\
\dfrac{\sigma_{k-1} + \theta + \sigma_k}{2} + \delta_{ij} - \gamma_{ij} & = & u_i - u_j, & (\boldsymbol{x}_i, \boldsymbol{x}_j) \in C_k, \quad k \in \{1,2,3,4,5\}, & (3.33)\\
\gamma_{ij}, \delta_{ij} & \geq & 0, & (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathbb{X}^2, & (3.34)\\
\sigma_i & \geq & 0, & i \in \{1,2,3,4,5\}, & (3.35)\\
u_j & \geq & 0, & j \in \{1,\ldots,n\}. & (3.36)
\end{array}$$

Constraint sets (3.27)–(3.30) are identical to (3.17)–(3.20), respectively, while constraint sets (3.31)–(3.32) are similarly equivalent to (3.21)–(3.22), respectively, except that the value of c has now been fixed at c_{min} (the value obtained by solving the linear programming problem of the first stage). Constraint set (3.33) furthermore ensures that the difference in attractiveness $u_i - u_j$ for each pair of alternatives $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in C_k$ is as close as possible to the midpoint of the corresponding difference interval $(\sigma_{k-1}, \sigma_k]$, for all $k \in \{1, 2, 3, 4, 5\}$. Constraint sets (3.34)–(3.36) finally ensure that all variables are non-negative.

A numerical example

The working of the MACBETH procedure is illustrated by means of a small example in this section.

Example 3.2 *Suppose a DM categorises as follows six alternatives $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_6$ (ranked* a priori *in decreasing order of preference) according to the category definitions Table 3.7:*

$$\begin{aligned}
C_1 &= \{(\boldsymbol{x}_2, \boldsymbol{x}_3), (\boldsymbol{x}_3, \boldsymbol{x}_4), (\boldsymbol{x}_4, \boldsymbol{x}_5), (\boldsymbol{x}_5, \boldsymbol{x}_6)\},\\
C_2 &= \{(\boldsymbol{x}_2, \boldsymbol{x}_4)\},\\
C_3 &= \{(\boldsymbol{x}_1, \boldsymbol{x}_2), (\boldsymbol{x}_2, \boldsymbol{x}_5), (\boldsymbol{x}_3, \boldsymbol{x}_5)\},\\
C_4 &= \{(\boldsymbol{x}_1, \boldsymbol{x}_3), (\boldsymbol{x}_1, \boldsymbol{x}_4)\},\\
C_5 &= \{(\boldsymbol{x}_1, \boldsymbol{x}_5), (\boldsymbol{x}_2, \boldsymbol{x}_6), (\boldsymbol{x}_3, \boldsymbol{x}_6), (\boldsymbol{x}_4, \boldsymbol{x}_6)\}, \text{ and}\\
C_6 &= \{(\boldsymbol{x}_1, \boldsymbol{x}_6)\}.
\end{aligned}$$

The pairwise comparison matrix corresponding to this categorisation is

$$\boldsymbol{A} = \begin{array}{c} \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \\ \boldsymbol{x}_5 \\ \boldsymbol{x}_6 \end{array} \begin{array}{c} \begin{array}{cccccc} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \boldsymbol{x}_3 & \boldsymbol{x}_4 & \boldsymbol{x}_5 & \boldsymbol{x}_6 \end{array} \\ \left[\begin{array}{cccccc} - & 3 & 4 & 4 & 5 & 6 \\ - & - & 1 & 2 & 3 & 5 \\ - & - & - & 1 & 3 & 5 \\ - & - & - & - & 1 & 5 \\ - & - & - & - & - & 1 \\ - & - & - & - & - & - \end{array}\right] \end{array}.$$

Solving the linear programming problem (3.16)–(3.25) for these data and for a value of $\theta = 0.001$ yields the variable values $c = 0.667$ and

$$\begin{array}{llllll}
\sigma_0 = 0.000, & \sigma_1 = 1.000, & \sigma_2 = 2.000, & \sigma_3 = 3.000, & \sigma_4 = 4.000, & \sigma_5 = 8.667,\\
u_1 = 8.001, & u_2 = 5.334, & u_3 = 3.667, & u_4 = 3.334, & u_5 = 1.667, & u_6 = 0.000.
\end{array}$$

The DM is therefore not considered to be perfectly consistent (since $c > 0$). The value function obtained from a linear interpolation of the above values of $\sigma_0, \ldots, \sigma_5$ and $u_1, \ldots, u_6$ is shown graphically in Figure 3.4.

The second phase of the MACBETH procedure involves solving the linear programming problem (3.26)–(3.36) for $c_{\text{min}} = 0.667$ and $\theta = 0.001$. This yields the variable values

$$\begin{array}{llllll}
\sigma_0 = 0.000, & \sigma_1 = 1.000, & \sigma_2 = 2.000, & \sigma_3 = 3.000, & \sigma_4 = 4.000, & \sigma_5 = 5.000,\\
u_1 = 6.834, & u_2 = 4.500, & u_3 = 4.000, & u_4 = 3.334, & u_5 = 1.667, & u_6 = 0.000
\end{array}$$

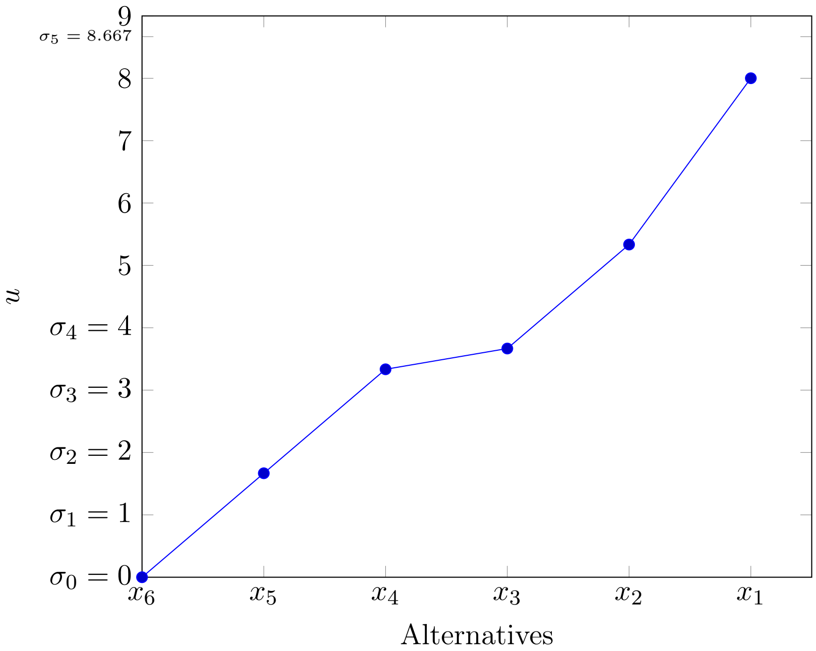


FIGURE 3.4: *First-phase MACBETH-generated value function and difference intervals for Example 3.2.*

together with the equitability auxiliary variables values

γ_{ij}	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$
$i=1$	0.1665	0.6660	0.0000	0.0000	—
$i=2$	—	0.0000	0.3340	0.0000	0.0000
$i=3$	—	—	0.0000	0.1675	0.5005
$i=4$	—	—	—	0.0000	1.1665
$i=5$	—	—	—	—	0.0000

δ_{ij}	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$
$i=1$	0.0000	0.0000	0.0000	0.6670	—
$i=2$	—	0.0000	0.0000	0.3330	0.0000
$i=3$	—	—	0.1655	0.0000	0.0000
$i=4$	—	—	—	1.1665	0.0000
$i=5$	—	—	—	—	1.1665

The updated value function obtained from a linear interpolation of the above values of $\sigma_0, \ldots, \sigma_5$ *and* $u_1, \ldots, u_6$ *is shown graphically in Figure 3.5.*

Note that there are only three pairs of auxiliary variables γ_{ij} *and* δ_{ij} *for which* $\gamma_{ij}+\delta_{ij} = 0$*, namely* $(i,j) = (1,4),(2,3),(2,6)$*. These pairs of variables indicate that the value function differences* $u_1 - u_4$*,* $u_2 - u_3$ *and* $u_2 - u_6$ *could be placed exactly in the centres of the difference intervals corresponding to the categories assigned by the DM for these pairs of alternatives (the intervals* $(3,4]$*,* $(0,1]$ *and* $(4,5]$*, respectively). In all other cases, value function differences under- or overachieved in terms of these centres in order to preserve the level of DM consistency measured during the first phase of the MACBETH procedure. For example, the underachievement variable value* $\gamma_{12} = 0.167$ *indicates that the difference* $u_1 - u_2$ *is* 0.167 *value function units left of the centre of the difference interval* $(2,3]$ *assigned by the DM to the pair of alternatives* $(\boldsymbol{x}_1, \boldsymbol{x}_2) \in C_3$*. Similarly, the overachievement variable value* $\delta_{15} = 0.667$ *indicates that the difference* $u_1 - u_5$ *is* 0.667 *value function units right of the centre of the difference interval* $(4,5]$ *assigned by the DM pair of alternatives* $(\boldsymbol{x}_1, \boldsymbol{x}_5) \in C_5$*.* □

Extension to multiple attributes

Although the MACBETH procedure has hitherto been described in the context of DM preferences in respect of a single decision attribute, it is possible to incorporate multiple attributes in the procedure, as in the case of the AHP. If there are m decision attributes, the procedure described

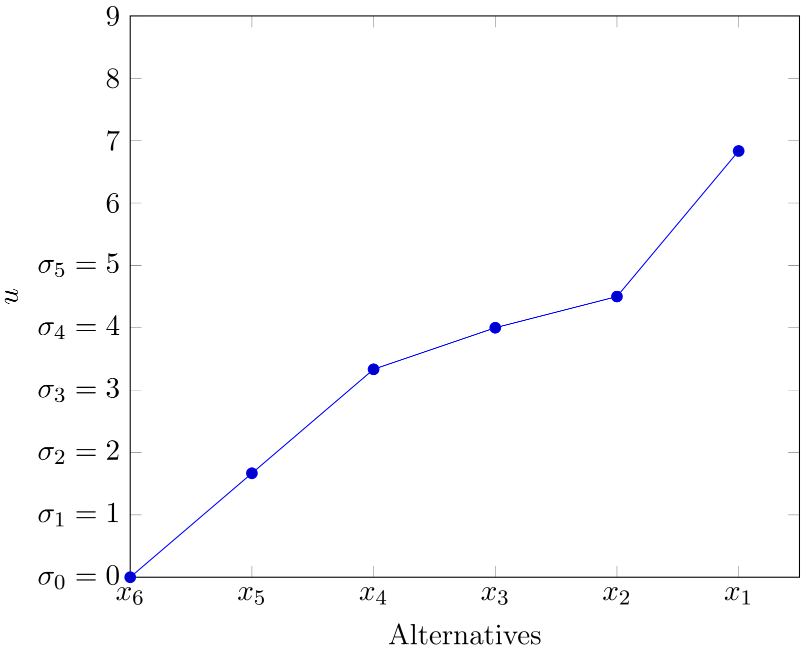


FIGURE 3.5: *Second-phase MACBETH-generated value function and difference intervals for Example 3.2.*

above is applied to construct a value function $u_k : \mathbb{X} \mapsto \mathbb{R}^+$ for each attribute $k \in \{1, \ldots, m\}$ independently of the other attributes. A set of functions $\{\phi_1, \ldots, \phi_m\}$, all defined on $\mathbb{R}^+$, is then constructed with the property that

$$\boldsymbol{x}_i \succ_{\text{comb}} \boldsymbol{x}_j \quad \text{if and only if} \quad \sum_{k=1}^{m} \phi_k[u_k(\boldsymbol{x}_i) - u_k(\boldsymbol{x}_j)] > 0, \tag{3.37}$$

where the binary relation $\succ_{\text{comb}}$ denotes a combined preference by the DM in respect of all m attributes. The sum in (3.37) may therefore be interpreted as a combined value function for the DM in terms of all the attributes.

For a discussion of how the functions $\phi_1, \ldots, \phi_m$ may be constructed, the reader is referred to 1969 paper by Tversky [214], the 1987 paper by Roy and Vincke [218], and the 1992 paper by Fishburn [66].

3.3 Interactive preference learning

Whereas the value function construction methods described in §3.2 are classic methodologies which were designed for application in the context of requiring DM preference information specification at the start of the MCDM problem solution process only, the focus in this penultimate section of the chapter turns to more modern methodologies which have been proposed for application in the context of requiring DM preference information specification throughout the MCDM problem solution process. The area in which methods of the latter type are researched is known as *interactive preference learning*, which is closely related to the fields of *interactive optimisation* (in operations research) and *reinforcement learning* (in computer science).

According to Sebag *et al.* [195], the area of computer science was preoccupied by specifications in the form of formal languages establishment and theorem proving during the 1970s and 1980s. During the 1990s and early 2000s, the mainstream focus gradually shifted to programming by examples, with pattern recognition and machine learning coming into their own. It is only since the early 2010s, however, that interactive learning and interactive optimisation have emerged as active research areas. This section is devoted to a very brief and high-level overview of research endeavours in the field of interactive preference learning. Although the field is young, the literature has already grown large. The objective in this section is therefore not to present an exhaustive review of this research field, but rather to highlight the variety of problems studied recently and solution approaches adopted recently within this field and certain other fields related to it.

The research field of preference learning is a subfield of the larger field of interactive optimisation. These fields are, in turn, related to the field of reinforcement learning, which is a subfield of the larger field of machine learning, as illustrated in Figure 3.6. Research activities related to learning the preferences of human subjects within each of these four fields are highlighted in the remainder of this section.

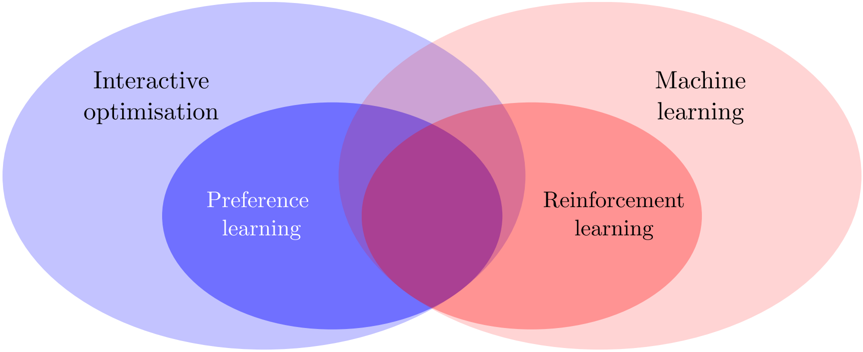


FIGURE 3.6: *The subfields of preference learning and reinforcement learning, as well as their respective "parent" fields of interactive optimisation and machine learning.*

3.3.1 Interactive optimisation

The origins of interactive optimisation can be traced back to a so-called black-box optimisation paradigm in which an underlying objective function of the form $u : \mathcal{D} \mapsto \mathbb{R}$ is assumed on a decision space $\mathcal{D}$ and in which the objective is to uncover an argument in $\mathcal{D}$ that maximises the value of u. The definitive feature of this type of optimisation problem, however, is that the function u is not known *a priori*, but is instead discovered as the optimisation process unfolds. This discovery has traditionally been facilitated by iteratively involving some form of registering or measurement of responses from the environment, and later on, exposure to human DMs. In some cases these DMs act as impartial agents relaying information about the environment to the learning system, and in others they convey subjective preferences to the learning system. The challenge is generally to prompt as few responses as possible from the environment or human in the loop by pursuing an effective trade-off between exploration and exploitation of the decision space $\mathcal{D}$ [3].

An example of this kind of optimisation is the problem of online path planning for optimal sensing by a mobile robot, considered by Martinez-Cantin *et al.* [148] in 2009, in which the objective of the robot is to learn as much as possible about its environment, given a time constraint. The learning process is facilitated by the iterative construction of a utility function as the robot moves through its environment, allowing it to replan its responses as more of the environment is experienced in this finite-horizon planning problem. Robot responses were determined by maximising the most recent utility function estimate, which required solution of a high-dimensional, continuous, non-differentiable, nonlinear optimisation problem in real time. There was no human in the loop during the robot's learning process.

Garnett *et al.* [68] studied two interesting interactive binary classification problems in 2012, again not involving a human subject in the optimisation process. In the first problem, called *active search*, the objective was to uncover as many members of a given class of objects as possible in a given search space. In the second, called *active surveying*, the objective was to query points with the aim of ultimately predicting the proportion of a given class of objects in some search space. In both cases, it is not known *a priori* which areas of the search space contain the desired class objects, but it is possible to successively request binary observations at any point in the search space. The authors demonstrated that numerous real-world problems take one of these forms, and they tackled these problems *via* a Bayesian decision theoretic approach. After selecting natural utility functions, they derived optimal policies for solving these problems.

In 2013, Knox *et al.* [122] presented a case study in which they applied a framework for training a physically enabled robot from numeric human feedback. They also provided the first demonstration of the possibility of eliciting multiple behaviours by the robot from a single instance of such feedback without algorithmic modifications and of a robot learning from free-form feedback generated by a human without any additional guidance or evaluative feedback.

3.3.2 Preference learning

In preference learning, the objective is to learn the subjective preferences of a specific human subject in some task performing or decision making context. Preference learning is therefore, by definition, interactive — in the absence of interaction with the human subject, his preferences cannot be elicited. Preference learning may furthermore be considered an interactive optimisation process in which the human subject's value function (the objective function) is discovered as the optimisation process unfolds.

There have been many instances in the literature over the last decade in which preference learning system designs have been put forward. In 2010, for example, Viappiani *et al.* [216] considered the problem of maximising the expected value of information gained from repeated interaction with a human DM. Since the problem of expected value of information maximisation is, however, typically prohibitively expensive from a computational point of view, the authors investigated the option of involving so-called *choice queries* in the optimisation process (the situation where the DM is repeatedly asked to identify his or her most preferred alternative from a set). It was shown that under very general assumptions, solutions to this problem coincide with an optimal recommendation set (a set maximising the expected utility function of the DM).

In 2011, Gulwani [81] considered the problem of designing a string expression language which supports restricted forms of regular expressions[6]. The objective was that the language should be expressive enough to represent a wide variety of string manipulation tasks with which end-users often struggle. To this end, he designed an algorithm for synthesising a desired program in this language from input-output examples. The algorithm is interactive, can rank multiple solutions, can detect noise in the user input, and supports an interaction model in which the user is prompted to provide outputs corresponding to inputs that may have multiple computational interpretations.

[6] In theoretical computer science and in formal language theory, a *regular expression* (sometimes also called a *rational expression*) is a sequence of characters which defines a search pattern. Such a pattern is normally used by a string searching algorithm to *find* or to *find and replace* operations on strings [2].

The notion of *coactive learning* was proposed in 2012 by Shivaswamy and Joachims [200] as a model of the interaction between a learning system and a human DM which both have the common goal of providing results of maximum value to the DM. At each step of the iterative process, the system (such as a search engine, for example) receives a context (such as a query) and predicts a result (such as a ranking of resources). The DM then responds by correcting the system, if such an action is deemed necessary, providing it with a slightly improved (but not necessarily optimal) object or resource as feedback. The authors argued that such feedback can often be inferred from observable DM behaviour, such as from the DM's clicks in a web-search, for example.

In 2013, Jain *et al.* [107] considered the problem of learning good search trajectories for manipulation tasks. This is a challenging problem because the criteria underlying the definition of a good search trajectory vary for different users, tasks and environments. The authors adopted a coactive online learning approach towards teaching robots the preferences of humans for object manipulation tasks. The novelty in their approach is the type of feedback expected from the user: The human in the loop is not required to demonstrate optimal search trajectories as training data. Instead, he is merely required to provide trajectories in an iterative fashion that slightly improve the trajectory currently proposed by the machine. This coactive preference feedback may be elicited more easily from the user than demonstrations of optimal search trajectories. The generalisability of the approach toward a variety of household tasks was demonstrated.

In 2015, Branke [25] proposed an interactive multi-objective evolutionary algorithm for learning a DM's value function. At regular intervals, the user is prompted to rank a single pair of solutions in terms of preference. This information is then used to update the internal value function model of the algorithm. The model is used in subsequent generations of the search process to rank solutions according to dominance in the Pareto sense. In this way, the algorithm converges quicker toward the region of the Pareto front which is most desirable to the DM.

In practical machine learning systems, performance tuning typically requires a more nuanced approach than merely minimising a single expected loss objective. This performance tuning process may often be considered more profitably as a multi-objective optimisation problem. In 2016, Dewancker *et al.* [53] proposed a novel generative model for estimating scalar-valued utility functions in pursuit of capturing human preferences in a multi-objective optimisation context. They also proposed an interactive learning system that iteratively refines the understanding of DMs' ideal utility functions by employing binary preference queries.

3.3.3 Machine learning

The origins of machine learning may be traced back to the 1940s and 1950s, and may be described as the field of study in which computers are given the ability to learn without being programmed explicitly as to the learning that should take place. It may be regarded as a general approach towards achieving artificial intelligence [44, 156]. The parsing of data, learning meaning from it, and drawing a conclusion (such as implementing a decision or making a prediction, for example) are considered to form the foundations of machine learning [44]. One of the key contributors to the usefulness of machine learning is the ability to extrapolate patterns that are detectable in data so as to be applicable to new problems [181]. One of the major drawbacks associated with the field, however, is the data-hungry nature of traditional machine learning methods which usually require large volumes of data in order to facilitate effective algorithmic learning [191]. There are four main learning paradigms within the field of machine learning. These are supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning.

In *supervised learning*, the machine is presented with both input data and output data (in other words, labelled data) and the objective is to approximate an underlying function that maps a number of input features to target variables. Classification and regression are popular supervised machine learning techniques which have found application in a variety of areas, such as handwriting classification, speech recognition, computer vision, and spam detection [46, 72, 108, 132, 196]. In *unsupervised learning*, on the other hand, unlabelled data are instead provided and the machine is required to discover structure or patterns within the data. Self-organising maps, nearest-neighbour mapping, and k-means clustering are popular supervised machine learning techniques [104, 106]. In *semi-supervised learning*, a combination of labelled and unlabelled data is employed. Classification, regression and clustering techniques may also be implemented within a semi-supervised machine learning paradigm. Face recognition from a video feed and web-page classification are two examples of semi-supervised learning applications [191]. Finally, in *reinforcement learning*, the machine is exposed to a dynamic environment and is expected to decide on a course of action or policy that maximises a suitable reward quantification by a process of trial and error. Robot movement and disease detection are two examples of reinforcement learning applications [208, 228].

There are many examples of the application of machine learning methods in pursuit of mimicking human behaviour. In 2010, for example, Liang *et al.* [138] expressed an interest in machine learning programs for multiple, but related, tasks, given only a small number of training examples per task which are observed from human behaviour. They introduced a nonparametric hierarchical Bayesian prior distribution over programs which share statistical properties across multiple tasks. The key challenge was to parameterise this multi-task sharing and to transfer knowledge gained from training examples in the context of one task to that of another task.

Computational creativity is a relatively new subfield of artificial intelligence in which computational models of creative thought in science and the arts are studied. From an engineering point of view, it is important to be able to assess the progress made from one version of a computational creativity machine learning program to another so that different software systems for the same creative task can be compared and contrasted in a concrete manner. In 2011, Pease and Colton [165] described a number of versions of the Turing test[7] which have been used to measure progress in computational creativity. The authors argued, however, that the Turing test is not appropriate for use within the context of computational creativity, because it tends to homogenise creativity into a single (human) style, does not take cognisance of the importance of background and contextual information for a creative act, and encourages superficial and uninteresting advances. They introduced two descriptive models as alternatives to the Turing test for evaluating creative software.

Although most preference learning methodologies in the literature reside within the area of reinforcement learning, there are examples of other machine learning systems aimed at preference learning. In 2014, for example, Akrour *et al.* [3] advocated a machine learning-based programming framework, called *programming by feedback*, which involves a sequence of interactions between a machine and a human DM in which the DM only provides preference judgements on pairs of solutions supplied to him or her by the machine. The actions carried out by the machine

[7] Designed by Alan Turing in 1950, the *Turing test* is a test of a machine's ability to exhibit intelligent behaviour that may be considered equivalent to, or indistinguishable from, that of a human [43]. According to the test, a natural language conversation between a human and a machine designed to generate human-like responses is evaluated by a human evaluator. The evaluator is aware that one of the two partners in the conversation is a machine (but does not know which is which). The conversation is limited to a text-only channel (such as a computer keyboard and screen, so that the result of the test can be independent of the machine's ability to render words as speech). If the evaluator cannot reliably tell the machine from the human, the machine is said to have passed the test.

are based on two components. The first is a learning component which estimates the DM's value function and accounts for the DM's comparison incompetence (*i.e.* inconsistency with respect to pairwise comparison articulations). The second is an optimisation component which explores the search space and returns what it considers the most appropriate candidate solution.

In 2016, Holzinger [93] noted that whereas automatic machine learning approaches typically benefit from large quantities of data and many training sets, practitioners in the health domain are sometimes confronted with a small number of data sets or very rare events, where automatic machine learning approaches often fail. He argued that interactive machine learning may be applied profitably in such cases, having its roots in preference learning and active learning. A human-in-the-loop approach can be beneficial when attempting to solve computationally hard problems such as subspace clustering, protein folding, or k-anonymisation of health data. In these applications, human expertise can aid in reducing exponential search spaces through the heuristic selection of samples. In this way, what would otherwise be an NP-hard problem, may reduce greatly in complexity as a result of the assistance of a human agent involved in the machine learning phase.

3.3.4 Reinforcement learning

As mentioned in the previous section, reinforcement learning is a machine learning paradigm in which a virtual agent is expected to respond to feedback from a (typically dynamic) environment by deciding on actions that maximize some notion of cumulative reward by a process of trial and error [209]. The agent is not programmed as to which actions to take, but is rather expected to discover which actions yield the best reward by implementing them and observing the response of the environment. Agent actions may affect not only the immediate reward, but also future rewards. The trial-and-error nature of the search process for good actions and the notion of delayed or future reward are the main distinguishing features of a reinforcement learning approach. One of the main challenges that result from these characteristics is that it is often hard to find the right balance between exploration and exploitation during the trial-and-error search process of the agent.

In addition to the learning agent and environment in which the agent finds itself, there are three further main elements of any reinforcement learning system: a policy, a reward function and a value function [209]. The *policy* is a mapping from the perceived states of the environment to actions that can be taken when the environment is in a given state. The policy therefore defines the agent's behaviour for any given state of the environment. The *reward function* represents the goal of the reinforcement learning problem — it maps each perceived state of the environment to a corresponding reward value which represents the desirability of that state. Whereas the reward function quantifies which environment states are desirable in the short term, the *value function* measures which environment states will be good in the long run. The value function score of a state is usually interpreted as the total reward the agent can possibly accumulate in the future, discounted appropriately. The ultimate goal of the agent, as it searches through the action space by trial and error, is to maximise the value function, not the immediate reward. For an overview of early work in reinforcement learning, the reader is referred to the excellent survey paper by Kaelbling *et al.* [111]. More recent survey papers are also available [28, 123].

A number of interesting preference-based reinforcement learning approaches have recently been proposed in the literature. In 2016, for example, Wirth *et al.* [225] adopted an approach to preference-based reinforcement learning that requires only pairwise comparisons between trajectories as feedback over the standard approach of numeric reward function specification, which usually results in considerable hand-tuning of parameters by a human expert. They integrated preference-based estimation of the reward function into a model-free reinforcement learning algorithm. The algorithm utilises stochastic policies to control the greediness of the policy updates directly. Preference-based reward function estimation is performed *via* a sample-based Bayesian approach, which is also able to estimate the uncertainty associated with utility. The authors showed that it is possible to learn non-parametric continuous action policies from a small number of elicited preferences.

Also in 2016, Holladay *et al.* [92] noted that in order to learn human preferences, robots have to interact with humans. They employed comparison-based learning to facilitate this interaction (*i.e.* prompting the articulation of learning preferences by asking a user to compare several alternatives). They argued that in some settings it may be difficult for a human to articulate which alternative he or she prefers. To minimise user burden, they therefore proposed active learning, but avoided forcing the human to articulate preferences when considering what he deemed to be difficult comparisons. Instead, they harnessed this information (on which comparisons are hard to perform) to learn DM uncertainty. Based on this information, the authors designed an algorithm for modelling uncertainty and used the resulting uncertainty estimates to select and process comparison queries.

In 2017, Christiano *et al.* [38] recognised that for sophisticated reinforcement learning systems to interact effectively with real-world environments, it is necessary to communicate complex goals to these systems. They explored goals defined in terms of human preferences between pairs of search trajectory segments and showed that this approach can resolve complex reinforcement learning tasks effectively without access to a reward function. They applied this approach to Atari games and to simulated robot locomotion, providing feedback in less than 1% of the learning agent's interactions with the environment, thus reducing the cost of human oversight far enough that it could be applied to state-of-the-art reinforcement learning systems.

3.3.5 The learning approach adopted in this dissertation

The DSS proposed later in this dissertation for the facilitation of SPBP instances may be described most accurately as residing within that portion of the field of preference learning in Figure 3.6 that does not overlap with the field of machine learning. Instead of adopting a sophisticated machine learning approach toward preference learning, it is investigated in this dissertation whether the much simpler approach of extending a standard MCDM methodology, such as the AHP or MACBETH described in §3.2, to within the realm of interactive optimisation can hold benefits in terms of a reduced cognitive burden on the part of the human in the loop when learning a DM's value function in an iterative fashion.

3.4 Chapter summary

The purpose of this chapter was to provide the reader with a brief overview of certain central notions from the realm of preference learning which are required in order to facilitate a general understanding of the material presented in the remainder of this dissertation.

The chapter opened in §3.1 with an introduction to various basic concepts related to preference modelling. These concepts included an overview and classification of the different types of MCDM problems considered in the literature, the standard practice of DM preference articulation in the form of expressing binary relations with respect to decision alternatives, and the fundamental notion of a value function as a numerical representation of DM preference.

Detailed descriptions of two of the best-known methods for value function construction were then provided in §3.2. These methods were the seminal AHP (a ratio-based value function construction method) and the subsequent MACBETH procedure (a difference-based value function construction method).

A review of modern developments in the area of interactive preference learning then followed in §3.3. This review touched upon the topics of interactive preference modelling (in §3.3.1), preference learning (in §3.3.2), machine learning (in §3.3.3) and reinforcement learning (in §3.3.4). The specific learning approach adopted in this dissertation was finally elucidated in §3.3.5 within the context of the aforementioned material contained in the literature review.

CHAPTER 4

Decision support systems

Contents

Since the main goal in this dissertation is to put forward an interactive DSS for facilitating the streamlined solution of SPBP instances, it is important to take cognisance of the different types of DSSs available in the literature, their constituent components, methodologies that have been documented for their orderly design, as well as techniques available for the verification and validation of DSSs. This chapter is devoted to a review of the academic literature related to these topics. After describing in §4.1 a number of DSS taxonomies that have been proposed in the literature, the focus shifts in §4.2 to a discussion on the main components of such systems. Three major design paradigms for DSSs are then reviewed briefly in §4.3, and this is followed in §4.4 by descriptions of a number of techniques that are routinely used during the verification and validation of DSSs.

4.1 Types of DSSs

DSSs have evolved substantially over the last sixty years from rather simple model-oriented systems to very advanced multi-function platforms. During the 1960s, DSSs were mostly based on powerful and rather expensive mainframe computers [155]. These systems typically provided managers and DMs with structured, periodic reports facilitating informed decision making. The arrival of management information systems theory during the 1970s saw the evolution of DSSs into more elaborate computerised systems that supported decisions related to production, promotion, pricing, marketing and logistical functions in businesses [5]. By the early 1980s, DSSs started enjoying more interest from academia, which resulted in knowledge and principles of sound practice related to DSSs being expanded considerably [173]. It was only during the 1990s, however, that a significant paradigm shift occurred in the theory of DSSs, which saw the design of considerably more complex systems incorporating advanced database technologies as well as client/server capabilities. As organisations worldwide started upgrading their computer network infrastructure, object-oriented technology and data warehousing started to influence the design characteristics of DSSs. The rapid expansion of the internet towards the end of the twentieth century provided additional opportunities for the scope of development of DSSs [192], which saw these systems embracing online and cloud analytical processing, as well as web accessibility, at the start of the twenty first century.

Various taxonomies have been put forward in the literature for DSSs. These taxonomies differ based on the criterion selected to distinguish between the different types of DSS. Haettenschwiler [82], for example, suggested the following differentiation between DSSs based on the *nature of their interaction* with users as the distinguishing criterion: passive DSSs, active DSSs and cooperative DSSs. A *passive DSS* is designed to aid in the process of decision making, but does not suggest explicit solutions or decision recommendations. An *active DSS*, on the other hand, can make recommendations with respect to decision suggestions or courses of action. A *cooperative DSS* accommodates an iterative process between the user and the system, aimed at achieving a common goal or consolidated solution. In this interaction, the user can modify, correct, complete or refine the decision recommendations of the system, before resubmitting these recommendations back into the system for validation. The system then again has the opportunity to improve, complete or refine the updated suggestions, and this process can be repeated.

Power [169] suggested the following alternative DSS taxonomy, this time according to the *mode of assistance* of the system as the distinguishing criterion:

A communication-driven DSS facilitates cooperation by supporting multiple users working on a task of common interest. Examples include integrated tools such as Google Docs [77] or Microsoft Groove [203].

A data-driven DSS, also sometimes referred to as a data-oriented DSS, facilitates access to and manipulation of internal company data (or, in some cases, external data) in the form of a time series.

A document-driven DSS facilitates management, retrieval and manipulation of volumes of unstructured information in a variety of electronic formats.

A knowledge-driven DSS provides access to an array of specialised problem-solving expertise which is usually stored as a collection of facts, rules and/or procedures.

A model-driven DSS provides access to a combination of statistical, financial, optimisation or simulation models. These systems take as input data parameters provided by the user to aid in the analysis of some situation.

Taking *scope* as the distinguishing criterion, Power [168] differentiated between enterprise-wide DSSs and desktop DSSs. An *enterprise-wide DSS* is typically linked to a large database or data warehouse and usually serves a variety of managers in a company. A *desktop DSS*, on the other hand, is a small-scale system that is able to run on an individual user's personal computer.

Based on the aforementioned taxonomies, the DSS put forward in this dissertation for supporting decisions related to the solution of SPBP instances may be described as a cooperative, model-driven, desktop DSS.

4.2 Typical components of a DSS

According to Marakas [146], as well as Sprague and Carlson [202], there are three fundamental hardware and software components of a typical model-driven DSS architecture.

These components are the following:

The database and the data warehouse, together also known as the *knowledge base*, are storage facilities for schemas, tables, reports, views and other information elements related to the purpose of the DSS, as well as the software and protocols necessary to query and update these data.

The model base is a collection of various representations of the real underlying situation or system being studied, composed of a number of concepts which may be used by DMs to know, understand, or simulate the subject supported by the system. The model base captures the decision context and the various user criteria for decision making.

The user interface is the space allowing for interaction between humans and machines. The objective of this interaction is to facilitate effective operation and control of the DSS from the human end, whilst the DSS is simultaneously able to feed back information *via* this space to aid DMs.

It is, however, acknowledged that the (human) users of a DSS may themselves also be considered important components of a DSS architecture. This is especially true in the decision support context of this dissertation.

4.2.1 The database and the data warehouse

It has long been recognised that a DSS is only as good as the underlying database supporting it [155]. In order to provide a DSS with sensible and correct data, there should be a mechanism for facilitating access to the data contained in the database, as well as for the storage, manipulation and retrieval of large amounts of data. The database may be considered the information feeder of the DSS and sound database design is therefore crucial in order to ensure overall high-quality performance of the DSS. A good database management system should ensure data integrity, attempt to reduce data redundancy, follow a logical sequence of actions and exhibit consistency in its performance [192].

Relational databases are usually the paradigm of choice when it comes to the design of a DSS database. The reason for this is not only the flexibility associated with a relational database, but also its facilitation of data normalisation (and the resulting reduction in data duplication), which is essential in the maintenance of large databases [173]. The ability of such a database to identify and manipulate relationships between data entities renders the facilitation of effective information analyses an important desirable feature of a relational database. Although hierarchical databases and network databases are still in use today, this is primarily because of the exorbitant costs typically involved in the migration of data from such databases to new database platforms (these costs are often significantly more than the maintenance costs incurred by legacy systems) [192].

Since the database constitutes such an important link in the overall capability and effectiveness of any DSS, its structure and design should be considered carefully and implemented with due consideration of the various applications in the model base that depend on it. There was a general trend during the early years of the twenty first century to migrate to web-based DSSs and subsequently to employ task-specific search engines, or to build DSSs around a so-called "thin client, fat server environment," as well as employing network and web-based search and storage technologies [159]. This situation was exacerbated by the rise of the internet, the most expansive network of interconnected databases and web pages in existence.

Databases usually contain *current* information about transactional and other processes of the underlying real-world system, but they often fail to provide rich information that is sometimes more important in decision making processes than stand-alone islands of information [192]. The notion of a *data warehouse* fulfils this role by capturing historical operational data and decision actions, representing it in a context provision format within a relational database [164]. In this way, a data warehouse is capable of complimenting the functions of the database within the DSS. More specifically, the data warehouse and the database coexist to provide synergistic outcomes that support the various information requirements of the DSS, superimposed on a systems platform with the context providing the benefit of knowing and truly understanding the unfolding of the past. Research into data warehouses has resulted in the use of intelligent virtual agents that mine meaning out of large volumes of past data in order to assist in terms of the speed and simplicity of *ad hoc* data queries.

4.2.2 The model base

The model base may be considered as the mathematical or computational power house of a model-based DSS. It typically houses a suite of mathematical or statistical models for which parameters may be specified by the user *via* the user interface. Once populated with parameter values, these models may be solved. The resulting model solutions may then be combined and interpreted, and these interpretations may be relayed back to the user *via* the user interface. Such interpretations may take the form of complex system response explanations or recommendations in terms of future courses of action.

The models in the DSS model base are usually designed for use in either static or dynamic analyses of the underlying real-world system supported by the DSS [224]. In a *static analysis*, no consideration is given to the long-term temporal response of the underlying system. Such an analysis typically involves taking a single snapshot in time of the prevailing situation in the underlying system, assuming that it will remain constant over time. This type of analysis is typically much simpler than its dynamic counterpart. In a *dynamic analysis*, on the other hand, the repercussions over time of a potential decision taken in the underlying system are considered. In other words, this type of analysis takes into account how the situation in the underlying system may change over time due to changes in costs, rules, resource availability, *etc.*

There are four types of models that may populate the model base of a DSS [158, 174]:

An algebraic model solves sets of sophisticated equations or performs complex numerical analytic procedures that are difficult for a human to perform by hand.

A contemplative model forecasts results, situations or outcomes that may realise in the underlying system as a result of adopting a specific set of parameters.

A descriptive model explains why some aspect of the underlying system is the way it is, as well as why and how it works the way it does.

A prescriptive model is designed to perform some kind of optimisation procedure in order to come up with a recommended course of action within the underlying system.

Popular examples of models that are routinely included in the model bases of model-driven DSSs are [89, 144]: Accounting and financial models (including break-even analyses and capital budgeting models), decision analysis models (including multi-criteria decision models, machine learning models, decision trees and influence diagrams), forecasting models (including moving average and exponential smoothing methods, time series extrapolation, regression methods and econometric models), network and optimisation models (including linear programs, combinatorial optimisation models and graph theoretic models), and simulation models (including discrete-event simulation models, agent-based models and system dynamics models).

Prompted by the literature reviewed above, the DSS for SPBP solution facilitation put forward in the next chapter comprises four main components: a database, a model base (consisting of multiple components), a user interface and the user whose preferences are being considered.

4.2.3 The user interface

The effectiveness of a *computerised* DSS depends largely on its user interface design [145]. This is the case for both a routine software program on the low end of the sophistication spectrum and for a high-end DSS. It is universally acknowledged that the user's interaction with the DSS should be as simple and effective as possible [110].

The most important role of a DSS user interface is that it determines how users interact with the DSS. The desirable features of such a user interface are therefore that it [110, 138]:

- balances technicality (*i.e.* the functionality of the DSS) with mentality (*i.e.* the state of mind and the ability of the user),
- provides the user a picture-oriented (or visual) means to interact with the DSS,
- facilitates flawless to-and-fro communication and interaction between the user and the DSS,
- reduces errors, increases operational speed, and supports sound decision making, and
- is at the same time both logical and intuitive.

There are six dominant user interface technologies, although it is common that these technologies are not employed in isolation, but rather in combination [145]:

A command-line user interface primarily makes use of text commands to induce actions by the system. The user would, for example, enter a command such as "run," and the system would subsequently execute the command. Well-known examples of this type of interface are the MS-DOS [150], UNIX [210] and LINUX [139] operating systems. Whereas such interfaces may be powerful and fast, the user is required to learn the commands available in order to operate the system, which can be a very severe restriction.

A menu user interface typically offers users a variety of functionalities in the form of a list of choices. A drop-down menu is usually employed to make it easier for the user to select a functionality for execution. The advantage of this type of interface is that the user does not have to learn commands, but the disadvantage is that space restrictions often limit the number of options that can be included in a drop-down menu.

A graphical user interface allows the user to issue commands through visible objects by either pointing at or clicking on images, icons or other symbols in order to perform actions. A distinguishing feature of this type of interface is that it typically focuses more on multimedia rather than on text communication.

A question-and-answer user interface allows the DSS to ask questions directly to the user, who then answers these questions in some form of dialogue. This type of interface is the result of an attempt at mimicking human-to-human interaction. A major challenge associated with this type of interface, however, is the facilitation of structured responses by the user in order to avoid the situation where the DSS does not "understand" the responses by the human.

A voice user interface facilitates human-machine interaction through speech. A human voice is required to operate the DSS or to cause it to perform an action. This type of interface has become commonplace as speech recognition systems have improved.

A touch screen user interface is the most popular and also the most recent type of interface. This type of interface relies upon the sensing of human touch, directing the system to perform a selected action when a user touches a particular visual object.

Since the focus of the DSS design put forward in this dissertation is on concept demonstration rather than on actual, practical computer implementation, its user interface is not particularly emphasised. The most accurate description of the type of user interface envisaged for inclusion in the DSS proposed in this dissertation, however, is that it would be a combination of a menu user interface and a graphical user interface.

4.3 DSS design methodologies

A *systems development methodology* is defined by Walters *et al.* [222] as a collection of operations, tools, techniques and documentation methods which may assist a systems analyst in the development of a DSS. Systems development methodologies usually share a number of core elements or phases in their development processes which together form the so-called *systems development life cycle* [114, 52]. Although there is no general consensus in the literature on the number of phases in such a systems development life cycle, Kendall and Kendall [114] described the seven phases illustrated in Figure 4.1.

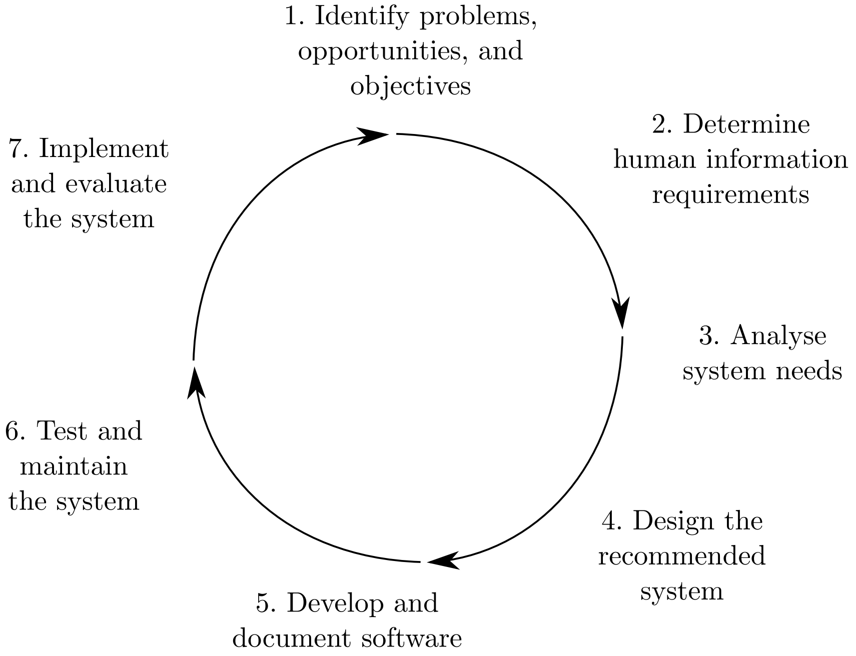

1. Identify problems, opportunities, and objectives
2. Determine human information requirements
3. Analyse system needs
4. Design the recommended system
5. Develop and document software
6. Test and maintain the system
7. Implement and evaluate the system

FIGURE 4.1: *The systems development life cycle according to Kendall and Kendall [114].*

While the systems development life cycle depicted in Figure 4.1 certainly forms the foundations of systems design and development, a number of variations on the theme may be found in the literature. The major differences of approach involve the order in which design and development phases are performed, as well as the amount of time and the number of resources allocated to each phase. The characteristics of arguably the three most popular and well-documented systems development life cycle incarnations are described in this section, namely the structured waterfall design methodology, the agile design methodology, and the object-orientated design methodology.

4.3.1 The structured waterfall design methodology

According to Modha *et al.* [152], the first systems development methodologies emerged toward the end of the so-called pre-methodological era, when computers were introduced to the business world in the 1960s. During this time, computer systems were developed without following any pre-defined design methodologies; instead, developers rather relied on their experience. Despite the fact that the systems analysts of the time generally had adequate programming skills, Avison and Fitzgerald [8] argued that they often failed to achieve a true understanding of the organisations for which they designed systems or the contexts within which these systems were to be used. They claimed that this shortcoming arose predominantly because of a lack of adequate communication between the systems analysts and their clients or the final users of their systems. This lack of communication reportedly often resulted in poorly documented user requirements [222] which, in turn, resulted in systems that did not provide the value to the users that they could have [9]. This left the market desiring a better structured and more concise approach to the design and development of information systems and DSSs. This desire marked the start of the early methodological era, resulting in the establishment of the first systems development methodologies, known as *structured methodologies* [8, 9]. These methodologies are distinguished by their carefully documented processes and systematically arranged design and development phases.

One such structured methodology is the *waterfall design methodology*, described by Dennis *et al.* [52] as a top-down and precise approach according to which the methodology's phases are completed in a sequential fashion. Rob [176] explained that it is not possible to return to previous phases upon reaching any particular phase in the waterfall methodology. Instead, one can only proceed through the various stages as if descending in phases over a series of cliffs. During each of the phases, the approval of the DSS stakeholders is required before the process may proceed to a next phase — hence the descriptor *waterfall.* Several variations on this methodology have been advocated since the original model proposed by Royce [180], but the version by Dennis *et al.* [52] partitions the waterfall methodology into four distinct phases, namely *planning*, *analysis*, *design* and *implementation*, as illustrated in Figure 4.2.

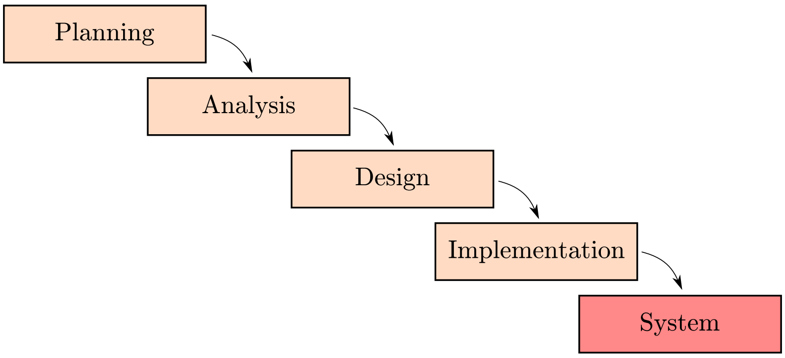


FIGURE 4.2: *The structured waterfall design methodology according to Dennis* et al. *[52].*

An advantage associated with the waterfall methodology is the fact that it is often easier to manage the overall development process as a result of the systematic development phases of the methodology, because milestones may be set for each of the phases and the achievement of these milestones may be measured at regular development updates. Another advantage is that the requirements of the users and system specifications are known in advance as a result of careful documentation and so the design process can commence knowing the end goal [52].

There are also a number of drawbacks associated with the waterfall design methodology [52, 222]. The benefits of the structured waterfall design and development process may, in fact, become liabilities. If there is not much time to revise design work, the challenge of designing an entire DSS theoretically on paper may be rather daunting. Furthermore, if errors were to be discovered later during the development process, or further system requirements come to light, moving back up to the waterfall in order to accommodate these additional requirements may present significant challenges. Finally, the rather long development process may sometimes result in a system that meets the original user requirements, but one that is no longer of particular use, since there may have been significant changes to the business environment during the design process.

4.3.2 The agile design methodology

After the early era in system design described above, the so-called *methodological era* emerged and with it there were major advances with respect to tools and techniques that could be harnessed by systems analysts during the design process of information systems and DSSs. These additional tools included, but were not limited to, data dictionary software, project management software, and computer assisted software engineering. These tools employed normalisation methods, entity relationship diagrams, and data flow diagrams [8].

As a result of the disadvantages of the traditional design methodologies of the previous era, new methodologies were proposed during the next era which made use of many of these newly developed tools and software [9, 222]. One of the first systems development methodologies which came about during this era was the so-called *agile methodology*, which was put forward in an attempt at speeding up the process of system development. The agile methodology furthermore places considerable emphasis on the identification and appropriate adherence to system user requirements, and also favours systems that are practical and work effectively, as opposed to those designed hand in hand with voluminous documentation [52]. Kendall and Kendall [114] explained that the result of embracing the agile design methodology is faster system delivery. Moreover, system developers are able to introduce changes and improvements to the system at any time during the development, a desirable trait in the software development environment.

The disadvantages associated with the agile methodology are often more the result of an incorrect implementation thereof or because of human error, as opposed to being deficiencies of the methodology *per se*. Examples of such “disadvantages” include developers who demand spending long periods of time documenting their desired design plans and developers who do not provide working portions of code or prototypes which demonstrate that the larger design vision is indeed practical.

4.3.3 The object-oriented design methodology

Another system design methodology that emerged during the methodological era is the *object-oriented methodology*. According to Rob [176], this method may be described as a bottom-up approach to system design, as opposed to the aforementioned top-down waterfall methodology. Kendall and Kendall [114] explained that according to the object-oriented methodology, the system is iteratively partitioned into objects or entities (containing data) by means of so-called use case models (which describe how the system is expected to perform). The data in these objects may be locations, events, people, or actual components of the system, all of which may be classified into classes of similar objects that share certain characteristics.

A *use case model* consists of four elements, namely *actors* (usually denoted by stick men), *use cases* (usually denoted by ovals), *relationships* (usually denoted by curves or lines), and the *system boundary* (usually denoted by a box). The actors represent particular roles played by individuals or groups of individuals in the underlying system supported by the DSS. The use cases are the various functions performed within the underlying system, while the relationship curves and lines indicate which actors are associated with which use cases. The system boundary finally represents the scope of the system in relation to the entire underlying system [52]. For each use case model, an associated activity diagram is also created, enveloping all possible use case situations. *Activity diagrams*, comprising so-called *swim lanes* illustrate which actors are required to perform which activities, showcase the progression of the main activities of the underlying system as well as indicate the relations between these activities [114]. This is accomplished by employing the symbols shown in Figure 4.3.

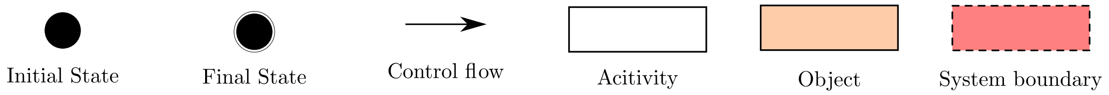


Figure 4.3: *Activity diagram symbols used in the object-oriented methodology [52, 114].*

Use case models and activity diagrams are often employed in conjunction with the well-known *unified modelling language*. This language may be thought of as providing an assortment of tools that may be used to capture the analysis and design phases of system development. The unified modelling language is incorporated in the object-oriented design methodology [52, 114, 176].

Comparison of system design methodologies

There is consensus in the literature on information system and DSS design that no single one of the aforementioned development methodologies may be considered universally superior [222]. Instead, each design methodology brings with it specific advantages and disadvantages [114]. Furthermore, these strengths and weaknesses result in each methodology being best suited for different types of scenarios and underlying systems, as highlighted in Table 4.1.

	Methodology		
Ability to develop systems	Waterfall	Agile	Object-oriented
with unfamiliar technology	Poor	Poor	Poor
with unclear user requirements	Poor	Excellent	Average
that are reliable	Good	Good	Good
that are complex	Good	Poor	Good
with schedule visibility	Poor	Good	Average
within a short time frame	Poor	Excellent	Good

Table 4.1: *Strengths and weaknesses of the DSS design methodologies described in this section [52, 114].*

The DSS put forward in the next chapter of this dissertation was designed according to a combination of elements from the agile system design methodology and the object-oriented system design methodology described above.

4.4 DSS verification and validation

The credibility of a DSS is evaluated during the processes of verification and validation. As Kleijnen [121] states, however, there is no universally agreed-upon definition in the literature of the distinction between the processes of verification and validation. The most widely adopted definitions of these terms, by Finlay and Wilson [65], are adopted in this dissertation.

Verification is the process of demonstrating that a framework or system is capable of performing as intended, by comparing its performance with user requirements. Verification is, therefore, ideally assessed on a continual basis during framework or system development. A graphical illustration of four different types of verification and validation that should ideally be carried out during the development of a DSS is provided in Figure 4.4. The inner circle of the figure indicates the individuals who should form part of the relevant testing process.

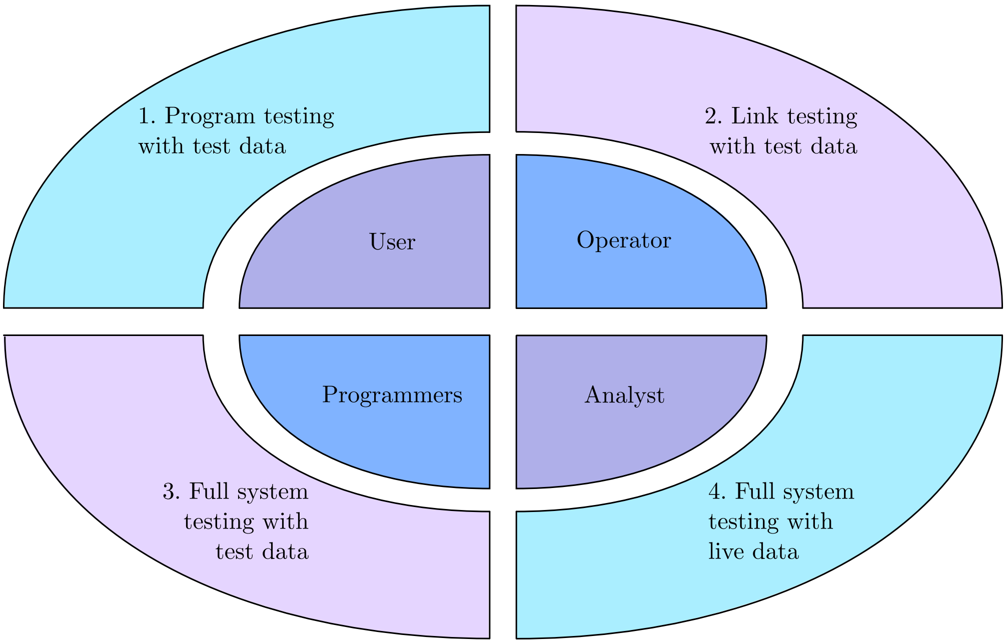


FIGURE 4.4: *Four steps of system verification (adapted from Kendall and Kendall [114]).*

The first step in Figure 4.4 involves system verification by means of test data. This type of verification should occur continually throughout the implementation phase of a system. Every new portion of executable code should be verified in isolation, using small samples of test data. These test data may be artificial samples of data generated by the systems analyst. The second step is to link the various testing efforts by means of test data. While the previous step involves isolated testing of small portions of executable code, this step involves verification of all code portions together in an ensemble in order to ensure that the entire system functions as planned. The third step involves full system verification using test data. In this step, the user is involved and is requested to run a few scenarios through the system, using a large set of test data. The final step in Figure 4.4 is full system testing with live data instead of test data. Live data here refers to real past data gathered from the true underlying system, for which the correct DSS output is known. In this manner, the results of the new system can be compared with the results of whichever system came before it.

Banks *et al.* [17], as well as Law and David [130], proposed the following pertinent verification techniques in the context of simulation modelling which are equally valid within the context of general DSS design:

1. Generate a flow chart of all logically possible actions that the system can take for all the different eventualities that may occur. For each of these actions associated with each event type, evaluate the system logic thoroughly.

2. Evaluate the system output for soundness based on a range of possible input parameter values. The system should display a range of output data or statistics which should be evaluated carefully.

3. Print the system input parameters upon system initialisation, as well as intermediate results computed by the system at certain strategic junctures, and ensure that these parameter values and results are sensible/correct.

4. Provide a clear definition of every parameter and variable used in the system implementation, as well as a description of the purpose of each system component or procedure.

5. Execute the system for an extensive period of time so as to eliminate the possibility of missing errors that may not occur during short runs.

According to Finlay and Wilson [65], the aim of *validation* is to establish a potential user's level of confidence in the system. It is essentially the process of ensuring that a system is a true representation of the real world and its requirements, thereby ensuring that the system is adequately calibrated, and that the output of the system is "correct" and able to meet the real needs of users. Like verification, the process of validation must also preferably be performed during the development phase of a decision support framework or system, which is usually iterative and incremental by nature.

Law and Kelton [129] suggested three types of validity to be considered during the validation process, namely *conceptual validity, operational validity* and *credibility.*

Conceptual validity is related to whether a model or system is, in fact, a valid representation of the underlying real-world system [129]. The most popular type of validation technique employed to confirm conceptual validity is *face validation*, which involves asking a collection of individuals who are knowledgeable in terms of the underlying real-world system whether the model or system (and its behaviour) are reasonable from a practical point of view [190]. A *Turing test* may also be employed for this purpose. In this case, outputs from the real underlying system, as well as model or system outputs, are provided to a subject matter expert (someone who is knowledgeable about the underlying real-world system), and the expert is asked to distinguish between the real and model or system outputs [131, 190]. If this cannot reliably be done, then the system is considered conceptually validated.

Operational validity is concerned with the question of whether the output of a model or DSS is in line with the behavioural data of the underlying real-world system [129]. This type of validity is typically confirmed by means of *results validation*, and is usually only possible if real-world data are available for comparison purposes. The comparison analysis may involve a wide range of statistical analyses in order to assess whether or not the model or DSS output is significantly different (in a statistical sense) from that of the real-world system [131, 190]. Additionally, an operationally valid model should exhibit reasonableness, in the sense that the model should exhibit *continuity, consistency* and *degeneracy.* Three types of tests may be included under the heading of operational validity, namely continuity tests, consistency tests and degeneracy tests.

Continuity implies, if small changes are affected to the input parameters of the model or DSS, that these should be reflected in the corresponding outputs and variable values as similarly small changes [129].

Consistency implies that the output should be similar for separate runs of the model or DSS with the same input parameters (in other words, the model output should, for example, not change significantly as a result of a change in the random number generator seed).

Degeneracy requires that the model or DSS should reflect the removal of one or more objects appropriately. For example, if a supermarket outlet has three tellers, and one of these tellers is closed, the effect should be reflected appropriately in the model or DSS output [129, 190]. A test that may be employed for testing degeneracy is known as the *extreme condition test.* When carrying out this test, inappropriate input parameter values are specifically chosen in order to ascertain whether or not an appropriate effect is exhibited by the model or DSS. If, for example, the inventory level of some raw material is set to zero in a simulation model of a production plant, the production rate of all products containing that raw material should also be equal to zero until inventory replenishment of the raw material in question occurs [190].

Credibility is determined by the end-user of the model or DSS by answering the questions related to the project objectives. The user will trust a credible model or DSS. If, however, the user does not trust a model or DSS, the course of action recommended by such a model or DSS will likely not be implemented [129].

4.5 Chapter summary

The literature review in this chapter focused on guidelines from the literature in terms of DSS design, verification and validation. The objective was to draw inspiration from the literature in terms of these guidelines so as to be in an informed position when proposing a DSS for solving SPBP instances later in this dissertation. The various types of DSSs documented in the literature were outlined in §4.1. These DSS types are communication-driven DSSs, data-driven DSSs, document-driven DSSs, knowledge-driven DSSs and model-driven DSSs. Thereafter, the main components found in a model-driven DSS (the kind of DSS proposed later in this dissertation) were described, paying special attention to their individual functions and combined interaction. These components are a database, the model base and the user interface. Three well-known DSS design methodologies were then reviewed in §4.3, namely the structured waterfall design methodology, the agile design methodology and the object-oriented design methodology. Methods of DSS verification and validation were finally outlined in §4.4.

Part II

Decision Support framework

CHAPTER 5

Framework Design

Contents

As mentioned in Chapter 1, the aim in this dissertation is to propose an effective and efficient framework for evaluating a set of subjective alternatives and learning user preference, as well as to build a predictive preference model and embed this model (in conjunction with an optimisation or exploration algorithm) within a DSS in support of solving SPBP instances with large solution spaces in which no obvious objective functions are available. This chapter contains a detailed description of the design of a DSS put forward to achieve this aim.

The chapter opens in §5.1 with a high-level overview of the envisaged working of the DSS after which the system component architecture is elucidated in a generic manner in §5.2, with a focus on the flow of data between the various system components. These system components are described in more detail in §5.3, referring to the various functional building blocks that make up the components. The efficiency enhancement element of the proposed DSS hinges on the method according to which value function estimates of alternatives already considered by the DM are updated as the DM becomes aware of new alternatives. This updating method is described in detail in §5.4. The focus of the discussion then shifts to a temporal description in §5.5 of the sequence of events proposed to take place when executing the DSS. The chapter finally closes with a brief summary of its contents in §5.6.

5.1 High-level process overview

At a high level of abstraction, the working of the DSS put forward in this chapter conforms to the interactive learning process illustrated in Figure 5.1. This process consists of three sub-processes, namely *preference evaluation*, *utility learning* and *alternatives discovery*.

The sub-process of *preference evaluation* is carried out by a human DM, also called the *user*, and typically entails the specification of pairwise comparison information in respect of a subset of solution alternatives, adopting either a ratio-based or a difference-based approach, as described in §3.2.1 or §3.2.2, respectively. This information is then aggregated according to a method such as those described in §3.2 in order to form scores for each of the alternatives. This sub-process may take place in the context of either a single selection criterion or multiple selection criteria.

The alternative scores serve as input data to the *value function learning* sub-process which is carried out by a machine, based on a user-preferred interpolation method. During this sub-process, the user's value function in respect of the particular SPBP instance is estimated over the entire solution space. This estimated value function may then be used to predict user preference of alternatives in parts of the solution space not yet seen or considered explicitly by the user.

The predictive ability of the approximate value function learned above, according to which anticipated user preference scores may be attached to hitherto unexplored regions of the solution space, is utilised during the *alternatives discovery* sub-process. An exploration of the solution space is carried out by the machine during this sub-process. This exploration is accompanied by an exploitation of those areas of the solution space found to contain promising alternatives. This exploration and exploitation of the solution space involves application of an optimisation technique, such as those described in §2.4.

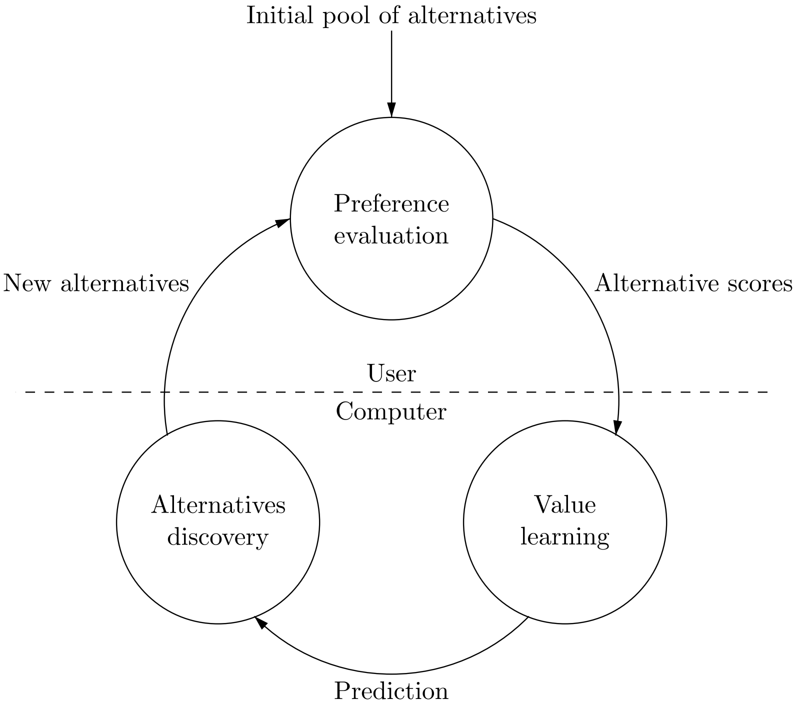


Figure 5.1: *High-level overview schematic for the process of interactive preference learning adopted in the DSS proposed in this dissertation.*

Based on the results of the above-mentioned exploration and exploitation of the solution space, a small subset of strategically selected new solution alternatives is then presented to the user for pairwise comparison. This subset of new alternatives is purposefully selected so as to facilitate rapid and meaningful updates to the user's value function estimate. The selection may, for example, comprise a combination of particularly good and poor solution alternatives encountered thus far during the learning process (according to the current estimate of the user's value function) so as to promote exploitation of promising regions of the solution space, as well as randomly generated new alternatives (so as to promote exploration of unknown regions of the solution space).

The interactive learning cycle described above, which involves cooperation between the human DM and the machine, is repeated sequentially until some termination criterion is met, typically specified in terms of an appropriate measure of value function estimation convergence or in terms of the number of pairwise comparisons of solution alternatives that the user is prepared to carry out.

5.2 Decision support system architecture

The preference selection decision support framework put forward in this chapter for proposing desirable preference alternatives in the context of single-criterion or multi-criteria SPBP instances with large solution spaces in which no obvious objective functions are available, relies heavily on the DM and his input. As outlined in §3.2.1 and §3.2.2, either a ratio-based or a difference-based approach may be adopted to capture a DM's preferences in the context of the type of SPBP described above.

It is proposed that the three high-level interactive preference learning sub-processes of Figure 5.1 are facilitated by four DSS components, namely the *human DM or user*, a *system configuration* component, a *database* and a *data management* component, as shown in grey in Figure 5.2. The output data of the system configuration component should serve as input data for the data management component *via* the database.

Four sub-components make up the system configuration component, namely a *method of value function construction* sub-component, an *exploration & exploitation method and parameters* sub-component, an *initialisation method for set of prediction alternatives* sub-component, and a *search termination features* sub-component. Each of these sub-components is assumed to be pre-populated with a selection of suitable methods, parameters and criteria from which the user can choose a preferred combination to configure the DSS for application in the context of a particular SPBP instance. This entire system configuration is then stored in the database from whence it can be retrieved by the data management component upon DSS execution. The user-specified system configuration determines the manner in which the DSS functions. Examples of constituent methods, parameters and criteria housed in each of the four sub-components of the system configuration component are described in some detail in the following section.

The data management component furthermore consists of five sub-components. An *extraction of alternatives* sub-component is responsible for selecting, from a large set of potentially desirable solution alternatives for which estimated or user-specified value function data have not yet been learned, a suitable subset of alternatives to be presented to the user for which pairwise comparison information is sought. Upon receipt from the user, this pairwise comparison information is aggregated to form alternative scores by a *score aggregation* sub-component and combined with existing alternative score information so as to construct an updated value function estimate by a *value function construction* sub-component. Based on this value function estimate, a search is launched in solution space by an *exploration and exploitation* sub-component with the aim of discovering alternatives that score highly in terms of the approximate value function learned thus far. As mentioned in §5.1, this may be achieved within an optimisation paradigm (where the value function assumes the role of a fitness function), by applying a technique such as the PHC algorithm of §2.4.2, for example. Desirable alternatives thus discovered are stored in the database and the entire process is repeated iteratively. During each iteration, the user is presented with a

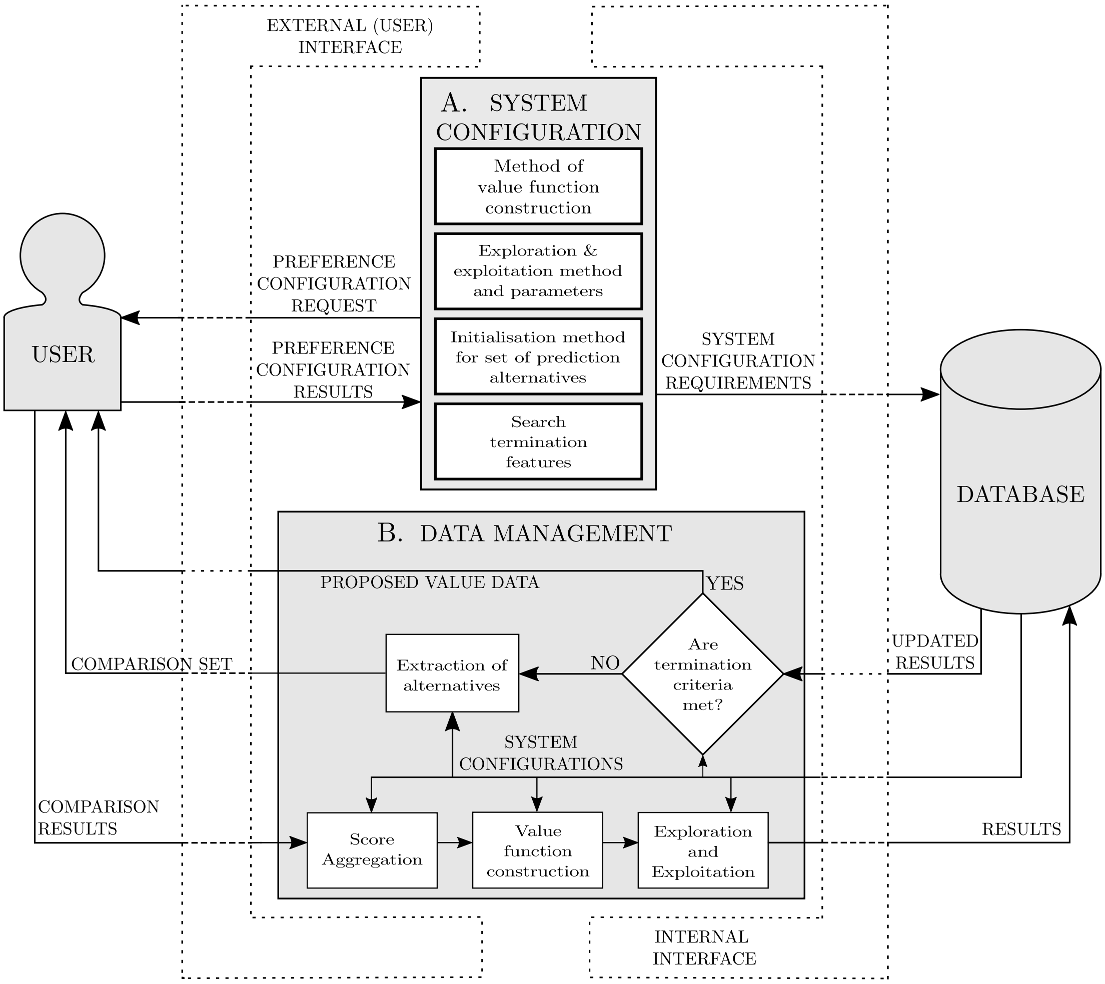

EXTERNAL (USER) INTERFACE
A. SYSTEM CONFIGURATION
Method of value function construction
Exploration & exploitation method and parameters
Initialisation method for set of prediction alternatives
Search termination features
USER
PREFERENCE CONFIGURATION REQUEST
PREFERENCE CONFIGURATION RESULTS
SYSTEM CONFIGURATION REQUIREMENTS
DATABASE
B. DATA MANAGEMENT
PROPOSED VALUE DATA
YES
NO
Are termination criteria met?
Extraction of alternatives
COMPARISON SET
UPDATED RESULTS
SYSTEM CONFIGURATIONS
COMPARISON RESULTS
Score Aggregation
Value function construction
Exploration and Exploitation
RESULTS
INTERNAL INTERFACE

Figure 5.2: *Proposed DSS architecture together with an indication of data flow between components.*

new set of promising alternatives for which pairwise comparisons are sought, hopefully leading to a better estimate of the user's value function, and a new exploration and exploitation of the solution space. A final sub-component of the DSS data management component is responsible for determining when to terminate this iterative process. A more detailed description of the working of and interaction between the constituent sub-components of the data management component is also provided in the next section.

Note, therefore, that whereas the system configuration component is accessed only once by the user in respect of a particular SPBP instance, before DSS application within the context of that specific problem instance, the user thereafter interacts repeatedly with the data management component as it iterates through the learning process of Figure 5.1.

Finally, the presence of two interfaces is assumed to form part of the DSS. The first of these is an *external (or user) interface* which is responsible for presenting alternatives to the user in an intelligible manner and facilitating the retrieval of pairwise comparison information from the user in an effective manner. The second interface, an *internal interface*, is assumed between the DSS components and the database. The function of this interface is to convert data into specific formats required by the various DSS sub-components and to protect the database from infiltration by spurious data.

5.3 Components of the decision support system

In order for the DSS proposed in §5.2 to function as desired, each sub-component of the data management component has to be specified and configured appropriately by the user. As mentioned in §5.1, the (iterative) method of working of the DSS data management component is influenced by the once-off, user-specified DSS configuration facilitated by the system configuration component. The sub-components and working of each of these two components are described in more detail in this section, starting with the latter component.

5.3.1 The system configuration component

As briefly alluded to in §5.2, the system configuration component consists of four sub-components. The elements of these subcomponents are elucidated in Figure 5.3.

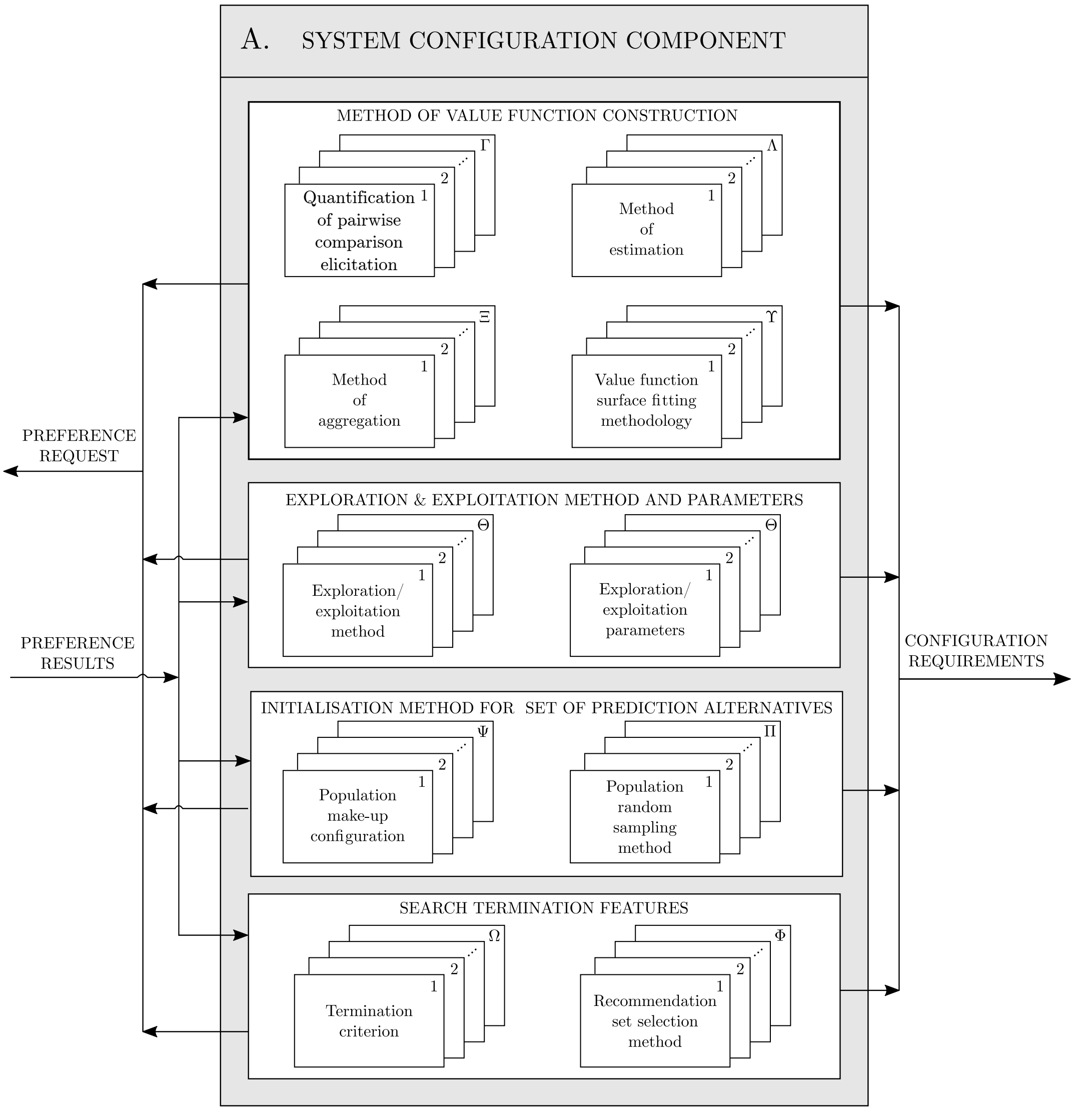

A. SYSTEM CONFIGURATION COMPONENT
METHOD OF VALUE FUNCTION CONSTRUCTION
Quantification of pairwise comparison elicitation
1
2
Γ
Method of estimation
1
2
Λ
Method of aggregation
1
2
Ξ
Value function surface fitting methodology
1
2
ϒ
EXPLORATION & EXPLOITATION METHOD AND PARAMETERS
Exploration/ exploitation method
1
2
Θ
Exploration/ exploitation parameters
1
2
Θ
INITIALISATION METHOD FOR SET OF PREDICTION ALTERNATIVES
Population make-up configuration
1
2
Ψ
Population random sampling method
1
2
Π
SEARCH TERMINATION FEATURES
Termination criterion
1
2
Ω
Recommendation set selection method
1
2
Φ
PREFERENCE REQUEST
PREFERENCE RESULTS
CONFIGURATION REQUIREMENTS

FIGURE 5.3: *DSS configuration sub-components and associated data flow.*

The *method of value function construction* sub-component facilitates specification by the DM of the method to be adopted during *quantification of pairwise comparison elicitation* of alternatives, the *method of estimation* to be used for the calculation of alternative scores, the *aggregation method* to be employed (in the case of multiple criteria being present in the SPBP instance under consideration), and the *value function surface fitting methodology* to be employed during value function estimation.

As illustrated in Figure 5.3, the *method of value function construction* sub-component is assumed to be pre-populated with Γ different methods of quantification of alternative pairwise comparison elicitation, from which the user is required to select one. Examples of such methods were provided in Tables 2.2 and 3.7. It is also assumed that Λ different methods of estimation are available, from which the user is again required to select one. Examples of these methods of estimation are the technique of linear programming described in §3.2.2 (in the case of adopting the MACBETH technique for value function construction) or the methods listed in Table 3.2 (in the case of adopting the AHP for value function construction). A total of Ξ pre-populated aggregation methods are also assumed to be available during alternative score calculation, from which the user should select one. Examples of such aggregation methods include construction of the aggregation functions $\phi_1, \ldots, \phi_m$ referred to in §3.2.2 (in the case of employing the MACBETH procedure for value function construction purposes), or the additive and multiplicative methods described in §3.2.1 (when employing the AHP for value function construction). Similarly, the user must select one of Υ possible methods of value hypersurface fitting. Examples of hypersurface fitting methods may include a simple linear interpolation of the value function score estimates computed during the value function aggregation process, or some more sophisticated interpolation method or regression method. This concludes the user's choices required within the *method of value function construction* sub-component.

The *exploration & exploitation method and parameters* sub-component of the system configuration component is similarly assumed to be pre-populated with Θ solution space exploration and exploitation methodologies (from which the user must again choose one) and their corresponding constituent operator sets and required parameters (for which the user may specify choices and values, respectively). Examples of such methodologies include the PHC algorithm described in §2.4.2, or alternatively the method of SA described in §2.4.3 (in the case of exploitation). In the case of the PHC algorithm, the constituent operator sets and required parameters include specification of the neighbourhood operator and the search termination criterion, while in the case of SA these include the initial temperature, the cooling schedule, the reheating schedule and the termination criterion.

The *initialisation method for set of prediction alternatives* sub-component is assumed to be pre-populated with Ψ population make-up configurations. These configurations may, for example, specify that different comparison set proportions should be reserved for exploitation alternatives and for exploration alternatives so as to pursue a suitable trade-off between search convergence and search diversity. In the case of a non-zero population proportion being allocated to randomly generated alternatives in a comparison set make-up configuration, it is also assumed that the *initialisation method for alternative prediction* sub-component is pre-populated with Π population random sampling methods, from which the user is required to select one. Examples of such sampling methods include sampling according to a uniform distribution or according to some other, specialised distribution, such as the method of Latin hypercube sampling (LHS)[1].

[1] A sampling method ensuring a full coverage of the entire specified region of a sample space. This is done by partitioning the sample space into n intervals along each dimension of the sample space and sampling so as to ensure a single observation in each axis-aligned hyperplane [149]

The *search termination features* sub-component is assumed to be pre-populated with Ω termination criteria, of which the user must select one. These criteria determine the number of learning cycles of Figure 5.1 to be executed. The *search termination features* sub-component is finally assumed to contain a set of pre-populated recommendation set selection methods according to which a final set of high-quality solution alternatives may be selected, based on estimated value function local maxima, for presentation to the DM as decision support. A simple example of such a method may be to select a pre-specified number of solution alternatives that achieve the largest estimated value function scores. Alternatively, a clustering approach, such as the k-means algorithm of §2.3.2, may be adopted according to which the solution space may be clustered into regions of similarly performing alternatives (based on the final value function estimate), returning representatives of each cluster only, and ranked according to decreasing estimated value function scores.

5.3.2 The data management component

After having captured the user's preferences in terms of the methods and parameters to be utilised during the evaluation of solution alternatives *via* the system configuration component, as described in the previous section, these preferences are passed to the data management component for use during all learning iterations of Figure 5.1 carried out by the latter component. The working of the data management component is illustrated graphically in Figure 5.4, summarised in pseudocode form in Algorithm 5.1 and elaborated upon in this section.

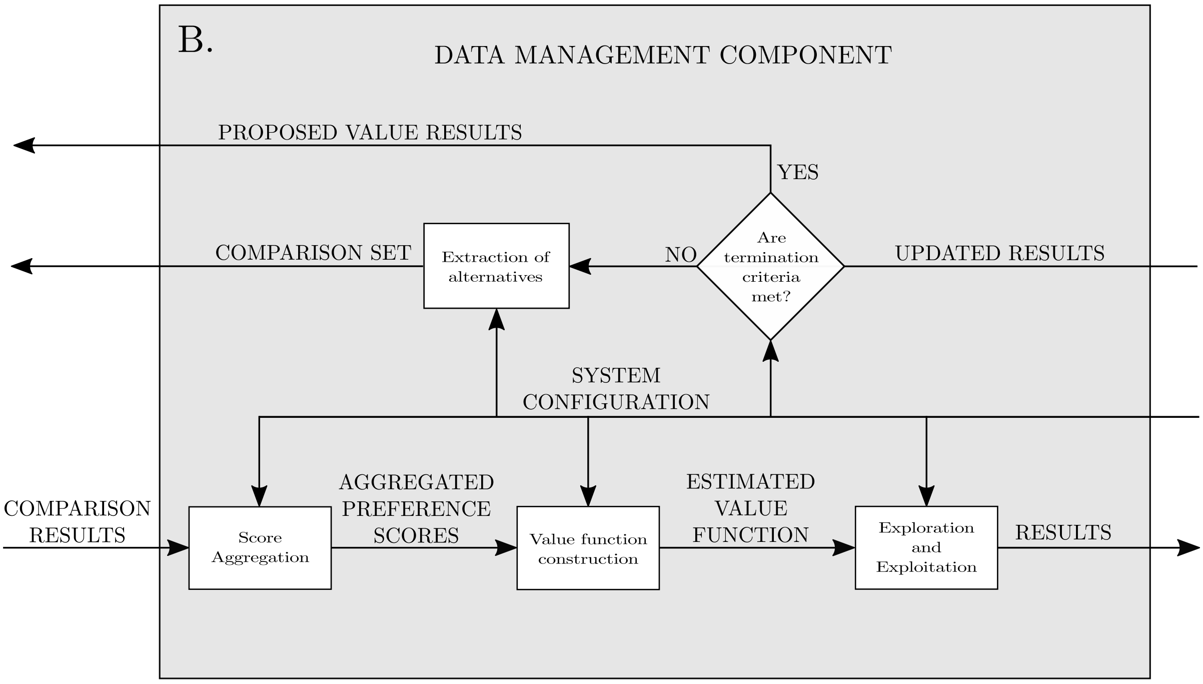

B.
DATA MANAGEMENT COMPONENT
PROPOSED VALUE RESULTS
YES
COMPARISON SET
Extraction of alternatives
NO
Are termination criteria met?
UPDATED RESULTS
SYSTEM CONFIGURATION
COMPARISON RESULTS
Score Aggregation
AGGREGATED PREFERENCE SCORES
Value function construction
ESTIMATED VALUE FUNCTION
Exploration and Exploitation
RESULTS

Figure 5.4: *Data flow within the DSS data management component.*

In the case of multiple selection criteria, the process starts in Steps 1–3 of Algorithm 3.1 with the elicitation of pairwise comparison information in respect of the various selection criteria and

Algorithm 5.1: Data management component

```
Input  : System configuration parameters & methods as specified by the DM via the
         System configuration component.
Output: Recommended alternative set.
1  if there is more than one selection criterion then
2      request pairwise comparison matrix for selection criteria;
3      compute selection criteria weights or aggregation functions;
4  anchor set ← an arbitrary subset of solution alternatives;
5  archive ←anchor set;
6  comparison set ← anchor subset ∪ {randomly generated additional alternatives};
7  for each selection criterion do
8      request pairwise comparison matrix for alternatives in comparison set;
9      compute scores for alternatives in comparison set according to estimation method;
10 compute value function scores for alternatives in comparison set according to aggregation
   method;
11 for each solution alternative $\boldsymbol{x}$ archive\anchor set do
12     rescale the value function score of $\boldsymbol{x}$;
13 archive ← archive ∪ comparison set;
14 fit a value function hypersurface through the (alternative, value function score) pairs in
   archive;
15 execute exploration & exploitation method i.r.o. the initial population with the value
   hypersurface estimate as fitness function or reward function;
16 best ← subset of solutions returned by exploration & exploitation method;
17 $\mathcal{S}$ ← subset of randomly generated (exploration) solutions generated according to
   population random sampling method;
18 comparison set ← best ∪ $\mathcal{S}$ ∪ anchor set according to population make-up configuration;
19 if stopping criterion is satisfied then
20     output recommendation alternative set according to recommendation set selection
       method and stop;
21     else go to Step 7
```

the computation of weights for these criteria and in terms of the judgement scale specified by the DM via the system configuration component as described in §3.2.1 (in the case of employing the AHP for value function construction purposes), or aggregation functions $\phi_1, \ldots, \phi_m$ as described in §3.2.2 (in the case of employing the MACBETH technique for value function construction).

The set of alternatives presented to the user over the course of working of the data management component is partitioned into various *comparison sets*. One such comparison set is presented to the user during each learning cycle of Figure 5.1. A fixed, arbitrarily chosen set of solution alternatives, called *anchors*, is included as subset in each comparison set. Under the assumption of a perfectly consistent user and with a view to decrease the pairwise comparison workload of the DM, this anchor set facilitates the generation of implicit comparison information in respect of solution alternatives from different comparison sets without requiring the user to compare alternatives in the different comparison sets explicitly. The anchor set is selected in Step 4 of Algorithm 5.1.

An *archive* is maintained dynamically over the entire learning process of the data management component. This archive contains all solution alternatives presented at various times to the user for pairwise comparison purposes, together with their estimated value function scores. The archive is initialised to contain only the anchor set in Step 5 of Algorithm 5.1. This anchor set is also included (together with additional solution alternatives) in the first comparison set in Step 6 of the algorithm. The additional alternatives are generated according to the *population random sampling method* specified earlier in the *initialisation method for set of prediction alternatives* sub-component of the system configuration component.

The learning cycle of Figure 5.1 spans Steps 7–18 of Algorithm 5.1. In Steps 7–8, pairwise comparison information is requested from the user for each selection criterion in respect of the current comparison set (again in terms of his preferred judgement scale, as specified earlier *via* the *method of value function construction* sub-component of the system configuration component). Each pairwise comparison matrix of Step 8 is used in Step 9 to estimate scores for the solution alternatives in the comparison set with respect to each selection criterion. The score value elicitation of Step 9 proceeds according to the user's preferred *method of estimation*, as specified earlier *via* the *method of value function construction* sub-component of the system configuration component. Using the aggregation weights or functions together with the scores computed in respectively Steps 3 and 9, a value function score is computed in Step 10 for each alternative in the comparison set. This computation occurs according to the user's preferred *method of aggregation*, as specified earlier *via* the *method of value function construction* sub-component of the system configuration component.

The archived value function score rescaling process in Steps 11–12 is nontrivial in each subsequent repetition of the loop spanning Steps 7–18, other than during the first iteration. In these steps, the aforementioned generation of implicit comparison information in respect of alternatives in the current comparison is brought in line with that of alternatives in previous comparison sets from a consistency point of view. This crucial step of the learning process is aimed at improving the efficiency of the process (in terms of the pairwise comparison burden imposed on the DM) and its working is described in detail in the following section.

The archive is updated in Step 13 to include the comparison set for which user preference information has most recently been obtained and in Step 14 a value function hypersurface estimate is fitted through the value function scores of solution alternatives in the archive. This hypersurface fitting occurs according to the user's preferred *value function surface fitting methodology* specified earlier in the *method of value function construction* sub-component of the system configuration component.

Depending on the user's preferred *exploration/exploitation method*, as specified earlier *via* the *exploration and exploitation method and parameters* sub-component of the system configuration component, the estimated value function fitted in Step 14 is considered as a fitness function over the decision space. Using the user's preferred *exploration/exploitation method* populated with preferred *exploration/exploitation parameters*, as specified earlier *via* the *exploration & exploitation method and parameters* sub-component of the system configuration component. Exploration and exploitation of the solution space is launched in Step 15.

The next comparison set is generated in Steps 16–18 of Algorithm 5.1. The make-up of this comparison set is structured according to the user's preferred *population make-up configuration* as specified earlier *via* the *initialisation method for set of prediction alternatives* sub-component of the system configuration component. The comparison set contains a pre-specified number of exploitation alternatives, as well as the anchor set and a number of randomly generated solution alternatives (again generated according to the user's preferred *population random sampling method* specified earlier *via* the *initialisation method for set of prediction alternatives* sub-component of the system configuration component).

If the user's preferred *termination criterion* (as specified earlier *via* the *search termination features* sub-component of the system configuration component) is met in Step 19, then a subset of the best alternatives is recommended to the user as desirable solutions to the SPBP instance in Step 20. This recommendation set is selected according to the user's preferred *recommendation set selection method*, as specified earlier *via* the *search termination features* sub-component of the system configuration component. At this point the algorithm terminates. Otherwise, if the stopping criterion is not satisfied, the data management component returns to Step 7 of Algorithm 5.1, repeating the learning cycle of Figure 5.1 contained in Steps 7–18.

5.4 Proposed method of DSS efficiency enhancement

Suppose the anchor set $\mathcal{X}$, described in the previous section, contains the χ alternatives $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_\chi$ and denote the remaining alternatives in the comparison set $\mathcal{C}_i$ presented in Step 8 of Algorithm 5.1 to the DM during iteration i of the learning cycle by $\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,n-\chi}$. Then there are n alternatives in $\mathcal{C}_i$, because $\mathcal{X} \subset \mathcal{C}_i$, and so the archive

$$\mathcal{A}_i = \bigcup_{j=0}^{i} \mathcal{C}_j$$

contains a total of

$$n + i(n - \chi) \tag{5.1}$$

distinct alternatives at the end of iteration i of the learning cycle in Figure 5.1, for all $i = 0, 1, 2, \ldots$ Denote the value function score of an alternative $\boldsymbol{y} \in \mathcal{C}_i$ during the i-th iteration of the learning cycle by $u_i(\boldsymbol{y})$.

As mentioned in §5.3.2, a method is required for rescaling the value function scores of the alternatives in the archive $\mathcal{A}_{i-1}$ during iteration i of the learning cycle in order to facilitate an implicit comparison of these alternatives with the new alternatives in $\mathcal{C}_i \setminus \mathcal{X}$ before augmenting the archive to $\mathcal{A}_i$.

5.4.1 Value function rescaling for a ratio-based approach

The method proposed for value function rescaling in the case of adopting a ratio-based value function estimation method (such as the AHP described in §3.2.1) employs the ratio between the geometric mean

$$\mu_i = \sqrt[\chi]{\prod_{j=1}^{\chi} u_i(\boldsymbol{a}_j)} \tag{5.2}$$

of the anchor value function scores aggregated during iteration i of the learning cycle and the geometric mean μ_{i-1} of the value function scores of these same alternatives during iteration $i-1$ (defined similarly). The ratio μ_i/μ_{i-1} is a measure of the level of consistency in the pairwise comparison information provided by the DM during iteration i of the learning cycle relative to that provided during iteration $i-1$. For a perfectly consistent DM it would hold that $\mu_i/\mu_{i-1} = 1$ for all $i = 1, 2, 3, \ldots$ The further this ratio deviates from unity, however, the more inconsistency is introduced by the DM when progressing from iteration $i-1$ to iteration i of the learning cycle.

The value function score of each alternative within the archive is therefore updated in Steps 11–12 of Algorithm 5.1 by means of the rescaling operations

$$u_i(\boldsymbol{y}) \leftarrow \left(\frac{\mu_i}{\mu_{i-1}}\right)^{\zeta} u_{i-1}(\boldsymbol{y}), \qquad \boldsymbol{y} \in \mathcal{A}_{i-1} \setminus \mathcal{X} \tag{5.3}$$

$$u_i(\boldsymbol{y}) \leftarrow \left(\frac{\mu_{i-1}}{\mu_i}\right)^{1-\zeta} u_i(\boldsymbol{y}), \qquad \boldsymbol{y} \in \mathcal{C}_i \tag{5.4}$$

for some parameter $\zeta \in (0, 1]$, called the *recency preference update weight*[2], before enlarging the archive in Step 13 of the algorithm during iteration i of the learning cycle. In this way, the relative desirability of the new alternatives $\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,n-\chi}$ are brought in line with that of the alternatives in $\mathcal{A}_{i-1} \setminus \mathcal{X}$ previously considered. The anchor set $\mathcal{X}$ therefore plays the role of intermediary alternatives through whose value function scores the desirability of previously considered alternatives may be updated as a result of the DM being exposed to new alternatives.

The function of the recency preference update weight parameter ζ in (5.3)–(5.4) is to allow the DM to specify the relative importance of the anchor value function scores resulting from pairwise comparisons during iteration i (*i.e.* the values $u_i(\boldsymbol{x}_1), \ldots, u_i(\boldsymbol{x}_\chi)$) when attempting to address inconsistency discrepancies between these scores and those determined during iteration $i-1$ (*i.e.* the values $u_{i-1}(\boldsymbol{x}_1), \ldots, u_{i-1}(\boldsymbol{x}_\chi)$). As illustrated in Figure 5.5, the aforementioned DM inconsistency is dealt with by inflating the scores of all (non-anchor) alternatives in the archive $\mathcal{A}_{i-1}$ by a factor μ_i/μ_{i-1} scaled exponentially by raising this factor to the power ζ, and deflating the scores of all alternatives in the most recent comparison set $\mathcal{C}_i$ by the same factor[3] raised to the power $1-\zeta$.

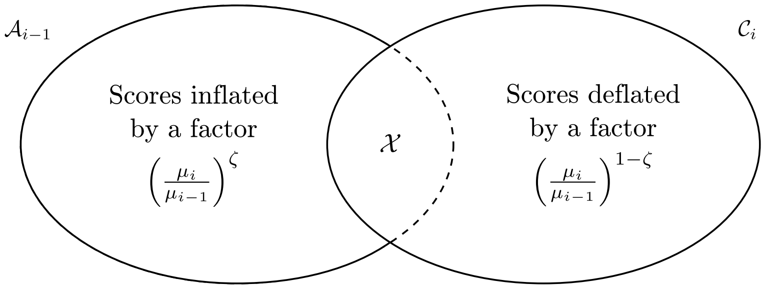

$\mathcal{A}_{i-1}$
$\mathcal{C}_i$
Scores inflated by a factor $\left(\frac{\mu_i}{\mu_{i-1}}\right)^{\zeta}$
$\mathcal{X}$
Scores deflated by a factor $\left(\frac{\mu_i}{\mu_{i-1}}\right)^{1-\zeta}$

FIGURE 5.5: *The updating mechanism in (5.3)–(5.4) aimed at dealing with DM inconsistency.*

If, for example, $\zeta = 1$, then the most recent anchor value function scores (*i.e.* those computed during iteration i) are considered "correct," in which case the non-anchor alternatives in $\mathcal{A}_{i-1}$ (which are considered to be the "source" of inconsistency) are merely brought in line with the value function scores of the alternatives in $\mathcal{C}_i$ by scaling the scores of the alternatives in the former set multiplicatively by the factor μ_i/μ_{i-1}, while leaving the scores of the alternatives in the most recent comparison set unchanged.

[2] A value of zero is precluded for this parameter, because such a value would not conform with the general spirit of the iterative learning process of Figure 5.1. It is assumed that a limited amount of DM inconsistency may result during *later* iterations of the learning process when the DM becomes more familiar with previously unexplored regions of the decision space, resulting in a situation where he would, in hindsight, have viewed the relative attractiveness values between pairs of alternatives expressed early on during the learning process in a different light, given what he has subsequently learnt later during the learning cycle. Therefore, it is expected that the DM would wish to alter earlier pairwise comparison evaluations slightly, rather than wishing to alter later pairwise comparison evaluations.

[3] Note that deflating the score of alternative in the set $\mathcal{C}_i$ by the factor μ_i/μ_{i-1} is equivalent to inflating the score of that alternative by the factor μ_{i-1}/μ_i.

Alternatively, if $\zeta = \frac{1}{2}$, then the value function scores of non-anchor alternatives in the archive $\mathcal{A}_{i-1}$ and those of the most recently considered alternatives in $\mathcal{C}_i$ are considered to have contributed in equal measure the the current DM inconsistency, in which case the scores of the alternatives in the former set are scaled multiplicatively by the factor $\sqrt{\mu_i/\mu_{i-1}}$, while scaling the scores of the alternatives in the latter set multiplicatively by the factor $\sqrt{\mu_{i-1}/\mu_i}$.

5.4.2 Value function rescaling for a difference-based approach

The same general rescaling approach as that proposed in the previous section is also followed when adopting a difference-based value function estimation method (such as the MACBETH procedure described in §3.2.2), except that the arithmetic mean

$$\delta_i = \frac{1}{\chi}\sum_{j=1}^{\chi} u_i(\boldsymbol{a}_j) \tag{5.5}$$

is used instead of the geometric mean in (5.2). In this case, the difference $\delta_i - \delta_{i-1}$ is a measure of the level of inconsistency exhibited by the DM when providing pairwise comparison information during iteration i of the learning cycle with respect to that provided during the previous cycle. For a perfectly consistent DM it would hold that $\delta_i - \delta_{i-1} = 0$ for all $i = 1, 2, 3\ldots$ The further this difference deviates from zero, however, the more inconsistency is introduced by the DM when progressing from iteration $i-1$ to iteration i of the learning cycle.

The value function score of each alternative within the archive is therefore updated in Steps 11–12 of Algorithm 5.1 by means of the rescaling operations

$$u_i(\boldsymbol{y}) \leftarrow u_{i-1}(\boldsymbol{y}) + \zeta(\delta_i - \delta_{i-1}), \qquad \boldsymbol{y} \in \mathcal{A}_{i-1} \setminus \mathcal{X} \tag{5.6}$$

$$u_i(\boldsymbol{y}) \leftarrow u_i(\boldsymbol{y}) + (1-\zeta)(\delta_{i-1} - \delta_i), \qquad \boldsymbol{y} \in \mathcal{C}_i, \tag{5.7}$$

where $\zeta \in (0, 1]$ is again a recency preference update weight, before enlarging the archive in Step 13 of the algorithm during iteration i of the learning cycle. In this way, the relative desirability of the new alternatives $\boldsymbol{y}_{i,1}, \ldots, \boldsymbol{x}_{i,n-\chi}$ are again brought in line with that of the alternatives in $\mathcal{A}_{i-1} \setminus \mathcal{X}$ previously considered by the DM.

The role of the recency preference update weight parameter ζ in (5.6)–(5.7) is similar to that in (5.3)–(5.4). In this case, DM inconsistency is dealt with by increasing the scores of all (non-anchor) alternatives in the archive $\mathcal{A}_{i-1}$ by the difference $\delta_i - \delta_{i-1}$, scaled multiplicatively by a factor ζ, while decreasing the scores of all alternatives in the most recent comparison set $\mathcal{C}_i$ by the same difference[4], scaled multiplicatively by the factor $1-\zeta$, as illustrated in Figure 5.6.

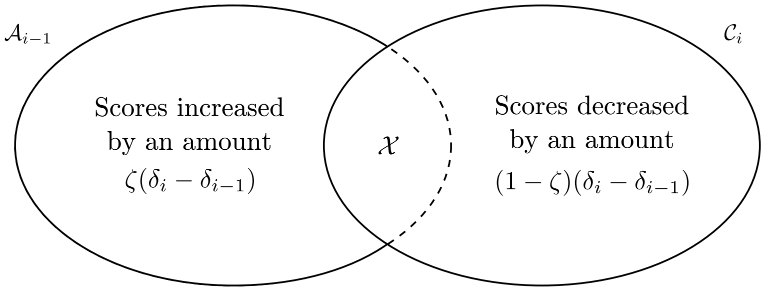

$\mathcal{A}_{i-1}$
$\mathcal{C}_i$
Scores increased by an amount $\zeta(\delta_i - \delta_{i-1})$
$\mathcal{X}$
Scores decreased by an amount $(1-\zeta)(\delta_i - \delta_{i-1})$

Figure 5.6: *The updating mechanism in (5.6)–(5.7) aimed at dealing with DM inconsistency.*

[4] Note that decreasing the score of an alternative in the set $\mathcal{C}_i$ by the difference $\delta_i - \delta_{i-1}$ is equivalent to increasing that score by the difference $\delta_{i-1} - \delta_i$.

If, for example, $\zeta = 1$, then the most recent anchor value function scores are again considered "correct," in which case the value function scores of the non-anchor alternatives in the archive $\mathcal{A}_{i-1}$ (the "source" of the inconsistency) are merely brought in line with the value function scores of the alternatives in $\mathcal{C}_i$ by increasing the scores of the alternatives in the former set additively by the difference $\delta_i - \delta_{i-1}$, while leaving the scores of the alternatives in the most recent comparison set unchanged.

Alternatively, if $\zeta = \frac{1}{2}$, then the value function scores of non-anchor alternatives in the archive $\mathcal{A}_{i-1}$ and those of the most recently considered alternatives in $\mathcal{C}_i$ are considered to have contributed in equal measure to the current DM inconsistency, in which case the scores of the alternatives in the former set are increased by the mean difference $\frac{1}{2}(\delta_i - \delta_{i-1})$, while the scores of the alternatives in the latter set are decreased by the same mean difference $\frac{1}{2}(\delta_i - \delta_{i-1})$.

5.4.3 The degree of DSS efficiency enhancement

If ω learning cycles are performed in total, it follows from (5.1) that the value function estimate u_ω utilised by the DSS to make a final recommendation to the DM is fitted through a total of $n+\omega(n-\chi)$ value function scores. If these alternatives all had to be compared with one another explicitly, this would have required $\binom{n+\omega(n-\chi)}{2}$ pairwise comparisons by the DM. Instead, the DM is only required to perform $(\omega+1)\binom{n}{2}$ pairwise comparisons according to the process outlined in Algorithm 5.1 in the case of only $m = 1$ selection criterion being present in the SPBP instance under consideration — $\binom{n}{2}$ pairwise comparisons for each of the comparison sets $\mathcal{C}_0, \ldots, \mathcal{C}_\omega$. This translates into a saving of

$$\Delta(n, \chi, \omega) = \binom{n + \omega(n-\chi)}{2} - (\omega+1)\binom{n}{2} \tag{5.8}$$

pairwise comparisons required by the DM, which may be a considerable number, as shown in Table 5.1 for small values of n, χ and ω.

The value $\Delta(5, 2, 3)$ in Table 5.1 is elucidated in Figure 5.7. The fourteen vertices in the graph representation of the learning cycle in this figure represent solution alternatives and the edges represent pairwise comparisons of alternatives carried out by the DM according to the scheme of Algorithm 5.1 described above. There are

$$(\omega+1)\binom{n}{2} = 4\binom{5}{2} = 40$$

such edges. Since $\chi = 2$, a total of

$$\binom{n+\omega(n-\chi)}{2} = \binom{14}{2} = 91$$

pairwise comparisons would, however, have been required if all the alternatives were to be compared explicitly. The number of savings of pairwise comparisons brought about by the learning process proposed in Algorithm 5.1 is therefore $\Delta(5, 2, 3) = 91 - 40 = 51$ in accordance with (5.8). This is a significant saving of approximately 56%.

Since $\Delta(n, \chi, \omega)$ in (5.8) is a decreasing function of χ, it is clear that by minimising the number of alternatives χ in the anchor set $\mathcal{X}$, the pairwise comparison burden of the DM is decreased. The reader may therefore rightly question whether the inclusion of more than one anchor in the set $\mathcal{X}$ is not superfluous. The reason for allowing for the possibility that $\chi > 1$ is an attempt at increasing the robustness of the DSS in terms of accommodating possible inconsistency by the

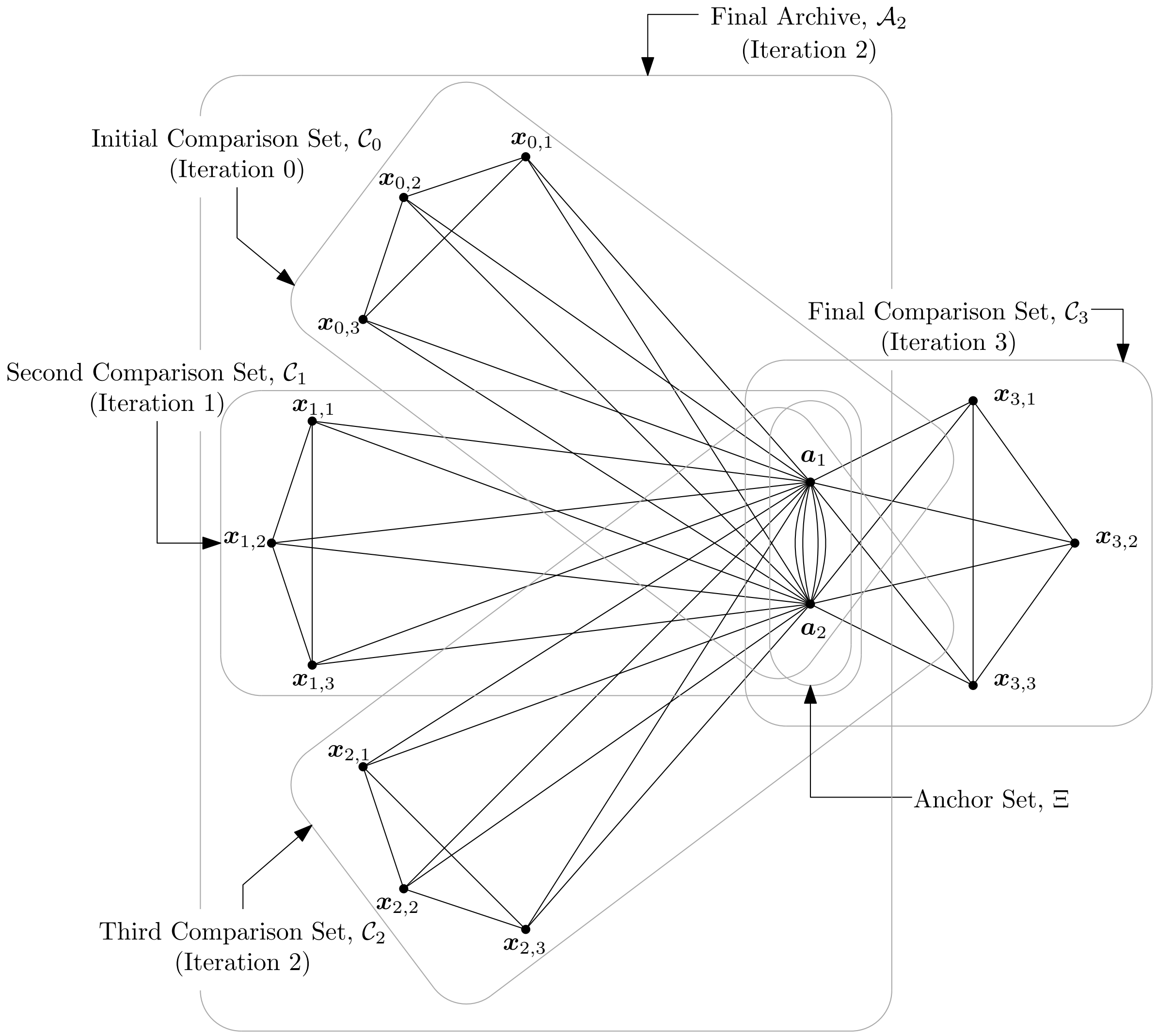


FIGURE 5.7: *Graphical representation of the forty pairwise comparisons that would be required by the DM if $n = 5$ alternatives were to be included in each comparison set, of which $\chi = 2$ are anchors, and if $\omega = 3$ iterations of the learning cycle of Figure 5.1 were to be carried out according to Algorithm 5.1 for an SPBP instance involving a single selection criterion. Each edge in this graph representation represents a single pairwise comparison of two alternatives required by the DM.*

	$n = 5$	$n = 10$	$n = 15$	$n = 20$	$n = 25$
$\omega = 1$	16 (44%)	81 (47%)	196 (48%)	361 (49%)	576 (49%)
$\omega = 2$	48 (62%)	243 (64%)	588 (65%)	1 083 (66%)	1 728 (66%)
$\omega = 3$	96 (71%)	486 (73%)	1 176 (74%)	2 166 (74%)	3 456 (74%)
$\omega = 4$	160 (76%)	810 (78%)	1 960 (79%)	3 610 (79%)	5 760 (79%)
$\omega = 5$	240 (80%)	1 215 (82%)	2 940 (82%)	5 415 (83%)	8 640 (83%)

(a) For an anchor set of size $\chi = 1$

	$n = 5$	$n = 10$	$n = 15$	$n = 20$	$n = 25$
$\omega = 1$	8 (29%)	63 (41%)	168 (44%)	323 (46%)	528 (47%)
$\omega = 2$	25 (45%)	190 (58%)	505 (62%)	970 (63%)	1 585 (64%)
$\omega = 3$	51 (56%)	381 (68%)	1 011 (71%)	1 941 (72%)	3 171 (73%)
$\omega = 4$	86 (63%)	636 (74%)	1 686 (76%)	3 236 (77%)	5 286 (78%)
$\omega = 5$	130 (68%)	955 (78%)	2 530 (80%)	4 855 (81%)	7 930 (82%)

(b) For an anchor set of size $\chi = 2$

	$n = 5$	$n = 10$	$n = 15$	$n = 20$	$n = 25$
$\omega = 1$	1 (5%)	46 (34%)	141 (40%)	286 (43%)	481 (44%)
$\omega = 2$	6 (17%)	141 (51%)	426 (57%)	861 (60%)	1 446 (62%)
$\omega = 3$	15 (27%)	285 (61%)	855 (67%)	1 725 (69%)	2 895 (71%)
$\omega = 4$	28 (36%)	478 (68%)	1 428 (73%)	2 878 (75%)	4 828 (76%)
$\omega = 5$	45 (43%)	720 (73%)	2 145 (77%)	4 320 (79%)	7 245 (80%)

(c) For an anchor set of size $\chi = 3$

TABLE 5.1: *The reduction $\Delta(n, \chi, \omega)$ in the number of pairwise comparisons in (5.8) if a comparison set of size n (containing χ anchors) were to be presented to the DM during each of ω iterations of the learning cycle of Figure 5.1 for an SPBP instance involving a single selection criterion. The percentage values represent the savings as proportions of the number $\binom{n+\omega(n-\chi)}{2}$.*

DM when performing pairwise comparisons. As χ increases, the effectiveness with which the data management component of the DSS is capable of detecting and coping with DM inconsistency is expected to improve. Increased robustness of the DSS in respect of being able to cope with DM inconsistency is, in fact, also the reason for requiring DM comparisons of *all* pairs of alternatives in each comparison set presented to the user. Whereas it would, of course, have been possible to require comparison of each new alternative $\boldsymbol{x}_{i,j}$ in the comparison set $\mathcal{C}_i$ with a single anchor only (and multiplicatively rescaling the relative preferences of previously compared alternatives) in the case of a perfectly consistent DM, the requirement that the DM should compare all pairs of alternatives of the form $\boldsymbol{x}_{i,j}$ and $\boldsymbol{x}_{i,k}$ for $j, k \in \{1, \ldots, n-\chi\}$ with $j \neq k$ (in addition to comparing these alternatives with each anchor $\boldsymbol{a}_\ell \in \mathcal{X}$ for $\ell \in \{1, \ldots, \chi\}$) is aimed at enhancing the capacity of the DSS to deal effectively with DM inconsistency. The validity of this expectation is tested in a sensitivity analysis involving the parameter χ later in this dissertation.

5.5 Process-flow decision support process

The purpose of this section is to elucidate the temporal sequence of events involved in an application of the DSS design put forward in §5.2 and expounded upon in §5.3–§5.4.

Referring to the architecture of §5.2 and the components described in §5.3, it is now possible to elaborate in more detail on the high-level learning process cycle of Figure 5.1 facilitated by Algorithm 5.1. Figure 5.8 contains an illustration of the envisaged sequential flow of events when applying the DSS. The system configuration is initially provided by the user *via* the system configuration component and stored within the database. This is a once-off occurrence and is,

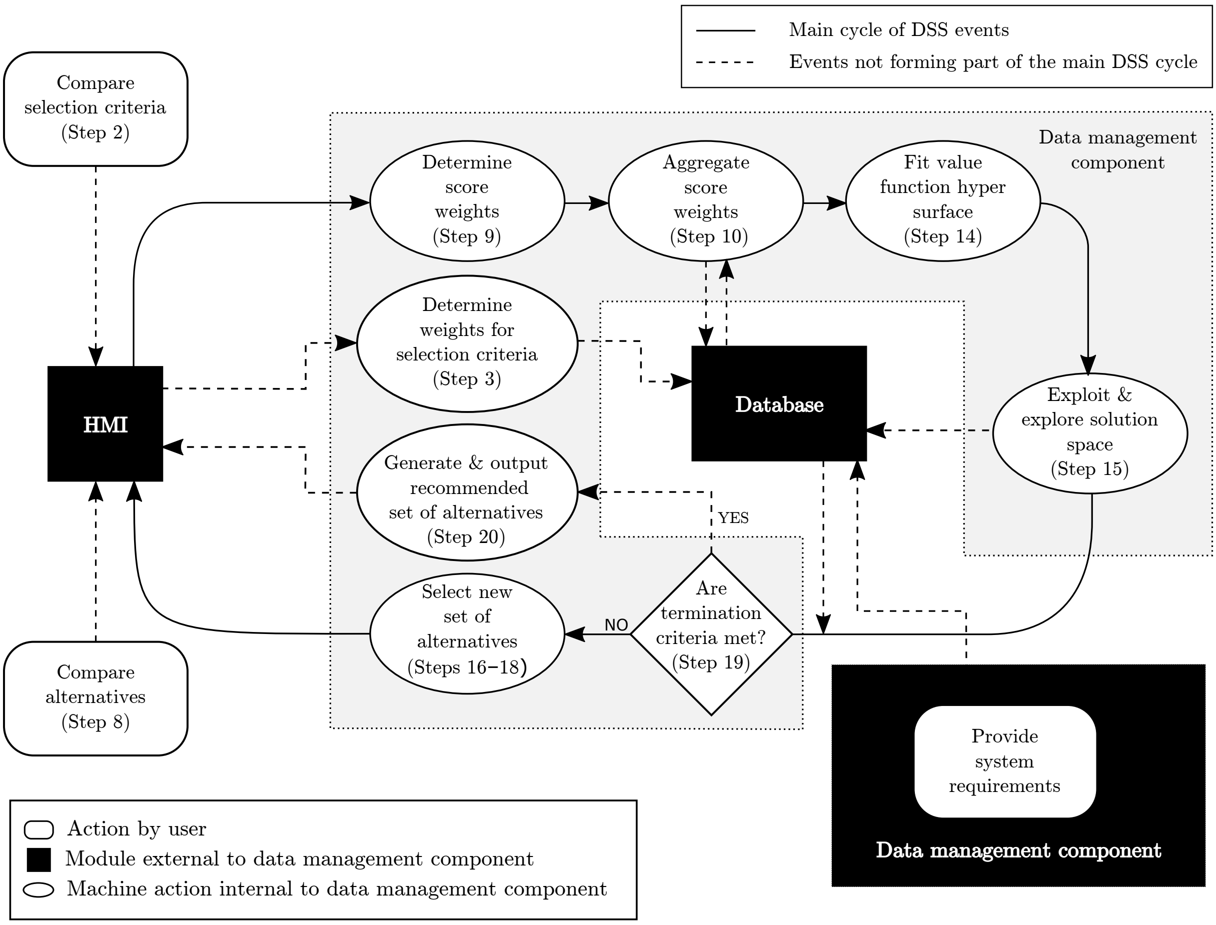

Main cycle of DSS events
Events not forming part of the main DSS cycle
Compare selection criteria (Step 2)
Data management component
Determine score weights (Step 9)
Aggregate score weights (Step 10)
Fit value function hyper surface (Step 14)
Determine weights for selection criteria (Step 3)
HMI
Database
Exploit & explore solution space (Step 15)
Generate & output recommended set of alternatives (Step 20)
YES
Select new set of alternatives (Steps 16–18)
NO
Are termination criteria met? (Step 19)
Compare alternatives (Step 8)
Provide system requirements
Data management component
Action by user
Module external to data management component
Machine action internal to data management component

FIGURE 5.8: *Process flow diagram for the proposed DSS. The step numbers included in the figure correspond to those in Algorithm 5.1.*

therefore, depicted as entering the database from outside the main process loop in the figure. The same is true for the elicitation of pairwise comparisons of the relative importance of selection criteria and the computation of weights or aggregation functions for these criteria in the event of considering a multi-criteria instance of the SPBP.

Once the system requirements have been provided and the relative importance of the selection criteria have been quantified, and these have been stored in the database, a set of solution alternatives can be presented to the user for the purpose of pairwise comparison in respect of each selection criterion. The user then compares these alternatives according to the specified quantification method for pairwise comparison elicitation and reads preference data into the system through a *human machine interface* (HMI). The next step is that of determining value function scores for the alternatives in respect of each selection criterion. These scores are found by applying the pre-specified estimation method to the pairwise comparison quantification for the alternatives, as specified by the user. The scores are combined by means of the weights or aggregation functions associated with the selection criteria in order to arrive at overall value function scores for each solution alternative according to the pre-specified aggregation method. The value function score for each alternative is then stored in the database.

Value function estimation is now possible by fitting a hypersurface through all of the (compared alternative value score) coordinates. The exploitation and exploration process commences next, based on the current value function estimate. The nature and duration of this process may vary, depending on the type of algorithm employed and the parameters contained within the algorithm. As soon as the current value function estimate has been exploited and explored satisfactorily, the database is updated and a new set of alternatives can once again be selected for presentation to the user. This process is repeated until some termination criterion is met, upon which a final, recommended set of solution alternatives is generated and presented to the user as decision support *via* the HMI.

Note that the data management subsystem contributes more significantly towards the DSS process activities in Figure 5.8 than the system configuration component. This is because the generation of a new set of alternatives, the value function score estimation and aggregation, and hypersurface fitting activities reside within the data management component. The user's methodological preferences contribute towards the initial system configuration requirements, while his solution alternative preferences contribute repeatedly to the activities encapsulated in the data management component.

5.6 Chapter summary

An overview was provided in this chapter of a newly proposed DSS for solving SPBP instances. The interaction between the human DM (the user) and the machine computations carried out according to the DSS design were highlighted in §5.1 after which the data flow between the various components of the system was described in a high-level DSS architecture overview in §5.2. The constituent elements of the DSS components were thereafter elaborated upon in §5.3. A method for enhancing the DSS efficiency (by decreasing the pairwise comparison burden on the part of the DM) was described in some detail in §5.4 after which a brief process-flow description of the DSS working was provided in §5.5, elucidating the chronological order of events during application of the system.

CHAPTER 6

Worked examples

Contents

In order to illustrate the working of the DSS framework put forward in the previous chapter, two worked examples of the application of the framework are presented in some detail in this chapter. The examples are both related to a problem of selecting an aesthetically pleasing colour within the green and blue colour spectrum of the RGB colour coding scheme. This single-objective, SPBP instance is henceforth referred to as the 2DCSP. The cardinality of the decision space of the 2DCSP is $256 \times 256 = 65\,536$ since any of the 256 values in the set $\{0, 1, 2, \ldots, 255\}$ may be assigned to each of the green and blue colour attributes independently. For validation purposes, a method of modelling the role of the DM interacting with the framework of the previous chapter is described in §6.1 and two metrics are established for quantifying the degree to which learning takes place as the iterative process of Figure 5.1 progresses according to Algorithm 5.1.

A concept demonstrator of the DSS of §5.2–§5.5 was implemented by the author in Wolfram's Mathematica 11 [226]. The implementation was verified according to a selection of the techniques described in §4.4. In particular, the Mathematica implementation's output was evaluated for soundness at face value for a range of input parameter values. The input parameters, as well as intermediate results, were further printed and scrutinised at certain strategic execution junctures. The implementation was also run for a large number of iterations of the learning cycle in the hope of eliminating errors that are unlikely to occur during short runs. The concept demonstrator was used to produce the numerical work presented in the worked examples of this chapter.

The difference between the two worked examples presented in this chapter lies in the method adopted for value function score estimation. In the first example (in §6.2) the ratio-based AHP approach (of §3.2.1) is adopted for this purpose, while the difference-based MACBETH approach (of §3.2.2) is adopted in the second example (in §6.3). The chapter closes with an assessment (in §6.4) of the differences between these two approaches used as descriptive examples and a brief summary (in §6.5).

6.1 Performance assessment procedure

This section contains a description (in §6.1.1) of how the human DM is modelled for the purposes of analysing the framework performance in §6.4, DSS validation and effectiveness assessment in the following chapters. The methodology adopted in the quality assessment of the value function estimated during a single iteration of the learning cycle of Figure 5.1, carried out by the data management component of the DSS, is also described (in §6.1.2).

6.1.1 Modelling the DM

As mentioned in the chapter introduction, the efficacy of the SPBP DSS framework of §5 is evaluated in this chapter within the context of the 2DCSP. For the purposes of this evaluation, the role of the DM is modelled by assuming that he or she performs pairwise comparisons of decision alternatives according to a value function which is known *a priori*, but which is, of course, not made available to the DSS. The particular true value function considered in this dissertation is the superposition

$$U(g,b) = \exp\left[-\tfrac{5^2}{2}\left(\left(\tfrac{g-63}{255}\right)^2 + \left(\tfrac{b-63}{255}\right)^2\right)\right] + \tfrac{2}{3}\exp\left[-\tfrac{20^2}{18}\left(\left(\tfrac{g-191}{255}\right)^2 + \left(\tfrac{b-191}{255}\right)^2\right)\right] \tag{6.1}$$

of two Gaussian functions, where $g \in \{0,\dots,255\}$ is a green value and $b \in \{0,\dots,255\}$ is a blue value in the decision space $\mathcal{D} = \{0,\dots,255\} \times \{0,\dots,255\}$ of the 2DCSP. This value function is depicted in Figure 6.1. As may be seen in the figure, the function exhibits a global maximum at $(g,b) = (63,63)$ (representing the colour ■) and a local maximum at $(g,b) = (191,191)$ (representing the colour 🟦).

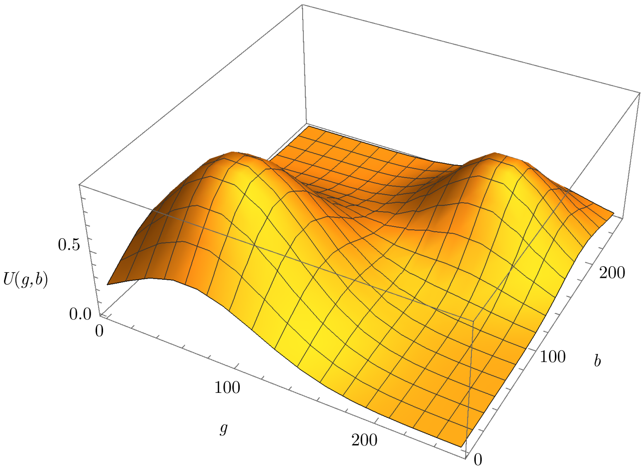


FIGURE 6.1: *The true value function $U(g,b)$ assumed in the numerical investigations of this chapter.*

In order to understand how a pairwise comparison of two alternatives by the DM is modelled in the case of employing a ratio-based value function estimation approach, note that the quotient of two values of the aforementioned function is bounded from above by

$$\theta = \frac{\max_{(g,b)\in\mathcal{D}} U(g,b)}{\min_{(g,b)\in\mathcal{D}} U(g,b)}. \tag{6.2}$$

Suppose the real interval $\mathcal{R} = [1,\theta]$ is partitioned into nine subintervals $\mathcal{R}_1,\dots,\mathcal{R}_9$ according to the preferred judgement scale of the DM, as described in §2.2.5.

Consider two solution alternatives (g_i, b_i) and (g_j, b_j) and suppose, without loss of generality, that $U(g_i, b_i) \geq U(g_j, b_j)$. Then the entry in row i and column j, and that in row j and column i, of the pairwise comparison matrix involving the relative attractiveness of these two alternatives is taken as k and $1/k$, respectively, if $U(g_i, b_i)/U(g_j, b_j) \in \mathcal{R}_k$ in the case of a perfectly consistent DM.

Suppose, however, a difference-based value function estimation approach is adopted, then the absolute difference between two values of the value function in (6.1) resides in the unit interval $\mathcal{R}' = [0, 1]$. In this case, the unit interval is partitioned into six intervals $\mathcal{R}'_1, \ldots, \mathcal{R}'_6$ of equal length, as described in §3.2.2. If (g_i, b_i) and (g_j, b_j) are again two solution alternatives with $U(g_i, b_i) \geq U(g_j, b_j)$, then the entry in row i and column j of the pairwise comparison matrix involving the relative attractiveness of the alternatives is taken as k if $U(g_i, b_i) - U(g_j, b_j) \in \mathcal{R}'_k$.

6.1.2 Learning performance indicators

The value function estimate $u_i(g, b)$ computed during iteration i of the learning cycle is scaled to the same range as that of the true value function $U(g, b)$ by applying the linear transformation

$$u_i^*(g, b) = \left(\frac{U^{max} - U^{min}}{u_i^{max} - u_i^{min}}\right)[u_i(g, b) - u_i^{min}] + U^{min},$$

where U^{min} and U^{max} denote respectively the minimum value and the maximum value of the true value function $U(g, b)$ on the domain $(g, b) \in \mathcal{D}$, while u_i^{min} and u_i^{max} denote respectively the minimum value and the maximum value of the value function estimate $u_i(g, b)$ on the domain $(g, b) \in \mathcal{D}$. Note that $u_i^*(g, b) = U^{min}$ if $u_i(g, b) = u_i^{min}$ and that $u_i^*(g, b) = U^{max}$ if $u_i(g, b) = u_i^{max}$.

The closeness to $U(g, b)$ of the scaled value function estimate $u_i^*(g, b)$ learned during iteration i of the learning cycle in Figure 5.1 is measured by the *global learning performance measure indicator* (GLPMI)

$$Q(u_i^*) = \frac{1}{65\,536} \sum_{g=0}^{255} \sum_{b=0}^{255} |U(g, b) - u_i^*(g, b)|. \tag{6.3}$$

Note that $Q(u_i^*) = 0$ if and only if $u_i^*(g, b) = U(g, b)$ for all $(g, b) \in \mathcal{D}$. Otherwise $Q(u_i^*) \in (0, 1]$. Furthermore, the smaller the value of $Q(u_i^*)$, the closer $u_i^*(g, b)$ approximates the presumed value function $U(g, b)$ of the DM. Finally, the GLPMI is normalised to within the range $[0, 1]$ by division by the constant $256 \times 256 = 65\,536$ in (6.3). Learning is therefore perceived to take place according to the GLPMI in (6.3) if the sequence $Q(u_0^*), Q(u_1^*), Q(u_2^*), \ldots$ is decreasing. Detection of such learning indicates that the value function estimates $u_0^*, u_1^*, u_2^*, \ldots$ are improving in terms of the closeness of their imitation of the shape of the true value function U across the *entire* decision space D.

Whereas the GLPMI in (6.3) measures the quality of $u_i^*(g, b)$ as an approximation of $U(g, b)$ *everywhere* in the decision space $\mathcal{D}$, the DM may, however, be more concerned with the approximation quality of $u_i^*(g, b)$ in the neighbourhood of the global maximum $(g, b) = (63, 63)$, and less concerned with the approximation quality of $u_i^*(g, b)$ in areas of the decision space that are far away from $(63, 63)$ which may be of little or no interest to him or her. For this reason, the *local learning performance measure indicator* (LLPMI)

$$D(u_i^*) = \sqrt{(g_i^* - 63)^2 + (b_i^* - 63)^2} \tag{6.4}$$

is also employed, where $(g_i^*, b_i^*) \in \mathcal{D}$ is the global maximum of $u_i^*(g, b)$ and $u_i(g, b)$. This performance indicator therefore measures the Euclidean distance in decision space between the DM's presumed globally best decision alternative and the best decision alternative according to the scaled value function estimate $u_i^*(g,b)$ learned during iteration i of the learning cycle. The LLPMI (6.4) therefore measures the quality of decision support to the DM in the form of a single best decision alternative based on the DM's value function estimate $u_i^*(g,b)$ in the sense that the quality of such decision support is judged to increase as $D(u_i^*)$ decreases towards zero. Learning is perceived to take place according to the LLPMI in (6.4) if the sequence $D(u_0^*), D(u_1^*), D(u_2^*), \ldots$ is decreasing. The level of correlation between the GLPMI in (6.3) and the LLPMI in (6.4) is expected to be determined by a trade-off between exploration and exploitation of the decision space $\mathcal{D}$. More exploration should lead to better shape approximation of the true value function U by an estimate u_i^* over the entire decision space $\mathcal{D}$, resulting in an improvement of the GLPMI of (6.3). More exploitation, on the other hand, should lead to increased refinement of a value function estimate u_i^* in the region of its global maximum, resulting in an improvement in the LLPMI of (6.4).

6.2 Adopting a ratio-based function estimation approach

After describing in detail the type of decisions required by the DM, as prompted by the system configuration component, in terms of specifying the system's working according to his or her preferences in §6.2.1, the discussion turns in §6.2.2 to an elucidation of the learning process of Figure 5.1 in the context of the 2DCSP, as driven by the data management component of the DSS concept demonstrator in conjunction with iterative responses by the DM in terms of pairwise comparison information in respect of colour alternatives.

6.2.1 System configuration

Recall that the *method of value function construction* sub-component of the system configuration component prompts the DM for specifications in terms of his or her preferred quantification method for pairwise comparison elicitation, method of alternative score estimation, method of aggregation and value function surface fitting methodology. Since the 2DCSP has only one selection criterion, no specification is required in respect of the preferred method of aggregation. Suppose the DM prefers to adopt a ratio-based approach of value function score estimation, using the geometric judgement scale (with parameter $a = 2$) in Table 6.1 in terms of which to specify pairwise comparison information for colour alternatives.

a_{ij}	Description
1	Alternatives i and j are equally desirable
4	Alternative i is weakly more desirable than alternative j
16	Alternative i is strongly more desirable than alternative j
64	Alternative i is very strongly more desirable than alternative j
256	Alternative i is absolutely more desirable than alternative j

Table 6.1: *Descriptions of the meaning of an entry a_{ij} in row i and column j of a pairwise comparison matrix involving colour alternative i relative to colour alternative j, according to the geometric judgement scale (base 2) when adopting a ratio-based approach of value function score estimation. Values in the set $\{2, 8, 32, 128\}$ may be used as intermediate specifications.*

Suppose, furthermore, the DM prefers use of the LLSM. Recall, from (2.16), that the estimated value function score of colour alternative i for which pairwise preferences a_{ij} have been specified in respect of all other colour alternatives $j \in \{1, \ldots, n\}$ in an $n \times n$ pairwise comparison matrix

$\boldsymbol{A} = [a_{ij}]_{i,j=1,\ldots,n}$ is given by

$$u_i = \frac{\left(\prod_{j=1}^{n} a_{ij}\right)^{\frac{1}{n}}}{\sum_{i=1}^{n}\left(\prod_{j=1}^{n} a_{ij}\right)^{\frac{1}{n}}} \tag{6.5}$$

for all $i = 1, \ldots, n$. Suppose also the DM specifies that the method of linear interpolation should be used to estimate value function surfaces, based on the value function scores computed in (6.5).

Recall next that the *exploration & exploitation method and parameters* sub-component of the system configuration component prompts the DM for specifications in terms of his preferred exploitation/exploration method and its corresponding parameters. Suppose the DM specifies the following simple exploitation/exploration scheme, henceforth referred to as the *Basic exploitation and exploration* (Basic E&E) method, to be used during iteration i of the learning cycle of Figure 5.1:

1. A point (g_i^*, b_i^*) is found in decision space which (globally) maximises the current value function surface estimate.

2. The decision space $\mathcal{D}$ is partitioned into two sets $\mathcal{E}_i$ and $\overline{\mathcal{E}}_i$. The set $\mathcal{E}_i$ contains all those alternatives within an exploitation radius r_i from (g_i^*, b_i^*), that is, all alternatives $(g, b) \in \{0, \ldots, 255\} \times \{0, \ldots, 255\}$ for which

$$\sqrt{(g - g_i^*)^2 + (b - b_i^*)^2} \leq r_i.$$

   The set $\overline{\mathcal{E}}_i$ is merely the complement of $\mathcal{E}_i$ (*i.e.* $\overline{\mathcal{E}}_i = \mathcal{D} \setminus \mathcal{E}_i$).

3. The exploitation alternatives are selected from $\mathcal{E}_i$, while the exploration alternatives are selected from $\overline{\mathcal{E}}_i$.

In this case, the exploitation/exploration parameters are the exploitation radii for the various iterations of the learning cycle. Suppose the DM specifies the exponentially decreasing exploitation radii $r_1 = 120$, $r_2 = 64$, $r_3 = 34$, $r_4 = 18$, $\ldots$ (or as many of these radii as are required) in a bid to encourage exploration during early iterations of the learning cycle, but rather favouring exploitation during later iterations.

The *initialisation method for set of prediction alternatives* sub-component of the system configuration component prompts the DM for specifications in terms of his preferred population make-up configuration for the comparison sets $\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \ldots$ of the various iterations of the learning cycle (*i.e.* the method of choosing new colour alternatives for presentation to the DM in pursuit of pairwise comparison information) as well as for his or her preferred population random sampling method.

Suppose the DM is willing to compare $n = 12$ colour alternatives in a pairwise fashion during each iteration of the learning cycle (that is, $|\mathcal{C}_i| = 12$ for all $i = 0, 1, 2, \ldots$). This will require the specification of $\binom{12}{2} = 66$ pairwise comparison entries from Table 6.1 during each iteration.

Suppose further the DM specifies that the initial comparison set should consist of the four corner points $(0, 0)$, $(0, 255)$, $(255, 0)$ and $(255, 255)$ of the decision space $\mathcal{D}$ — so as to be able to linearly interpolate a value function surface estimate over the entire decision space — together with a further eight randomly sampled alternatives, and that $\chi = 1$ alternative (the colour $\boldsymbol{a}_1 = (0, 0)$) should serve the purpose of anchor throughout (that is, $\mathcal{X} = \{(0, 0)\}$).

Suppose, however, the DM specifies that during subsequent iterations, the population make-up should include the anchor, together with six exploitation alternatives and five exploration alternatives. Suppose also that the DM prefers sampling colour alternatives according to a uniform distribution.

The *search termination features* sub-component of the system configuration component finally prompts the DM for specifications in terms of his or her preferred termination criterion and recommendation set selection method. Suppose the DM specifies, as termination criterion, that $\omega = 2$ learning cycles of Figure 5.1 should be carried out and, as recommendation set selection method, that a palette of twelve sample colours should be recommended to him or her in order of decreasing desirability. This palette should contain the anticipated best colour alternative and the anticipated worst colour alternative. Using the k-means algorithm, its remaining colours should be determined by clustering the decision space into ten clusters according to the estimated value function scores of alternatives and returning a cluster representative in each case.

6.2.2 Data management

Whereas the specifications in the previous section are sought from the DM in a once-off fashion by the system configuration component of the DSS, the data management component of the DSS now commences with its iterative learning process as illustrated in Figure 5.1. Suppose the alternatives in Table 6.2 are selected as the initial comparison set $\mathcal{C}_0$ by the data management component according to the DM's preferred method of population configuration.

Alt	Coordinates	Colour	Role	Alt	Coordinates	Colour	Role
$\boldsymbol{a}_1$	$(0, 0)$	■	Anchor	$\boldsymbol{x}_{0,1}$	$(0, 143)$	■	Exploration
$\boldsymbol{x}_{0,2}$	$(0, 255)$	■	Corner point	$\boldsymbol{x}_{0,3}$	$(65, 91)$	■	Exploration
$\boldsymbol{x}_{0,4}$	$(135, 200)$	■	Exploration	$\boldsymbol{x}_{0,5}$	$(201, 123)$	■	Exploration
$\boldsymbol{x}_{0,6}$	$(214, 187)$	■	Exploration	$\boldsymbol{x}_{0,7}$	$(216, 128)$	■	Exploration
$\boldsymbol{x}_{0,8}$	$(232, 198)$	■	Exploration	$\boldsymbol{x}_{0,9}$	$(253, 229)$	■	Exploration
$\boldsymbol{x}_{0,10}$	$(255, 0)$	■	Corner point	$\boldsymbol{x}_{0,11}$	$(255, 255)$	■	Corner point

TABLE 6.2: *The comparison set $\mathcal{C}_0$ selected during Iteration 0 of the learning cycle, together with the roles of its members in the 2DCSP decision space.*

Suppose, furthermore, that after careful consideration of the colour alternatives in Table 6.2, the DM returns the pairwise comparison matrix

$$\boldsymbol{A}_r^{(0)} = \begin{array}{c|cccccccccccc} & \boldsymbol{a}_1 & \boldsymbol{x}_{0,1} & \boldsymbol{x}_{0,2} & \boldsymbol{x}_{0,3} & \boldsymbol{x}_{0,4} & \boldsymbol{x}_{0,5} & \boldsymbol{x}_{0,6} & \boldsymbol{x}_{0,7} & \boldsymbol{x}_{0,8} & \boldsymbol{x}_{0,9} & \boldsymbol{x}_{0,10} & \boldsymbol{x}_{0,11} \\ \hline \boldsymbol{a}_1 & 1 & 2 & 64 & \frac{1}{2} & \frac{1}{2} & 2 & \frac{1}{2} & 2 & \frac{1}{2} & 2 & 64 & 2 \\ \boldsymbol{x}_{0,1} & \frac{1}{2} & 1 & 64 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 2 & 64 & 2 \\ \boldsymbol{x}_{0,2} & \frac{1}{64} & \frac{1}{64} & 1 & \frac{1}{256} & \frac{1}{64} & \frac{1}{64} & \frac{1}{256} & \frac{1}{64} & \frac{1}{128} & \frac{1}{32} & 2 & \frac{1}{16} \\ \boldsymbol{x}_{0,3} & 2 & 2 & 256 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 256 & 4 \\ \boldsymbol{x}_{0,4} & 2 & 2 & 64 & \frac{1}{2} & 1 & 2 & \frac{1}{2} & 2 & \frac{1}{2} & 2 & 64 & 2 \\ \boldsymbol{x}_{0,5} & \frac{1}{2} & 2 & 64 & \frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} & 2 & \frac{1}{2} & 2 & 64 & 2 \\ \boldsymbol{x}_{0,6} & 2 & 2 & 256 & \frac{1}{2} & 2 & 2 & 1 & 2 & 2 & 2 & 256 & 2 \\ \boldsymbol{x}_{0,7} & \frac{1}{2} & 2 & 64 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} & 2 & 64 & 2 \\ \boldsymbol{x}_{0,8} & 2 & 2 & 128 & \frac{1}{2} & 2 & 2 & \frac{1}{2} & 2 & 1 & 2 & 128 & 2 \\ \boldsymbol{x}_{0,9} & \frac{1}{2} & \frac{1}{2} & 32 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 & 32 & 2 \\ \boldsymbol{x}_{0,10} & \frac{1}{64} & \frac{1}{64} & \frac{1}{2} & \frac{1}{256} & \frac{1}{64} & \frac{1}{64} & \frac{1}{256} & \frac{1}{64} & \frac{1}{128} & \frac{1}{32} & 1 & \frac{1}{16} \\ \boldsymbol{x}_{0,11} & \frac{1}{2} & \frac{1}{2} & 16 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 16 & 1 \end{array} \tag{6.6}$$

in terms of the judgement scale in Table 6.1 for these alternatives during Iteration 0 of the learning cycle. Applying the LLSM formula (6.5) to this pairwise comparison matrix yields the estimated value function scores in Table 6.3. The set of colour alternatives in this table also forms the archive $\mathcal{A}_0$ during Iteration 0 of the learning cycle.

Alt	Coordinates	Colour	Score	Alt	Coordinates	Colour	Score
$\boldsymbol{a}_1$	$(0,0)$	⬛	0.092 276	$\boldsymbol{x}_{0,1}$	$(0,143)$	🟦	0.065 249
$\boldsymbol{x}_{0,2}$	$(0,255)$	🟦	0.001 286	$\boldsymbol{x}_{0,3}$	$(65,91)$	🟦	0.195 525
$\boldsymbol{x}_{0,4}$	$(135,200)$	🟦	0.103 576	$\boldsymbol{x}_{0,5}$	$(201,123)$	🟩	0.082 208
$\boldsymbol{x}_{0,6}$	$(214,187)$	🟦	0.164 416	$\boldsymbol{x}_{0,7}$	$(216,128)$	🟩	0.073 239
$\boldsymbol{x}_{0,8}$	$(232,198)$	🟦	0.130 497	$\boldsymbol{x}_{0,9}$	$(253,229)$	🟦	0.051 788
$\boldsymbol{x}_{0,10}$	$(255,0)$	🟩	0.001 144	$\boldsymbol{x}_{0,11}$	$(255,255)$	🟦	0.038 797

TABLE 6.3: *The archive $\mathcal{A}_0$ during Iteration 0 of the learning cycle, together with the value function scores of its members estimated by the LLSM formula (6.5), as applied to the pairwise comparison matrix $\boldsymbol{A}_r^{(0)}$ in (6.6).*

A linear interpolation of these points in value function space returns the (unscaled) estimated value function surface $u_0^r(g,b)$ depicted in Figure 6.2.

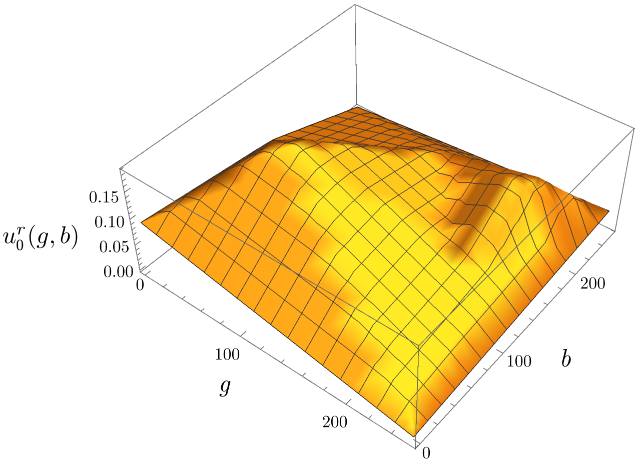


FIGURE 6.2: *The (unscaled) value function surface $u_0^r(g,b)$ estimated during Iteration 0 of the learning cycle, based on a linear interpolation of the value function score estimates in Table 6.3.*

Application of the method of SA (as described in §2.4.3 and illustrated in Example 2.9) yields the global maximum value on this surface as the point $(g_1^*, b_1^*) = (65,91)$ in decision space. Six exploitation alternatives are therefore randomly selected from the set $\mathcal{E}_1$ of colour alternatives within an exploitation radius $r_1 = 120$ from the global maximum $(65,91)$ shown in Figure 6.3, while five colour alternatives are selected from the set $\overline{\mathcal{E}}_1$ of alternatives outside the radius $r_1 = 120$ from $(65,91)$.

Suppose these eleven colour alternatives are selected according to a uniform distribution, together with the anchor $(0,0)$, as listed in Table 6.4 to form the comparison set $\mathcal{C}_1$ presented to the DM during Iteration 1 of the learning cycle.

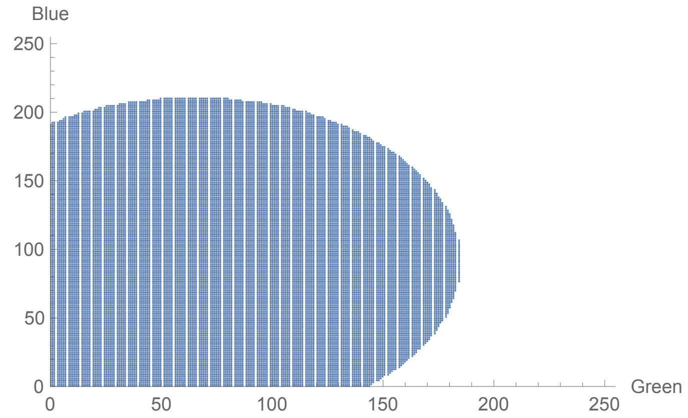


FIGURE 6.3: *The area of exploitation $\mathcal{E}_1$ in the 2DCSP decision space identified in preparation for Iteration 1 of the learning cycle (all colour alternatives within a radius of $r_1 = 120$ from the global maximum $(g_1^*, b_1^*) = (65, 91)$ in Figure 6.2).*

Alt	Coordinates	Colour	Role	Alt	Coordinates	Colour	Role
$\boldsymbol{a}_1$	$(0,0)$	■	Anchor	$\boldsymbol{x}_{1,1}$	$(12,246)$	■	Exploration
$\boldsymbol{x}_{1,2}$	$(44,89)$	■	Exploitation	$\boldsymbol{x}_{1,3}$	$(78,255)$	■	Exploration
$\boldsymbol{x}_{1,4}$	$(83,209)$	■	Exploitation	$\boldsymbol{x}_{1,5}$	$(86,19)$	■	Exploitation
$\boldsymbol{x}_{1,6}$	$(145,112)$	■	Exploitation	$\boldsymbol{x}_{1,7}$	$(153,27)$	■	Exploitation
$\boldsymbol{x}_{1,8}$	$(170,148)$	■	Exploitation	$\boldsymbol{x}_{1,9}$	$(215,177)$	■	Exploration
$\boldsymbol{x}_{1,10}$	$(215,240)$	■	Exploration	$\boldsymbol{x}_{1,11}$	$(221,97)$	■	Exploration

TABLE 6.4: *The comparison set $\mathcal{C}_1$ selected during Iteration 1 of the learning cycle, together with the roles of its members in the 2DCSP decision space.*

If the DM returns the pairwise comparison matrix

$$\boldsymbol{A}_r^{(1)} = \begin{array}{c} \\ \boldsymbol{a}_1 \\ \boldsymbol{x}_{1,1} \\ \boldsymbol{x}_{1,2} \\ \boldsymbol{x}_{1,3} \\ \boldsymbol{x}_{1,4} \\ \boldsymbol{x}_{1,5} \\ \boldsymbol{x}_{1,6} \\ \boldsymbol{x}_{1,7} \\ \boldsymbol{x}_{1,8} \\ \boldsymbol{x}_{1,9} \\ \boldsymbol{x}_{1,10} \\ \boldsymbol{x}_{1,11} \end{array} \begin{array}{cccccccccccc} \boldsymbol{a}_1 & \boldsymbol{x}_{1,1} & \boldsymbol{x}_{1,2} & \boldsymbol{x}_{1,3} & \boldsymbol{x}_{1,4} & \boldsymbol{x}_{1,5} & \boldsymbol{x}_{1,6} & \boldsymbol{x}_{1,7} & \boldsymbol{x}_{1,8} & \boldsymbol{x}_{1,9} & \boldsymbol{x}_{1,10} & \boldsymbol{x}_{1,11} \\ \left[\begin{matrix} 1 \\ \frac{1}{32} \\ 2 \\ \frac{1}{8} \\ \frac{1}{2} \\ 2 \\ \frac{1}{2} \\ \frac{1}{2} \\ 2 \\ 2 \\ 2 \\ \frac{1}{2} \end{matrix} \right. & \begin{matrix} 32 \\ 1 \\ 128 \\ 2 \\ 4 \\ 64 \\ 32 \\ 32 \\ 64 \\ 64 \\ 32 \\ 4 \end{matrix} & \begin{matrix} \frac{1}{2} \\ \frac{1}{128} \\ 1 \\ \frac{1}{32} \\ \frac{1}{4} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{matrix} & \begin{matrix} 8 \\ \frac{1}{2} \\ 32 \\ 1 \\ 2 \\ 32 \\ 8 \\ 8 \\ 16 \\ 32 \\ 16 \\ 2 \end{matrix} & \begin{matrix} 2 \\ \frac{1}{4} \\ 4 \\ \frac{1}{2} \\ 1 \\ 4 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{matrix} & \begin{matrix} \frac{1}{2} \\ \frac{1}{64} \\ 2 \\ \frac{1}{32} \\ \frac{1}{4} \\ 1 \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{4} \end{matrix} & \begin{matrix} 2 \\ \frac{1}{32} \\ 2 \\ \frac{1}{8} \\ \frac{1}{2} \\ 2 \\ 1 \\ \frac{1}{2} \\ 2 \\ 2 \\ 2 \\ \frac{1}{2} \end{matrix} & \begin{matrix} 2 \\ \frac{1}{32} \\ 2 \\ \frac{1}{8} \\ \frac{1}{2} \\ 2 \\ 2 \\ 1 \\ 2 \\ 2 \\ 2 \\ \frac{1}{2} \end{matrix} & \begin{matrix} \frac{1}{2} \\ \frac{1}{64} \\ 2 \\ \frac{1}{16} \\ \frac{1}{2} \\ 2 \\ \frac{1}{2} \\ \frac{1}{2} \\ 1 \\ 2 \\ \frac{1}{2} \\ \frac{1}{2} \end{matrix} & \begin{matrix} \frac{1}{2} \\ \frac{1}{64} \\ 2 \\ \frac{1}{32} \\ \frac{1}{2} \\ 2 \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ 1 \\ \frac{1}{2} \\ \frac{1}{2} \end{matrix} & \begin{matrix} \frac{1}{2} \\ \frac{1}{32} \\ 2 \\ \frac{1}{16} \\ \frac{1}{2} \\ 2 \\ \frac{1}{2} \\ \frac{1}{2} \\ 2 \\ 2 \\ 1 \\ \frac{1}{2} \end{matrix} & \left. \begin{matrix} 2 \\ \frac{1}{4} \\ 4 \\ \frac{1}{2} \\ \frac{1}{2} \\ 4 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 1 \end{matrix} \right] \end{array} \tag{6.7}$$

for the colour alternatives in comparison set $\mathcal{C}_1$ of Table 6.4 presented to him or her during Iteration 1 of the learning cycle, then the score values in Table 6.5 may be computed *via* the LLSM formula (6.5).

The alternatives in $\mathcal{A}_0$ (see Table 6.3) and in $\mathcal{C}_1$ (see Table 6.5) now have to be combined in order to form the archive $\mathcal{A}_1$ of Iteration 1 of the learning cycle. Before this can be done, however, the value function scores of the alternatives in $\mathcal{A}_0$ have to be rescaled multiplicatively according to the method described in §5.4.1 in order to take into account the new pairwise comparison

Alt	Coordinates	Colour	Score	Alt	Coordinates	Colour	Score
$\boldsymbol{a}_1$	$(0,0)$	■	0.079 131	$\boldsymbol{x}_{1,1}$	$(12,246)$	■	0.002 941
$\boldsymbol{x}_{1,2}$	$(44,89)$	■	0.199 397	$\boldsymbol{x}_{1,3}$	$(78,255)$	■	0.007 851
$\boldsymbol{x}_{1,4}$	$(83,209)$	■	0.033 270	$\boldsymbol{x}_{1,5}$	$(86,19)$	■	0.167 672
$\boldsymbol{x}_{1,6}$	$(145,112)$	■	0.070 497	$\boldsymbol{x}_{1,7}$	$(153,27)$	■	0.062 806
$\boldsymbol{x}_{1,8}$	$(170,148)$	■	0.111 908	$\boldsymbol{x}_{1,9}$	$(215,177)$	■	0.133 081
$\boldsymbol{x}_{1,10}$	$(215,240)$	■	0.094 103	$\boldsymbol{x}_{1,11}$	$(221,97)$	■	0.037 345

Table 6.5: *The comparison set $\mathcal{C}_1$ selected during Iteration 1 of the learning cycle, together with the (unscaled) value function scores of its members estimated by the LLSM formula (6.5), as applied to the pairwise comparison matrix $\boldsymbol{A}_r^{(1)}$ in (6.7).*

information gained about the decision space. Recall that the purpose of this rescaling is to bring in line implicitly the pairwise comparison information for the comparison set $\mathcal{C}_0$ of Iteration 0 of the learning cycle with that of the comparison set $\mathcal{C}_1$ of Iteration 1 without having to compare the entries of these distinct comparison sets with one another explicitly. A recency preference update weight value of $\zeta = 1$ is adopted for this purpose in (5.3)–(5.4). Since the anchor set $\mathcal{X}$ contains only the single alternative $\boldsymbol{a}_1 = (0,0)$, the geometric means in (5.2) reduce to the value function score estimates $\mu_0 = u_0^r(\boldsymbol{a}_1) = 0.092\,276$ and $\mu_1 = u_1^r(\boldsymbol{a}_1) = 0.079\,131$ for iterations $i = 0$ and $i = 1$, respectively. Therefore, the estimated value function score $u_0(\boldsymbol{x})$ of a colour alternative $\boldsymbol{x}$ in Table 6.3 is updated as

$$u_1^r(\boldsymbol{x}) \leftarrow \frac{0.079\,131}{0.092\,276}\, u_0^r(\boldsymbol{x}) = 0.857\,546\, u_0^r(\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathcal{A}_0 \setminus \{\boldsymbol{a}_1\} \tag{6.8}$$

and these alternatives are then merged together with the alternatives in Table 6.5 (whose scores remain unchanged) to obtain the archive $\mathcal{A}_1 = \mathcal{A}_0 \cup \mathcal{C}_1$ of Iteration 1 shown in Table 6.6.

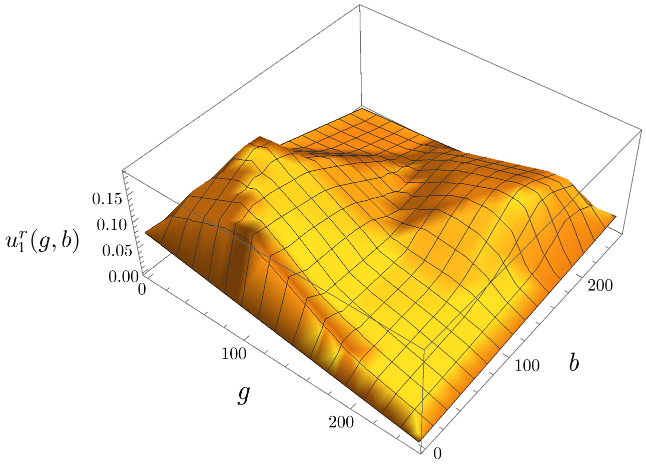


Figure 6.4: *The (unscaled) value function surface $u_1^r(g,b)$ estimated during Iteration 1 of the learning cycle, based on a linear interpolation of the value function score estimates in Table 6.6.*

A linear interpolation of the points in Table 6.6 within value function space returns the updated (unscaled) value function surface estimate $u_1(g,b)$ during Iteration 1 of the learning cycle depicted in Figure 6.4. For this function, the method of SA returns the global maximum as the point $(g_2^*, b_2^*) = (44,89)$. Six exploitation alternatives are therefore randomly selected from the set $\mathcal{E}_2$ of colour alternatives within the reduced exploitation radius $r_2 = 64$ from $(44,89)$ shown in Figure 6.5, while five alternatives are randomly selected from the set $\overline{\mathcal{E}}_2$ of alternatives outside a radius $r_1 = 64$ from $(44,89)$. Suppose these eleven colour alternatives are selected according

Alt	Coordinates	Colour	Score	Alt	Coordinates	Colour	Score
$\boldsymbol{a}_1$	$(0,0)$	■	0.079 131	$\boldsymbol{x}_{0,1}$	$(0,143)$	■	$\underline{0.055\,954}$
$\boldsymbol{x}_{0,2}$	$(0,255)$	■	$\underline{0.001\,102}$	$\boldsymbol{x}_{0,3}$	$(65,91)$	■	$\underline{0.167\,672}$
$\boldsymbol{x}_{0,4}$	$(135,200)$	■	$\underline{0.088\,821}$	$\boldsymbol{x}_{0,5}$	$(201,123)$	■	$\underline{0.070\,497}$
$\boldsymbol{x}_{0,6}$	$(214,187)$	■	$\underline{0.140\,995}$	$\boldsymbol{x}_{0,7}$	$(216,128)$	■	$\underline{0.062\,806}$
$\boldsymbol{x}_{0,8}$	$(232,198)$	■	$\underline{0.111\,908}$	$\boldsymbol{x}_{0,9}$	$(253,229)$	■	$\underline{0.044\,411}$
$\boldsymbol{x}_{0,10}$	$(255,0)$	■	$\underline{0.000\,981}$	$\boldsymbol{x}_{0,11}$	$(255,255)$	■	$\underline{0.033\,270}$
$\boldsymbol{x}_{1,1}$	$(12,246)$	■	0.002 941	$\boldsymbol{x}_{1,2}$	$(44,89)$	■	0.199 397
$\boldsymbol{x}_{1,3}$	$(78,255)$	■	0.007 851	$\boldsymbol{x}_{1,4}$	$(83,209)$	■	0.033 270
$\boldsymbol{x}_{1,5}$	$(86,19)$	■	0.167 672	$\boldsymbol{x}_{1,6}$	$(145,112)$	■	0.070 497
$\boldsymbol{x}_{1,6}$	$(153,27)$	■	0.062 806	$\boldsymbol{x}_{1,8}$	$(170,148)$	■	0.111 908
$\boldsymbol{x}_{1,9}$	$(215,177)$	■	0.133 081	$\boldsymbol{x}_{1,10}$	$(215,240)$	■	0.094 103
$\boldsymbol{x}_{1,11}$	$(221,97)$	■	0.037 345				

Table 6.6: *The archive $\mathcal{A}_1$ during Iteration 1 of the learning cycle, together with the estimated (unscaled) value function scores of its members. The underlined value function score estimates have been updated from Table 6.5 by means of the multiplicative rescaling formula (6.8). The remaining values are taken directly from Table 6.5.*

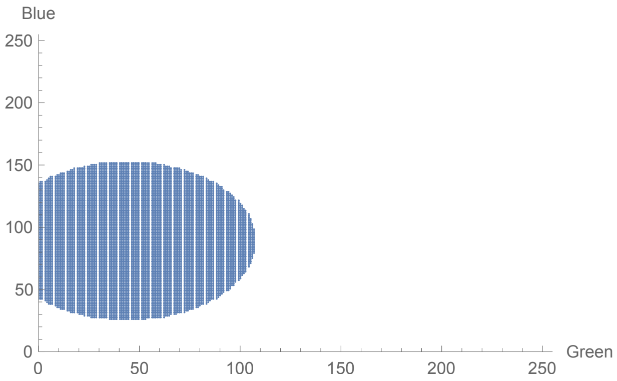


Figure 6.5: *The area of exploitation $\mathcal{E}_2$ in the 2DCSP decision space identified in preparation for Iteration 2 of the learning cycle (all colour alternatives within a radius of $r_2 = 64$ from the global maximum $(g_2^*, b_3^*) = (44, 89)$ in Figure 6.4).*

Alt	Coordinates	Colour	Role	Alt	Coordinates	Colour	Role
$\boldsymbol{a}_1$	$(0,0)$	⬛	Anchor	$\boldsymbol{x}_{2,1}$	$(23,135)$	🟦	Exploitation
$\boldsymbol{x}_{2,2}$	$(24,105)$	🟦	Exploitation	$\boldsymbol{x}_{2,3}$	$(27,148)$	🟦	Exploitation
$\boldsymbol{x}_{2,4}$	$(57,124)$	🟦	Exploitation	$\boldsymbol{x}_{2,5}$	$(66,100)$	🟦	Exploitation
$\boldsymbol{x}_{2,6}$	$(85,53)$	🟩	Exploitation	$\boldsymbol{x}_{2,7}$	$(170,85)$	🟩	Exploration
$\boldsymbol{x}_{2,8}$	$(173,237)$	🟦	Exploration	$\boldsymbol{x}_{2,9}$	$(200,42)$	🟩	Exploration
$\boldsymbol{x}_{2,10}$	$(221,201)$	🟦	Exploration	$\boldsymbol{x}_{2,11}$	$(223,187)$	🟦	Exploration

TABLE 6.7: *The comparison set $\mathcal{C}_2$ selected during Iteration 2 of the learning cycle, together with the roles of its members in the 2DCSP decision space.*

Alt	Coordinates	Colour	Score	Alt	Coordinates	Colour	Score
$\boldsymbol{a}_1$	$(0,0)$	⬛	0.056 779	$\boldsymbol{x}_{2,1}$	$(23,135)$	🟦	0.063 733
$\boldsymbol{x}_{2,2}$	$(24,105)$	🟦	0.120 311	$\boldsymbol{x}_{2,3}$	$(27,148)$	🟦	0.050 585
$\boldsymbol{x}_{2,4}$	$(57,124)$	🟦	0.101 169	$\boldsymbol{x}_{2,5}$	$(66,100)$	🟦	0.135 045
$\boldsymbol{x}_{2,6}$	$(85,53)$	🟩	0.151 583	$\boldsymbol{x}_{2,7}$	$(170,85)$	🟩	0.045 066
$\boldsymbol{x}_{2,8}$	$(173,237)$	🟦	0.071 538	$\boldsymbol{x}_{2,9}$	$(200,42)$	🟩	0.033 761
$\boldsymbol{x}_{2,10}$	$(221,201)$	🟦	0.090 132	$\boldsymbol{x}_{2,11}$	$(223,187)$	🟦	0.080 298

TABLE 6.8: *The comparison set $\mathcal{C}_2$ selected during Iteration 2 of the learning cycle, together with their (unscaled) value function scores estimated by the LLSM formula (6.5) applied to the matrix $\boldsymbol{A}_r^{(2)}$ in (6.9).*

to a uniform distribution, together with the anchor $(0,0)$, as listed in Table 6.7, to form the comparison set $\mathcal{C}_2$ presented to the DM during Iteration 2 of the learning cycle.

Suppose, furthermore, that the DM returns the pairwise comparison matrix

$$\boldsymbol{A}_r^{(2)} = \begin{array}{cc} & \begin{array}{cccccccccccc} \boldsymbol{a}_1 & \boldsymbol{x}_{2,1} & \boldsymbol{x}_{2,2} & \boldsymbol{x}_{2,3} & \boldsymbol{x}_{2,4} & \boldsymbol{x}_{2,5} & \boldsymbol{x}_{2,6} & \boldsymbol{x}_{2,7} & \boldsymbol{x}_{2,8} & \boldsymbol{x}_{2,9} & \boldsymbol{x}_{2,10} & \boldsymbol{x}_{2,11} \end{array} \\ \begin{array}{c} \boldsymbol{a}_a \\ \boldsymbol{x}_{2,1} \\ \boldsymbol{x}_{2,2} \\ \boldsymbol{x}_{2,3} \\ \boldsymbol{x}_{2,4} \\ \boldsymbol{x}_{2,5} \\ \boldsymbol{x}_{2,6} \\ \boldsymbol{x}_{2,7} \\ \boldsymbol{x}_{2,8} \\ \boldsymbol{x}_{2,9} \\ \boldsymbol{x}_{2,10} \\ \boldsymbol{x}_{2,11} \end{array} & \left[\begin{array}{cccccccccccc} 1 & \frac{1}{2} & \frac{1}{2} & 2 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 2 & \frac{1}{2} & 2 & \frac{1}{2} & \frac{1}{2} \\ 2 & 1 & \frac{1}{2} & 2 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 2 & \frac{1}{2} & 2 & \frac{1}{2} & \frac{1}{2} \\ 2 & 2 & 1 & 2 & 2 & \frac{1}{2} & \frac{1}{2} & 2 & 2 & 4 & 2 & 2 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 2 & \frac{1}{2} & 2 & \frac{1}{2} & \frac{1}{2} \\ 2 & 2 & \frac{1}{2} & 2 & 1 & \frac{1}{2} & \frac{1}{2} & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 1 & \frac{1}{2} & 2 & 2 & 4 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 & 1 & 2 & 2 & 4 & 2 & 2 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} & 2 & \frac{1}{2} & \frac{1}{2} \\ 2 & 2 & \frac{1}{2} & 2 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 2 & 1 & 2 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{4} & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} & \frac{1}{2} \\ 2 & 2 & \frac{1}{2} & 2 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 2 & 2 & 2 & 1 & 2 \\ 2 & 2 & \frac{1}{2} & 2 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 2 & 2 & 2 & \frac{1}{2} & 1 \end{array}\right] \end{array} \tag{6.9}$$

in respect of the colour alternatives in the comparison set $\mathcal{C}_2$ of Table 6.7. Then the value function scores in Table 6.8 may be computed *via* the LLSM formula (6.5). Thereafter, the estimated value function score $u_1^r(\boldsymbol{x})$ of an alternative $\boldsymbol{x}$ in Table 6.6 is updated as

$$u_2^r(\boldsymbol{x}) \leftarrow \frac{0.056\,779}{0.079\,131}\, u_1^r(\boldsymbol{x}) = 0.717\,532\, u_1(\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathcal{A}_1 \setminus \{\boldsymbol{a}_1\} \tag{6.10}$$

and these alternatives are then combined with the alternatives in Table 6.8 (whose value function scores remain unchanged) to obtain the archive $\mathcal{A}_2$ during Iteration 2 of the learning cycle shown in Table 6.9. A linear interpolation of these points in value function space returns the updated (unscaled) value function surface estimate $u_2(g,b)$ depicted in Figure 6.6.

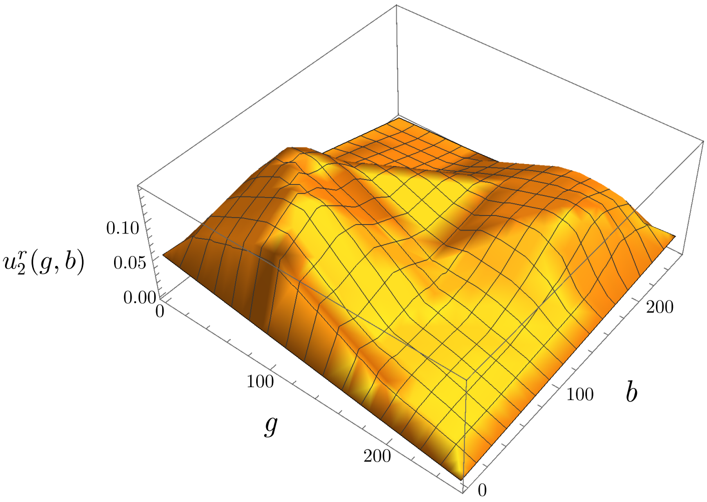


FIGURE 6.6: *The (unscaled) value function surface $u_2^r(g,b)$ estimated during Iteration 2 of the learning cycle, based on a linear interpolation of the value function score estimates in Table 6.9.*

Since $\omega = 2$ full learning cycles have now been completed, the cyclic process of Figure 5.1 terminates with a recommendation of suitable alternatives to the DM, based on the value function surface estimate $u_2^r(g,b)$ in Figure 6.6. According to the method of SA, the global maximum and minimum values on the surface in Figure 6.6 are achieved by $(56,67)$ and $(255,0)$, respectively. These two alternatives are therefore estimated to be respectively the most and least desirable colours in the opinion of the DM.

Alt	Coordinates	Colour	Score	Alt	Coordinates	Colour	Score
$\boldsymbol{a}_1$	$(0,0)$	⬛	$\underline{0.056\,779}$	$\boldsymbol{x}_{0,1}$	$(0,143)$	🟦	$\underline{0.040\,149}$
$\boldsymbol{x}_{0,2}$	$(0,255)$	🟦	$\underline{0.000\,790}$	$\boldsymbol{x}_{0,3}$	$(65,91)$	🟦	$\underline{0.120\,311}$
$\boldsymbol{x}_{0,4}$	$(135,200)$	🟦	$\underline{0.063\,733}$	$\boldsymbol{x}_{0,5}$	$(201,123)$	🟩	$\underline{0.050\,585}$
$\boldsymbol{x}_{0,6}$	$(214,187)$	🟩	$\underline{0.101\,169}$	$\boldsymbol{x}_{0,7}$	$(216,128)$	🟩	$\underline{0.045\,066}$
$\boldsymbol{x}_{0,8}$	$(232,198)$	🟩	$\underline{0.080\,298}$	$\boldsymbol{x}_{0,9}$	$(253,229)$	🟩	$\underline{0.031\,866}$
$\boldsymbol{x}_{0,10}$	$(255,0)$	🟩	$\underline{0.000\,704}$	$\boldsymbol{x}_{0,11}$	$(255,255)$	🟦	$\underline{0.023\,872}$
$\boldsymbol{x}_{1,1}$	$(12,246)$	🟦	$\underline{0.002\,110}$	$\boldsymbol{x}_{1,2}$	$(44,89)$	🟦	$\underline{0.143\,075}$
$\boldsymbol{x}_{1,3}$	$(78,255)$	🟦	$\underline{0.005\,633}$	$\boldsymbol{x}_{1,4}$	$(83,209)$	🟦	$\underline{0.023\,873}$
$\boldsymbol{x}_{1,5}$	$(86,19)$	🟩	$\underline{0.120\,311}$	$\boldsymbol{x}_{1,6}$	$(145,112)$	🟩	$\underline{0.050\,585}$
$\boldsymbol{x}_{1,7}$	$(153,27)$	🟩	$\underline{0.045\,066}$	$\boldsymbol{x}_{1,8}$	$(170,148)$	🟩	$\underline{0.080\,298}$
$\boldsymbol{x}_{1,9}$	$(215,177)$	🟩	$\underline{0.095\,491}$	$\boldsymbol{x}_{1,10}$	$(215,240)$	🟦	$\underline{0.067\,523}$
$\boldsymbol{x}_{1,11}$	$(221,97)$	🟩	$\underline{0.026\,796}$	$\boldsymbol{x}_{2,1}$	$(23,135)$	🟦	0.063 733
$\boldsymbol{x}_{2,2}$	$(24,105)$	🟦	0.120 311	$\boldsymbol{x}_{2,3}$	$(27,148)$	🟦	0.050 585
$\boldsymbol{x}_{2,4}$	$(57,124)$	🟦	0.101 169	$\boldsymbol{x}_{2,5}$	$(66,100)$	🟦	0.135 045
$\boldsymbol{x}_{2,6}$	$(85,53)$	🟩	0.151 583	$\boldsymbol{x}_{2,7}$	$(170,85)$	🟩	0.045 066
$\boldsymbol{x}_{2,8}$	$(173,237)$	🟦	0.071 538	$\boldsymbol{x}_{2,9}$	$(200,42)$	🟩	0.033 761
$\boldsymbol{x}_{2,10}$	$(221,201)$	🟩	0.090 132	$\boldsymbol{x}_{2,11}$	$(223,187)$	🟩	0.080 298

TABLE 6.9: *The archive $\mathcal{A}_2$ during Iteration 2 of the learning cycle, together with the (unscaled) value function scores of its members. The underlined value function score estimates have been updated from Table 6.5 by means of the multiplicative rescaling formula (6.10). The remaining values are taken directly from Table 6.8.*

In addition, the decision space is partitioned into $k = 10$ clusters based on the estimated value function scores in Figure 6.6 by means of the k-means algorithm, described in §2.3.2. Representatives from each of these clusters are then recommended as a representative colour palette, together with the aforementioned best and worst alternatives, as decision support to the DM. These decision support recommendations are listed in Table 6.10.

No	Coordinates	Colour	Score	Status
a	$(56, 67)$	■	0.165 048	Best colour alternative
b	$(29, 108)$	■	0.120 433	Cluster 1 representative
c	$(46, 119)$	■	0.107 969	Cluster 2 representative
d	$(84, 38)$	■	0.100 488	Cluster 3 representative
e	$(179, 238)$	■	0.071 005	Cluster 4 representative
f	$(13, 7)$	■	0.067 585	Cluster 5 representative
g	$(145, 21)$	■	0.052 974	Cluster 6 representative
h	$(124, 216)$	■	0.051 652	Cluster 7 representative
i	$(164, 65)$	■	0.048 052	Cluster 8 representative
j	$(105, 209)$	■	0.040 617	Cluster 9 representative
k	$(66, 196)$	■	0.031 906	Cluster 10 representative
ℓ	$(255, 0)$	■	0.000 724	Worst colour alternative

Table 6.10: *Palette of colours recommended (in order of decreasing desirability) as decision support to the DM.*

6.3 Adopting a difference-based function estimation approach

A similar approach to that in the previous section is adopted in this section when presenting the second worked example of this chapter, which also involves solving the 2DCSP of the previous section, but this time using a different method of value function score estimation.

6.3.1 System configuration

Suppose the DM specifies exactly the same system configuration as that described in §6.2.1, except that he or she prefers to use the difference-based MACBETH procedure described in §3.2.2 for value function score estimation instead of the ratio-based AHP, adopting the MACBETH categories in Table 3.7 as quantification method for pairwise comparison articulation instead of the geometric judgement scale in Table 6.1.

The alternatives considered during each iteration of the learning cycle in Figure 5.1 in this section correspond exactly to those considered during the previous section, so as to showcase clearly the different value function surface estimates produced during each iteration by the AHP and by the method of MACBETH.

6.3.2 Data management

The iterative learning process of Figure 5.1 may commence after the once-off system configuration has been completed in accordance with the DM's specifications. Suppose again the initial comparison set $\mathcal{C}_0$ in Table 6.2 is selected according to the DM's preferred method of population configuration. During iteration 0, the DM carefully considers these alternatives and ranks them in order of decreasing desirability as $\boldsymbol{x}_{0,3}$(■), $\boldsymbol{x}_{0,6}$(■), $\boldsymbol{x}_{0,8}$(■), $\boldsymbol{x}_{0,4}$(■), $\boldsymbol{a}_1$(■), $\boldsymbol{x}_{0,5}$(■), $\boldsymbol{x}_{0,7}$(■), $\boldsymbol{x}_{0,1}$(■), $\boldsymbol{x}_{0,9}$(■), $\boldsymbol{x}_{0,11}$(■), $\{\boldsymbol{x}_{0,10}$(■), $\boldsymbol{x}_{0,2}$(■)$\}$, not preferring $\boldsymbol{x}_{0,2}$ or $\boldsymbol{x}_{0,10}$ above the other. The DM thereafter registers the relative pairwise desirabilities of the alternatives in terms of the difference categories in Table 3.7, resulting in the pairwise comparison matrix

$$\boldsymbol{A}_d^{(0)} = \begin{array}{c} \\ \boldsymbol{x}_{0,3} \\ \boldsymbol{x}_{0,6} \\ \boldsymbol{x}_{0,8} \\ \boldsymbol{x}_{0,4} \\ \boldsymbol{a}_1 \\ \boldsymbol{x}_{0,5} \\ \boldsymbol{x}_{0,7} \\ \boldsymbol{x}_{0,1} \\ \boldsymbol{x}_{0,9} \\ \boldsymbol{x}_{0,11} \\ \boldsymbol{x}_{0,10} \end{array} \begin{array}{c} \begin{array}{ccccccccccc} \boldsymbol{x}_{0,3} & \boldsymbol{x}_{0,6} & \boldsymbol{x}_{0,8} & \boldsymbol{x}_{0,4} & \boldsymbol{a}_1 & \boldsymbol{x}_{0,5} & \boldsymbol{x}_{0,7} & \boldsymbol{x}_{0,1} & \boldsymbol{x}_{0,9} & \boldsymbol{x}_{0,11} & \boldsymbol{x}_{0,10} \end{array} \\ \left[\begin{array}{ccccccccccc} 1 & 2 & 3 & 4 & 4 & 5 & 5 & 5 & 5 & 5 & 6 \\ 0 & 1 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 4 & 4 \\ 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 3 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \end{array}. \tag{6.11}$$

Note that the matrix only represents eleven alternatives. This is due to the fact that alternatives $\boldsymbol{x}_{0,10}$ and $\boldsymbol{x}_{0,2}$ are considered indistinguishable in terms of desirability and may, therefore, be considered as the same alternative, for computational purposes. Solving the linear programming problem (3.16)–(3.36) of the MACBETH method yields the estimated value function scores in Table 6.11, which also serve as the archive $\mathcal{A}_0$ during Iteration 0 of the learning cycle.

Alt	Coordinates	Colour	Score	Alt	Coordinates	Colour	Score
$\boldsymbol{a}_1$	$(0, 0)$	⬛	0.068 029	$\boldsymbol{x}_{0,1}$	$(0, 143)$	🟦	0.033 971
$\boldsymbol{x}_{0,2}$	$(0, 255)$	🟦	0.000 000	$\boldsymbol{x}_{0,3}$	$(65, 91)$	🟦	0.339 073
$\boldsymbol{x}_{0,4}$	$(135, 200)$	🟦	0.101 614	$\boldsymbol{x}_{0,5}$	$(201, 123)$	🟩	0.034 039
$\boldsymbol{x}_{0,6}$	$(214, 187)$	🟩	0.219 609	$\boldsymbol{x}_{0,7}$	$(216, 128)$	🟩	0.033 971
$\boldsymbol{x}_{0,8}$	$(232, 198)$	🟩	0.135 740	$\boldsymbol{x}_{0,9}$	$(253, 229)$	🟦	0.033 835
$\boldsymbol{x}_{0,10}$	$(255, 0)$	🟩	0.000 000	$\boldsymbol{x}_{0,11}$	$(255, 255)$	🟦	0.000 184

TABLE 6.11: *The archive $\mathcal{A}_0$ during Iteration 0 of the learning cycle, together with the (unscaled) value function scores of its members as estimated by the MACBETH method of §3.2.2.*

Fitting a linear interpolation through the points in $\mathcal{A}_0$ returns the estimated (unscaled) value function surface $u_0^d(g, b)$ illustrated in Figure 6.7.

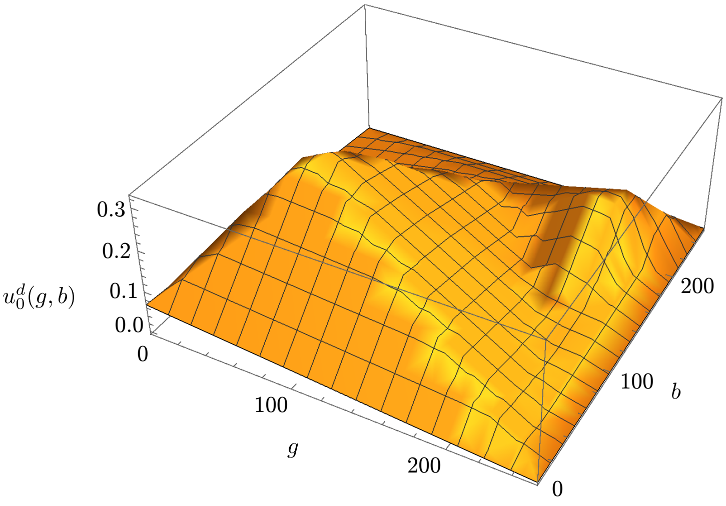


FIGURE 6.7: *The (unscaled) value function surface $u_0^d(g, b)$ estimated during Iteration 0 of the learning cycle, based on a linear interpolation of the value function score estimates in Table 6.11.*

In §6.3 the method of SA (described in §2.4.3) was applied to determine the global maximum of the estimated value function $u_0^r(g, b)$ in the decision space. New alternatives which constituted the comparison set $\mathcal{C}_1$ (together with the anchor value $(0, 0)$) were selected by exploitation within and exploration outside a radius $r = 120$ around this maximum alternative $(65, 91)$. In this section this same set of new alternatives, shown in Table 6.4, is taken as the comparison set $\mathcal{C}_1$ presented to the DM during Iteration 1 for comparison purposes.

Suppose the DM considers these alternatives, sorts them in order of decreasing attractiveness as $\boldsymbol{x}_{1,2}$(🟦), $\boldsymbol{x}_{1,5}$(🟩), $\boldsymbol{x}_{1,9}$(🟩), $\boldsymbol{x}_{1,8}$(🟩), $\boldsymbol{x}_{1,10}$(🟦), $\boldsymbol{a}_1$(⬛), $\boldsymbol{x}_{1,6}$(🟩), $\boldsymbol{x}_{1,7}$(🟩), $\boldsymbol{x}_{1,11}$(🟩), $\boldsymbol{x}_{1,4}$(🟦), $\boldsymbol{x}_{1,3}$(🟦), $\boldsymbol{x}_{1,1}$(🟦) and registers their relative pairwise attractiveness according to the difference categories in Table 3.7 to obtain the comparison matrix

$$\boldsymbol{A}_d^{(1)} = \begin{array}{c} \\ \boldsymbol{x}_{1,2} \\ \boldsymbol{x}_{1,5} \\ \boldsymbol{x}_{1,9} \\ \boldsymbol{x}_{1,8} \\ \boldsymbol{x}_{1,10} \\ \boldsymbol{a}_1 \\ \boldsymbol{x}_{1,6} \\ \boldsymbol{x}_{1,7} \\ \boldsymbol{x}_{1,11} \\ \boldsymbol{x}_{1,4} \\ \boldsymbol{x}_{1,3} \\ \boldsymbol{x}_{1,1} \end{array} \begin{array}{c} \begin{array}{cccccccccccc} \boldsymbol{x}_{1,2} & \boldsymbol{x}_{1,5} & \boldsymbol{x}_{1,9} & \boldsymbol{x}_{1,8} & \boldsymbol{x}_{1,10} & \boldsymbol{a}_1 & \boldsymbol{x}_{1,6} & \boldsymbol{x}_{1,7} & \boldsymbol{x}_{1,11} & \boldsymbol{x}_{1,4} & \boldsymbol{x}_{1,3} & \boldsymbol{x}_{1,1} \end{array} \\ \left[\begin{array}{cccccccccccc} 1 & 2 & 2 & 3 & 4 & 4 & 4 & 4 & 5 & 5 & 5 & 5 \\ 0 & 1 & 1 & 2 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 \\ 0 & 0 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 4 & 4 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \end{array}. \qquad (6.12)$$

Solving the linear programming problems (3.16)–(3.36) yields the value function scores shown in Table 6.12.

Alt	Coordinates	Colour	Score	Alt	Coordinates	Colour	Score
$\boldsymbol{a}_1$	$(0, 0)$	⬛	0.074 394	$\boldsymbol{x}_{1,1}$	$(12, 246)$	🟦	0.000 000
$\boldsymbol{x}_{1,2}$	$(44, 89)$	🟦	0.239 738	$\boldsymbol{x}_{1,3}$	$(78, 255)$	🟦	0.008 238
$\boldsymbol{x}_{1,4}$	$(83, 209)$	🟦	0.016 477	$\boldsymbol{x}_{1,5}$	$(86, 19)$	🟩	0.190 045
$\boldsymbol{x}_{1,6}$	$(145, 112)$	🟩	0.066 156	$\boldsymbol{x}_{1,7}$	$(153, 27)$	🟩	0.041 363
$\boldsymbol{x}_{1,8}$	$(170, 148)$	🟩	0.099 209	$\boldsymbol{x}_{1,9}$	$(215, 177)$	🟩	0.165 217
$\boldsymbol{x}_{1,10}$	$(215, 240)$	🟦	0.082 633	$\boldsymbol{x}_{1,11}$	$(221, 97)$	🟩	0.016 526

TABLE 6.12: *The comparison set $\mathcal{C}_1$ selected during Iteration 1 of the learning cycle, together with the (unscaled) value function scores of its members estimated by the MACBETH method in §3.2.2.*

In order to form a new archive $\mathcal{A}_1$ it is required that the alternatives in $\mathcal{A}_0$ (see Table 6.11) and the alternatives in $\mathcal{C}_1$ (see Table 6.12) have to be combined. As mentioned in §6.2, the value function scores of $\mathcal{A}_0 \setminus \{\boldsymbol{a}_1\}$ and/or $\mathcal{C}_1$ have to be rescaled for this purpose. This section, however, requires the application of the additive rescaling method described in §5.4.2. A recency preference update weight parameter $\zeta = 1$ is again assumed, meaning the most recent anchor value (in $\mathcal{C}_1$) is seen as "correct" and the estimated value function score values in $\mathcal{A}_0 \setminus \{\boldsymbol{a}_1\}$ are, therefore, brought in line with the value function score estimates in Iteration 1, while leaving the most recent value function scores of the alternatives in $\mathcal{C}_1$ unchanged. The estimated value function score $u_0^d(\boldsymbol{x})$ of a colour alternative $\boldsymbol{x}$ in Table 6.11 is updated as

$$u_1^d(\boldsymbol{x}) \leftarrow (0.074\,394 - 0.068\,029) + u_0^d(\boldsymbol{x}) = 0.006\,364 + u_0^d(\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathcal{A}_0^d \setminus \{\boldsymbol{a}_1\} \qquad (6.13)$$

and all these alternatives are merged together with the alternatives in Table 6.12 (whose value function scores remain unchanged) to form a new archive $\mathcal{A}_1 = (\mathcal{A}_0 \setminus \{\boldsymbol{a}_1\}) \cup \mathcal{C}_1$ of Iteration 1, shown in Table 6.13.

Alt	Coordinates	Colour	Score	Alt	Coordinates	Colour	Score
$\boldsymbol{a}_1$	$(0, 0)$	⬛	0.074 394	$\boldsymbol{x}_{0,1}$	$(0, 143)$	🟦	$\underline{0.040\,268}$
$\boldsymbol{x}_{0,2}$	$(0, 255)$	🟦	$\underline{0.006\,364}$	$\boldsymbol{x}_{0,3}$	$(65, 91)$	🟦	$\underline{0.345\,438}$
$\boldsymbol{x}_{0,4}$	$(135, 200)$	🟦	$\underline{0.107\,979}$	$\boldsymbol{x}_{0,5}$	$(201, 123)$	🟩	$\underline{0.040\,404}$
$\boldsymbol{x}_{0,6}$	$(214, 187)$	🟦	$\underline{0.225\,974}$	$\boldsymbol{x}_{0,7}$	$(216, 128)$	🟩	$\underline{0.040\,336}$
$\boldsymbol{x}_{0,8}$	$(232, 198)$	🟦	$\underline{0.142\,105}$	$\boldsymbol{x}_{0,9}$	$(253, 229)$	🟦	$\underline{0.040\,200}$
$\boldsymbol{x}_{0,10}$	$(255, 0)$	🟩	$\underline{0.006\,364}$	$\boldsymbol{x}_{0,11}$	$(255, 255)$	🟦	$\underline{0.006\,549}$
$\boldsymbol{x}_{1,1}$	$(12, 246)$	🟦	0.000 000	$\boldsymbol{x}_{1,2}$	$(44, 89)$	🟦	0.239 738
$\boldsymbol{x}_{1,3}$	$(78, 255)$	🟦	0.008 238	$\boldsymbol{x}_{1,4}$	$(83, 209)$	🟦	0.016 477
$\boldsymbol{x}_{1,5}$	$(86, 19)$	🟩	0.190 045	$\boldsymbol{x}_{1,6}$	$(145, 112)$	🟩	0.066 156
$\boldsymbol{x}_{1,6}$	$(153, 27)$	🟩	0.041 363	$\boldsymbol{x}_{1,8}$	$(170, 148)$	🟩	0.099 209
$\boldsymbol{x}_{1,9}$	$(215, 177)$	🟦	0.165 217	$\boldsymbol{x}_{1,10}$	$(215, 240)$	🟦	0.082 633
$\boldsymbol{x}_{1,11}$	$(221, 97)$	🟩	0.016 526				

TABLE 6.13: *The archive $\mathcal{A}_1$ during Iteration 1 of the learning cycle, together with the estimated (unscaled) value function scores of its members. The underlined value function score estimates have been updated from Table 6.11 by means of the additive rescaling formula (6.13). The remaining values are taken directly from Table 6.12.*

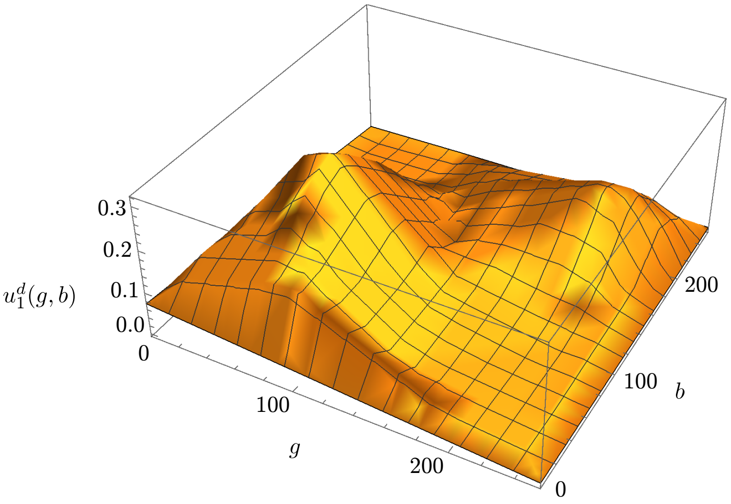


FIGURE 6.8: *The (unscaled) value function surface $u_1^d(g, b)$ estimated during Iteration 1 of the learning cycle, based on a linear interpolation of the value function score estimates in Table 6.13.*

A linear interpolation through the scores of the alternatives in $\mathcal{A}_1$ within value function space yields an updated (unscaled) value function estimate $u_1^d(g, b)$. This value function estimate is shown in Figure 6.8. As in the construction of the new set of alternatives during Iteration 1, the exact same alternatives as in §6.2.2 are taken as the new comparison set $\mathcal{C}_2$ shown in Table 6.7 (instead of a combination of alternatives constructed by exploration/exploitation of the region around an estimated maximum point of $u_1^d(g, b)$ by applying the method of SA).

Suppose the DM ranks the alternatives in C_2 according to decreasing attractiveness as $\boldsymbol{x}_{2,6}$(🟩), $\boldsymbol{x}_{2,5}$(🟦), $\boldsymbol{x}_{2,2}$(🟦), $\boldsymbol{x}_{2,4}$(🟦), $\boldsymbol{x}_{2,10}$(🟦), $\boldsymbol{x}_{2,11}$(🟦), $\boldsymbol{x}_{2,8}$(🟦), $\boldsymbol{x}_{2,1}$(🟦), $\boldsymbol{a}_1$(⬛), $\boldsymbol{x}_{2,3}$(🟦), $\boldsymbol{x}_{2,7}$(🟩), $\boldsymbol{x}_{2,9}$(🟩) and returns the relative pairwise attractiveness of the alternatives according to the difference categories in Table 3.7 to form the comparison matrix

$$\boldsymbol{A}_d^{(2)} = \begin{array}{r|cccccccccccc|} & \boldsymbol{x}_{2,6} & \boldsymbol{x}_{2,5} & \boldsymbol{x}_{2,2} & \boldsymbol{x}_{2,4} & \boldsymbol{x}_{2,10} & \boldsymbol{x}_{2,11} & \boldsymbol{x}_{2,8} & \boldsymbol{x}_{2,1} & \boldsymbol{a}_{1} & \boldsymbol{x}_{2,3} & \boldsymbol{x}_{2,7} & \boldsymbol{x}_{2,9} \\ \boldsymbol{x}_{2,6} & 1 & 1 & 3 & 3 & 3 & 3 & 4 & 4 & 5 & 5 & 5 & 6 \\ \boldsymbol{x}_{2,5} & 0 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 4 & 4 & 4 & 5 \\ \boldsymbol{x}_{2,2} & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 4 \\ \boldsymbol{x}_{2,4} & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 \\ \boldsymbol{a}_{2,10} & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 \\ \boldsymbol{x}_{2,11} & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 2 & 2 & 3 & 3 \\ \boldsymbol{x}_{2,8} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 \\ \boldsymbol{x}_{2,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 \\ \boldsymbol{a}_{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 \\ \boldsymbol{x}_{2,3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 \\ \boldsymbol{x}_{2,7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ \boldsymbol{x}_{2,9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} . \qquad (6.14)$$

Solving the linear programming problems (3.16)–(3.36) yields the value function scores shown in Table 6.14.

The value function scores of the alternatives in the archive $\mathcal{A}_1$ and those of the alternatives in the comparison set $\mathcal{C}_2$ are again brought in line by performing the updating procedure

$$u_2^d(\boldsymbol{x}) \leftarrow (0.038\,157 - 0.074\,394) + u_1^d(\boldsymbol{x}) = -0.036\,237 + u_1^d(\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathcal{A}_1 \qquad (6.15)$$

to obtain the new archive $\mathcal{A}_2$ shown in Table 6.15. A linear interpolation of the points in Table 6.15 within the value function space returns the updated value function surface estimate $u_2^d(g, b)$ in Figure 6.9 during Iteration 2 of the learning cycle.

Alt	Coordinates	Colour	Score	Alt	Coordinates	Colour	Score
$\boldsymbol{a}_1$	$(0, 0)$	■	0.0381577	$\boldsymbol{x}_{2,1}$	$(23, 135)$	■	0.057 141
$\boldsymbol{x}_{2,2}$	$(24, 105)$	■	0.114 321	$\boldsymbol{x}_{2,3}$	$(27, 148)$	■	0.038 119
$\boldsymbol{x}_{2,4}$	$(57, 124)$	■	0.114 245	$\boldsymbol{x}_{2,5}$	$(66, 100)$	■	0.171 386
$\boldsymbol{x}_{2,6}$	$(85, 53)$	■	0.190 522	$\boldsymbol{x}_{2,7}$	$(170, 85)$	■	0.019 059
$\boldsymbol{x}_{2,8}$	$(173, 237)$	■	0.057 179	$\boldsymbol{x}_{2,9}$	$(200, 42)$	■	0.000 000
$\boldsymbol{x}_{2,10}$	$(221, 201)$	■	0.104 569	$\boldsymbol{x}_{2,11}$	$(223, 187)$	■	0.095 299

TABLE 6.14: *The comparison set $\mathcal{C}_2$ selected during Iteration 2 of the learning cycle, together with their (unscaled) value function scores estimated by the MACBETH method in §3.2.2 applied to the matrix $\boldsymbol{A}_d^{(2)}$ in (6.14).*

To conclude the difference-based worked example of this section, the decision space is again partitioned into $k = 10$ clusters based on the estimated value function scores in Figure 6.9 by invoking the k-means algorithm, described in §2.3.2. Representatives from each of these clusters are recommended as decision support to the DM in the form of a representative colour palette, together with the aforementioned best and worst alternatives. These decision support recommendations are listed in Table 6.16.

Alt	Coordinates	Colour	Score	Alt	Coordinates	Colour	Score
$\boldsymbol{a}_1$	$(0,0)$	■	0.038 157	$\boldsymbol{x}_{0,1}$	$(0,143)$	■	$\underline{0.004\,031}$
$\boldsymbol{x}_{0,2}$	$(0,255)$	■	$\underline{-0.029\,872}$	$\boldsymbol{x}_{0,3}$	$(65,91)$	■	$\underline{0.309\,201}$
$\boldsymbol{x}_{0,4}$	$(135,200)$	■	$\underline{0.071\,741}$	$\boldsymbol{x}_{0,5}$	$(201,123)$	■	$\underline{0.004\,167}$
$\boldsymbol{x}_{0,6}$	$(214,187)$	■	$\underline{0.189\,737}$	$\boldsymbol{x}_{0,7}$	$(216,128)$	■	$\underline{0.004\,099}$
$\boldsymbol{x}_{0,8}$	$(232,198)$	■	$\underline{0.105\,868}$	$\boldsymbol{x}_{0,9}$	$(253,229)$	■	$\underline{0.003\,963}$
$\boldsymbol{x}_{0,10}$	$(255,0)$	■	$\underline{-0.029\,872}$	$\boldsymbol{x}_{0,11}$	$(255,255)$	■	$\underline{-0.029\,687}$
$\boldsymbol{x}_{1,1}$	$(12,246)$	■	$\underline{-0.036\,237}$	$\boldsymbol{x}_{1,2}$	$(44,89)$	■	$\underline{0.203\,501}$
$\boldsymbol{x}_{1,3}$	$(78,255)$	■	$\underline{-0.027\,998}$	$\boldsymbol{x}_{1,4}$	$(83,209)$	■	$\underline{-0.019\,760}$
$\boldsymbol{x}_{1,5}$	$(86,19)$	■	$\underline{0.153\,808}$	$\boldsymbol{x}_{1,6}$	$(145,112)$	■	$\underline{0.029\,919}$
$\boldsymbol{x}_{1,7}$	$(153,27)$	■	$\underline{0.005\,126}$	$\boldsymbol{x}_{1,8}$	$(170,148)$	■	$\underline{0.062\,972}$
$\boldsymbol{x}_{1,9}$	$(215,177)$	■	$\underline{0.128980}$	$\boldsymbol{x}_{1,10}$	$(215,240)$	■	$\underline{0.046\,396}$
$\boldsymbol{x}_{1,11}$	$(221,97)$	■	$\underline{-0.019\,710}$	$\boldsymbol{x}_{2,1}$	$(23,135)$	■	0.057 141
$\boldsymbol{x}_{2,2}$	$(24,105)$	■	0.114 321	$\boldsymbol{x}_{2,3}$	$(27,148)$	■	0.038 119
$\boldsymbol{x}_{2,4}$	$(57,124)$	■	0.114 245	$\boldsymbol{x}_{2,5}$	$(66,100)$	■	0.171 386
$\boldsymbol{x}_{2,6}$	$(85,53)$	■	0.190 522	$\boldsymbol{x}_{2,7}$	$(170,85)$	■	0.019 059
$\boldsymbol{x}_{2,8}$	$(173,237)$	■	0.057 179	$\boldsymbol{x}_{2,9}$	$(200,42)$	■	0.000 000
$\boldsymbol{x}_{2,10}$	$(221,201)$	■	0.104 569	$\boldsymbol{x}_{2,11}$	$(223,187)$	■	0.095 299

TABLE 6.15: *The archive $\mathcal{A}_2$ during Iteration 2 of the learning cycle, together with the (unscaled) value function scores of its members. The underlined value function score estimates have been updated from Table 6.13 by means of the additive rescaling formula (6.15). The remaining values are taken directly from Table 6.14.*

No	Coordinates	Colour	Score	Status
a	$(65,91)$	■	0.309 201	Best colour alternative
b	$(41,61)$	■	0.161 423	Cluster 1 representative
c	$(78,113)$	■	0.117 619	Cluster 2 representative
d	$(27,24)$	■	0.096 619	Cluster 3 representative
e	$(59,140)$	■	0.086 052	Cluster 4 representative
f	$(112,192)$	■	0.055 332	Cluster 5 representative
g	$(135,215)$	■	0.055 237	Cluster 6 representative
h	$(155,17)$	■	0.050 695	Cluster 7 representative
i	$(11,0)$	■	0.035 223	Cluster 8 representative
j	$(96,226)$	■	0.003 354	Cluster 9 representative
k	$(172,0)$	■	−0.007 729	Cluster 10 representative
ℓ	$(255,0)$	■	−0.029 872	Worst colour alternative

TABLE 6.16: *Palette of colours recommended (in order of decreasing desirability) as decision support to the DM after a difference-based method is applied.*

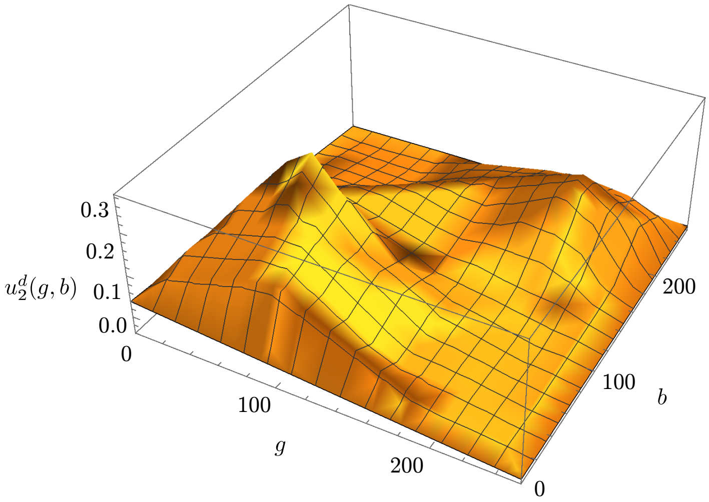


FIGURE 6.9: *The (unscaled) value function surface $u_2^d(g,b)$ estimated during Iteration 2 of the learning cycle, based on a linear interpolation of the value function score estimates in Table 6.15.*

6.4 Analysis of worked examples

The learning performance of the framework of Chapter 5 is measured in this section in the contexts of the ratio-based example in §6.2 and the difference-based example in §6.3. Both the GLPMI and LLPMI measures of learning performance introduced in §6.1.2 are employed to compare the relative performances of the framework in the aforementioned examples.

The three value function estimates $u_0^r(g,b)$, $u_1^r(g,b)$ and $u_2^r(g,b)$ of the true value function $U(g,b)$ in (6.1) produced in the worked example of §6.2 (and shown graphically in Figures 6.2, 6.4 and 6.6) achieved the GLPMI values

$$\begin{aligned} Q(\tilde{u}_0^r) &= 0.276\,852, \\ Q(\tilde{u}_1^r) &= 0.208\,619, \text{ and} \\ Q(\tilde{u}_2^r) &= 0.149\,776, \end{aligned}$$

respectively. The corresponding GLPMI values for the difference-based worked example of §6.3 are

$$\begin{aligned} Q(\tilde{u}_0^d) &= 0.180\,929, \\ Q(\tilde{u}_1^d) &= 0.098\,144, \text{ and} \\ Q(\tilde{u}_2^d) &= 0.109\,148. \end{aligned}$$

A graphical representation of the above GLPMI values may be found in Figure 6.10.

The GLPMI values for the ratio-based example exhibit a decrease from Iteration 0 to Iteration 1 and again from Iteration 1 to Iteration 2. The GLPMI values for the difference-based example similarly exhibit a decrease from Iteration 0 to Iteration 1, but not from Iteration 1 to 2. A reduction in GLPMI values indicates that learning of the true value function $U(g,b)$ does indeed take place. It may, therefore, be inferred that learning did indeed take place in the case of the worked example of §6.2 for two iterations of the learning cycle of Figure 5.1. The same may be concluded for the first iteration of the difference-based worked example in §6.3, after which learning stagnation seems to have set in.

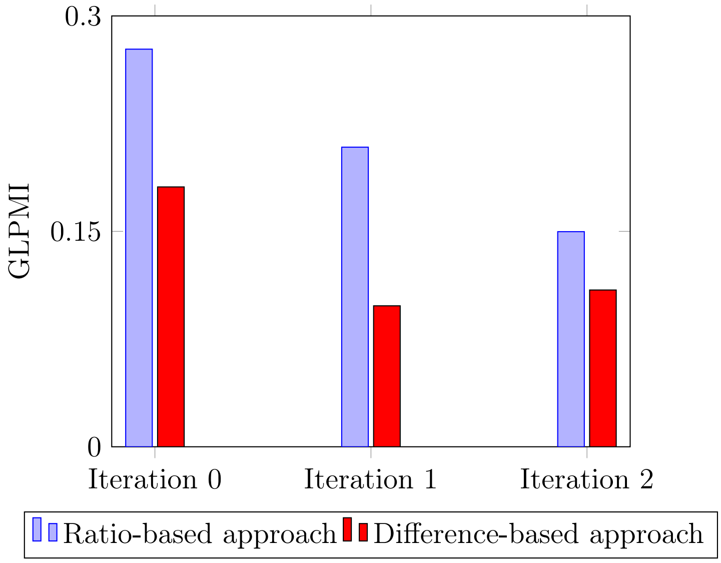


FIGURE 6.10: *GLPMI values (in 6.3) for $\zeta = 1$ in the context of the worked examples of §6.2 and §6.3, for each iteration of the learning cycle of the proposed DSS framework.*

The three value function estimates $u_0^r(g,b)$, $u_1^r(g,b)$ and $u_2^r(g,b)$ of the true value function $U(g,b)$ in (6.1) produced in the worked example of §6.2 achieved the LLPMI values

$$\begin{aligned} D(\tilde{u}_0^r) &= 28.071, \\ D(\tilde{u}_1^r) &= 32.202, \text{ and} \\ D(\tilde{u}_2^r) &= 8.062, \end{aligned}$$

while the corresponding LLPMI values for the difference-based worked example of §6.3 are

$$\begin{aligned} D(\tilde{u}_0^d) &= 28.071, \\ D(\tilde{u}_1^d) &= 28.071, \text{ and} \\ D(\tilde{u}_2^d) &= 28.071. \end{aligned}$$

A graphical representation of the above LLPMI values may be found in Figure 6.11.

It may be seen in the figure that the ratio-based approach managed to achieve a significant reduction in the LLPMI value over two iterations of the learning cycle in Figure 5.1, although not in a monotonic fashion. This was clearly not the case in the difference-based approach.

Recall that, in order to be able to compare the two worked examples in terms of value function shapes attributed specifically to differences in the working of the AHP and that of the method of MACBETH, the newly generated sets of alternatives proposed for comparison in the ratio-based example of §6.2 were also considered unaltered in the difference-based example of §6.3. This did not perhaps allow for appropriate exploitation of the optimal points of the value function estimates $u_0^d(g,b)$, $u_1^d(g,b)$ and $u_2^d(g,b)$, and may be the reason for the similarity between the LLPMI values of the difference-based example, illustrated in red in Figure 6.11.

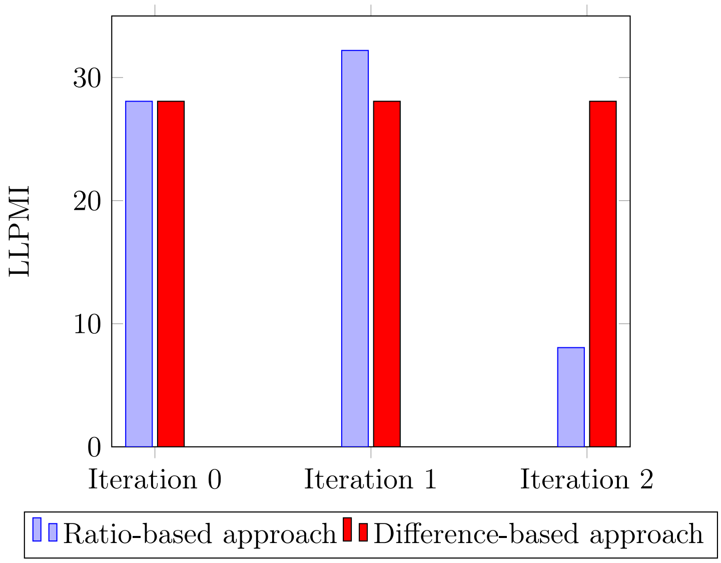


Figure 6.11: *LLPMI values (in 6.4) for $\zeta = 1$ in the context of the worked examples of §6.2 and §6.3, for each iteration of the learning cycle of the proposed DSS framework.*

6.5 Chapter summary

After having described a method of modelling the human DM for the purpose of system validation and having established two metrics for measuring the quality of estimated DM value functions (the GLPMI and the LLPMI), two detailed worked examples were presented in this chapter in order to demonstrate application of the DSS framework of Chapter 5 in the context of the simple 2DCSP, which asks for an aesthetically pleasing colour within the green and blue colour spectrum of the RGB coding scheme. The two worked examples differed only in the method of value function score estimation, otherwise assuming exactly the same system configuration. In the first example, in §6.2, the ratio-based AHP was employed for value function score estimation, while the difference-based MACBETH method was employed for this purpose in the second example, in §6.3. In each case, the working of the two main components of the DSS framework, namely the system configuration component and the data management component, was illustrated by mimicking numerically their functions in a concept demonstrator of the DSS framework of Chapter 5 implemented in Wolfram's Mathematica 11. The results returned by the concept demonstrator in each case were presented both graphically and in tabular format. The numerical results obtained in the two worked examples were finally analysed and discussed in §6.4.

Part III

Decision Support Framework Validation

CHAPTER 7

Decision Support System Validation

Contents

The purpose of this chapter is to assess the practical workability and potential effectiveness of the DSS proposed in §5. This assessment takes place in the context of a perfectly consistent DM wishing to solve the 2DCSP described in §6. The intrinsic ability of the DSS of §5 to learn within the context of the 2DCSP is ascertained in §7.2. Thereafter, the relative performances of the incorporation of various combinations of judgement scales and aggregation methods within the DSS framework of §5 are evaluated in §7.2 in terms of the metrics introduced in §6.1.2. This is followed in §7.4 by an investigation into the relative merits of incorporating various degrees of exploration *versus* exploitation of the 2DCSP decision space within the learning framework of the DSS framework. A sensitivity analysis is next performed in §7.5 with respect to the various parameters of the DSS framework of §5. A summary of the contents of the chapter is finally provided in §7.6.

Since stochastic elements are involved in the DSS framework of §5, an analysis of its performance is carried out at a particular level of statistical significance in this chapter. The methodology adopted for achieving this is first elucidated in §7.1.

7.1 Statistical analysis of learning performance indicator values

A hypothesis testing approach is adopted from the realm of inferential statistics in order to analyse the learning performance indicator values introduced in §6.1.2 returned by numerous applications of the DSS of §5.

When comparing the relative performance of the incorporation of various combinations of judgement scales and aggregation methods within the data management component, each execution of the DSS is replicated thirty times, each time with a different random number generator seed value. The same thirty seed values are, however, used in different replication runs of the DSS, when adopting other combinations of judgement scales and aggregation methods. The learning performance indicator values in (6.3) and (6.4) are reported in the form of box plots for each of the replication runs. Since the same corresponding seed replication values are reused in these replication runs, the experiments may each be thought of as producing a matched sample of independent and identically distributed observations of size thirty of the learning performance indicators according to certain underlying distributions. If the number of observations in each sample of learning performance indicator values is sufficiently large[1] (as in this case), then the underlying distribution may be approximated closely by a normal distribution according to the central limit theorem [178].

The first step in the statistical comparison of the samples of learning performance measure indicator values produced by the aforementioned replication runs is therefore to carry out an ANOVA test [80] (as described in §2.5.1) in respect of the LLPMI and GLPMI means of the samples obtained by employing different combinations of judgement scales and aggregation methods, assuming as null-hypothesis that *there is no significant difference between the means of any two samples*. This test indicates whether or not there are differences at a particular level of significance between at least two of the sample means. If such significant differences are detected, the ANOVA test is, unfortunately, incapable of indicating between which pairs of means these differences occur. A *post hoc* test is therefore required in order to determine between which means these differences actually occur if the ANOVA reveals that there are, in fact, such differences.

After having concluded from an ANOVA test that there are significant differences between the sample means of a learning performance measure indicator, a Levene test [136] is carried out (as described in §2.5.2), assuming as null-hypothesis that *there are no differences between the variances of the samples*, in order to test for homoscedasticity of the samples. If this null-hypothesis is rejected by the Levene test (*i.e.* the respective sample variances are found to be statistically different at a particular level of significance), then the Games-Howell *post hoc* test [99, 98] is employed (as described in §2.5.4) in order to determine the exact locations of the differences in the sample means established by the ANOVA test. Otherwise, if the sample variances are found not to be statistically different at the assumed level of significance, Fisher's LSD test [87] is employed (as described in §2.5.3) in order to determine the exact locations of the differences in the sample means established by the ANOVA test.

All the above statistical tests are carried out an $\alpha = 5\%$ level of statistical significance.

7.2 Testing the intrinsic ability of the DSS framework to learn

The objective in this section is to assess, within the context of the 2DCSP of §6, to what extent the DSS framework of §5 is capable of learning according to the global performance indicator as the number of iterations in the learning cycle of Figure 5.1 increases. The learning capability of the the DSS framework is assessed in this section when incorporating a difference-based approach (the AHP, described in §3.2.1) as well as a difference-based approach (MACBETH method, discussed in §3.2.2).

7.2.1 Adopting a Ratio-based approach

The AHP provides the possibility of incorporating various judgement scale and aggregation method combinations. This section, contains an investigation into the performance of the DSS when adopting the AHP and either the EM or LLSM as aggregation method in separate sections.

[1] A widely accepted rule of thumb is that this value should be at least 30 [116].

Learning based on the EM of aggregation

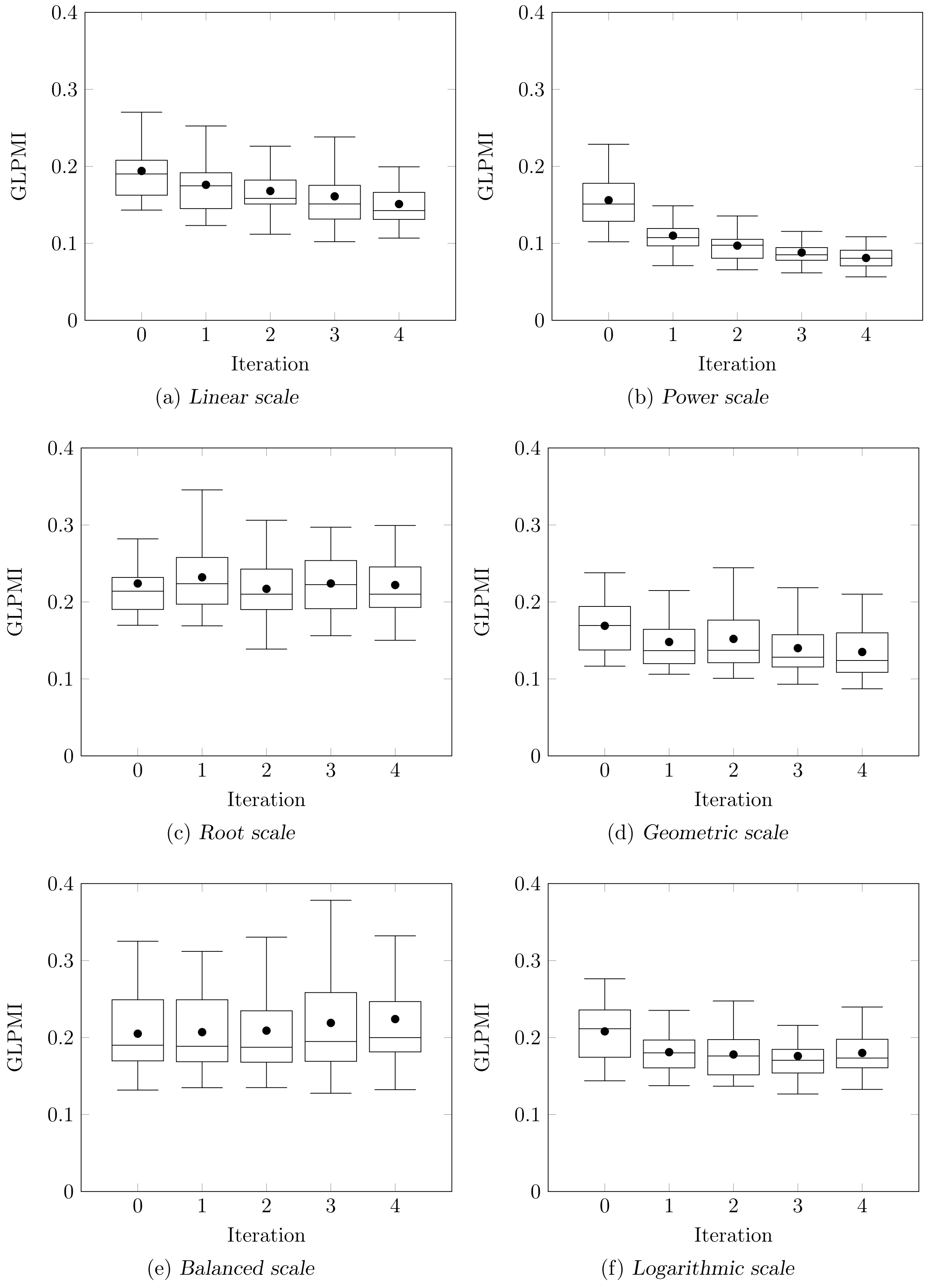
Figures 7.1(a)–(f) contain box plots of the GLPMI values in (6.3) associated with $\omega = 4$ iterations of the learning cycle of Figure 5.1 when populating the DSS framework of §5 with each of the six judgement scales of Table 2.2, respectively, in conjunction with the EM. In each of these cases, the comparison sets $\mathcal{C}_0$, $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$ and $\mathcal{C}_4$ each contains $n = 12$ colour alternatives. Only $\chi = 1$ anchor is employed (the alternative $(0, 0)$). In each of the six cases, a total of $N = \omega + 1 = 5$ samples of GLPMI values are therefore produced (one GLPMI value for each iteration of the learning cycle), each containing $M = 30$ observations. Since the objective is merely to assess the *intrinsic* ability of the DSS framework of §5 to learn, no optimisation (or exploitation) component is employed in any of the six cases when generating new alternatives for consideration by the DM, and so only the GLPMI is used to asses the learning capabilities of the framework corresponding to the six relevant judgement scales. In other words, the alternatives $\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,n-1}$ of each comparison set $\mathcal{C}_i$ are generated purely randomly (according to a uniform distribution on $\mathcal{D}$) without exploiting areas of the decision space that seem to be performing well in terms of the current DM value function estimate. The means of the GLPMI samples are reported in Table 7.1. The table also contains the p-values returned for the ANOVA and Levene tests for these samples.

			GLPMI				
	Iteration	Iteration	Iteration	Iteration	Iteration	ANOVA	Levene
Scale	0	1	2	3	4	test p-values	test p-values
Linear	0.194	0.176	0.168	0.161	0.151	3.03×10^{-05}	7.10×10^{-01}
Power	0.156	0.110	0.097	0.088	0.081	$< 1 \times 10^{-17}$	5.70×10^{-05}
Root	0.224	0.232	0.217	0.224	0.222	7.16×10^{-01}	6.50×10^{-01}
Geometric	0.169	0.148	0.152	0.140	0.135	2.48×10^{-03}	8.81×10^{-01}
Balanced	0.205	0.207	0.209	0.219	0.224	6.57×10^{-01}	2.90×10^{-01}
Logarithmic	0.208	0.181	0.178	0.176	0.180	4.71×10^{-04}	1.88×10^{-01}

Table 7.1: *GLPMI means associated with four iterations of the learning cycle when populating the DSS framework of §5 with the AHP ratio-based approach, using each of the six AHP judgement scales of Table 2.2 in conjunction with the EM. The corresponding p-values returned for the ANOVA and Levene tests are also included. Table entries smaller than [larger than, respectively]* 0.05 *are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the null-hypothesis (stating that the GLPMI means of every pair of learning iterations are equal), may be [may not be, respectively] rejected.*

It follows from the ANOVA test p-values in Table 7.1 that the relevant null-hypothesis (stating that the GLPMI means across all learning iterations are statistically indistinguishable) may be rejected at a statistical significance level of $\alpha = 5\%$ for each of the six judgement scales, except for the root and balanced judgement scales (the ANOVA test p-values for these scales are 0.716 and 0.657 respectively, which exceeds the 0.05 level of significance). Based on a visual inspection of the box plots in each of Figures 7.1(a), (b), (d) and (f), it may therefore be concluded with 95% confidence that *some* degree of learning takes place in the context of the 2DCSP if any of the linear, power, geometric or logarithmic AHP judgement scales are combined with the EM.

The Levene test p-values in Table 7.1 indicate that the null-hypothesis (stating that the GLPMI samples are homoscedastic across all learning iterations) may not be rejected at a statistical significance level of $\alpha = 5\%$ for a combination of the linear, root, geometric, balanced or logarithmic AHP judgement scales with the EM. That is, there is no statistical evidence in support of heteroscedasticity of the learning performance samples at a 95% level of confidence (their respective Levene test p-values 0.710, 0.650, 0.881, 0.290 and 0.188 all exceed the significance level of 0.05). It therefore cannot be claimed with 95% confidence that there is an improvement in



(a) *Linear scale*

(b) *Power scale*

(c) *Root scale*

(d) *Geometric scale*

(e) *Balanced scale*

(f) *Logarithmic scale*

FIGURE 7.1: *Box plots of the GLPMI values in (6.3) associated with four iterations of the learning cycle when populating the DSS framework of §5 with each of the six AHP judgement scales of Table 2.2 in conjunction with the EM in the absence of exploitation of promising areas of the 2DCSP decision space. Mean values are indicated by solid dots, while median values are denoted by central quartile dividers.*

	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Iteration 0	3.89×10^{-02}	3.58×10^{-03}	1.79×10^{-04}	1.85×10^{-06}
Iteration 1	—	3.82×10^{-01}	8.00×10^{-02}	4.47×10^{-03}
Iteration 2	—	—	3.77×10^{-01}	4.62×10^{-02}
Iteration 3	—	—	—	2.62×10^{-01}

(a) Fisher LSD test p-values for the linear judgement scale.

	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Iteration 0	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
Iteration 1	—	8.10×10^{-02}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
Iteration 2	—	—	3.75×10^{-01}	1.40×10^{-02}
Iteration 3	—	—	—	4.14×10^{-01}

(b) Games-Howell test p-values for the power judgement scale.

	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Iteration 0	4.34×10^{-01}	5.32×10^{-01}	9.97×10^{-01}	8.30×10^{-01}
Iteration 1	—	1.60×10^{-01}	4.36×10^{-01}	3.19×10^{-01}
Iteration 2	—	—	5.30×10^{-01}	6.82×10^{-01}
Iteration 3	—	—	—	8.27×10^{-01}

(c) Fisher LSD test p-values for the root judgement scale.

	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Iteration 0	1.81×10^{-02}	5.41×10^{-02}	1.38×10^{-03}	2.01×10^{-04}
Iteration 1	—	6.54×10^{-01}	3.85×10^{-01}	1.57×10^{-01}
Iteration 2	—	—	1.89×10^{-01}	6.31×10^{-02}
Iteration 3	—	—	—	5.81×10^{-01}

(d) Fisher LSD test p-values for the geometric judgement scale.

	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Iteration 0	8.91×10^{-01}	8.07×10^{-01}	3.50×10^{-01}	2.08×10^{-01}
Iteration 1	—	9.14×10^{-01}	4.26×10^{-01}	2.62×10^{-01}
Iteration 2	—	—	4.90×10^{-01}	3.10×10^{-01}
Iteration 3	—	—	—	7.44×10^{-01}

(e) Fisher LSD test p-values for the balanced judgement scale.

	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Iteration 0	8.99×10^{-04}	2.53×10^{-04}	1.45×10^{-04}	6.64×10^{-04}
Iteration 1	—	7.18×10^{-01}	6.09×10^{-01}	9.29×10^{-01}
Iteration 2	—	—	8.81×10^{-01}	7.85×10^{-01}
Iteration 3	—	—	—	6.73×10^{-01}

(f) Fisher LSD test p-values for the logarithmic judgement scale.

TABLE 7.2: *Fisher's LSD and Games-Howell test p-values for the GLPMI means in Table 7.1 resulting from aggregation by the EM. Table entries smaller than [larger than, respectively]* 0.05 *are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the null-hypothesis (stating that the GLPMI means of two learning iterations are equal), may be rejected [may not be rejected, respectively].*

the consistency of the level of learning effectiveness of the linear, root, geometric, balanced and logarithmic AHP judgement scales over the experimental replications when combined with the EM. Fisher's LSD *post hoc* test is therefore applicable when seeking to isolate pairs of learning iterations for which the GLPMI means differ at a statistical significance level of $\alpha = 5\%$ in the case of combining the linear, root, geometric, balanced and logarithmic AHP judgement scales with the EM.

The relevant Fisher LSD test p-values may be found in Tables 7.2(a) and 7.2(c)–(f). The results in Tables 7.2(c) and 7.2(e) reveal that for the root and balanced AHP judgement scales, the mean GLPMIs are statistically indistinguishable for each pair of iterations at a 5% level of statistical significance when combined with the EM. Hence it may be claimed with 95% confidence that population of the DSS data management component with the EM in conjunction with the root and balanced AHP judgement scales is incapable of learning the true underlying value function of the DM in an iterative fashion after four iterations of the learning cycle in the context of the 2DCSP. For the logarithmic AHP judgement scale, the mean GLPMIs are only different at a 5% level of statistical significance between Iterations 0 and 1, between Iterations 0 and 2, between Iterations 0 and 3, and also between Iterations 0 and 4, but the mean learning performance indicators are not distinguishable at a 5% level of statistical significance for any of the remaining pairs of iterations, as may be deduced from the red values in Tables 7.2(f). This means that it may be claimed with 95% confidence that incorporating the EM in conjunction with the logarithmic AHP judgement scale in the DSS data management component results in learning the true underlying value function of the DM (to some extent) within the context of the 2DCSP only for one iteration of the learning cycle, but that this learning process then stagnates from there onwards. A similar claim may be made in the case of incorporating the geometric AHP judgement scale in Table 7.2(d), but in this case it appears that after Iteration 1 the learning stagnates and in fact does worse during the second iteration, as may be seen from the p-value of 0.054 between Iterations 0 and 2 which exceeds the significance level of 5%. The results for the linear AHP judgement scale and EM in Table 7.2(a) suggests that the mean GLPMIs are statistically indistinguishable between the pairs of Iterations 1 and 2, between Iterations 1 and 3, between Iterations 2 and 3, and between Iterations 3 and 4 at a 5% level of statistical significance. The remaining iteration pairs can be claimed with 95% confidence as being statistically different and from a visual inspection of Figure 7.1(a) it may be deducted that the framework does indeed learn the DM's true underlying value function (to some extent), but does so only after every two iterations.

It furthermore follows from the Levene test p-values in Table 7.1 that the relevant null-hypothesis (stating that the GLPMI samples are homoscedastic across all learning iterations) may also be rejected at a statistical significance level of $\alpha = 5\%$ for the power AHP judgement scale. Based on a visual inspection of the box plots in Figure 7.1(b), this means that it can be concluded with 95% confidence that the apparent gradual decrease in inter-quartile ranges of the box plots in Figure 7.1(b) are confirmed statistically as more learning iterations are completed. The practical interpretation of this conclusion is that the consistency of the level of learning effectiveness improves as a function of increasing learning iterations if the power AHP judgement scale is combined with the EM within the context of the 2DCSP. It follows that the non-parametric Games-Howell *post hoc* test should be employed to isolate pairs of learning iterations for which the GLPMI means differ at a statistical significance level of $\alpha = 5\%$ in the case of combining the power AHP judgement scale with the EM.

The relevant Games-Howell test p-values may be found in Table 7.2(b). The results in Table 7.2(b) reveal that for the power AHP judgement scale, the GLPMI means are different for each pair of iterations at a 5% level of statistical significance when combined with the EM, except between the adjacent pairs after Iteration 1 (to be more specific, between Iterations 1 and 2, between Iterations 2 and 3, and between Iterations 3 and 4). Hence it may be claimed with 95% confidence that population of the DSS data management component with the EM in conjunction with the power AHP judgement scale is capable of learning the true underlying value function of the DM (to some extent) in an iterative fashion, but does so, with statistical confidence, over the first two iterations of the learning cycle only in the context of the 2DCSP, after which the learning process stagnates.

The findings of this section are summarised in Table 7.3.

Judgement scale	Does iterative learning take place?	Does the level of consistency of learning increase per iteration?
Linear	Yes, but only after every second iteration	No
Power	Yes, but only after every second iteration	Yes
Root	No	No
Geometric	Yes, only for two iterations and then stagnates	No
Balanced	No	No
Logarithmic	Yes, only for one iteration and then stagnates	No

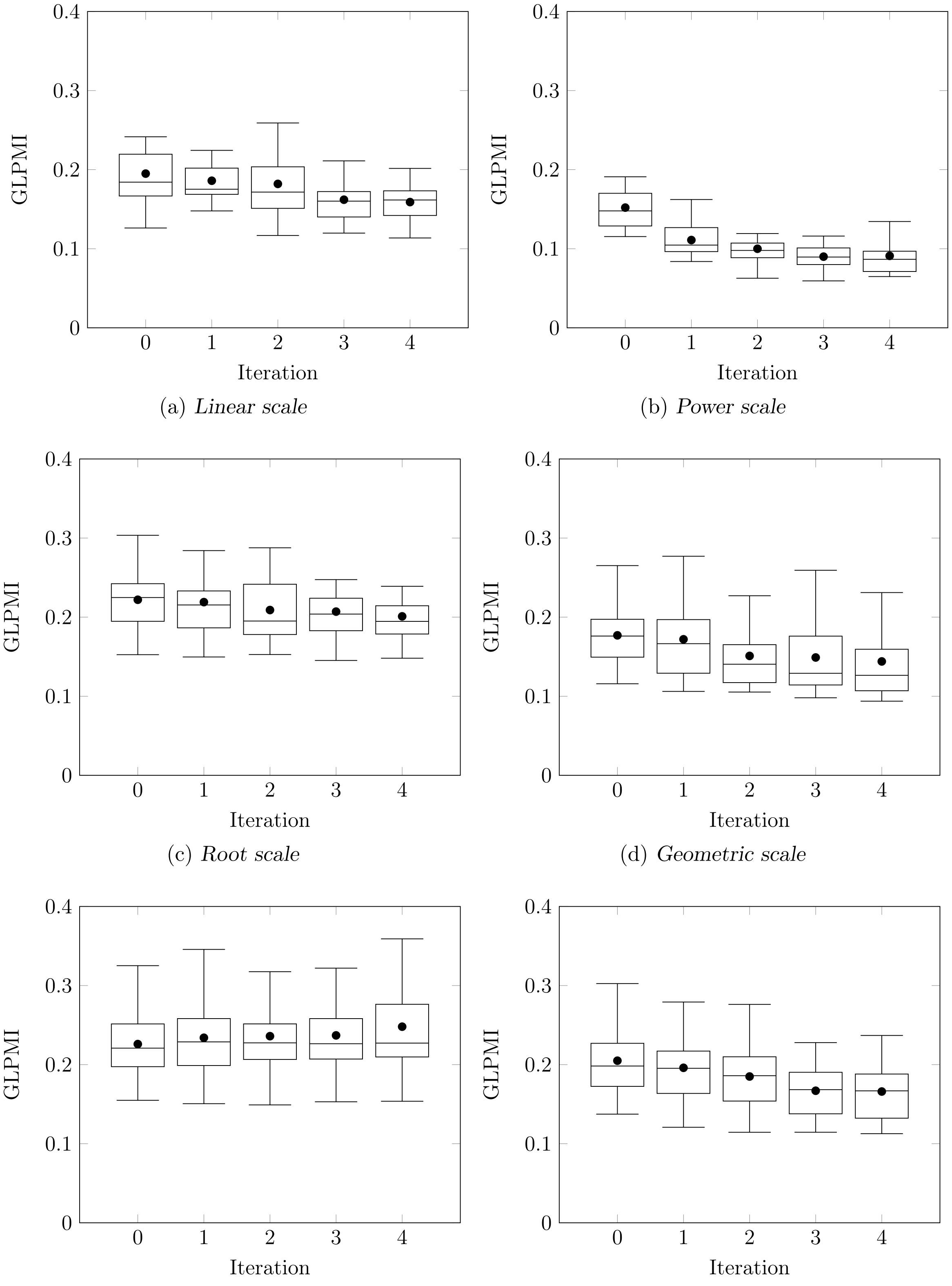
Table 7.3: *Summary of findings at a 95% level of confidence if the EM is adopted in a ratio-based approach combined with various AHP judgement scales within the context of the 2DCSP.*

Learning based on the LLSM of aggregation

The EM is replaced by the LLSM in this section in an assessment of the extent to which the DSS framework of §5 is capable of learning, similar to that carried out in the previous section. Figures 7.2(a)–(f) contain box plots of the GLPMI values in (6.3) associated with $\omega = 4$ iterations of the learning cycle of Figure 5.1 when populating the DSS framework of §5 with each of the six AHP judgement scales of Table 2.2, respectively, in conjunction with the LLSM. In each of these cases, the comparison sets $\mathcal{C}_0$, $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$ and $\mathcal{C}_4$ again contain $n = 12$ alternatives each. Only $\chi = 1$ anchor is employed (again the alternative $(0, 0)$). For each of the six judgement scales, a total of $N = \omega + 1 = 5$ samples of GLPMI values are therefore produced (one for each iteration of the learning cycle), each again containing $M = 30$ observations. Since the objective is similar to that in the previous section, namely to assess the *intrinsic* ability of the DSS framework of §5 to learn, no optimisation (or exploitation) component is again employed in any of the six cases. Therefore, the alternatives $\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,n-1}$ of each comparison set $\mathcal{C}_i$ are once again generated purely randomly (according to a uniform distribution on $\mathcal{D}$) without exploiting areas of the decision space that seem to be performing well in terms of the current DM value function estimate. The means of the GLPMI samples are reproduced in Table 7.4 along with the corresponding ANOVA and Levene test p-values for the samples.

It follows from the p-values of the ANOVA test in Table 7.4 that, for each of the six judgement scales, the relevant null-hypothesis (stating that the GLPMI means across all iterations are statistically indistinguishable) may be rejected at a significance level of $\alpha = 5\%$, except for the root and balanced judgement scales (as their p-values of 0.179 and 0.536, respectively, exceed the significance level of 0.05 for the ANOVA test). The box plots in Figures 7.2(a), (b), (d) and (e) provide a visual indication that *some* degree of learning takes place in the context of the 2DCSP if, similar to the results in the previous section, any of the linear, power, geometric or logarithmic AHP judgement scales are combined with the LLSM. This visual conclusion is therefore confirmed at a 95% level of confidence by the ANOVA test.

It furthermore follows from the Levene test statistics in Table 7.4 that the relevant null-hypothesis (stating that the GLPMI samples are homoscedastic across all iterations) may also be rejected



(a) *Linear scale*

(b) *Power scale*

(c) *Root scale*

(d) *Geometric scale*

(e) *Balanced scale*

(f) *Logarithmic scale*

FIGURE 7.2: *Box plots of the GLPMI values in (6.3) associated with four iterations of the learning cycle when populating the DSS framework of §5 with each of the six AHP judgement scales of Table 2.2 in conjunction with the LLSM in the absence of exploitation of promising areas of the 2DCSP decision space. Mean values are indicated by solid dots, while median values are denoted by central quartile dividers.*

	GLPMI mean						
	Iteration	Iteration	Iteration	Iteration	Iteration	ANOVA	Levene
Scale	0	1	2	3	4	test p-value	test p-value
Linear	0.195	0.186	0.182	0.162	0.159	2.53×10^{-05}	1.95×10^{-02}
Power	0.152	0.111	0.100	0.090	0.091	$< 1 \times 10^{-17}$	2.82×10^{-01}
Root	0.222	0.219	0.209	0.207	0.201	1.79×10^{-01}	8.89×10^{-01}
Geometric	0.177	0.172	0.151	0.149	0.144	1.77×10^{-02}	3.76×10^{-01}
Balanced	0.226	0.234	0.236	0.237	0.248	5.36×10^{-01}	6.26×10^{-01}
Logarithmic	0.205	0.196	0.185	0.167	0.166	2.26×10^{-04}	8.40×10^{-01}

TABLE 7.4: *GLPMI means associated with four iterations of the learning cycle when populating the DSS framework of §5 with the AHP ratio-based approach, using each of the six AHP judgement scales of Table 2.2 in conjunction with the LLSM. The corresponding p-values returned for the ANOVA and Levene tests are also included. Table entries smaller than [larger than, respectively]* 0.05 *are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the relevant null-hypothesis may be rejected [may not be rejected, respectively].*

at a statistical significance level of $\alpha = 5\%$ if the linear AHP judgement scale is combined with the LLSM. Hence, at a 95% level of confidence, there is statistical evidence that the level of consistency of learning across the 30 experimental replications changes as the learning cycle of Figure 5.1 progresses in the case of the 2DCSP (the respective Levene test p-value of 1.95×10^{-02} does not exceed the significance level of 0.05). The remaining Levene test p-values of 0.282, 0.889, 0.376, 0.626 and 0.840 for the cases of the power, root, geometric, balanced and logarithmic AHP judgement scales, respectively, combined with the LLSM exceed the significance level of 0.05 and it therefore cannot be claimed with 95% confidence that there is an improvement or worsening in the consistency of the level of global learning effectiveness of these AHP judgement scales over the experimental replications when combined with the LLSM. Fisher's LSD *post hoc* test is therefore employed to isolate pairs of learning iterations for which the GLPMI means differ at a statistical significance of $\alpha = 5\%$ in the case of combining the power, root, geometric, balanced and logarithmic AHP judgement scales with the LLSM, while the non-parametric Games-Howell test is employed for this purpose in the case of the remaining linear AHP judgement scale. The relevant Games-Howell and Fisher LSD test p-values may be found in Table 7.5.

According to Fisher's LSD test p-value results in Table 7.5(e) for the balanced AHP judgement scale, the GLPMI means are indistinguishable for each pair of iterations at a 5% level of statistical significance. In the case of the root AHP judgement scale in Table 7.5(c), Fisher's LSD test p-values indicate that the GLPMI means are all indistinguishable at a 5% level of significance, except between Iterations 0 and 4. It may therefore be claimed with 95% confidence that the balanced AHP judgement scale, when combined with the LLSM, is incapable of learning the true underlying value function of the DM in an iterative fashion within the context of the 2DCSP. The same may be claimed for the case of the root AHP judgement scale when combined with the LLSM, except after four iterations of the learning cycle, at which point global learning of the true underlying value function of the DM is detected (to some extent). In the case of the logarithmic AHP judgement scale in Table 7.5(f) it appears that it can only be claimed with 95% confidence that there exists a difference between Iterations 0 and 3, between Iterations 0 and 4, between Iterations 1 and 3, and between Iterations 1 and 4. This implies that global learning only occurs after three iterations of the learning cycle. Fisher's LSD test p-values for the geometric AHP judgement scale in Table 7.5(d) indicates that learning occurs only after two iterations of the learning cycle (as indicated by the black value between Iterations 0 and 2). The final Fisher's LSD test results for the power judgement scale in Table 7.5(b) seem to learn faster than any of the aforementioned scales, with learning occurring after one iteration (between Iterations 0 and 1) and then after two further iterations (between Iterations 1 and 3). It may therefore be claimed

	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Iteration 0	8.31×10^{-01}	6.81×10^{-01}	4.00×10^{-03}	1.00×10^{-03}
Iteration 1	—	9.84×10^{-01}	8.00×10^{-03}	1.00×10^{-03}
Iteration 2	—	—	1.83×10^{-01}	7.60×10^{-02}
Iteration 3	—	—	—	9.97×10^{-01}

(a) Games-Howell p-values for the linear judgement scale.

	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Iteration 0	5.37×10^{-10}	2.84×10^{-14}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
Iteration 1	—	7.54×10^{-02}	7.19×10^{-04}	1.10×10^{-03}
Iteration 2	—	—	9.80×10^{-02}	1.26×10^{-01}
Iteration 3	—	—	—	9.00×10^{-01}

(b) Fisher LSD p-values for the power judgement scale.

	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Iteration 0	7.24×10^{-01}	1.65×10^{-01}	1.21×10^{-01}	3.24×10^{-02}
Iteration 1	—	2.99×10^{-01}	2.30×10^{-01}	7.31×10^{-02}
Iteration 2	—	—	8.71×10^{-01}	4.46×10^{-01}
Iteration 3	—	—	—	5.49×10^{-01}

(c) Fisher LSD p-values for the root judgement scale.

	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Iteration 0	6.53×10^{-01}	3.18×10^{-02}	1.88×10^{-02}	6.19×10^{-03}
Iteration 1	—	8.80×10^{-02}	5.61×10^{-02}	2.13×10^{-02}
Iteration 2	—	—	8.35×10^{-01}	5.43×10^{-01}
Iteration 3	—	—	—	6.88×10^{-01}

(d) Fisher LSD p-values for the geometric judgement scale.

	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Iteration 0	5.32×10^{-01}	4.25×10^{-01}	3.55×10^{-01}	8.35×10^{-02}
Iteration 1	—	8.62×10^{-01}	7.64×10^{-01}	2.66×10^{-01}
Iteration 2	—	—	8.99×10^{-01}	3.48×10^{-01}
Iteration 3	—	—	—	4.16×10^{-01}

(e) Fisher LSD p-values for the balanced judgement scale.

	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Iteration 0	3.57×10^{-01}	5.12×10^{-02}	2.75×10^{-04}	1.57×10^{-04}
Iteration 1	—	2.99×10^{-01}	5.72×10^{-03}	3.61×10^{-03}
Iteration 2	—	—	8.01×10^{-02}	5.74×10^{-02}
Iteration 3	—	—	—	8.79×10^{-01}

(f) Fisher LSD p-values for the logarithmic judgement scale.

Table 7.5: *Fisher's LSD and Games-Howell test p-values for the GLPMI means in Table 7.4 resulting from aggregation by the LLSM. Table entries smaller than [larger than, respectively]* 0.05 *are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the null-hypothesis (stating that the GLPMI means of two learning iterations are equal), may be rejected [may not be rejected, respectively].*

with 95% confidence that the power AHP judgement scale, when combined with the LLSM, is incapable of learning the true underlying value function of the DM in an iterative fashion, within the context of the 2DCSP.

The relevant Games-Howell test p-values may be found in Tables 7.5(a). It follows from the results in Tables 7.5(a) that for the linear AHP judgement scale, the GLPMI means are different between Iterations 0 and 3, between Iterations 0 and 4, between Iterations 1 and 3, between Iterations 1 and 4, and between Iterations 2 and 4 at a 5% level of statistical significance. It may therefore be claimed with 95% confidence that incorporating the LLSM in conjunction with the linear AHP judgement scale in the DSS data management component also results in learning the true underlying value function of the DM (to some extent) in an iterative fashion within the context of the 2DCSP, but only after three iterations of the learning cycle.

The findings of the analysis in this section are summarised in Table 7.6 for a combination of the various AHP judgement scales with the LLSM.

Judgement scale	Does iterative learning take place?	Does the level of consistency of learning increase per iteration?
Linear	Yes, but only after 3 iterations	Yes
Power	Yes, but learning decreases per iteration	No
Root	Yes, but only after 4 iterations	No
Geometric	Yes, but only after 2 iterations	No
Balanced	No	No
Logarithmic	Yes, but only after 3 iterations	No

TABLE 7.6: *Summary of findings at a 95% level of confidence if the LLSM is adopted in a ratio-based approach combined with various AHP judgement scales within the context of the 2DCSP.*

Based on the results reported above, as well as in the previous section, it can be claimed with 95% confidence that the DSS framework of §5 is capable of intrinsic learning (to some extent) of the value function of a perfectly consistent DM in the context of the 2DCSP, but that the rate of learning differs significantly over the various AHP judgement scales. It would seem that the AHP judgement scale that best facilitates learning is the power judgement scale. This might be due to the considerable rate of increase that the power scale exhibits $(1, 4, 9, 16, 25, 36, 49, 64, 81)$. This begs the question as to why the geometric judgement scale performs so unsatisfactorily in terms of facilitating learning in the case of the 2DCSP, as it has an even larger rate of increase $(2, 4, 8, 16, 32, 64, 128, 256)$. The reason for this could perhaps be due to the assumed form of the true value function of the DM in (6.1), but this is mere speculation.

A natural question now arises as to which AHP judgement scale and aggregation method combinations perform the best in the context of approximating the DM's value function. This question is addressed in §7.3.

7.2.2 Adopting a difference-based approach

The MACBETH method incorporates a single six-point judgement scale and does not incorporate the EM or LLSM of aggregation. This section is, therefore, only dedicated to the assessment of the single case of populating the DSS framework over four iterations of the learning cycle of §5 with the MACBETH difference-based approach.

Figure 7.3 contains box plots of the GLPMI values in (6.3) associated with $\omega = 4$ iterations of the learning cycle of Figure 5.1 when populating the DSS framework of §5. Similar to the approach in §7.2.1, the comparison sets $\mathcal{C}_0$, $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$ and $\mathcal{C}_4$ each contains $n = 12$ colour alternatives with $\chi = 1$ anchor employed (the alternative $(0, 0)$). A total of $N = \omega + 1 = 5$ samples of GLPMI values are again produced for each iteration of the learning cycle, each containing $M = 30$ observations. As explained in §7.2.1, no optimisation (or exploitation) is employed within the DSS framework in this section, and so the alternatives in each comparison set are generated purely randomly (according to uniform distribution on $\mathcal{D}$).

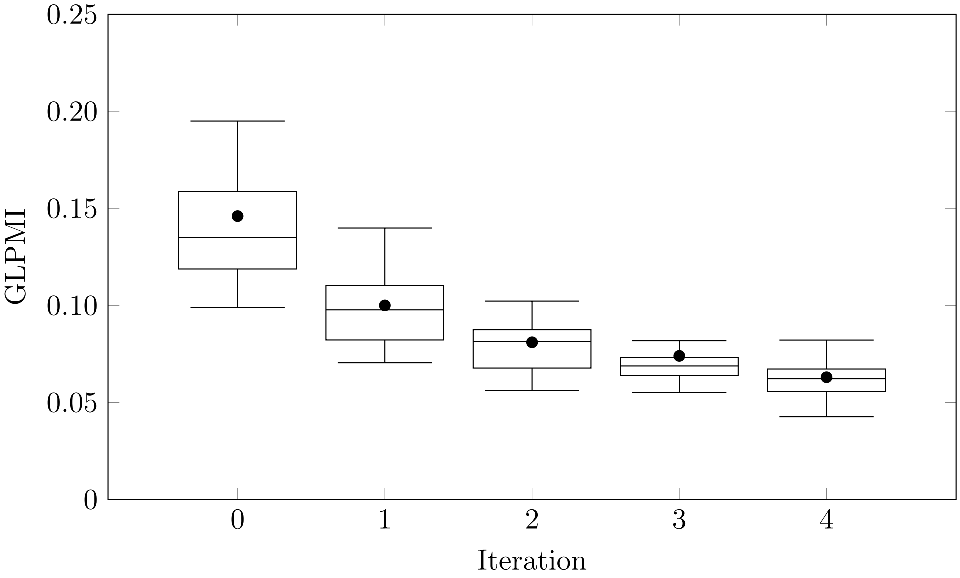


FIGURE 7.3: *Box plots of the GLPMI values in Table 7.17 associated with the best alternative uncovered during the fourth iteration of the learning cycle when populating the DSS framework of §5 with each of the six judgement scales of Table 2.2 in conjunction with the EM and the LLSM. Mean values are indicated by solid dots, while median values are denoted by a central quartile divider.*

It follows from the ANOVA test p-value in Table 7.7 that the relevant null-hypothesis (stating that the GLPMI means across all learning iterations are statistically indistinguishable) may be rejected at a statistical significance level of $\alpha = 5\%$ (the ANOVA test p-value is exceeded by the 0.05 level of significance). Based on a visual inspection of the box plots in Figure 7.3, it may therefore be concluded with 95% confidence that *some* degree of learning takes place in the context of the 2DCSP if the MACBETH difference-based method is incorporated in the DSS framework of §5.

	GLPMI mean						
Method	Iteration 0	Iteration 1	Iteration 2	Iteration 3	Iteration 4	ANOVA test p-value	Levene test p-value
MACBETH	0.146	0.100	0.081	0.074	0.063	$< 1 \times 10^{-17}$	1.76×10^{-04}

TABLE 7.7: *GLPMI means associated with four iterations of the learning cycle when populating the DSS framework of §5 with the MACBETH difference-based approach. The corresponding p-values returned for the ANOVA and Levene tests are also included. Table entries smaller than [larger than, respectively] 0.05 are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the relevant null-hypothesis may be rejected [may not be rejected, respectively].*

The Levene test p-value in Table 7.7 indicates that the null-hypothesis (stating that the GLPMI samples are homoscedastic across all learning iterations) may also be rejected at a statistical significance level of $\alpha = 5\%$. Based on a visual inspection of the box plots in Figure 7.3, this means that it can be concluded with 95% confidence that the apparent gradual decrease in inter-quartile ranges of the box plots are confirmed statistically as more learning iterations are completed. The practical interpretation of this conclusion is that the consistency of the level of learning effectiveness improves as a function of increasing learning iterations when the difference-based MACBETH method is incorporated within the context of the 2DCSP. It follows that the non-parametric Games-Howell *post hoc* test should be employed to isolate pairs of learning iterations for which the GLPMI means differ at a statistical significance level of $\alpha = 5\%$.

	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Iteration 0	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
Iteration 1	—	2.00×10^{-03}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
Iteration 2	—	—	6.10×10^{-01}	$< 1 \times 10^{-17}$
Iteration 3	—	—	—	1.01×10^{-01}

Table 7.8: *Games-Howell test p-values for the GLPMI means in Table 7.4 resulting from incorporating the MACBETH difference-based approach. Table entries smaller than [larger than, respectively]* 0.05 *are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the null-hypothesis (stating that the GLPMI means of two learning iterations are equal), may be rejected [may not be rejected, respectively].*

The relevant Games-Howell test p-values may be found in Table 7.8. The results in Table 7.8 reveal that for the MACBETH difference-based approach, the GLPMI means are different for all pairs of iterations at a 5% level of statistical significance, except between Iterations 2 and 3, and between Iterations 3 and 4. Hence it may be claimed with 95% confidence that population of the DSS data management component is capable of learning the true underlying value function of the DM (to some extent) in an iterative fashion when populated with the MACBETH difference-based approach, yet it appears that the learning process begins to stagnate after three iterations of the learning cycle in the context of the 2DCSP.

7.3 The AHP judgement scales and aggregation methods

Although the results in §7.2 certainly indicate an ability of the DSS framework of §5 to facilitate learning (to some extent) within the context of the 2DCSP if populated with the AHP ratio-based approach in conjunction with certain judgement scales, no indication was provided in §7.2 as to how well each judgement scale performs relative to the others, and which of the two aggregation methods is superior. That is, the framework of §5 might have been able to learn the true value function $U(g, b)$ in (4.1) to some extent, but the preferred judgement scale and aggregation method combination is not yet clear. Therefore, the relative performances of the various judgement scale and aggregation method combinations are once again measured by the GLPMI of (6.3) in this section, but this time only within the context of the fourth learning iteration of each of the experiments of the first part of §7.2.1.

Figure 7.4 contains box plots of the GLPMI values after four iterations of the learning cycle of Figure 5.1 for each of the six AHP judgement scales in combination with the EM and the LLSM. The boxes in this plot are exactly those corresponding to the twelve fourth-iteration boxes in the plots of Figures 7.1 and 7.2. A visual analysis of the figure reveals that the various judgement scale and aggregation method combinations seem not to perform equally well after the fourth iteration of the DSS learning cycle. The results in Table 7.9 confirm this, with the ANOVA test p-values revealing that the null hypothesis (stating that the GLPMI means are statistically indistinguishable across all judgement scales) may be rejected at a significance level of $\alpha = 5\%$ — for both the EM and the LLSM. From the Levene test p-values of Table 7.9, it furthermore follows that the null hypothesis (stating that the GLPMI samples are homoscedastic across all judgement scales) may also be rejected at a significance level of $\alpha = 5\%$, again for both the

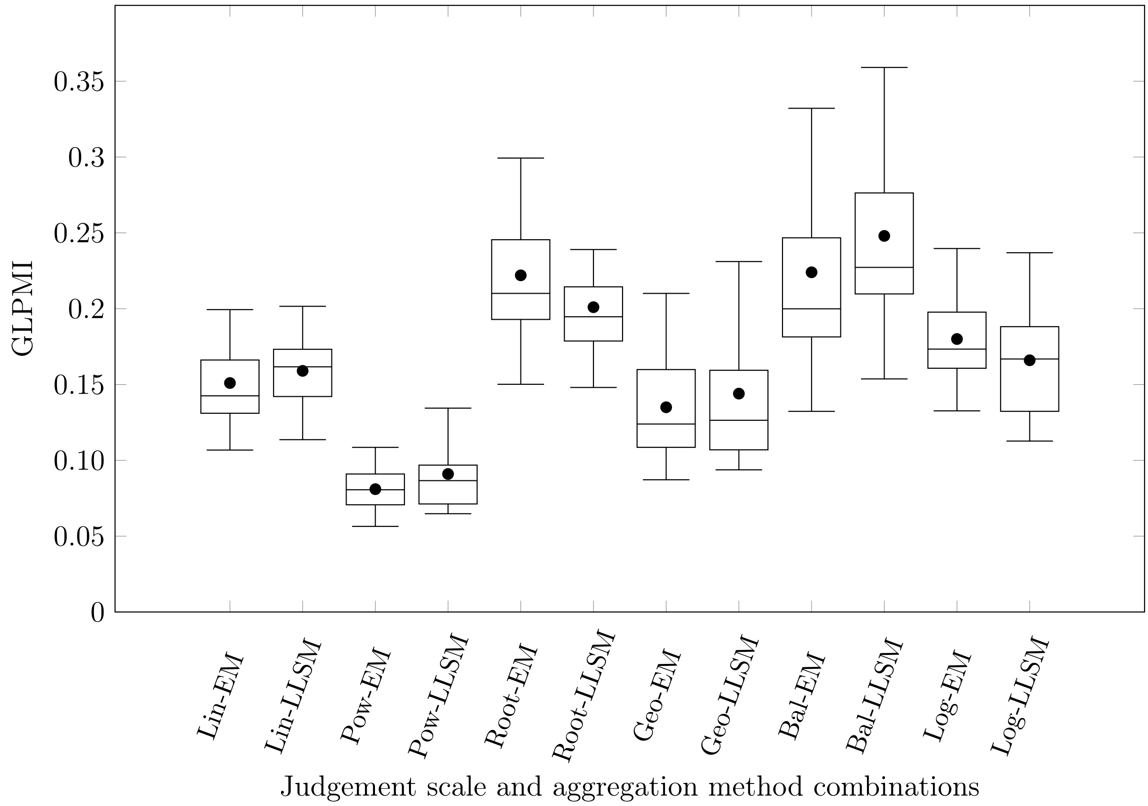


FIGURE 7.4: *Box plots of the GLPMI values in (6.3) associated with the fourth iteration of the learning cycle when populating the DSS framework of §5 with the ratio-based AHP, adopting each of the six judgement scales of Table 2.2 in conjunction with the EM and the LLSM. Judgement scale and aggregation method combinations are abbreviated in the form XY, where X denotes the judgement scale (with X = Lin for the linear scale, X = Pow for the power scale, X = Root for the root scale, X = Geo for the geometric scale, X = Bal for the balanced scale and X = Log for the logarithmic scale) and Y denotes the method of aggregation (with Y = EM for the EM and Y = LLSM for the LLSM). Mean values are indicated by solid dots, while median values are denoted by central quartile dividers.*

EM and the LLSM. It can therefore be claimed with 95% confidence that the various judgement scale and aggregation method combinations do not perform equally consistently over the 30 experimental replications during the fourth learning iteration of Figure 5.1.

Applying the non-parametric Games-Howell *post hoc* test in both cases in which the null hypotheses have been rejected, the pairs of judgement scales for which the GLPMI means differ at a significance level of $\alpha = 5\%$ may be isolated. The relevant Games-Howell *post hoc* test results may be found in Tables 7.10(a) and 7.10(b) for the EM and the LLSM, respectively. The Games-Howell test p-values in Table 7.10(a) reveal that for each pair of judgement scales, the GLPMI means are statistically different at a significance level of $\alpha = 5\%$ when combined with the EM, except between the linear and geometric judgement scales, and between the root and balanced judgement scales. It may therefore be claimed with 95% confidence that all pairs of combinations of judgement scales yield GLPMI means that differ significantly after four iterations of the learning cycle in the DSS framework of §5 when employing the ratio-based AHP in conjunction with the EM, except for the linear and geometric judgement scales, and for the root and balanced judgement scales. Exactly the same claim can be made for all combinations of the judgement scales in the context of the LLSM, except for three of the fifteen pairs, namely for the linear and geometric, linear and logarithmic, and geometric and logarithmic judgement scale combinations.

	GLPMI						ANOVA	Levene
	Linear	Power	Root	Geometric	Balanced	Log	p-value	p-value
EM	0.151	0.081	0.222	0.135	0.224	0.180	$< 1 \times 10^{-17}$	8.35×10^{-09}
LLSM	0.159	0.091	0.201	0.144	0.248	0.166	$< 1 \times 10^{-17}$	1.44×10^{-04}

TABLE 7.9: *GLPMI means associated with the fourth iteration of the learning cycle when populating the DSS framework of §5 with each of the six judgement scales of Table 2.2 in conjunction with the EM and the LLSM, together with the relevant p-values for the ANOVA and Levene statistical tests.*

	Power	Root	Geometric	Balanced	Logarithmic
Linear	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	3.85×10^{-01}	$< 1 \times 10^{-17}$	5.00×10^{-03}
Power	—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
Root	—	—	$< 1 \times 10^{-17}$	1.00×10^{-01}	$< 1 \times 10^{-17}$
Geometric	—	—	—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
Balanced	—	—	—	—	2.80×10^{-02}

(a) The EM

	Power	Root	Geometric	Balanced	Logarithmic
Linear	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	6.74×10^{-01}	$< 1 \times 10^{-17}$	9.70×10^{-01}
Power	—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
Root	—	—	$< 1 \times 10^{-17}$	5.00×10^{-03}	5.00×10^{-03}
Geometric	—	—	—	$< 1 \times 10^{-17}$	4.23×10^{-01}
Balanced	—	—	—	—	$< 1 \times 10^{-17}$

(b) The LLSM

TABLE 7.10: *Games-Howell post hoc p-values for the GLPMI means in Table 7.4 resulting from aggregation by the EM and the LLSM. All table entries are below 5% and hence denote significant differences in the sample data.*

A visual inspection of Figure 7.4 suggests that there are no significant differences between the GLPMI means returned by the two methods of aggregation for any fixed judgement scale. In order to confirm this statistically, ANOVA and Levene tests were again carried out for each of the six judgement scales combined with the EM and the LLSM. The results may be found in Table 7.11. The ANOVA test p-values in the table are all larger than 0.05, except for the p-value associated with the case of the root judgement scale. It may therefore be claimed with 95% confidence that the GLPMI means returned by the two methods of aggregation are statistically indistinguishable for all AHP judgement scales, save for the root judgement scale. It furthermore follows from the Levene test p-values in Table 7.11 that the aggregation methods yield GLPMI mean samples that are homoscedastic for all judgement scales with 95% confidence, as all these p-values are larger than the significance level of 0.05.

It is interesting to note that the two best-performing judgement scales in Figure 7.4 (the power and geometric scales) are both convex functions. Furthermore, the linear scale (which is both convex and concave) performs midway on the vertical axis scale of Figure 7.4. Finally, the two worst-performing judgement scales actually capable of intrinsic learning[2] in Figure 7.4 (the root and logarithmic scales) are both concave functions. Why convex judgement scales should fare better in learning the value function (6.1) — which is itself neither convex nor concave — is unclear. In order to maintain diversity in this respect, however, the best-performing convex judgement scale (the power scale), the linear judgement scale, and the best-performing concave judgement scale (the logarithmic scale) are taken forward in the analyses of the following section

[2] Recall, from §7.2, that the balanced judgement scale does not support learning as measured by the GLPMI (6.3).

	GLPMI		ANOVA	Levene
	EM	LLSM	p-value	p-value
Linear	0.151	0.159	2.15×10^{-01}	4.81×10^{-01}
Power	0.081	0.091	8.15×10^{-02}	2.15×10^{-01}
Root	0.222	0.201	3.25×10^{-02}	5.03×10^{-01}
Geometric	0.135	0.144	4.17×10^{-01}	1.59×10^{-01}
Balanced	0.224	0.248	1.54×10^{-01}	3.67×10^{-01}
Logarithmic	0.180	0.166	1.05×10^{-01}	2.83×10^{-01}

Table 7.11: *GLPMI means associated with the fourth iteration of the learning cycle when populating the DSS framework of §5 with each of the six judgement scales of Table 2.2 in conjunction with the EM and the LLSM, together with the relevant p-values for the ANOVA and Levene statistical tests. Table entries smaller than [larger than, respectively]* 0.05 *are typeset in black [red, respectively], show that the relevant probability of a Type I error is small [large, respectively] and hence that the null-hypothesis may be rejected [may not be rejected, respectively].*

aimed at quantifying the merits of exploration as opposed to those of exploitation during the learning cycle of Figure 5.1 within the context of the 2DCSP. Furthermore, only the LLSM is henceforth considered within the AHP (without loss of generality at 95% confidence in the context of the 2DCSP).

In conclusion, based on the statistical analysis of this section, it follows from a visual inspection of Figure 7.4 that there is a definitive ranking of judgement scale performance at a 95% level of confidence when solving the 2DCSP, as shown in Table 7.12, irrespective of the method of aggregation.

Rank	Judgement scale
Most preferred (1)	**Power scale**
Second-most preferred (2)	Geometric scale
Third-most preferred (2)	**Linear scale**
Fourth-most preferred (2)	**Logarithmic scale**
Fifth-most preferred (3)	Root scale (LLSM)
Least preferred (4)	Balanced scale

Table 7.12: *Preference ranking at a 95% confidence level of the AHP judgement scales in Table 2.2 for use in the 2DCSP, irrespective of the aggregation method employed.*

7.4 The merits of exploitation *versus* exploration

The GLPMI in (6.3) has hitherto been employed as the sole measure of learning success as more and more iterations of the learning cycle of Figure 5.1 are performed. This was because the analyses in §7.2 and §7.3 were concerned with the intrinsic ability of various DSS framework populations with respect to learning the true value function in (6.1) over the entire decision space $\mathcal{D}$. Furthermore, a purely explorative approach was adopted in §7.2 and §7.3 during the generation of colour alternatives not yet considered by the DM for inclusion in new comparison sets. The purpose of this section, however, is to investigate the merits of exploitation over mere exploration when generating new alternatives for presentation to the DM. The DM may arguably not be interested in the entire value function of Figure 6.1, but may instead be particularly concerned with the ability of the DSS to return colour alternatives that are close to the globally most preferred colour $(g, b) = (63, 63)$ (*i.e.* only that part of the value function close to its global maximum).

For this reason, both the GLPMI in (6.3) and the LLPMI in (6.4) are employed in this section to measure learning success.

As mentioned in the previous section, only the three AHP judgement scales typeset in boldface in Table 7.12 are considered in conjunction with the LLSM when populating the DSS framework with a ratio-based approach. Three methods of generation of new colour alternatives are considered — the purely explorative approach of the previous two sections (in which new colour alternatives are generated according to a uniform distribution over the entire decision space $\mathcal{D}$), a purely exploitative PHC approach (see §2.4.2), and the intermediate Basic E&E approach of §6 (combining elements of the previous two approaches). These three methods work as follows during the i-th iteration of the learning cycle (for $i > 0$):

Random approach (exploration). A total of $n-\chi$ alternatives are selected randomly according to a uniform distribution over the decision space $\mathcal{D}$ for inclusion in the i-th comparison set $\mathcal{C}_i$, where χ denotes the number of anchors employed. This method of alternative generation is henceforth referred to as *Random*, and was employed in §7.2 and §7.3.

Basic E&E approach (exploration and exploitation). A point (g_i^*, b_i^*) is found in the decision space $\mathcal{D}$ which globally maximises the value function estimate $w_{i-1}(g, b)$ produced during the previous iteration. The set $\mathcal{D}$ is partitioned into two subsets $\mathcal{E}_i$ and $\overline{\mathcal{E}}_i$. The set $\mathcal{E}_i$ contains all those alternatives within an exploitation radius r_i from (g_i^*, b_i^*), that is all alternatives (g, b) for which

$$\sqrt{(g - g_i^*)^2 + (b - b_i^*)^2} \leq r_i.$$

The set $\overline{\mathcal{E}}_i$ is merely the complement of $\mathcal{E}_i$. A total of $\sigma < n - \chi$ alternatives are drawn randomly from $\mathcal{E}_i$ according to a uniform distribution for inclusion in the i-th comparison set $\mathcal{C}_i$, and the remaining $n - \chi - \sigma$ alternatives are drawn similarly, but from $\overline{\mathcal{E}}_i$. This method of alternative generation is again referred to as *Basic E&E*, and was employed in the worked example of §6. The exploitation radii are taken here as $r_1 = 120$, $r_2 = 64$ and $r_3 = 34$ in order to allow for sufficient exploration early on during the learning cycle, but then to promote sharper exploitation later on.

PHC approach (exploitation). Suppose the previous comparison set $\mathcal{C}_{i-1}$ contained the non-anchor alternatives $(g_1, b_1), \ldots, (g_{n-\chi}, b_{n-\chi})$. Then a neighbourhood annulus

$$\mathcal{N}_i = \{(g, b) \in \mathcal{D} \mid r \leq \sqrt{(g - g_i)^2 + (b - b_i)^2} \leq R\}$$

of radii r and R is constructed around the alternative (g_i, b_i) or all $i = 1, \ldots, n-\chi$ (for some $r < R$). With probability p, a neighbouring alternative (g_i', b_i') of (g_i, b_i) is selected as a member of the annulus $\mathcal{N}_i$ for which the previous value function estimate $w_{i-1}(r, b)$ achieves a maximum, and with probability $1 - p$ the neighbouring alternative (g_i', b_i') is selected randomly from within $\mathcal{N}_i$ according to a uniform distribution. This method of alternative generation is therefore a form of PHC with the previous value function estimate $w_{i-1}(g, b)$ as objective function. The values of r and R are taken here as 8 and 35, respectively.

7.4.1 Global learning performance measure analysis

The analysis in this section is carried out separately for the cases of adopting a ratio-based or a difference-based approach.

Adopting a ratio-based approach

Figure 7.5 contains box plots of the GLPMI means of the fourth iterations (identical to those of Figure 7.2) for the linear, power and logarithmic AHP judgement scales in conjunction with the LLSM and the three aforementioned exploration/exploitation methods, namely Random, Basic E&E and PHC.

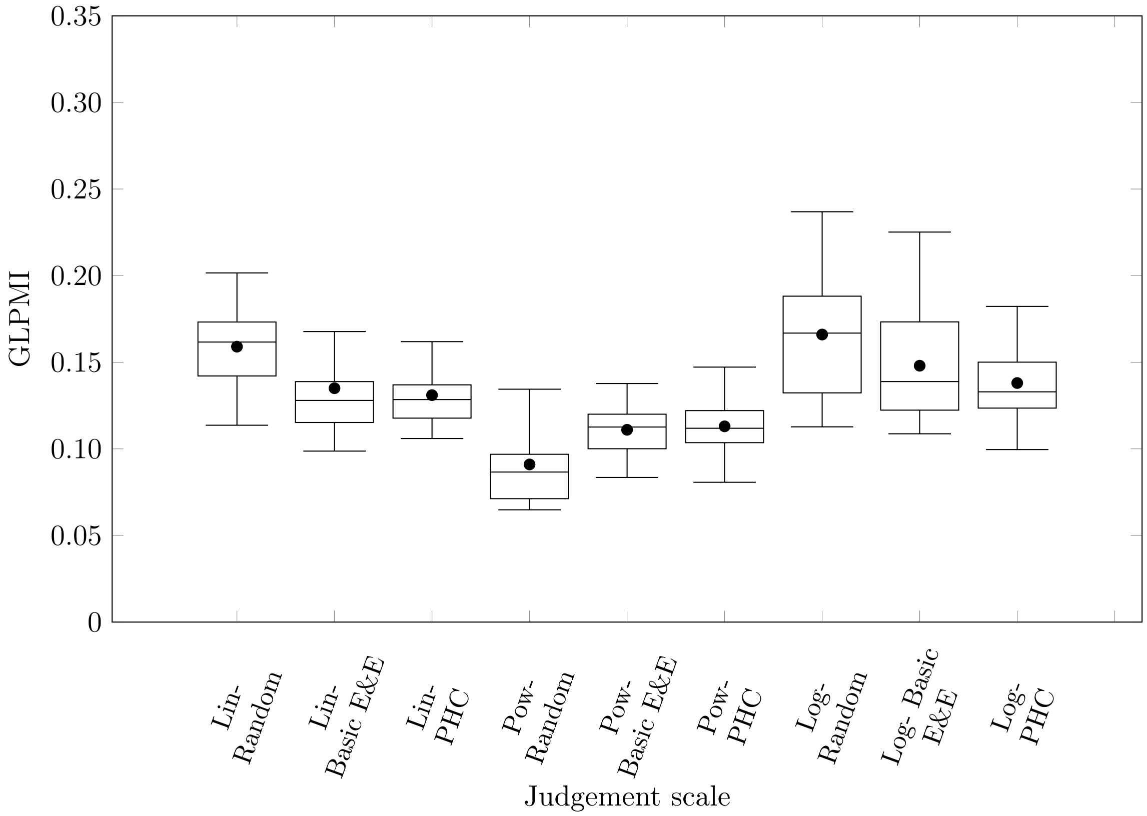


FIGURE 7.5: *Box plots of the GLPMI values in Table 7.13 associated with the fourth iteration of the learning cycle when populating the DSS framework of §5 with each of the three boldfaced AHP judgement scales of Table 7.12 in conjunction with the LLSM and when adopting the Random, Basic E&E and PHC exploration/exploitation methods. Mean values are indicated by solid dots, while median values are denoted by central quartile dividers.*

The GLPMI means are presented for the Random, Basic E&E and PHC exploration/exploitation methods for each of the three AHP judgement scales (linear, power and logarithmic) together with the corresponding ANOVA and Levene test p-values in Table 7.13. Since all of the ANOVA test p-values are smaller than 0.05, it can be claimed with 95% confidence that all three pairs of the exploration/exploitation methods differ statistically in terms of their global ability to learn the DM's underlying value function for all three AHP judgement scales. The Levene test p-values, however, suggest that for none of the judgement scales, may the null-hypothesis (stating that the GLPMI samples are homoscedastic across all methods of exploration/exploitation) be rejected. These corresponding Levene test p-values for the linear, power and logarithmic AHP judgement scales suggest at a 95% level of confidence that the variances of the GLPMI samples are statistically indistinguishable. Fisher's LSD *post hoc* test is therefore performed to determine between which exploration/exploitation methods a significant difference in GLPMI mean values occur.

		GLPMI		ANOVA	Levene
	Random	Basic E&E	PHC	*p*-value	*p*-value
Linear	0.159	0.135	0.131	5.67×10^{-05}	2.73×10^{-01}
Power	0.091	0.111	0.113	1.42×10^{-05}	1.68×10^{-01}
Logarithmic	0.166	0.148	0.138	4.29×10^{-03}	1.20×10^{-01}

Table 7.13: *GLPMI means associated with the fourth iteration of the learning cycle for the Random, Basic E&E and PHC exploration/exploitation methods of alternative generation when populating the DSS framework of §5 with the linear, root and geometric AHP judgement scales of Table 2.2 in conjunction with the LLSM, together with the relevant p-values for the ANOVA and Levene statistical tests. Table entries smaller than [larger than, respectively]* 0.05 *are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the null-hypothesis (stating that the GLPMI means of two learning iterations are equal), may be rejected [may not be rejected, respectively].*

The small Fisher's LSD test p-values in Table 7.14(a)–(c) indicate with 95% confidence that the GLPMI samples are statistically different for all pairs of the three exploration/exploitation methods when the linear, power and logarithmic AHP judgement scales are combined with the LLSM, except between the Basic E&E and PHC methods of exploration/exploitation (the p-values of 0.504, 0.716 and 0.244 are larger than the significance level of 0.05).

	Basic E&E	PHC
Random	4.07×10^{-04}	3.70×10^{-05}
Basic E&E	—	5.04×10^{-01}

(a) Fisher LSD test p-values for the linear judgement scale.

	Basic E&E	PHC
Random	7.03×10^{-05}	1.79×10^{-05}
Basic E&E	—	7.16×10^{-01}

(b) Fisher LSD test p-values for the power judgement scale.

	Basic E&E	PHC
Random	3.16×10^{-02}	1.17×10^{-03}
Basic E&E	—	2.44×10^{-01}

(c) Fisher LSD test p-values for the logarithmic judgement scale.

Table 7.14: *Fisher's LSD post hoc test p-values for the GLPMI means in Table 7.13 resulting from the Random, Basic E&E and PHC exploration/exploitation methods of alternative generation in the case of adopting a ratio-based approach. Table entries smaller than [larger than, respectively]* 0.05 *are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the null-hypothesis (stating that the GLPMI means of two exploration/exploitation methods are equal), may be rejected [may not be rejected, respectively].*

Upon visual inspection of Figure 7.5 it therefore follows, with 95% confidence, that the random method of alternative generation outperforms the other two methods of alternative generation, for the power AHP judgement scale, but not for the cases where the linear or logarithmic AHP judgement scales are employed. Furthermore, it may not be claimed with 95% confidence that one of either the Basic E&E method or the PHC method of alternative generation outperforms the other for any of the three AHP judgement scales. Although it would also seem from Figure 7.5 that the PHC method outperforms the Basic E&E method of alternative generation for the logarithmic AHP judgement scale, this claim cannot be made with 95% confidence.

The above finding for the power AHP judgement scale may be surprising at first glance, given the increase in level of sophistication embodied within the method of alternative generation as one moves from the Random paradigm to the Basic E&E paradigm, and again from the Basic

E&E paradigm to the PHC paradigm. If, however, one takes into account the global nature of the GLPMI (6.3) employed in these experiments, the results are not, in fact, surprising. Recall that the GLPMI rewards learning across the *entire* decision space $\mathcal{D}$. Less exploration (as indeed occurs when one moves from the Random paradigm to the Basic E&E paradigm, and again from the Basic E&E paradigm to the PHC paradigm) is therefore bound to be penalised by the GLPMI. The Basic E&E and PHC methods of alternative generation are, after all, expressly aimed at *exploitation* of *good* regions of the decision space according to value function estimation rather than at *exploration* of the *full* decision space. The reason why the random approach performs worse in this respect, in the case of the linear and logarithmic AHP judgement scales, than the exploration/exploitation approaches, is unclear. The question therefore arises whether there is any learning benefit, as measured according to the LLPMI (6.4), as one moves from the Random paradigm to the Basic E&E paradigm, and again from the Basic E&E paradigm to the PHC paradigm. This question is addressed later in this section.

Adopting a difference-based approach

Figure 7.6 contains box plots of the GLPMI means of the fourth iterations when adopting the difference-based MACBETH approach and the three aforementioned exploration/exploitation methods, namely Random, Basic E&E and PHC.

	GLPMI			ANOVA	Levene
	Random	Basic E&E	PHC	p-value	p-value
MACBETH	0.063	0.068	0.077	1.18×10^{-02}	4.27×10^{-01}

TABLE 7.15: *GLPMI means associated with the fourth iteration of the learning cycle for the Random, Basic E&E and PHC exploration/exploitation methods of alternative generation when populating the DSS framework of §5 with the difference-based approach, together with the relevant p-values for the ANOVA and Levene statistical tests. Table entries smaller than [larger than, respectively]* 0.05 *are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the null-hypothesis (stating that the GLPMI means of two learning iterations are equal), may be rejected [may not be rejected, respectively].*

The GLPMI means are presented for the Random, Basic E&E and PHC exploration/exploitation methods and for the difference-based approach, together with the corresponding ANOVA and Levene test p-values, in Table 7.15. Since the ANOVA test p-values are larger than 0.05, it cannot be claimed with 95% confidence that all three pairs of the exploration/exploitation methods differ statistically in terms of their global ability to learn the DM's underlying value function. The Levene test p-values suggest that the null-hypothesis (stating that the GLPMI samples are homoscedastic across all methods of exploration/exploitation) may not be rejected. This suggests at a 95% level of confidence that the variances of the GLPMI samples are statistically indistinguishable. Fisher's LSD *post hoc* test is therefore performed to determine between which exploration/exploitation methods significant differences in GLPMI mean values occur.

The resulting Fisher's LSD test p-values are provided in Table 7.14. The small p-value for the random *versus* PHC method indicates with 95% confidence that the GLPMI samples are statistically distinguishable in the case of the difference-based approach. The remaining two larger p-values of 0.443 and 0.196 are larger than the significance level of 0.05, and so it cannot be claimed with 95% confidence that the GLPMI values for these pairs of exploration/exploitation methods differ statistically. The difference-based approach therefore seems very robust in terms of which method of alternative generation is adopted.

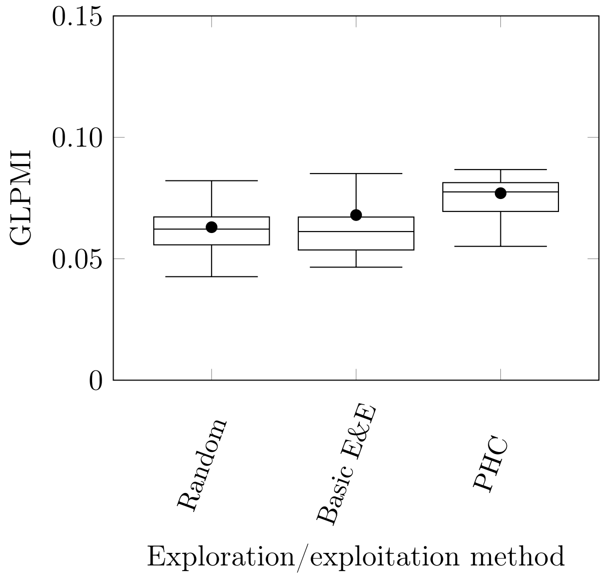


Figure 7.6: *Box plots of the GLPMI values in Table 7.15 associated with the fourth iteration of the learning cycle when populating the DSS framework of §5 with the difference-based MACBETH method. Mean values are indicated by solid dots, while median values are denoted by a central quartile divider.*

	Basic E&E	PHC
Random	4.43×10^{-01}	4.12×10^{-02}
Basic E&E	—	1.96×10^{-01}

Table 7.16: *Fisher's LSD post hoc test p-values for the GLPMI means in Table 7.15 resulting from the Random, Basic E&E and PHC exploration/exploitation methods of alternative generation in the case of adopting a difference-based approach. Table entries smaller than [larger than, respectively]* 0.05 *are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the null-hypothesis (stating that the GLPMI means of two exploration/exploitation methods are equal), may be rejected [may not be rejected, respectively].*

7.4.2 Local learning performance measure analysis

Whereas learning ability was measured in the aforementioned experiments by the quality of the estimated value function *everywhere* in the decision space $\mathcal{D}$ (*i.e.* in terms of GLPMI in (6.3)), learning ability is measured in this section in terms of the LLPMI in (6.4) which is more concerned with the globally best decision alternative than with learning a value function over the entire 2DCSP decision space.

Adopting a ratio-based approach

Figure 7.7 contains box plots of the LLPMI means in (6.4) returned for the best alternative during the fourth learning iterations after winsorising[3] for the linear, power and logarithmic AHP

[3] *Winsorisation* in statistics (also called *clipping* in signal processing) is the transformation of sample data so as to limit the effects of possibly spurious outliers. The process is named after the biostatistician Charles P Winsor (1895–1951). Dixon [54] described various ways in which winsorisation can be performed. The method adopted in this dissertation is closely related to the defining features of a box plot of sample data. In this context an outlier is defined as a data point below the lower cut-off point $\ell = t - \frac{3}{2}(s-t)$ or above the upper cut-off point $u = t + \frac{3}{2}(s-t)$, where t denotes the lower quartile (twenty fifth percentile) and s denotes the upper quartile (seventy fifth percentile). The quartiles t and s are lower and upper bounds on the boxed part of a box plot, respectively, while the cut-off points ℓ and u are lower and upper bounds on the lower and upper whiskers of

judgement scales in conjunction with the LLSM and the three different exploration/exploitation methods, namely Random, Basic E&E and PHC. These best alternatives themselves are provided in Tables 7.19, 7.20 and 7.21 for the linear, power and logarithmic AHP judgement scales respectively, and the corresponding LLPMI means and associated ANOVA and Levene test p-values may be found in Table 7.17.

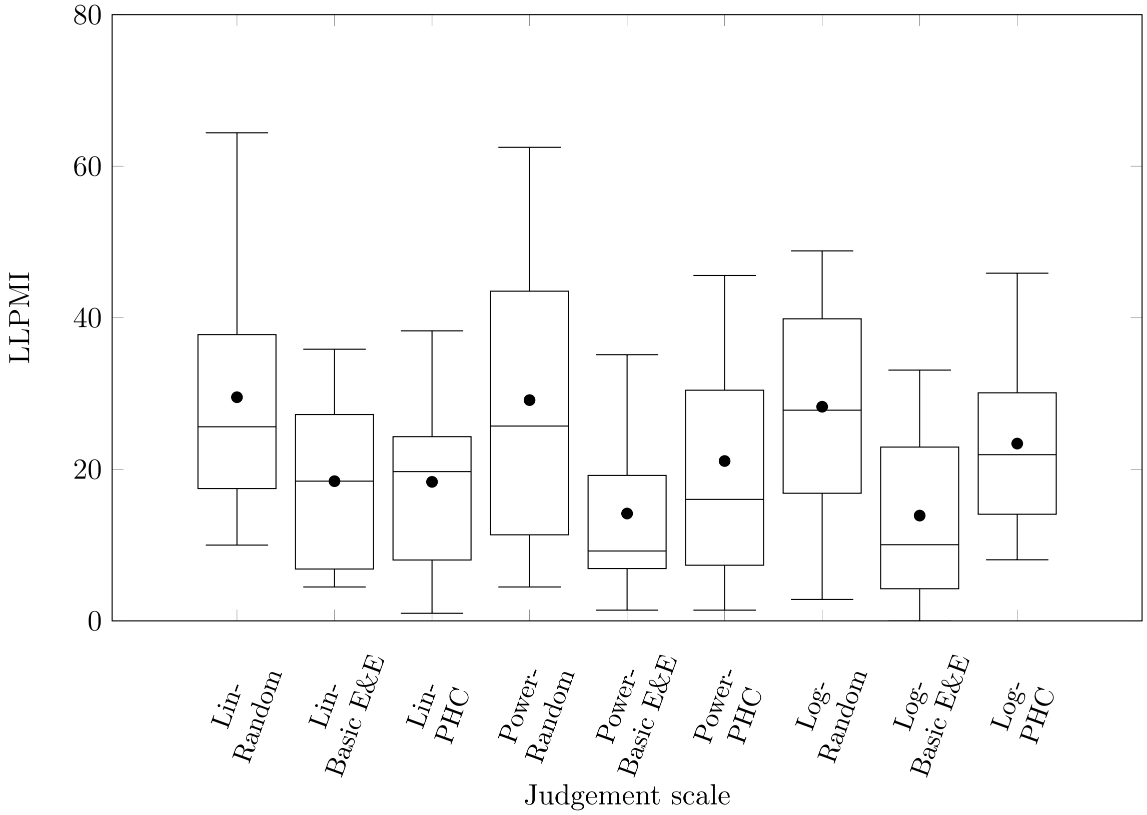


Figure 7.7: *Box plots of the winsorised LLPMI values in Table 7.17 associated with the best alternative uncovered during the fourth iteration of the learning cycle when populating the DSS framework of §5 with the linear, power and logarithmic AHP judgement scales of Table 2.2 in conjunction with the LLSM. Mean values are indicated by solid dots, while median values are denoted by a central quartile divider.*

The best alternatives after the fourth iteration of the learning cycle for each of the thirty instances, according to the LLPMI mean in the case of incorporating the power judgement scale in conjunction with the LLSM for all three methods of exploitation/exploration are presented in Figure 7.8.

Based on the ANOVA test p-values in Table 7.17, the relevant null-hypothesis (stating that the LLPMI means are statistically indistinguishable across the exploration/exploitation methods of alternative generation) may be rejected at a significance level of $\alpha = 5\%$. It furthermore follows from the Levene test p-values in Table 7.17 that the relevant null-hypothesis (stating that the LLPMI samples are homoscedastic) may be also rejected at a 5% level of significance in the case of the power AHP judgement scale, but not in the case of the linear and logarithmic judgement scales. The Games-Howell *post hoc* test is therefore used to determine between which pairs of exploration/exploitation methods of alternative generation differences are statistically detectable

the plot, respectively. Winsorisation is performed in this dissertation by replacing each outlier below the lower cut-off point ℓ with the value ℓ, and replacing each outlier above the upper cut-off point u with the value u. In winsorisation, unlike in trimming, the effects of outliers are therefore dampened or weakened, but still taken into account.

	LLPMI mean			ANOVA	Levene
	Random	Basic E&E	PHC	p-value	p-value
Linear	29.513	18.431	18.344	6.91×10^{-04}	1.29×10^{-01}
Power	29.132	14.165	21.095	1.45×10^{-03}	4.25×10^{-04}
Logarithmic	28.259	13.894	23.395	4.15×10^{-05}	2.93×10^{-01}

Table 7.17: *LLPMI means associated with the fourth iteration of the learning cycle for the Random, Basic E&E and PHC exploitation methods when populating the DSS framework of §5 with the boldfaced AHP judgement scales of Table 7.12 in conjunction with the LLSM, together with the relevant p-values for the ANOVA and Levene statistical tests. Table entries smaller than [larger than, respectively]* 0.05 *are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the null-hypothesis (stating that the LLPMI means of two learning iterations are equal), may be rejected [may not be, respectively].*

	Basic E&E	PHC
Random	9.14×10^{-04}	8.37×10^{-04}
Basic E&E	—	9.78×10^{-01}

(a) Fisher's LSD test p-values for the linear judgement scale.

	Basic E&E	PHC
Random	2.00×10^{-03}	1.85×10^{-01}
Basic E&E	—	1.08×10^{-01}

(b) Games-Howell test p-values for the power judgement scale.

	Basic E&E	PHC
Random	1.03×10^{-05}	1.16×10^{-01}
Basic E&E	—	2.62×10^{-03}

(c) Fisher's LSD test p-values for the logarithmic judgement scale.

Table 7.18: *Games-Howell and Fisher's LSD post hoc p-values for the LLPMI means in Table 7.17 resulting from the Random, Basic E&E and PHC methods of exploration and exploitation. Table entries smaller than [larger than, respectively]* 0.05 *are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the null-hypothesis (stating that the LLPMI means of two learning iterations are equal), may be rejected [may not be rejected, respectively].*

between the LLPMI sample means when employing the power AHP judgement scale, while the Fischer LSD *post hoc* test is used for this purpose in the case of the linear and logarithmic AHP judgement scales.

The Games-Howell post hoc test p-values in Table 7.18(b) indicate at a 5% level of significance that the LLPMI sample means are statistically distinguishable for the pair of the Random and Basic E&E exploration/exploitation method when adopting the power judgement scale, while the opposite is indicated between the remaining pairs. Based on Fisher's LSD post hoc test p-values in Table 7.18(a), the same claim can be made in the case of adopting the linear AHP judgement scale for the difference between the learning performance indicator sample means returned by the Random and Basic E&E methods of alternative aggregation, as well as for the Random and PHC methods of alternative aggregation, but the same cannot be claimed with 95% confidence when considering the Basic E&E and PHC methods of alternative aggregation. The results differ in the case of adopting the logarithmic judgement scale. Based on Fisher's LSD *post hoc* test p-values in Table 7.18(c) a statistical difference is present at a 5% level of significance between all pairs, except between the Random and PHC methods of alternative aggregation when adopting the logarithmic AHP judgement scale. In all cases, however, it is clear (visually and statistically) that there is indeed a learning benefit according to the LLPMI as one moves from the Random paradigm of alternative generation to a paradigm of alternative generation in which there is

	Random		Basic E&E		PHC	
Run	Coordinates	Colour	Coordinates	Colour	Coordinates	Colour
1	$(86, 23)$	■	$(79, 47)$	■	$(63, 71)$	■
2	$(65, 125)$	■	$(69, 63)$	■	$(69, 66)$	■
3	$(62, 52)$	■	$(91, 81)$	■	$(72, 65)$	■
4	$(172, 169)$	■	$(65, 80)$	■	$(63, 62)$	■
5	$(85, 54)$	■	$(75, 49)$	■	$(59, 59)$	■
6	$(98, 90)$	■	$(65, 80)$	■	$(167, 200)$	■
7	$(72, 43)$	■	$(45, 37)$	■	$(70, 56)$	■
8	$(47, 32)$	■	$(42, 58)$	■	$(42, 101)$	■
9	$(93, 43)$	■	$(34, 61)$	■	$(101, 71)$	■
10	$(98, 58)$	■	$(60, 59)$	■	$(64, 65)$	■
11	$(16, 54)$	■	$(56, 61)$	■	$(45, 77)$	■
12	$(53, 106)$	■	$(34, 43)$	■	$(87, 64)$	■
13	$(71, 73)$	■	$(32, 81)$	■	$(67, 63)$	■
14	$(105, 48)$	■	$(59, 55)$	■	$(65, 66)$	■
15	$(39, 120)$	■	$(54, 62)$	■	$(55, 83)$	■
16	$(39, 103)$	■	$(28, 61)$	■	$(85, 52)$	■
17	$(99, 43)$	■	$(43, 43)$	■	$(85, 60)$	■
18	$(52, 56)$	■	$(82, 81)$	■	$(67, 70)$	■
19	$(59, 66)$	■	$(36, 75)$	■	$(81, 71)$	■
20	$(43, 39)$	■	$(60, 67)$	■	$(74, 59)$	■
21	$(66, 43)$	■	$(44, 67)$	■	$(82, 57)$	■
22	$(106, 73)$	■	$(42, 67)$	■	$(45, 31)$	■
23	$(42, 68)$	■	$(50, 73)$	■	$(59, 40)$	■
24	$(75, 94)$	■	$(71, 80)$	■	$(46, 75)$	■
25	$(52, 69)$	■	$(60, 68)$	■	$(96, 51)$	■
26	$(193, 185)$	■	$(67, 68)$	■	$(48, 51)$	■
27	$(61, 33)$	■	$(51, 81)$	■	$(80, 68)$	■
28	$(56, 61)$	■	$(76, 67)$	■	$(31, 84)$	■
29	$(43, 68)$	■	$(64, 49)$	■	$(62, 71)$	■
30	$(77, 54)$	■	$(58, 60)$	■	$(89, 63)$	■

TABLE 7.19: *The best alternatives returned during the fourth iteration for 30 instances of the learning cycle when solving the 2DCSP by means of the linear AHP judgement scale in conjunction with the LLSM. The coordinates of each colour are reported in the form* (g, b)*, where* g *denotes the green value and* b *the blue value of the colour in the RGB colour scheme (recall that the red value of each colour is zero). The DM's true preferred colour is (63,63)* ■.

	Random		Basic E&E		PHC	
Run	Coordinates	Colour	Coordinates	Colour	Coordinates	Colour
1	$(62, 52)$	■	$(69, 63)$	■	$(62, 59)$	■
2	$(69, 38)$	■	$(66, 61)$	■	$(62, 64)$	■
3	$(214, 204)$	■	$(67, 56)$	■	$(68, 63)$	■
4	$(67, 74)$	■	$(73, 52)$	■	$(72, 53)$	■
5	$(47, 56)$	■	$(60, 98)$	■	$(67, 92)$	■
6	$(41, 85)$	■	$(56, 69)$	■	$(104, 69)$	■
7	$(82, 45)$	■	$(64, 64)$	■	$(59, 70)$	■
8	$(71, 55)$	■	$(54, 61)$	■	$(74, 45)$	■
9	$(56, 55)$	■	$(52, 72)$	■	$(64, 70)$	■
10	$(91, 19)$	■	$(209, 188)$	■	$(98, 78)$	■
11	$(95, 92)$	■	$(64, 72)$	■	$(63, 39)$	■
12	$(101, 45)$	■	$(55, 68)$	■	$(57, 70)$	■
13	$(63, 68)$	■	$(49, 57)$	■	$(60, 64)$	■
14	$(43, 84)$	■	$(45, 57)$	■	$(60, 54)$	■
15	$(45, 79)$	■	$(70, 67)$	■	$(197, 199)$	■
16	$(110, 45)$	■	$(74, 79)$	■	$(191, 192)$	■
17	$(52, 60)$	■	$(48, 57)$	■	$(71, 68)$	■
18	$(50, 65)$	■	$(63, 55)$	■	$(60, 58)$	■
19	$(206, 159)$	■	$(58, 69)$	■	$(86, 81)$	■
20	$(73, 74)$	■	$(31, 61)$	■	$(72, 61)$	■
21	$(76, 95)$	■	$(55, 90)$	■	$(66, 69)$	■
22	$(58, 66)$	■	$(76, 66)$	■	$(53, 35)$	■
23	$(201, 212)$	■	$(67, 64)$	■	$(83, 55)$	■
24	$(70, 64)$	■	$(66, 64)$	■	$(97, 88)$	■
25	$(79, 36)$	■	$(191, 180)$	■	$(86, 84)$	■
26	$(54, 11)$	■	$(58, 57)$	■	$(181, 187)$	■
27	$(61, 81)$	■	$(61, 62)$	■	$(105, 77)$	■
28	$(118, 43)$	■	$(31, 70)$	■	$(79, 64)$	■
29	$(43, 102)$	■	$(64, 61)$	■	$(70, 60)$	■
30	$(61, 67)$	■	$(23, 67)$	■	$(48, 87)$	■

TABLE 7.20: *The best alternatives returned during the fourth iteration for 30 instances of the learning cycle when solving the 2DCSP by means of the power AHP judgement scale in conjunction with the LLSM. The coordinates of each colour are reported in the form* (g, b)*, where* g *denotes the green value and* b *the blue value of the colour in the RGB colour scheme (recall that the red value of each colour is zero). The DM's true preferred colour is (63,63)* ■.

	Random		Basic E&E		PHC	
Run	Coordinates	Colour	Coordinates	Colour	Coordinates	Colour
1	$(28, 69)$	■	$(64, 64)$	■	$(73, 73)$	■
2	$(74, 33)$	■	$(71, 77)$	■	$(190, 190)$	■
3	$(111, 66)$	■	$(70, 54)$	■	$(76, 85)$	■
4	$(27, 85)$	■	$(17, 83)$	■	$(49, 53)$	■
5	$(88, 79)$	■	$(65, 57)$	■	$(69, 70)$	■
6	$(43, 58)$	■	$(71, 62)$	■	$(78, 79)$	■
7	$(88, 81)$	■	$(64, 60)$	■	$(52, 87)$	■
8	$(52, 72)$	■	$(36, 59)$	■	$(89, 86)$	■
9	$(46, 41)$	■	$(54, 64)$	■	$(112, 44)$	■
10	$(61, 86)$	■	$(63, 58)$	■	$(71, 71)$	■
11	$(74, 86)$	■	$(72, 84)$	■	$(50, 66)$	■
12	$(107, 67)$	■	$(57, 63)$	■	$(56, 59)$	■
13	$(84, 102)$	■	$(57, 65)$	■	$(81, 75)$	■
14	$(46, 39)$	■	$(47, 82)$	■	$(107, 50)$	■
15	$(108, 81)$	■	$(66, 64)$	■	$(71, 45)$	■
16	$(63, 75)$	■	$(37, 78)$	■	$(62, 72)$	■
17	$(52, 56)$	■	$(63, 63)$	■	$(104, 60)$	■
18	$(63, 94)$	■	$(32, 69)$	■	$(79, 58)$	■
19	$(23, 97)$	■	$(51, 58)$	■	$(85, 55)$	■
20	$(81, 68)$	■	$(66, 66)$	■	$(46, 45)$	■
21	$(66, 63)$	■	$(46, 51)$	■	$(180, 194)$	■
22	$(45, 55)$	■	$(54, 39)$	■	$(55, 48)$	■
23	$(91, 88)$	■	$(62, 59)$	■	$(64, 78)$	■
24	$(79, 46)$	■	$(73, 62)$	■	$(61, 39)$	■
25	$(74, 20)$	■	$(59, 64)$	■	$(39, 48)$	■
26	$(72, 65)$	■	$(60, 60)$	■	$(90, 80)$	■
27	$(65, 61)$	■	$(50, 58)$	■	$(42, 87)$	■
28	$(43, 45)$	■	$(192, 195)$	■	$(77, 63)$	■
29	$(72, 75)$	■	$(52, 73)$	■	$(72, 64)$	■
30	$(21, 43)$	■	$(64, 40)$	■	$(40, 56)$	■

Table 7.21: *The best alternatives returned during the fourth iteration for 30 instances of the learning cycle when solving the 2DCSP by means of the logarithmic AHP judgement scale in conjunction with the LLSM. The coordinates of each colour are reported in the form* (g, b)*, where* g *denotes the green value and* b *the blue value of the colour in the RGB colour scheme (recall that the red value of each colour is zero). The DM's true preferred colour is (63,63)* ■.

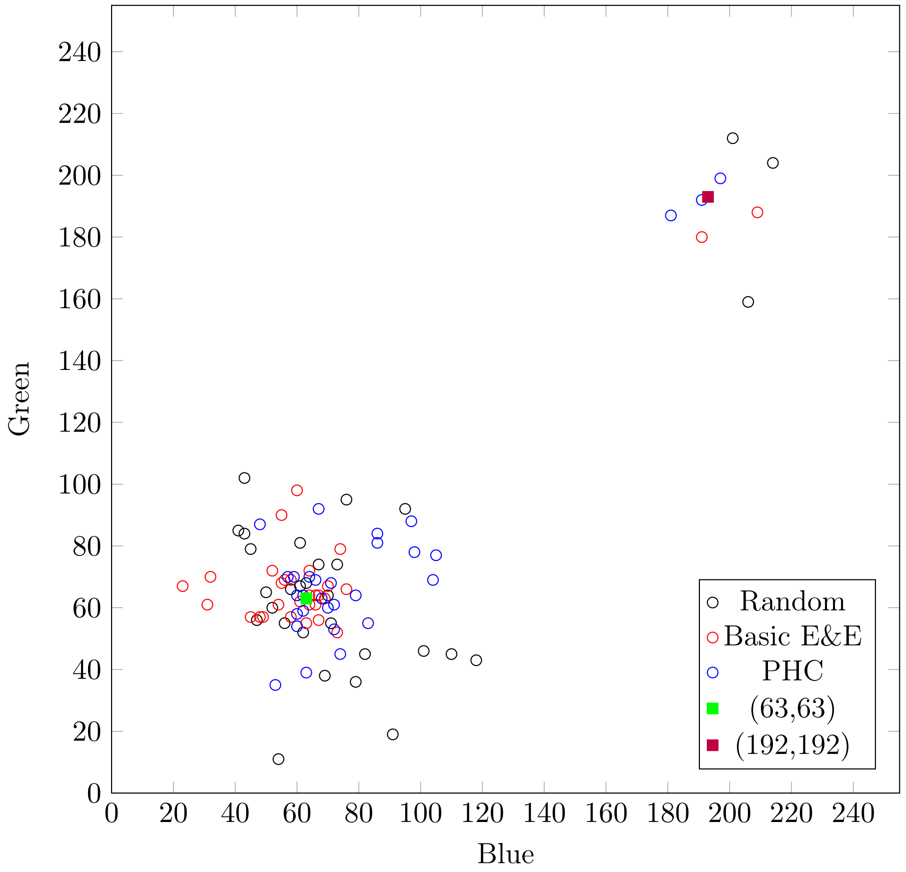


Figure 7.8: *Scatter plot of the best alternatives after four iterations of the learning cycle for the Random, Basic E&E and PHC approach when the power AHP judgement scale is incorporated in conjunction with the LLSM thirty times. The globally optimal point (63,63) and locally optimal point (192,192) are also presented.*

some form of exploitation of promising areas of the decision space according to the current value function estimate. The preferred method of exploitation when generating new alternatives is unfortunately not resolved by the aforementioned statistical analysis.

Adopting a difference-based approach

Figure 7.9 contains box plots of the LLPMI means during (6.4) returned for the best alternative during the fourth learning iterations after winsorising for the difference-based approach and the three different exploration/exploitation methods, namely Random, Basic E&E and PHC. These best alternatives themselves are provided in Table 7.22, and the corresponding LLPMI means and associated ANOVA and Levene test p-values may be found in Table 7.23.

The best alternatives after the fourth iteration of the learning cycle for each of the thirty instances, according to the LLPMI mean in the case of incorporating the difference-based MACBETH method for all three methods of exploitation/exploration are presented in Figure 7.10.

Based on the ANOVA test p-values in Table 7.23, the relevant null-hypothesis (stating that the LLPMI means are statistically indistinguishable across the exploration/exploitation methods of alternative generation) may be rejected at a significance level of $\alpha = 5\%$. It furthermore follows from the Levene test p-values in Table 7.23 that the relevant null-hypothesis (stating

	Random		Basic E&E		PHC	
Run	Coordinates	Colour	Coordinates	Colour	Coordinates	Colour
1	$(86, 101)$	■	$(62, 63)$	■	$(79, 47)$	■
2	$(30, 77)$	■	$(59, 71)$	■	$(59, 62)$	■
3	$(55, 81)$	■	$(56, 66)$	■	$(75, 75)$	■
4	$(73, 72)$	■	$(80, 54)$	■	$(71, 55)$	■
5	$(51, 65)$	■	$(61, 74)$	■	$(92, 40)$	■
6	$(47, 85)$	■	$(77, 62)$	■	$(71, 81)$	■
7	$(55, 91)$	■	$(60, 66)$	■	$(65, 65)$	■
8	$(81, 68)$	■	$(67, 61)$	■	$(83, 59)$	■
9	$(54, 65)$	■	$(55, 64)$	■	$(64, 52)$	■
10	$(60, 87)$	■	$(65, 58)$	■	$(52, 80)$	■
11	$(64, 58)$	■	$(60, 58)$	■	$(60, 67)$	■
12	$(65, 48)$	■	$(59, 31)$	■	$(84, 45)$	■
13	$(54, 25)$	■	$(66, 53)$	■	$(100, 63)$	■
14	$(64, 58)$	■	$(86, 62)$	■	$(92, 42)$	■
15	$(42, 60)$	■	$(53, 71)$	■	$(72, 57)$	■
16	$(83, 29)$	■	$(55, 63)$	■	$(75, 71)$	■
17	$(80, 52)$	■	$(78, 60)$	■	$(70, 72)$	■
18	$(56, 53)$	■	$(67, 77)$	■	$(76, 54)$	■
19	$(65, 81)$	■	$(63, 51)$	■	$(32, 79)$	■
20	$(41, 57)$	■	$(76, 78)$	■	$(70, 57)$	■
21	$(60, 41)$	■	$(59, 59)$	■	$(62, 58)$	■
22	$(45, 66)$	■	$(62, 63)$	■	$(90, 75)$	■
23	$(52, 57)$	■	$(69, 63)$	■	$(77, 71)$	■
24	$(78, 67)$	■	$(62, 56)$	■	$(69, 70)$	■
25	$(69, 57)$	■	$(60, 67)$	■	$(85, 55)$	■
26	$(66, 84)$	■	$(79, 68)$	■	$(71, 72)$	■
27	$(63, 58)$	■	$(69, 67)$	■	$(69, 63)$	■
28	$(88, 69)$	■	$(62, 65)$	■	$(85, 55)$	■
29	$(72, 68)$	■	$(55, 59)$	■	$(61, 53)$	■
30	$(66, 53)$	■	$(61, 72)$	■	$(83, 63)$	■

Table 7.22: *The best alternatives returned during the fourth iteration for 30 instances of the learning cycle when solving the 2DCSP by means of the difference-based MACBETH approach. The coordinates of each colour are reported in the form* (g, b)*, where* g *denotes the green value and* b *the blue value of the colour in the RGB colour scheme (recall that the red value of each colour is zero). The DM's true preferred colour is (63,63)* ■.

	LLPMI mean			ANOVA	Levene
	Random	Basic E&E	PHC	p-value	p-value
MACBETH	18.403	9.784	16.283	1.44×10^{-04}	3.09×10^{-02}

Table 7.23: *LLPMI means associated with the fourth iteration of the learning cycle for the Random, Basic E&E and PHC exploitation methods when populating the DSS framework of §5 with the difference-based approach, together with the relevant p-values for the ANOVA and Levene statistical tests. Table entries smaller than [larger than, respectively]* 0.05 *are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the null-hypothesis (stating that the LLPMI means of two learning iterations are equal), may be rejected [may not be, respectively].*

	Basic E&E	PHC
Random	1.00×10^{-03}	6.16×10^{-01}
Basic E&E	—	4.00×10^{-03}

Table 7.24: *Games-Howell post hoc p-values for the LLPMI means in Table 7.23 resulting from the Random, Basic E&E and PHC methods of exploration and exploitation. Table entries smaller than [larger than, respectively]* 0.05 *are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the null-hypothesis (stating that the LLPMI means of two learning iterations are equal), may be rejected [may not be rejected, respectively].*

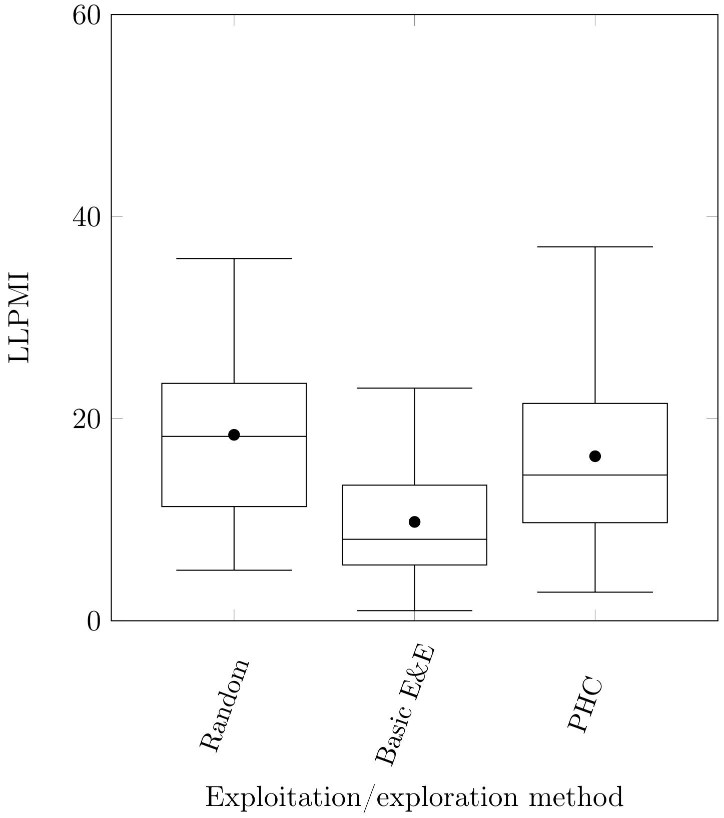


Figure 7.9: *Box plots of the winsorised LLPMI values in Table 7.23 associated with the best alternative uncovered during the fourth iteration of the learning cycle when populating the DSS framework of §5 with the difference-based MACBETH approach. Mean values are indicated by solid dots, while median values are denoted by a central quartile divider.*

that the LLPMI samples are homoscedastic) may be also rejected at a 5% level of significance. The Games-Howell *post hoc* test is therefore used to determine between which pairs of exploration/exploitation methods of alternative generation differences are statistically detectable between the LLPMI sample means when employing the difference-based method.

The Games-Howell post hoc test p-values in Table 7.16 confirm the visual conclusion from Figure 7.9, namely that the LLPMI sample means are statistically distinguishable at a 5% level of significance for the Random and Basic E&E and for the PHC and Basic E&E exploration/exploitation method pairs when adopting the difference-based approach.

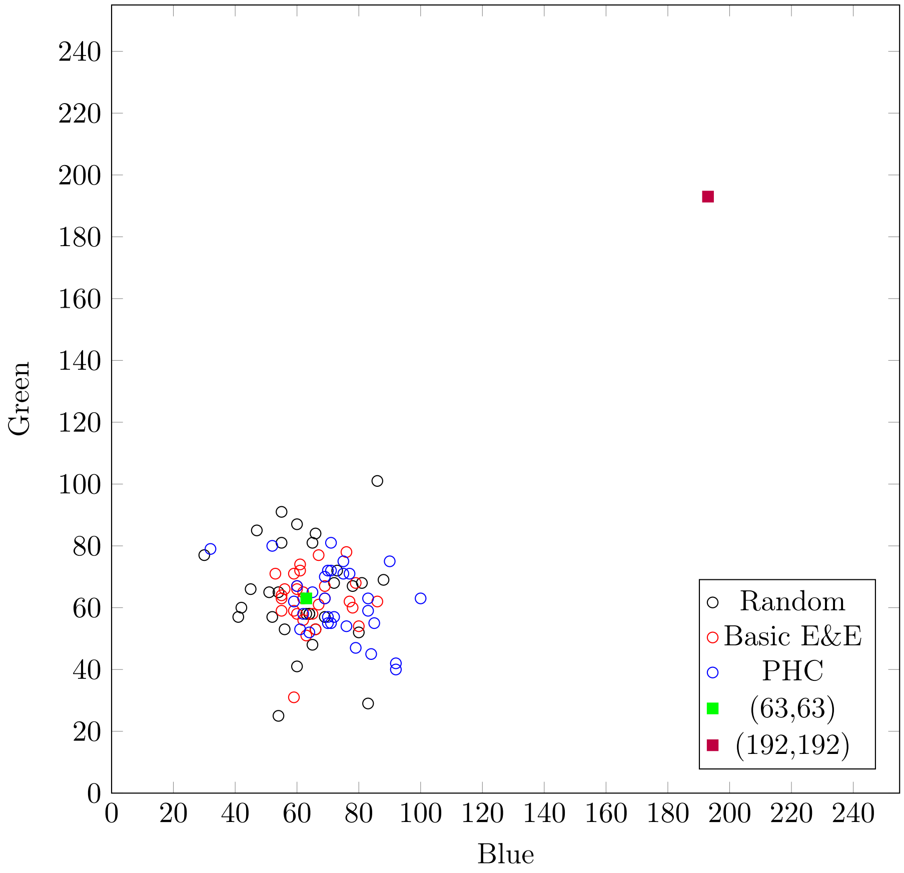


FIGURE 7.10: *Scatter plot of the best alternatives after four iterations of the learning cycle for the Random, Basic E&E and PHC approach when the difference-based MACBETH method is incorporated thirty times. The globally optimal point (63,63) and locally optimal point (192,192) are also presented.*

7.4.3 Discussion

Based on the statistical analysis of §7.4.1 for the ratio-based AHP and the difference-based MACBETH method, it follows from a visual inspection of Figure 7.5 that there is no definitive distinction between the Basic E&E and PHC method a 95% level of confidence when solving the 2DCSP in the case of the ratio-based approach, while for the difference-based approach in Figure 7.6, there is no distinction between the Random and Basic E&E methods if the aim is to explore the decision space and focus on learning of the DM's true value function over the entire decision space. This ranking is shown in Table 7.25.

	AHP			MACBETH
	Linear scale	Power scale	Logarithmic scale	
Most preferred	Basic E&E or PHC	Random	Basic E&E or PHC	Random or Basic E&E
Second-most preferred	Random	Basic E&E or PHC	Random	PHC

Table 7.25: *Preferred ranking at a 95% confidence level of the methods of alternative generation when pursuing exploration or global learning of the DM's value function in the context of the 2DCSP.*

Based on the statistical analysis of §7.4.2 for the ratio-based AHP and the difference-based MACBETH method, however, it follows from a visual inspection of Figures 7.7 (for the AHP) and 7.9 (for the MACBETH method) that there is a definitive ranking of the three methods of alternative generation at a 95% level of confidence when solving the 2DCSP if the aim is to exploit promising areas of the decision space in the hope of discovering local maxima of the DM's true value function. This ranking is shown in Table 7.26.

	AHP			MACBETH
	Linear scale	Power scale	Logarithmic scale	
Most preferred	Basic E&E or PHC	Basic E&E or PHC	Basic E&E	Basic E&E
Second-most preferred	Random	Random	PHC or Random	PHC or Random

Table 7.26: *Preferred ranking at a 95% confidence level of the methods of alternative generation when pursuing exploitation of promising areas of the decision space or local learning of the DM's value function.*

7.5 Parametric sensitivity analysis

The sensitivity of the DSS proposed in §5 with respect to the various parameters incorporated in the system is analysed in this section in the form of a classical parametric sensitivity analysis. The sensitivity analysis is conducted for both the ratio-based approach (AHP) in §7.5.1 and for the difference-based approach (MACBETH) in §7.5.2, and measured according to both the GLPMI and the LLPMI described in §6.1.2.

The following nine parameters are considered in the sensitivity analysis: The size of the comparison set during each iteration of the learning cycle, the number of learning cycles completed, the update weight parameter ζ, the exploitation radius of the Basic E&E method described in §7.4, the exploitation rate of the same method (*i.e.* the rate at which the exploitation radii decrease exponentially as a function of learning iterations), the exploitation size during each iteration (*i.e.* the proportion of new alternatives generated within the exploitation radius from the global maximum of the value function estimate), the inner and outer radii of the PHC exploitation/exploration method, and the so-called exploitation probability of this method (the proportion of newly generated alternatives that are local maxima within these annuli).

The parameter values for the base case of the sensitivity analysis are shown in boldface in Table 7.27. Three possible values are considered for each of the aforementioned nine parameters, namely a small value, a medium value (corresponding to the base case, except in the case of the update weight parameter), and a large value. These parameter values are also shown in Table 7.27. It follows that there are nineteen cases to consider in the sensitivity analysis.

No	Parameter	Small	**Medium**	Large
1	Comparison set size per iteration of the learning cycle	8	**12**	16
2	Number of iterations carried out per learning cycle	2	**4**	6
3	Update weight parameter	0.50	0.75	**1.00**
4	Exploitation radius (of the *Basic E&E* method)	90	**120**	150
5	Exploitation rate (of exponential decrease)	0.05	**0.15**	0.25
6	Exploitation set size (a proportion of the comparison set size)	0.3	**0.5**	0.7
7	Inner radius (of neighbourhood annuli in the *PHC* method)	4	**8**	12
8	Outer radius (of neighbourhood annuli in the *PHC* method)	20	**35**	50
9	Exploitation-probability (of selecting an annulus local maximum)	0.1	**0.2**	0.3

TABLE 7.27: *Parameter values considered in the parametric sensitivity analysis. Values typeset in boldface represent the base case.*

7.5.1 A ratio-based approach

The AHP judgement scale identified in §7.3 to be the most effective (*i.e.* the power scale) and the LLSM are employed throughout this analysis.

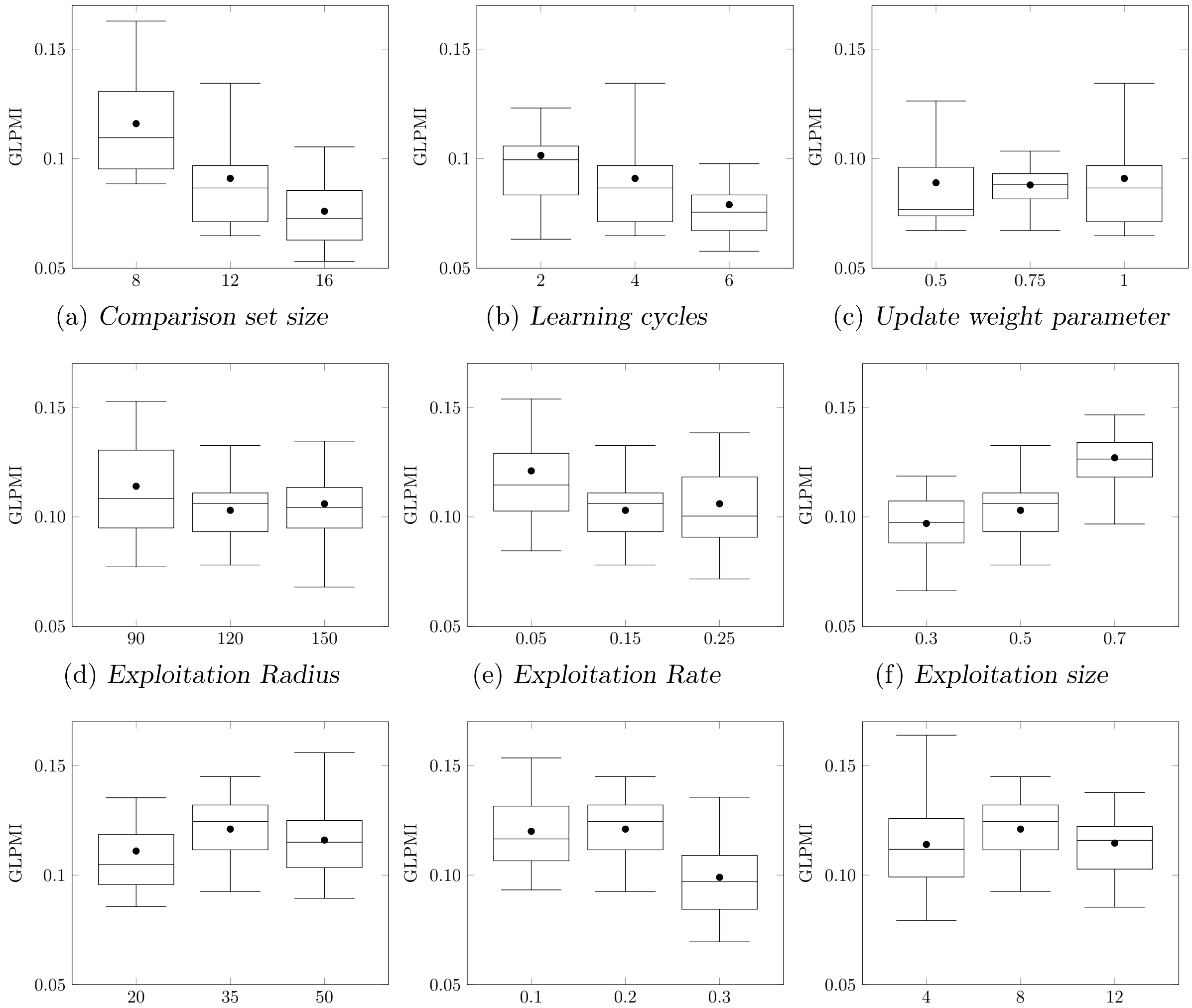
Thirty replications of the learning process are again carried out in each of the nineteen cases mentioned above, and each time the GLPMI and best LLPMI values are recorded. Box plots of the samples of GLPMI values are shown in Figure 7.11. It follows from the ANOVA p-values in Table 7.29 that the null-hypothesis stating that *the three GLPMI sample means are indistinguishable* may be rejected at a 5% level of significance in all of the cases, except for the update weight parameter, outer radius and inner radius. It may therefore be claimed with 95% confidence that the GLPMI is insensitive to the value of the update weight parameter ζ and the inner and outer radius within which the annulus local maximum is selected as new alternative in the PHC exploitation/exploration method. The GLPMI depends sensitively on the other seven parameters in Table 7.27.

The Levene test p-values in Table 7.29 furthermore reveal that the null-hypothesis stating that *the GLPMI samples are homoscedastic* may only be rejected at a 5% level of significance in the case of the exploitation radius. This means that the Games-Howell *post hoc* test should be used to detect between which samples the GLPMI means differ in the case of the exploitation radius parameter, while Fisher's LSD *post hoc* test may be used for this purpose in all other cases.

No	Parameter	Sensitive?	Best value
1	Comparison set size	Yes	Large (16)
2	Number of iterations	Yes	Large (6)
3	Update weight parameter	No	—
4	Exploitation radius	No	—
5	Exploitation rate	Yes	Large (0.25)
6	Exploitation set size	Yes	Small (0.3)
7	Inner radius	No	—
8	Outer radius	Yes	Medium (35)
9	Exploitation probability	No	—

TABLE 7.28: *Results of parametric sensitivity analysis in respect of the GLPMI mean.*

These *post hoc* test p-values may be found in Table 7.31. It follows from the results in Table 7.31(a) that the comparison set size may be ranked with 95% confidence in statistically significantly decreasing order of GLPMI mean quality as: First large, then medium and finally small.



(a) *Comparison set size*

(b) *Learning cycles*

(c) *Update weight parameter*

(d) *Exploitation Radius*

(e) *Exploitation Rate*

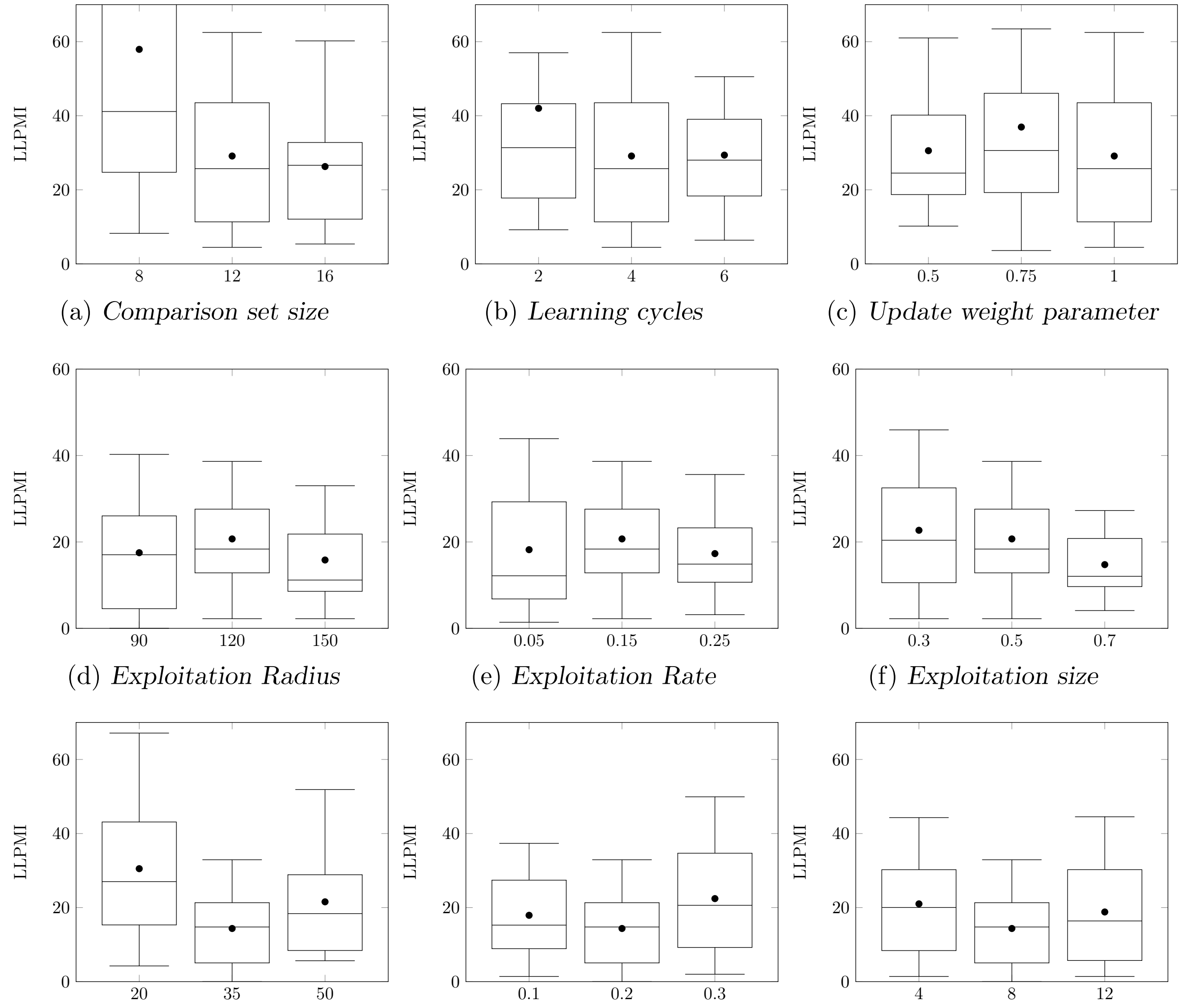
(f) *Exploitation size*

(g) *Outer Radius*

(h) *Exploitation probability*

(i) *Inner Radius*

FIGURE 7.11: *Parametric sensitivity analysis results for each of the nine parameters in Table 7.27 in respect of the GLPMI in (6.3). The middle box represents the base case in each plot (except for the update weight parameter where 1 is the base case). Mean values are denoted by solid dots, while median values are represented by central quartile lines.*



(a) *Comparison set size* (b) *Learning cycles* (c) *Update weight parameter*

(d) *Exploitation Radius* (e) *Exploitation Rate* (f) *Exploitation size*

(g) *Outer Radius* (h) *Exploitation probability* (i) *Inner Radius*

FIGURE 7.12: *Parametric sensitivity analysis results for each of the nine parameters in Table 7.27 in respect of the LLPMI in (6.4). The middle box represents the base case in each plot (except for the update weight parameter where 1 is the base case). Mean values are denoted by solid dots, while median values are represented by central quartile lines.*

Moreover, it follows with 95% confidence from Table 7.31(b) and (f) that the large values of both the number of learning iterations and exploitation probability parameters lead to a statistically superior GLPMI mean when compared to adopting the small or medium values of this parameter (which themselves lead to statistically indistinguishable GLPMI means). The same conclusion may be drawn for the exploitation size parameter in Figure 7.31(e), but in this case the small and medium values are statistically indistinguishable and both lead to a statistically superior GLPMI mean compared to the larger value. It furthermore follows with 95% confidence from Table 7.32(d) that the medium and large values of the exploitation rate parameter leads to a statistically better (indistinguishable, respectively) GLPMI mean than when adopting the small (large, respectively) value of this parameter. Table 7.32(c) finally reveals that the exploitation radius values lead to a statistically indistinguishable GLPMI mean. The conclusions of the sensitivity analysis in terms of the GLPMI are summarised in Table 7.28.

	GLPMI			ANOVA	Levene
Parameter	Small	Medium	Large	test p-values	test p-values
Comparison set size	0.116	0.091	0.076	6.40×10^{-09}	3.03×10^{-01}
Learning cycles	0.101	0.091	0.079	1.40×10^{-03}	3.20×10^{-01}
Update weight parameter	0.089	0.088	0.091	8.88×10^{-01}	7.12×10^{-02}
Exploitation radius	0.114	0.103	0.106	4.16×10^{-02}	5.82×10^{-03}
Exploitation rate	0.121	0.103	0.106	1.25×10^{-02}	1.18×10^{-01}
Exploitation size	0.097	0.103	0.127	2.78×10^{-12}	8.15×10^{-01}
Outer radius	0.111	0.121	0.116	1.47×10^{-01}	5.68×10^{-01}
Exploitation probability	0.120	0.121	0.099	1.20×10^{-06}	9.69×10^{-01}
Inner radius	0.114	0.121	0.114	2.20×10^{-01}	3.06×10^{-01}

Table 7.29: *Parametric sensitivity analysis GLPMI means returned by the DSS framework of §5 populated with the power AHP judgement scale and the LLSM. The corresponding p-values returned for the ANOVA/Levene tests are also included. Table entries smaller than [larger than, respectively]* 0.05 *are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the null-hypothesis (stating that the GLPMI means/variances of all pairs of learning iterations are equal), may be [may not be, respectively] rejected.*

Box plots of the samples of best LLPMI values are shown in Figure 7.12. This time it follows from the ANOVA p-values in Table 7.30 that the null-hypothesis stating that *the three best LLPMI sample means are indistinguishable* may be rejected at a 5% level of significance only in the cases of the comparison set size, the exploitation size, and the outer radius and exploitation probability parameters of the PHC method. It may therefore be claimed with 95% confidence that the LLPMI is insensitive to the values of all the remaining parameters in Table 7.27. The LLPMI, however, depends sensitively on the parameters mentioned above.

The Levene test p-values in Table 7.30 furthermore reveal that the null-hypothesis stating that *the best LLPMI samples are homoscedastic* may be rejected at a 5% level of significance in the cases of the comparison set radius, -rate and -size, the exploitation size, and the outer radius and exploitation probability parameters of the PHC method parameters. This means that the Games-Howell *post hoc* test may be used to detect between which samples the best LLPMI means differ in the cases of the comparison set size, the exploitation size, the outer radius and the exploitation probability parameters.

These *post hoc* test p-values may be found in Table 7.32. It follows with 95% confidence from the p-values in Table 7.32(a) that the large or medium values of the comparison set size parameter (which are statistically indistinguishable) leads to a statistically better best LLPMI mean than when adopting the small value of this parameter. It similarly follows with 95% confidence from Table 7.32(b) that the large value of the exploitation size parameter leads to a statistically better best LLPMI mean value than when the medium or small value of this parameter is employed.

	LLPMI			ANOVA	Levene
Parameter	Small	Medium	Large	test p-values	test p-values
Comparison set size	57.92	29.13	26.28	1.39×10^{-04}	3.06×10^{-05}
Learning cycles	42.01	29.13	29.37	1.40×10^{-01}	8.75×10^{-02}
Update weight parameter	30.57	36.95	29.13	3.02×10^{-01}	2.82×10^{-01}
Exploitation radius	17.51	20.70	15.82	2.47×10^{-01}	3.60×10^{-02}
Exploitation rate	18.20	20.70	17.30	4.82×10^{-01}	4.08×10^{-02}
Exploitation size	22.69	20.70	14.75	1.81×10^{-02}	1.82×10^{-03}
Outer radius	30.51	14.36	20.56	1.55×10^{-04}	5.09×10^{-03}
Exploitation probability	17.92	14.36	22.43	4.91×10^{-02}	8.92×10^{-03}
Inner radius	21.01	14.36	18.82	1.12×10^{-01}	1.51×10^{-01}

TABLE 7.30: *Parametric sensitivity analysis LLPMI means returned by the DSS framework of §5 populated with the power judgement scale and the LLSM. The corresponding p-values returned for the ANOVA/Levene tests are also included. Table entries smaller than [larger than, respectively]* 0.05 *are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the null-hypothesis (stating that the LLPMI means/variances of all pairs of learning iterations are equal), may be [may not be, respectively] rejected.*

	Medium (12)	Large (16)
Small (8)	5.59×10^{-05}	1.23×10^{-09}
Medium (12)	—	1.20×10^{-02}

(a) Fisher LSD test p-values for comparison set size.

	Medium (4)	Large (6)
Small(2)	8.96×10^{-02}	3.06×10^{-04}
Medium (4)	—	4.39×10^{-02}

(b) Fisher LSD test p-values for the number of iterations.

	Medium (120)	Large (150)
Small (90)	5.10×10^{-02}	2.12×10^{-01}
Medium (120)	—	7.74×10^{-01}

(c) Games-Howell test p-values for the exploitation radius.

	Medium (0.15)	Large (0.25)
Small (0.05)	6.05×10^{-03}	1.88×10^{-02}
Medium (0.15)	—	6.76×10^{-01}

(d) Fisher LSD test p-values for the exploitation rate.

	Medium (0.5)	Large (0.7)
Small (0.3)	1.10×10^{-01}	3.19×10^{-12}
Medium (0.5)	—	5.34×10^{-09}

(e) Fisher LSD test p-values for the exploitation subset size.

	Medium (8)	Large (12)
Small (4)	7.75×10^{-01}	7.94×10^{-06}
Medium (8)	—	2.54×10^{-06}

(f) Fisher LSD test p-values for the exploitation probability.

TABLE 7.31: *Fisher's LSD and Games-Howell test p-values for the GLPMI means in Table 7.29. Table entries smaller than [larger than, respectively]* 0.05 *are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the null-hypothesis (stating that the GLPMI means of two learning iterations are equal), may be rejected [may not be rejected, respectively].*

	Medium (120)	Large (150)
Small (90)	1.00×10^{-02}	3.00×10^{-03}
Medium (120)	—	7.97×10^{-01}

(a) Games-Howell test p-values for the exploitation radius.

	Medium (0.15)	Large (0.25)
Small (0.05)	8.11×10^{-01}	2.10×10^{-02}
Medium (0.25)	—	4.60×10^{-02}

(b) Games-Howell test p-values for the exploitation set size.

	Medium (120)	Large (150)
Small (90)	1.00×10^{-04}	8.20×10^{-02}
Medium (120)	—	6.10×10^{-02}

(c) Games-Howell test p-values for the outer radius.

	Medium (0.15)	Large (0.25)
Small (90)	3.95×10^{-01}	4.08×10^{-01}
Medium (120)	—	5.90×10^{-02}

(d) Games-Howell test p-values for the exploitation probability.

Table 7.32: *Games-Howell test p-values for the LLPMI means in Table 7.30. Table entries smaller than [larger than, respectively]* 0.05 *are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the null-hypothesis (stating that the LLPMI means of two learning iterations are equal), may be rejected [may not be rejected, respectively].*

No	Parameter	Sensitive?	Best value
1	Comparison set size	Yes	Large (16)
2	Number of iterations	No	—
3	Update weight parameter	No	—
4	Exploitation radius	No	—
5	Exploitation rate	Yes	Small (0.05)
6	Exploitation set size	Yes	Large (0.7)
7	Inner radius	No	—
8	Outer radius	No	—
9	Exploitation probability	No	—

Table 7.33: *Results of parametric sensitivity analysis in respect of the best LLPMI mean.*

The *post hoc* values in Figures 7.32(c) and (d) reveal that there seems to be no improvement in the best LLPMI mean for any of the parameter values, although it does appear from a visual inspection of Figure 7.12 that the medium value performs better than the smaller and larger values in both cases. The conclusions of the sensitivity analysis in terms of the best LLPMI are summarised in Table 7.33.

7.5.2 A difference-based approach

The single judgement scale of the MACBETH method is considered throughout this section in which a similar analysis to that of the previous section is performed.

Thirty replications of the learning process are again carried out in each of these nineteen cases of Table 7.27, again recorded for the GLPMI and best LLPMI values. Box plots of the samples of GLPMI values are shown in Figure 7.13. It follows from the ANOVA p-values in Table 7.34 that the null-hypothesis stating that *the three GLPMI sample means are indistinguishable* may not be rejected at a 5% level of significance for any of the cases, except for the Comparison set size parameter and the number of learning cycles. It may therefore be claimed with 95% confidence that the GLPMI is insensitive to any of the exploration/exploitation method parameters except for the comparison set size and number of learning cycles.

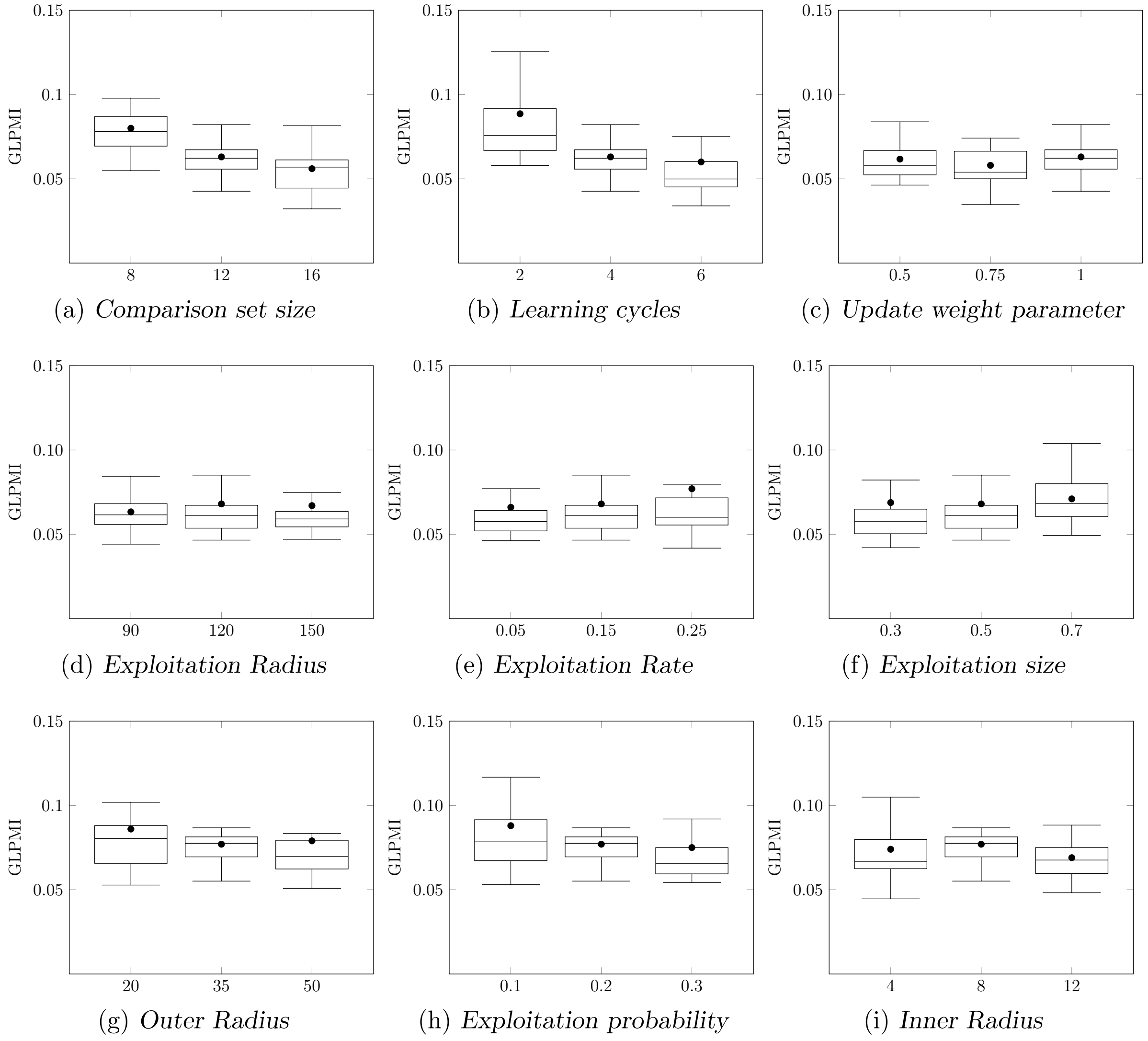
The Levene test p-values in Table 7.34 furthermore reveal that the null-hypothesis stating that *the GLPMI samples are homoscedastic* may not be rejected at a 5% level of significance in any of the parameter cases. This means that the Fisher LSD *post hoc* test should be used to detect between which samples the GLPMI means differ in the case of the exploitation radius parameter.

The *post hoc* test p-values for the comparison set size and number of learning cycles may be found in Table 7.35. It follows from the results in Table 7.31(a) and (b) that for the comparison set size and number of iterations it may be claimed with 95% confidence that a larger parameter value causes the GLPMI mean quality to perform statistically better in both cases, while the small and medium parameter values are determined to yield statistically indistinguishable results. The conclusions of the sensitivity analysis in terms of the GLPMI are summarised in Table 7.36.

		GLPMI		ANOVA	Levene
Parameter	Small	Medium	Large	test p-values	test p-values
Comparison set size	0.080	0.063	0.058	2.93×10^{-08}	6.85×10^{-01}
Learning cycles	0.088	0.063	0.060	9.65×10^{-04}	6.56×10^{-02}
Update weight parameter	0.061	0.058	0.063	3.03×10^{-01}	9.68×10^{-01}
Exploitation radius	0.063	0.068	0.067	8.41×10^{-01}	5.24×10^{-01}
Exploitation rate	0.066	0.068	0.077	6.38×10^{-01}	3.96×10^{-01}
Exploitation size	0.068	0.068	0.071	9.31×10^{-01}	3.90×10^{-01}
Outer radius	0.086	0.077	0.079	4.99×10^{-01}	1.59×10^{-01}
Exploitation probability	0.088	0.077	0.075	2.25×10^{-01}	1.66×10^{-01}
Inner radius	0.074	0.077	0.069	1.35×10^{-01}	7.76×10^{-02}

Table 7.34: *Parametric sensitivity analysis GLPMI means returned by the DSS framework of §5 populated with the difference-based approach. The corresponding p-values returned for the ANOVA/Levene tests are also included. Table entries smaller than [larger than, respectively]* 0.05 *are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the null-hypothesis (stating that the GLPMI means/variances of all pairs of learning iterations are equal), may be [may not be, respectively] rejected.*

Box plots of the samples of best LLPMI values are shown in Figure 7.14. Similar to the ANOVA results of the GLPMI, it follows from the ANOVA p-values in Table 7.37 that the null-hypothesis stating that *the three best LLPMI sample means are indistinguishable* may be rejected at a 5%



(a) *Comparison set size*

(b) *Learning cycles*

(c) *Update weight parameter*

(d) *Exploitation Radius*

(e) *Exploitation Rate*

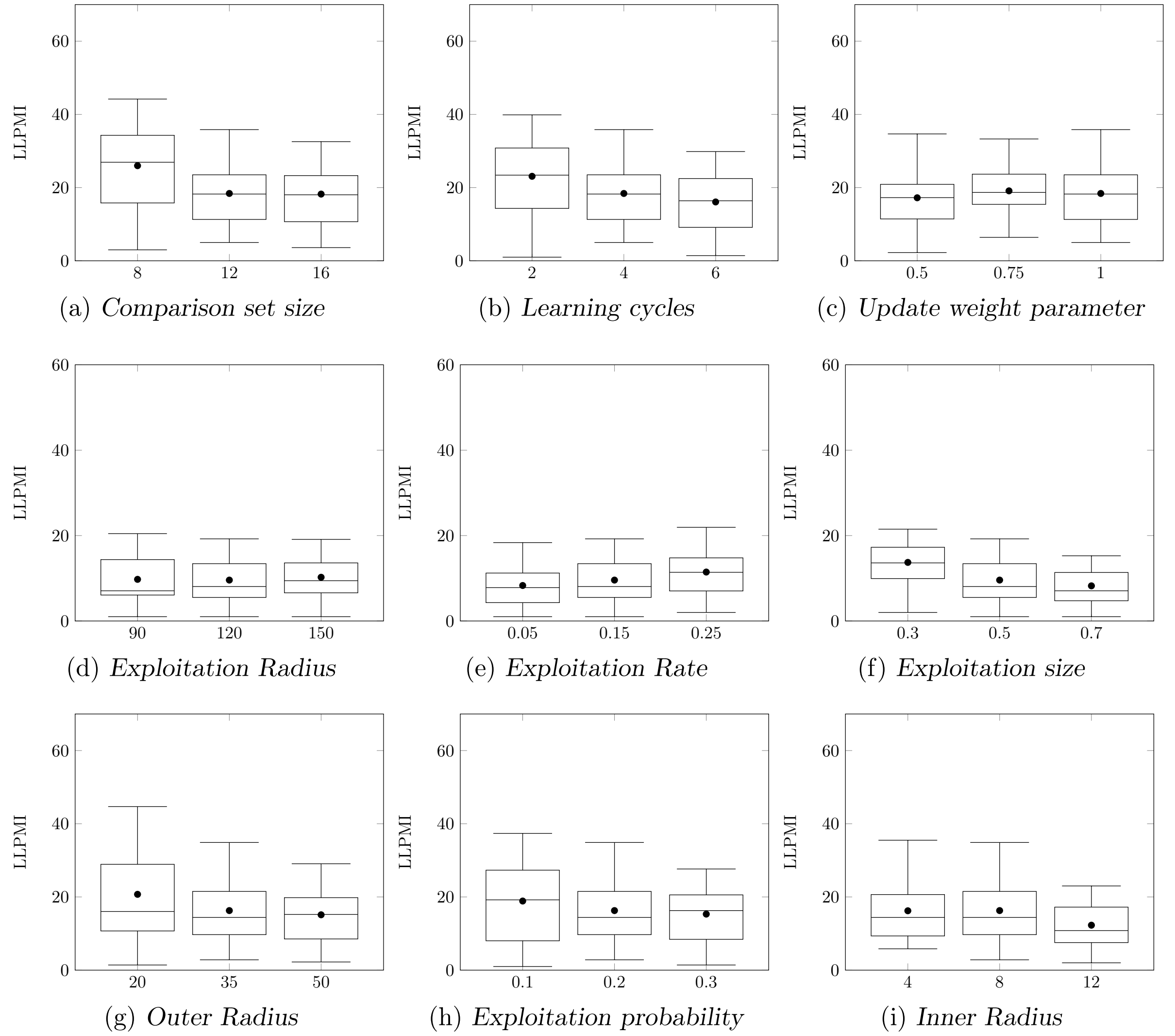
(f) *Exploitation size*

(g) *Outer Radius*

(h) *Exploitation probability*

(i) *Inner Radius*

FIGURE 7.13: *Parametric sensitivity analysis results for each of the nine parameters in Table 7.27 in respect of the GLPMI in (6.3) for the difference-based MACBETH method. The middle box represents the base case in each plot (except for the update weight parameter where 1 is the base case). Mean values are denoted by solid dots, while median values are represented by central quartile lines.*



(a) *Comparison set size* (b) *Learning cycles* (c) *Update weight parameter*

(d) *Exploitation Radius* (e) *Exploitation Rate* (f) *Exploitation size*

(g) *Outer Radius* (h) *Exploitation probability* (i) *Inner Radius*

FIGURE 7.14: *Parametric sensitivity analysis results for each of the nine parameters in Table 7.27 in respect of the LLPMI in (6.4) for the difference-based MACBETH method. The middle box represents the base case in each plot (except for the update weight parameter where 1 is the base case). Mean values are denoted by solid dots, while median values are represented by central quartile lines.*

	Medium (120)	Large (150)
Small (90)	1.52×10^{-05}	1.09×10^{-08}
Medium (120)	—	8.60×10^{-02}

(a) Fisher's LSD test p-values for the comparison set size.

	Medium (0.15)	Large (0.25)
Small (0.05)	1.97×10^{-03}	7.20×10^{-04}
Medium (0.25)	—	7.54×10^{-01}

(b) Fisher's test p-values for the number of learning cycles.

TABLE 7.35: *Fisher's LSD test p-values for the LLPMI means in Table 7.34. Table entries smaller than [larger than, respectively]* 0.05 *are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the null-hypothesis (stating that the LLPMI means of two learning iterations are equal), may be rejected [may not be rejected, respectively].*

No	Parameter	Sensitive?	Best value
1	Comparison set size	Yes	Large (16)
2	Number of iterations	Yes	Large (6)
3	Update weight parameter	No	—
4	Exploitation radius	No	—
5	Exploitation rate	No	—
6	Exploitation set size	No	—
7	Inner radius	No	—
8	Outer radius	No	—
9	Exploitation probability	No	—

TABLE 7.36: *Results of parametric sensitivity analysis in respect of the GLPMI mean for the MACBETH method.*

level of significance only in the cases of the comparison set size and the exploitation size, but unlike the GLPMI results, the null hypothesis for the exploitation size parameter sample means may also be rejected (in the case of the LLPMI) with 95% confidence. It may therefore be claimed with 95% confidence that the LLPMI is insensitive to the values of all the remaining parameters in Table 7.27. The LLPMI, however, depends sensitively on the comparison set size, number of learning cycles and exploitation size parameters.

The Levene test p-values in Table 7.37 furthermore reveal that the null-hypothesis stating that *the best LLPMI samples are homoscedastic* may be rejected at a 5% level of significance only in the case of the outer radius parameter of the PHC method. This means that Fisher's LSD *post hoc* test may be used to detect between which samples the best LLPMI means differ in the cases of the comparison set size, number of learning cycles and exploitation size parameters.

These *post hoc* test p-values may be found in Table 7.38. It follows with 95% confidence from the p-values in Table 7.38(a) and (c) that the large or medium values of the comparison set size parameter and the exploitation set size parameter (which are statistically indistinguishable) leads to a statistically better best LLPMI mean than when adopting the small value of this parameter. It follows with 95% confidence from Table 7.38(b) that the large value of the number of learning cycles parameter leads to a statistically better best LLPMI mean value than when the small value of this parameter is employed. The conclusions of the sensitivity analysis in terms of the best LLPMI are summarised in Table 7.39.

	LLPMI			ANOVA	Levene
Parameter	Small	Medium	Large	test p-values	test p-values
Comparison set size	25.99	18.40	18.23	1.66×10^{-03}	2.35×10^{-01}
Learning cycles	23.09	18.40	16.08	1.43×10^{-02}	3.58×10^{-01}
Update weight parameter	17.22	19.11	18.40	6.50×10^{-01}	3.80×10^{-01}
Exploitation radius	9.75	9.58	10.24	8.84×10^{-01}	4.06×10^{-01}
Exploitation rate	8.31	9.58	11.46	7.63×10^{-02}	9.93×10^{-01}
Exploitation size	13.72	9.58	8.23	1.32×10^{-04}	5.13×10^{-01}
Outer radius	20.72	16.28	15.12	8.22×10^{-02}	5.53×10^{-03}
Exploitation probability	18.89	16.28	15.32	3.17×10^{-01}	8.56×10^{-02}
Inner radius	16.22	16.28	12.28	8.27×10^{-02}	1.47×10^{-01}

TABLE 7.37: *Parametric sensitivity analysis LLPMI means returned by the DSS framework of §5 populated with the difference-based approach. The corresponding p-values returned for the ANOVA/Levene tests are also included. Table entries smaller than [larger than, respectively] 0.05 are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the null-hypothesis (stating that the LLPMI means/variances of all pairs of learning iterations are equal), may be [may not be, respectively] rejected.*

	Medium (120)	Large (150)
Small (90)	2.03×10^{-03}	1.64×10^{-03}
Medium (120)	—	9.45×10^{-01}

(a) Fisher's LSD test p-values for the comparison set size.

	Medium (0.15)	Large (0.25)
Small (0.05)	5.30×10^{-02}	4.30×10^{-03}
Medium (0.25)	—	3.35×10^{-01}

(b) Fisher's test p-values for the number of learning cycles.

	Medium (0.15)	Large (0.25)
Small (0.05)	1.80×10^{-03}	4.90×10^{-05}
Medium (0.25)	—	2.95×10^{-01}

(c) Fisher's test p-values for the exploitation set size.

TABLE 7.38: *Fisher's LSD test p-values for the LLPMI means in Table 7.37. Table entries smaller than [larger than, respectively] 0.05 are typeset in black [red, respectively], show that the probability of a Type I error is small [large, respectively] and hence that the null-hypothesis (stating that the LLPMI means of two learning iterations are equal), may be rejected [may not be rejected, respectively].*

No	Parameter	Sensitive?	Best value
1	Comparison set size	Yes	Large (16)
2	Number of iterations	Yes	Large (6)
3	Update weight parameter	No	—
4	Exploitation radius	No	—
5	Exploitation rate	Yes	Small (0.05)
6	Exploitation set size	Yes	Large (0.7)
7	Inner radius	No	—
8	Outer radius	No	—
9	Exploitation probability	No	—

TABLE 7.39: *Results of parametric sensitivity analysis in respect of the best LLPMI mean of the MACBETH method.*

7.6 Chapter summary

As mentioned in the introduction to this chapter, the purpose of the chapter was to assess the potential effectiveness and practical workability of the SPBP DSS of §5 for a perfectly consistent DM within the context of the 2DCSP in the context of the ratio-based AHP and the difference-based MACBETH method. A validation experiment was outlined (which included the use of various statistical tests). The entire validation experiment was carried out at a 95% level of confidence.

The intrinsic ability of the DSS of §5 with respect to learning the true value function of the DM in the context of the 2DCSP was first assessed when populating the system with six AHP judgement scales, two methods of aggregation and a uniformly distributed random generation of alternatives. It was found for all these combinations of DSS configuration elements, except when adopting the balanced AHP judgement scale, that learning takes place in an iterative fashion according to the GLPMI after varying numbers of iterations, as summarised in Tables 7.3 and 7.6.

A similar analysis was conducted to measure the intrinsic ability of the DSS to learn when populating the framework of §5 with a difference-based method and it was found that when incorporating the MACBETH method, learning does indeed take place in an iterative fashion according to the GLPMI.

Thereafter, the relative performances of the various aforementioned DSS configuration elements were evaluated. It was found that whereas the system is insensitive to the method of aggregation employed, there is a definitive preference ranking of the judgement scales considered in terms of the quality of global learning achievable within a fixed number of learning iterations. This ranking may be found in Table 7.12.

Up to this point in the DSS validation experiments, all new alternatives presented to the DM model for pairwise comparisons were generated purely randomly according to a uniform distribution over the entire decision space. The next aspect of the validation experiment involved assessing the merit of including some form of bias towards or exploitation of promising areas of the decision space according to the current value function estimate when generating new alternatives. Three methods of alternative generation were compared — the previously considered random method, together with an exploitation method, called the Basic E&E method (biasing sampling around the global maximum of the value function estimate), and the method of PHC. These experiments were conducted for the ratio-based as well as difference-based approaches and it was found that there is a preference ranking of these alternative generation methods, but that this ranking depends on the goal of the learning process (*i.e.* whether emphasis is placed on the global quality of the value function estimate or merely on the local quality of the estimate around its local maximum). The two alternative generation method rankings were summarised in Tables 7.25 and 7.26.

All of the above aspects of the DSS validation experiment were carried out with sensible or medium-ranged values of the parameters contained in the various AHP and MACBETH judgement scales, exploration/exploitation algorithms and DSS configuration. A sensitivity analysis of these parameter values was finally performed within the context of the best-performing AHP judgement scale (the power scale) and the MACBETH difference scale. The best parameter values thus uncovered may be found in Tables 7.28 and 7.33 for the AHP and in Tables 7.36 and 7.39 for the MACBETH method.

The entire DSS validation experiment described above was, however, carried out under the (unrealistic) assumption of a perfectly consistent DM. The potential effects on the validation results of varying degrees of inconsistency on the part of the DM are investigated in the following chapter.

In summary, the main findings resulting from the numerical experiments conducted in this chapter were the following:

- When populating the DSS of §5 with any of the AHP judgement scales in Table 2.2 (except the balanced judgement scale) in conjunction with the LLSM or the EM, or when populating the DSS with the MACBETH difference scale, the system is able to learn the value function of the DM (to some extent) both globally and locally (in the vicinity of well-performing areas of the decision space) within a small number of iterations of the learning cycle in Figure 5.1.
- The consistency with which the above learning takes place, increases marginally as the number of learning iterations increases.
- The convex (power) judgement scale performed better in terms of learning the DM's value function globally than did the linear judgement scale which, in turn, fared better in this respect than the concave (root) judgement scale.
- There is a benefit in the sense of improved global value function estimation quality when exploration is emphasised during the generation of new alternatives.
- There is a benefit in the sense of improved local value function estimation quality (in the vicinity of well-performing areas of the decision space) when exploitation is emphasised during the generation of new alternatives.

CHAPTER 8

The Effect of DM Inconsistency

Contents

Whereas the assessment in §7 of the practical workability and potential effectiveness of the DSS proposed in §5 took place under the strong assumption of a perfectly consistent DM, the purpose of this chapter is to assess the effects on the iterative learning ability of the DSS in §5 of introducing small to medium levels of DM inconsistency. After describing the method in which DM inconsistency is modelled and the experimental setup of the latter assessment in §8.1, the assessment itself is carried out in §8.2. In order to avoid undue repetitiveness, not all judgement scale and aggregation method combinations of §7 are, however, considered in this chapter. More specifically, the effects of DM inconsistency on the GLPMI of (6.3) is assessed in §8.2.1 for two representative combinations of the power AHP judgement scale and aggregation methods. The entire assessment is then repeated in §8.2.2 for the difference-based MACBETH method again analysing the effects of DM inconsistency on the GLPMI of (6.3). The section closes with a brief discussion of the results in §8.2.3.

The focus shifts in §8.3 to a study of the potential of increased learning effectiveness when accommodating DM inconsistency by increasing the number of anchors in the anchor set. More specifically, it is investigated whether there is any benefit in increasing the anchor set cardinality in terms of learning the DM's value function in the presence of various levels of DM inconsistency. This analysis is also carried out in respect of both the ratio-based AHP method with the best-performing power AHP judgement scale and aggregation method of the previous section (in §8.3.1) and the difference-based MACBETH method (in §8.3.2). The section again closes with a brief discussion of the results in §8.3.3. The chapter finally closes in §8.4 with a brief summary.

8.1 Modelling DM inconsistency and experimental setup

This section contains a detailed description in §8.1.1 of the approach taken towards modelling DM inconsistency for the purpose of assessing the effectiveness with which the DSS of §5 is able to accommodate small to medium levels of inconsistency in the specification of pairwise comparison information returned by the DM for a set of alternatives presented to him or her. Thereafter, the experimental setup of the assessment procedure is described briefly in §8.1.2.

8.1.1 DM inconsistency

In order to introduce a small to medium level of DM inconsistency in the pairwise comparison matrix information "returned" by a virtual DM for the purposes of testing the effectiveness of the DSS if the assumption of perfect DM consistency of §7 is relaxed, the following approach is adopted. It is still assumed that the DM bases pairwise comparison matrix entries on the "true" value function $U(g,b)$ given in algebraic form in (6.1) and depicted in Figure 6.1. The real interval $[1,\theta]$ is still partitioned into nine subintervals $\mathcal{R}_1,\ldots,\mathcal{R}_9$ (in the case of the AHP) and into six subintervals $\mathcal{R}_1,\ldots,\mathcal{R}_6$ (in the case of the MACBETH method) according to the judgement scale under consideration, where θ is defined as in (6.2).

The comparison set $\mathcal{C}$ under consideration is partitioned into two subsets: A subset $\mathcal{C}^\epsilon$ of cardinality $\epsilon \ll |\mathcal{C}|$ and the subset $\mathcal{C}\setminus\mathcal{C}^\epsilon$. The number $\epsilon \in \mathbb{N}_0$ is called the *extent of inconsistency* of the DM. Suppose an inconsistent DM considers alternatives (g_i,b_i), $(g_j,b_j)\in\mathcal{C}$, and assume (again without loss of generality) that $U(g_i,b_i)\geq U(g_j,b_j)$. Then the entry in row i and column j, and that in row j and column i, of the pairwise comparison matrix are taken as $a(\ell)$ and $1/a(\ell)$, respectively, for the ratio-based approach if $U'(g_i,b_i)/U'(g_j,b_j)\in\mathcal{R}_\ell$, where

$$U'(g_i,b_i)=\begin{cases}U(g_i,b_i) & \text{if } (g_i,b_i)\in\mathcal{C}\setminus\mathcal{C}^\epsilon\\ (1+\sigma)U(g_i,b_i) & \text{if } (g_i,b_i)\in\mathcal{C}^\epsilon\end{cases} \tag{8.1}$$

and where $a(1),\ldots,a(9)$ denote the AHP judgement scale entries under consideration. In the case of the difference-based approach, the inconsistent DM would categorise pairs of alternatives based on the difference between their true value function values $U'(g_i,b_i)$ and $U'(g_j,b_j)$, giving rise to the value k in row i and column j of the comparison matrix if $U'(g_i,b_i)-U'(g_j,b_j)\in\mathcal{R}'_k$, for the six MACBETH intervals $\mathcal{R}'_1,\ldots,\mathcal{R}'_6$, when $U'(g_i,b_i)\geq U'(g_j,b_j)$.

The value σ in (8.1) is an inconsistency offset and is randomly generated according to a uniform distribution on the real interval $[-\overline{\sigma},\overline{\sigma}]$. That is, instead of basing the pairwise comparison entries in row i and column j, and that in row j and column i, of the pairwise comparison matrix directly on the interval $\mathcal{R}_k$ in which the ratio $U(g_i,b_i)/U(g_j,b_j)$ falls or on the interval $\mathcal{R}'_k$ in which the difference $U(g_i,b_i)-U(g_j,b_j)$ falls, a random perturbation of this value is taken instead. Note that in the case of the AHP the reciprocal-symmetry property of the pairwise comparison matrix is preserved (*i.e.* the entry in row i and column j, and that in row j and column i, are reciprocals), but that the requirement of transitivity of pairwise comparisons for all alternative triples is no longer necessarily satisfied for any of the value function estimation methods.

If the perturbation in (8.1) leads to a different order of alternatives when ranked in order of non-increasing desirability, as required by the MACBETH method, the altered order is used to index the rows and columns of the pairwise comparison matrix and the difference $U'(g_j,b_j)-U'(g_i,b_i)$ is used instead of $U'(g_i,b_i)-U'(g_j,b_j)$ to determine the relevant difference category $\mathcal{R}'_k$. Similarly, if the perturbation in (8.1) results in the inequality $U'(g_i,b_i)<U'(g_j,b_j)$, then the ratio $U'(g_j,b_j)/U'(g_i,b_i)$ is used instead of $U'(g_i,b_i)/U'(g_j,b_j)$ to determine the relevant AHP category $\mathcal{R}_\ell$.

The attention of the reader is drawn to the fact that the extent of inconsistency ϵ should be relatively small compared to the comparison set size, for otherwise the DM would be so inconsistent as to return meaningless (*i.e.* almost random) preference information. Finally, note that if $\epsilon = 0$ or $\overline{\sigma} = 0$, then the DM is again perfectly consistent as was the case in §7.

8.1.2 Experimental setup

A selection of the experiments of §7 are repeated in this chapter for small and medium levels of DM inconsistency in order to ascertain the effects of the above inconsistency on the ability of the DSS of §5 to learn the DM's true value function when adopting a ratio-based (AHP) approach or a difference-based (MACBETH) approach. More specifically, the values $\epsilon = 2$ (a small extent of DM inconsistency) and $\epsilon = 4$ (a medium extent of DM inconsistency) in (8.1) are considered, in addition to the values $\overline{\sigma} = 0.5$ (a medium degree of DM inconsistency) and $\overline{\sigma} = 0.9$ (a large degree of DM inconsistency).

The same global learning performance indicator introduced in §6.1.2 is again used to measure the effectiveness of the iterative learning process of the DSS. The analysis is again carried out within the same inferential statistics framework as that described in §7.1 for the ratio-based AHP (in 8.2.1) and the difference-based MACBETH method (in §8.2.2). Instead of employing all the AHP judgement scale and aggregation method combinations that were considered in §7, only a representative selection of the combinations are, however, considered in a bid to avoid unnecessary repetitiveness. These combinations involve both aggregation methods of §7, but only the convex (power) AHP judgement scale which performed the best in §7. Hence the following two AHP judgement scale and aggregation method combinations are considered for the ratio-based approach in §8.2.1:

- The power AHP judgement scale in conjunction with the LLSM, and
- The same power judgement scale in conjunction with the EM.

In the assessment experiments of this chapter, the number of anchors included in the anchor set is $\chi = 1$ or $\chi = 2$, while update weights $\zeta = 0.7$ and $\zeta = 1.0$ are adopted. The parameter values considered in the experiments of this chapter are outlined in Table 8.1. A total of 24 experiments are carried out in §8.2 for the ratio-based AHP and difference-based MACBETH method, after which the best performing AHP power judgement scale and aggregation method combination, and the difference-based MACBETH method are taken forward in the anchor set cardinality experiments of §8.3.1 which constitutes a further 16 experiments. Thirty replications of $\omega = 4$ iterations of the learning cycle in Figure 5.1 are carried out in each of the 40 experiments.

All analyses are again carried out at a 5% level of statistical significance.

8.2 How DM inconsistency affects DSS learning ability

As mentioned in the chapter introduction, the objective in this section is to assess to what extent the DSS framework of §5 is capable of learning as the number of iterations in the learning cycle of Figure 5.1 increases in the presence of DM inconsistency as modelled in §8.1. This assessment takes place according to GLPMI for the AHP in §8.2.1 and MACBETH method in §8.2.2.

8.2.1 The effect of DM inconsistency on the AHP

The results of two sets of experiments are reported and interpreted in this section. These two sets of experiments correspond to the power AHP judgement scale and two aggregation method

Parameter	No of values	Values considered	
§8.2			
Inconsistency amount	2	$\epsilon = 2$	$\epsilon = 4$
Inconsistency offset	2	$\overline{\sigma} = 0.5$	$\overline{\sigma} = 0.9$
Weight update	2	$\zeta = 0.7$	$\zeta = 1.0$
Aggregation methods	2	LLSM	EM
Anchor set cardinality	1	$\chi = 1$	—
Judgement scales	1	Power	
Learning performance measures	1	GLPMI	—
§8.3			
Inconsistency amount	2	$\epsilon = 2$	$\epsilon = 4$
Inconsistency offset	2	$\overline{\sigma} = 0.5$	$\overline{\sigma} = 0.9$
Weight update	2	$\zeta = 0.7$	$\zeta = 1.0$
Aggregation methods	1	—	EM
Anchor set cardinality	2	$\chi = 1$	$\chi = 2$
Judgement scales	1	Power	
Learning performance measures	1	GLPMI	—

TABLE 8.1: *A summary of the experimental parameter value combinations considered in the assessment of this chapter.*

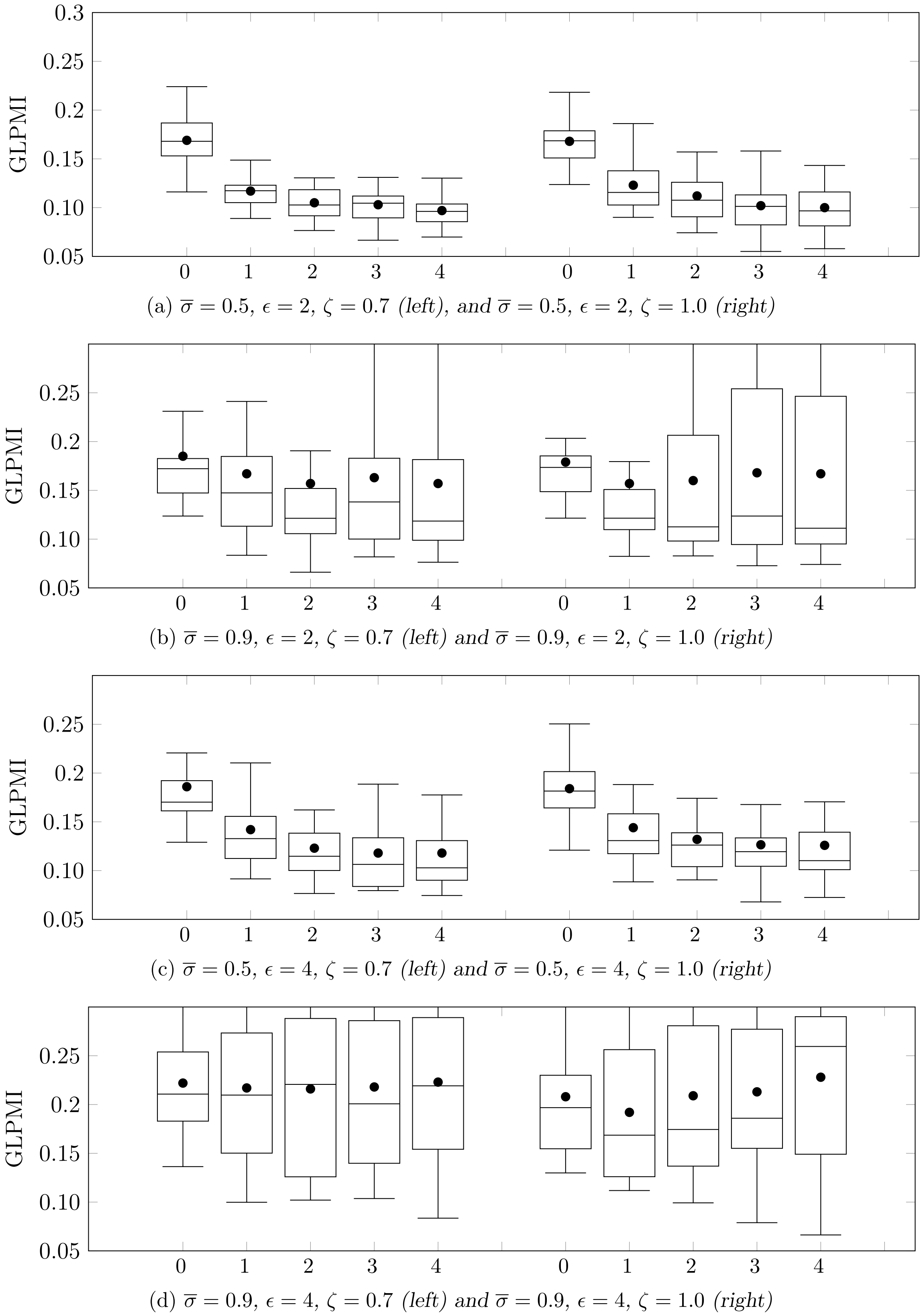
combinations mentioned in §8.1.2. In each set, the effectiveness with which the DSS of §5 can learn the DM's underlying value function is assessed in eight experiments, corresponding to the various combinations of employing $\chi = 1$ anchor with extents of inconsistency of $\epsilon = 2$ or $\epsilon = 4$ and degrees of inconsistency of $\sigma = 0.7$ or $\sigma = 1.0$ in (8.1), as well as update weights of $\zeta = 0.7$ or $\zeta = 1.0$. Learning effectiveness is measured according to the GLMPI of (6.3).

The power judgement scale combined with the LLSM

	GLPMI mean					ANOVA	Levene
	It 0	It 1	It 2	It 3	It 4	test p-values	test p-values
$\zeta = 0.7$, $\epsilon = 2$, $\overline{\sigma} = 0.5$	0.169	0.117	0.105	0.103	0.097	$< 1 \times 10^{-17}$	3.05×10^{-01}
$\zeta = 1.0$, $\epsilon = 2$, $\overline{\sigma} = 0.5$	0.168	0.123	0.112	0.102	0.100	$< 1 \times 10^{-17}$	8.52×10^{-01}
$\zeta = 0.7$, $\epsilon = 2$, $\overline{\sigma} = 0.9$	0.185	0.167	0.157	0.163	0.157	5.40×10^{-01}	1.37×10^{-01}
$\zeta = 1.0$, $\epsilon = 2$, $\overline{\sigma} = 0.9$	0.179	0.157	0.160	0.168	0.167	8.42×10^{-01}	6.86×10^{-05}
$\zeta = 0.7$, $\epsilon = 4$, $\overline{\sigma} = 0.5$	0.186	0.142	0.123	0.118	0.118	2.69×10^{-10}	8.65×10^{-01}
$\zeta = 1.0$, $\epsilon = 4$, $\overline{\sigma} = 0.5$	0.184	0.144	0.132	0.126	0.125	5.33×10^{-08}	7.81×10^{-01}
$\zeta = 0.7$, $\epsilon = 4$, $\overline{\sigma} = 0.9$	0.222	0.217	0.216	0.218	0.223	9.93×10^{-01}	1.18×10^{-02}
$\zeta = 1.0$, $\epsilon = 4$, $\overline{\sigma} = 0.9$	0.208	0.192	0.209	0.213	0.228	4.14×10^{-16}	2.31×10^{-02}

TABLE 8.2: *ANOVA and Levene test p-values for the GLPMI values over 4 iterations of the learning cycle in Figure 5.1 when populating the DSS of §5 with the power AHP judgement scale in conjunction with the LLSM of aggregation and employing $\chi = 1$ anchor at extents of inconsistency $\epsilon = 2$ and 4, degrees of inconsistency $\sigma = 0.5$ and 0.9, and for update weights $\zeta = 0.7$ and 1.0.*

Figures 8.1(a)–(d) contain box plots of the GLPMI values in (6.3) associated with $\omega = 4$ iterations of the learning cycle of Figure 5.1 when populating the DSS framework of §5 with the power AHP judgement scale in conjunction with the LLSM and employing $\chi = 1$ anchor at $\sigma = 0.5$ or



(a) $\overline{\sigma} = 0.5$, $\epsilon = 2$, $\zeta = 0.7$ *(left), and* $\overline{\sigma} = 0.5$, $\epsilon = 2$, $\zeta = 1.0$ *(right)*

(b) $\overline{\sigma} = 0.9$, $\epsilon = 2$, $\zeta = 0.7$ *(left) and* $\overline{\sigma} = 0.9$, $\epsilon = 2$, $\zeta = 1.0$ *(right)*

(c) $\overline{\sigma} = 0.5$, $\epsilon = 4$, $\zeta = 0.7$ *(left) and* $\overline{\sigma} = 0.5$, $\epsilon = 4$, $\zeta = 1.0$ *(right)*

(d) $\overline{\sigma} = 0.9$, $\epsilon = 4$, $\zeta = 0.7$ *(left) and* $\overline{\sigma} = 0.9$, $\epsilon = 4$, $\zeta = 1.0$ *(right)*

FIGURE 8.1: *GLPMI samples and means associated with Iterations 0, 1, 2, 3 and 4 of the learning cycle for the power AHP judgement scale in conjunction with the LLSM for degrees of inconsistency* $\overline{\sigma} = 0.5$ *or 1.0, extents of inconsistency* $\epsilon = 2$ *or 4, and update weights* $\zeta = 0.7$ *or 1.0.*

0.9 levels of inconsistency, degrees of inconsistency $\epsilon = 2$ or 4, and update weights $\zeta = 0.7$ (on the left) or $\zeta = 0.7$ (on the right). The means of the GLPMI samples, as well as the p-values returned for the ANOVA and Levene tests in respect of these samples are reported in Table 8.2.

It follows from the ANOVA test p-values in Table 8.2 that the relevant null-hypothesis (stating that the GLPMI means across all learning iterations are statistically indistinguishable) may be rejected at a statistical significance level of $\alpha = 5\%$ for the four cases where $\sigma = 0.5$. Based on a visual inspection of the box plots in Figures 8.1(a) and (c), it may therefore be concluded with 95% confidence that some degree of learning still takes place if any of the aforementioned combinations of the extent of inconsistency and weight updating level is combined with the AHP power judgement scale and the LLSM. The ANOVA p-values in Table 8.2 indicate that in the case of an increased degree of inconsistency $\sigma = 0.9$ in Figures 8.1(b) and (d) it may not be concluded with 95% confidence that the GLPMI means differ and that some degree of learning takes place.

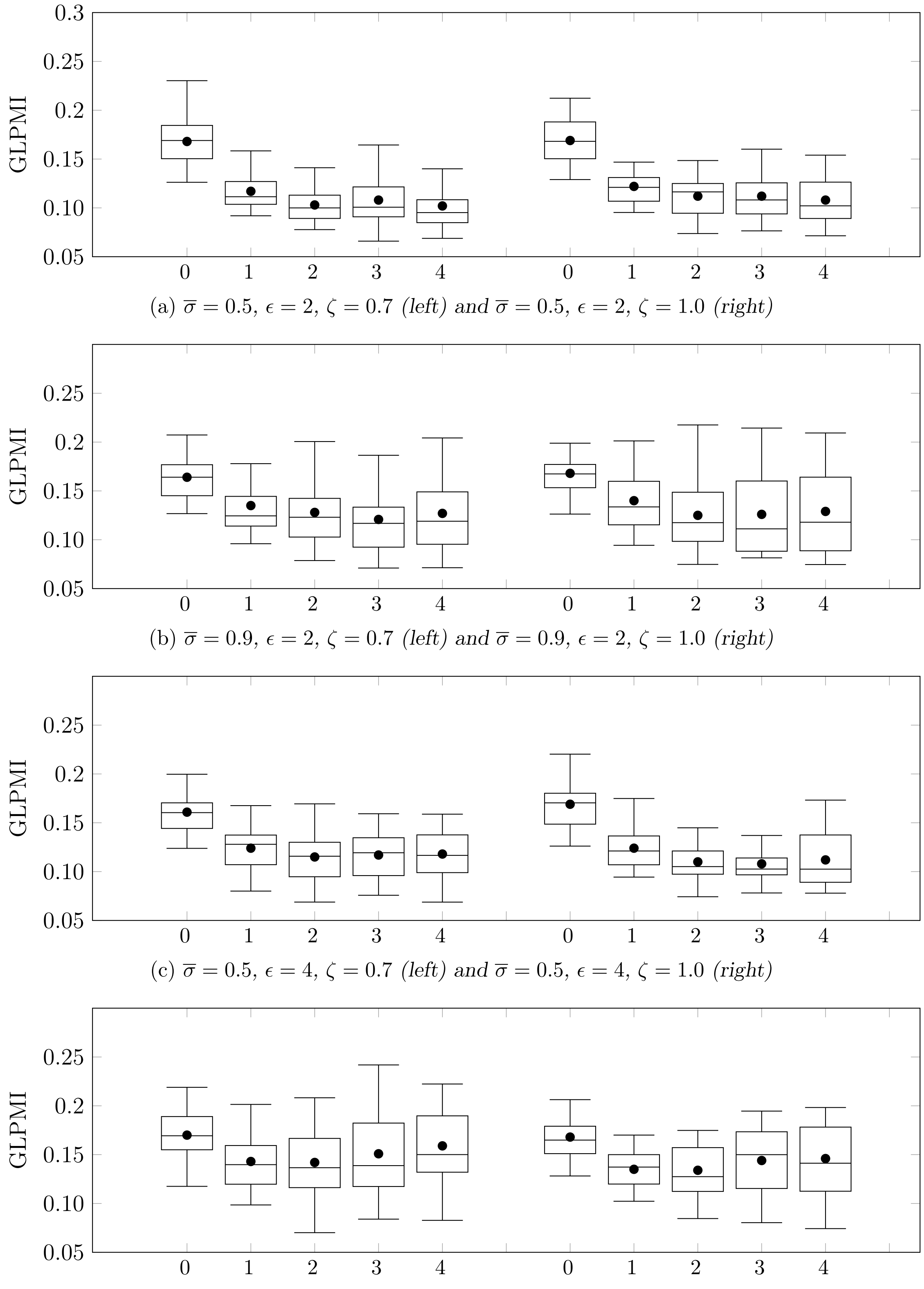
The Levene test p-values in Figure 8.2 indicate that the null-hypothesis (stating that the GLPMI samples are homoscedastic across all learning iterations) may not be rejected at a statistical significance level of $\alpha = 5\%$ for all extents and degrees of inconsistency, as well as weight update combinations, except for the cases where $\zeta = 1.0$, $\epsilon = 2$ and $\overline{\sigma} = 0.9$, where $\zeta = 0.7$, $\epsilon = 4$ and $\overline{\sigma} = 0.9$, and where $\zeta = 1.0$, $\epsilon = 4$ and $\sigma = 0.9$ (the Levene test p-value for these cases do not exceed the 0.05 level of significance). That is, there is no statistical evidence in support of heteroscedasticity of the learning performance samples at a 95% level of confidence (their respective Levene test p-values all exceed the significance level of 0.05). It would typically be necessary at this point to isolate the pairs of iterations by Fisher's LSD or Games-Howell *post hoc* test p-values. The purpose of this section is, however, only to investigate which parameter combination is capable of learning the DM's true value function to some extent over four iterations of the learning cycle. A visual inspection of Figure 8.1 therefore provides sufficient evidence that learning does not take place in Figures 8.1(b) and (d) when the various parameter combinations are combined with the power AHP judgement scale and the LLSM.

The power judgement scale combined with the EM

Figures 8.2(a)–(d) contain box plots of the GLPMI values in (6.3) associated with $\omega = 4$ iterations of the learning cycle of Figure 5.1 when populating the DSS framework of §5 with the AHP power judgement scale in conjunction with the EM in the case of employing $\chi = 1$ anchor at degrees of inconsistency $\sigma = 0.5$ or 0.9, and extents of inconsistency $\epsilon = 2$ or 4 for update weights $\zeta = 0.7$ (on the left) and $\zeta = 0.7$ (on the right). The means of the GLPMI samples are reported in Table 8.3 as are the p-values returned by the ANOVA and Levene tests.

It follows from the ANOVA p-values in Table 8.3 that the relevant null-hypothesis (stating that the GLPMI means across all learning iterations are statistically indistinguishable) may be rejected at a statistical significance level of $\alpha = 5\%$ for each of the eight cases. Based on a visual inspection of the box plots in each of Figures 8.2(a)–(d), it may therefore be concluded with 95% confidence that some degree of learning takes place if any of the inconsistency degree or extent values are combined with any of the two update weight values in conjunction with the power judgement scale and the EM.

The Levene test p-values in Table 8.3 furthermore indicate that the null-hypothesis (stating that the GLPMI samples are homoscedastic across all learning iterations) may not be rejected at a statistical significance level of $\alpha = 5\%$ for the parameter combinations of (a) $\zeta = 0.7$, $\epsilon = 2$ and $\overline{\sigma} = 0.5$, (b) $\zeta = 1.0$, $\epsilon = 2$ and $\overline{\sigma} = 0.5$, (c) $\zeta = 0.7$, $\epsilon = 2$ and $\overline{\sigma} = 0.9$, or (d) $\zeta = 0.7$, $\epsilon = 4$ and $\overline{\sigma} = 0.5$. That is, there is no statistical evidence in support of heteroscedasticity of the learning



(a) $\overline{\sigma} = 0.5$, $\epsilon = 2$, $\zeta = 0.7$ *(left)* and $\overline{\sigma} = 0.5$, $\epsilon = 2$, $\zeta = 1.0$ *(right)*

(b) $\overline{\sigma} = 0.9$, $\epsilon = 2$, $\zeta = 0.7$ *(left)* and $\overline{\sigma} = 0.9$, $\epsilon = 2$, $\zeta = 1.0$ *(right)*

(c) $\overline{\sigma} = 0.5$, $\epsilon = 4$, $\zeta = 0.7$ *(left)* and $\overline{\sigma} = 0.5$, $\epsilon = 4$, $\zeta = 1.0$ *(right)*

(d) $\overline{\sigma} = 0.9$, $\epsilon = 4$, $\zeta = 0.7$ *(left)* and $\overline{\sigma} = 0.9$, $\epsilon = 4$, $\zeta = 1.0$ *(right)*

FIGURE 8.2: *GLPMI samples and means associated with Iterations 0, 1, 2, 3 and 4 of the learning cycle for the power AHP judgement scale in conjunction with the EM for degrees of inconsistency $\overline{\sigma} = 0.5$ or 1.0, extents of inconsistency $\epsilon = 2$ or 4, and update weights $\zeta = 0.7$ or 1.0.*

	GLPMI mean					ANOVA	Levene
	It 0	It 1	It 2	It 3	It 4	test p-values	test p-values
$\zeta = 0.7$, $\epsilon = 2$, $\overline{\sigma} = 0.5$	0.168	0.117	0.103	0.108	0.102	$< 1 \times 10^{-17}$	3.85×10^{-01}
$\zeta = 1.0$, $\epsilon = 2$, $\overline{\sigma} = 0.5$	0.169	0.122	0.112	0.112	0.108	$< 1 \times 10^{-17}$	1.38×10^{-01}
$\zeta = 0.7$, $\epsilon = 2$, $\overline{\sigma} = 0.9$	0.164	0.135	0.128	0.121	0.127	6.21×10^{-06}	1.06×10^{-01}
$\zeta = 1.0$, $\epsilon = 2$, $\overline{\sigma} = 0.9$	0.168	0.140	0.125	0.126	0.129	6.02×10^{-06}	1.52×10^{-04}
$\zeta = 0.7$, $\epsilon = 4$, $\overline{\sigma} = 0.5$	0.161	0.124	0.115	0.117	0.118	9.87×10^{-14}	4.16×10^{-01}
$\zeta = 1.0$, $\epsilon = 4$, $\overline{\sigma} = 0.5$	0.169	0.124	0.110	0.108	0.112	$< 1 \times 10^{-17}$	2.01×10^{-02}
$\zeta = 0.7$, $\epsilon = 4$, $\overline{\sigma} = 0.9$	0.170	0.143	0.142	0.151	0.159	8.07×10^{-03}	2.03×10^{-03}
$\zeta = 1.0$, $\epsilon = 4$, $\overline{\sigma} = 0.9$	0.168	0.135	0.134	0.144	0.146	3.42×10^{-05}	3.88×10^{-05}

TABLE 8.3: *ANOVA and Levene test p-values for the GLPMI values over 4 iterations of the learning cycle in Figure 5.1 when populating the DSS of §5 with the power AHP judgement scale in conjunction with the EM of aggregation and employing $\chi = 1$ anchor at extents of inconsistency $\epsilon = 2$ and 4, degrees of inconsistency $\sigma = 0.5$ and 0.9, and for update weights $\zeta = 0.7$ and 1.0.*

performance samples at a 95% level of confidence (their respective Levene test p-values all exceed the significance level of 0.05). It therefore cannot be claimed with 95% confidence that there is an improvement in the consistency of the level of learning effectiveness in these four cases over the experimental replications when combined with the power judgement scale and the EM.

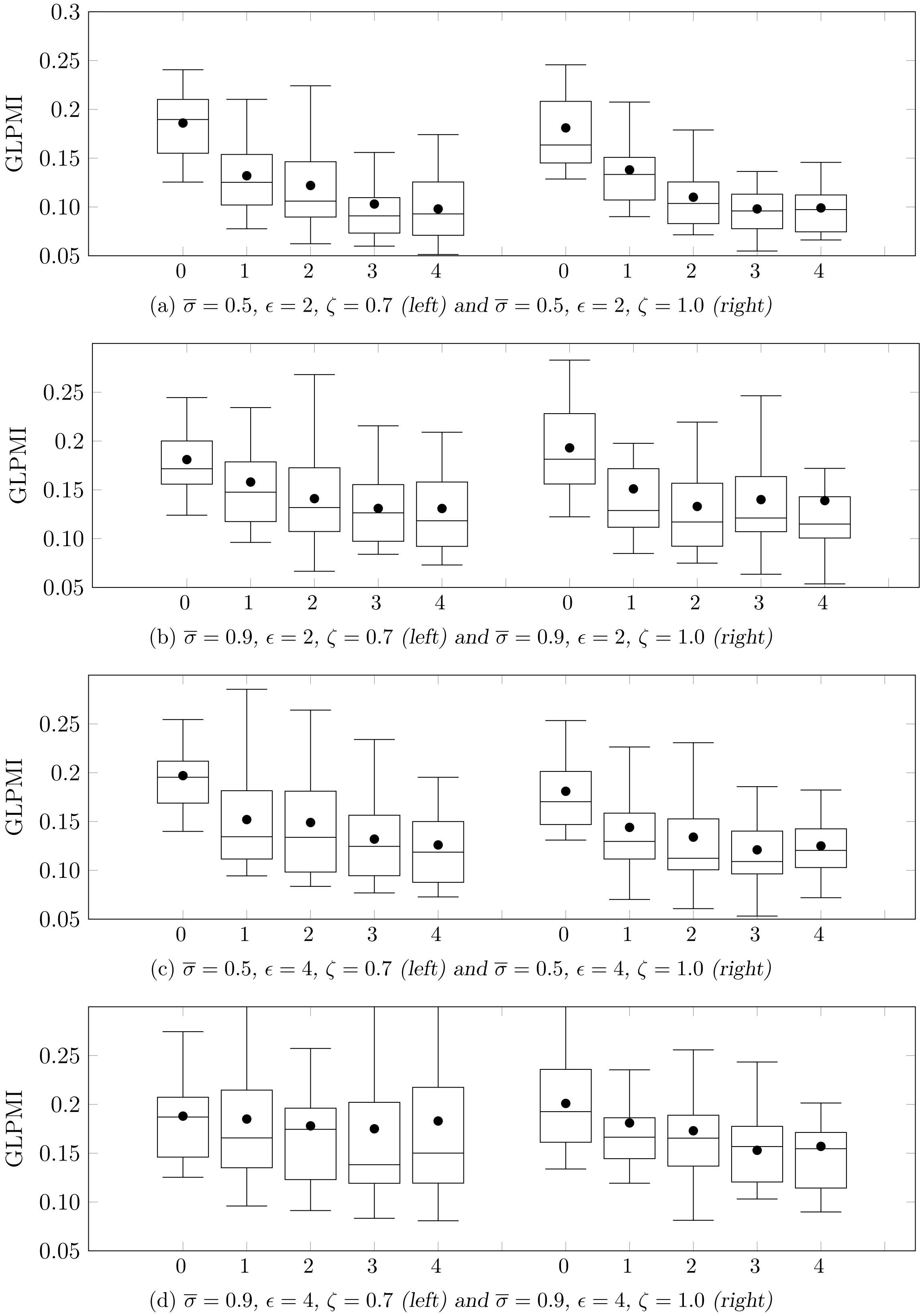
8.2.2 The effect of DM inconsistency on MACBETH

The experimental results are reported and interpreted in this section for the case of populating the DSS of §5 with the difference-based MACBETH method. In these experiments, the effectiveness with which the DSS can estimate the DM's true underlying value function (according to the GLPMI of (6.3)) is measured after four iterations of the learning cycle in Figure 5.1, corresponding to the various combinations of employing $\chi = 1$ anchor with extents of inconsistency $\epsilon = 2$ or $\epsilon = 4$ at degrees of inconsistency of $\overline{\sigma} = 0.7$ or $\overline{\sigma} = 1.0$ in (8.1) in conjunction with update weights $\zeta = 0.7$ or $\zeta = 1.0$.

	GLPMI mean					ANOVA	Levene
	It 0	It 1	It 2	It 3	It 4	test p-values	test p-values
$\zeta = 0.7$, $\epsilon = 2$, $\overline{\sigma} = 0.5$	0.186	0.132	0.122	0.103	0.098	1.67×10^{-14}	4.45×10^{-01}
$\zeta = 1.0$, $\epsilon = 2$, $\overline{\sigma} = 0.5$	0.181	0.138	0.110	0.098	0.099	1.11×10^{-16}	5.68×10^{-02}
$\zeta = 0.7$, $\epsilon = 2$, $\overline{\sigma} = 0.9$	0.181	0.158	0.141	0.131	0.131	4.76×10^{-05}	4.53×10^{-01}
$\zeta = 1.0$, $\epsilon = 2$, $\overline{\sigma} = 0.9$	0.193	0.151	0.133	0.140	0.139	6.78×10^{-04}	6.82×10^{-01}
$\zeta = 0.7$, $\epsilon = 4$, $\overline{\sigma} = 0.5$	0.197	0.152	0.149	0.132	0.126	7.74×10^{-08}	1.86×10^{-01}
$\zeta = 1.0$, $\epsilon = 4$, $\overline{\sigma} = 0.5$	0.181	0.144	0.134	0.121	0.125	3.32×10^{-06}	2.92×10^{-01}
$\zeta = 0.7$, $\epsilon = 4$, $\overline{\sigma} = 0.9$	0.188	0.185	0.178	0.175	0.183	9.56×10^{-01}	8.34×10^{-02}
$\zeta = 1.0$, $\epsilon = 4$, $\overline{\sigma} = 0.9$	0.201	0.181	0.173	0.153	0.157	1.47×10^{-03}	6.90×10^{-01}

TABLE 8.4: *ANOVA and Levene test p-values for the GLPMI values over 4 iterations of the learning cycle in Figure 5.1 when populating the DSS of §5 with the MACBETH method and employing $\chi = 1$ anchor at extents of inconsistency $\epsilon = 2$ and 4, degrees of inconsistency $\overline{\sigma} = 0.5$ and 0.9, and for update weights $\zeta = 0.7$ and 1.0.*

Figure 8.3 contains box plots of the GLPMI means returned during four iterations of the learning cycle of Figure 5.1 for the difference-based MACBETH method when employing $\chi = 1$ anchor



(a) $\overline{\sigma} = 0.5$, $\epsilon = 2$, $\zeta = 0.7$ *(left) and* $\overline{\sigma} = 0.5$, $\epsilon = 2$, $\zeta = 1.0$ *(right)*

(b) $\overline{\sigma} = 0.9$, $\epsilon = 2$, $\zeta = 0.7$ *(left) and* $\overline{\sigma} = 0.9$, $\epsilon = 2$, $\zeta = 1.0$ *(right)*

(c) $\overline{\sigma} = 0.5$, $\epsilon = 4$, $\zeta = 0.7$ *(left) and* $\overline{\sigma} = 0.5$, $\epsilon = 4$, $\zeta = 1.0$ *(right)*

(d) $\overline{\sigma} = 0.9$, $\epsilon = 4$, $\zeta = 0.7$ *(left) and* $\overline{\sigma} = 0.9$, $\epsilon = 4$, $\zeta = 1.0$ *(right)*

FIGURE 8.3: *GLPMI samples and means associated with Iterations 0, 1, 2, 3 and 4 of the learning cycle for the difference-based MACBETH method for degrees of inconsistency* $\sigma = 0.5$ *or 1.0, extents of inconsistency* $\epsilon = 2$ *or 4, and update weights* $\zeta = 0.7$ *or 1.0.*

at degrees of inconsistency $\overline{\sigma} = 0.5$ or 0.9, at extents of inconsistency $\epsilon = 2$ or 4, and for update weights $\zeta = 0.7$ (on the left) and $\zeta = 0.7$ (on the right). The mean GLPMI values, as well as the ANOVA and Levene p-values, are provided in Figure 8.4.

It follows from the ANOVA p-values in Figure 8.4, that the relevant null-hypothesis (stating that the LLPMI means are statistically indistinguishable across all levels of DM inconsistency) may be rejected at a significance level of $\alpha = 5\%$ in all cases, except for the case where $\zeta = 0.7$, $\epsilon = 4$, $\sigma = 0.9$. It furthermore follows from the Levene test p-values that the relevant null-hypothesis (stating that the LLPMI samples are homoscedastic) may not be rejected at a 5% level of significance in any of the eight cases corresponding to $\chi = 1$. Fisher's LSD *post hoc* test could therefore be used to determine between which pairs of iterations the GLPMI values are statistically distinguishable when employing various inconsistency parameter combinations. The purpose of the present section is, however, merely to investigate the impact of increased extents and degrees of inconsistency and varying update weights on the global learning ability of the DSS when incorporating the difference-based MACBETH method.

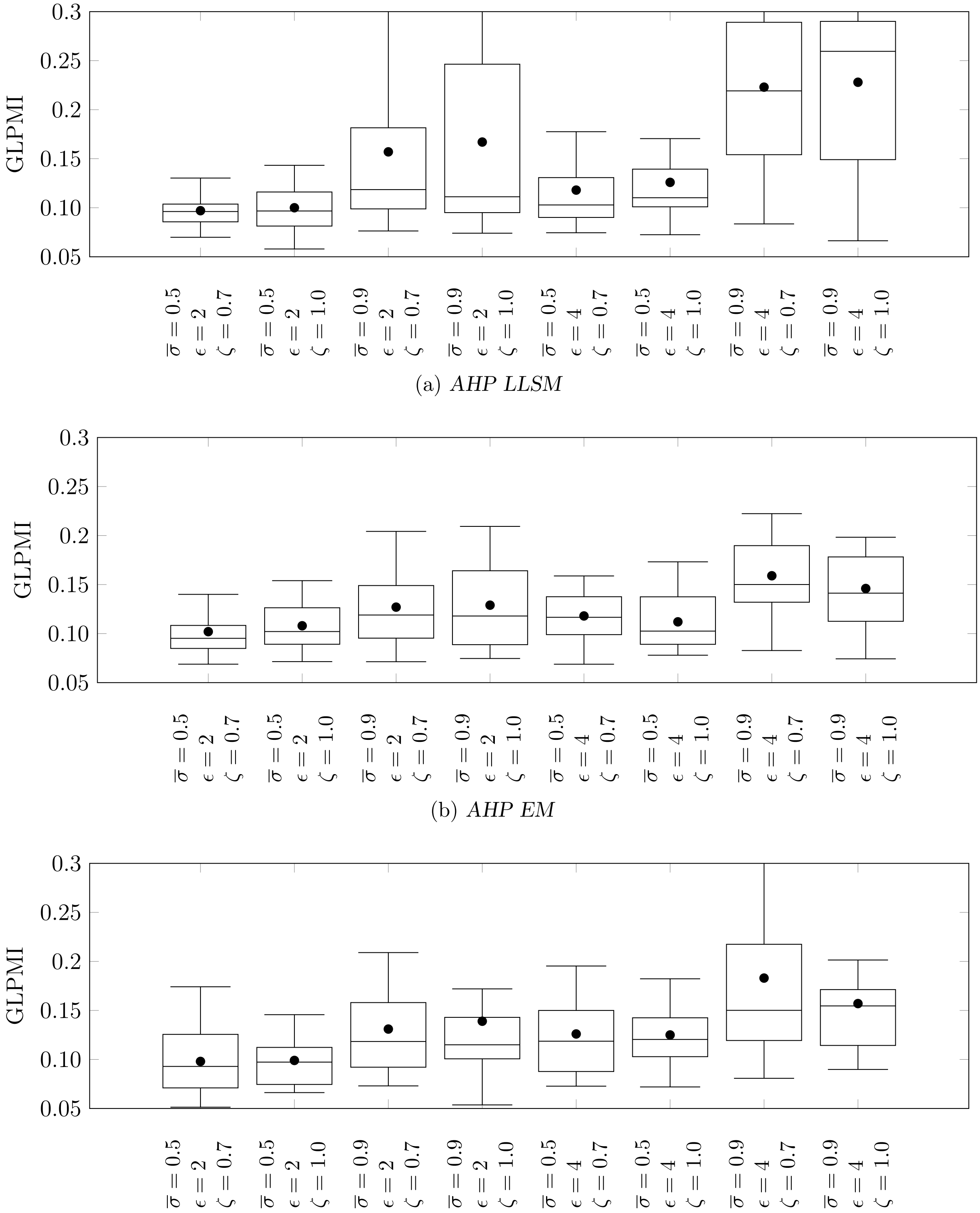
8.2.3 Discussion of results

It was found in §8.2.1 that the DSS of §5 is able to learn the DM's value function to some extent (according to the GLPMI) over the course of four iterations of the learning cycle for the power AHP judgement scale and LLSM combination in the presence of a small and medium extent of DM inconsistency ($\epsilon = 2$ and $\epsilon = 4$) combined with a small degree of inconsistency ($\overline{\sigma} = 0.5$) regardless of the weight update value, but the system failed to learn the DM's true value function when the inconsistency degree was increased to $\overline{\sigma} = 0.9$. It would appear that the GLPMI is, although still affected, less sensitive to variation in the extent of inconsistency than to variation in the degree of inconsistency exhibited by the DM. This may be seen by noting that the four sets of box plots in Figures 8.1(a) and (c) look rather similar, while the results in Figures 8.1(b) and (d) seem to be relatively unpredictable. Although the same conclusion may be drawn for the second set of experiments in this section, when the power AHP judgement scale is combined with the EM, it would appear that the EM performs better when subjected to a higher degree of inconsistency $\overline{\sigma} = 0.9$. The power AHP judgement scale and EM of aggregation combination performs slightly worse for the larger degree of inconsistency $\overline{\sigma} = 0.9$, but compared to the values of the power judgement scale and LLSM combination for this level of inconsistency, it performs significantly better. This may be seen when comparing Figures 8.2(b) and (d) with Figures 8.1(b) and (d).

The results reported in §8.2.2 for the difference-based MACBETH method in conjunction with the various extents and degrees of inconsistency, and update weights, were similar to the results for the ratio-based AHP method when combined with the power AHP judgement scale and EM of aggregation. The MACBETH approach appeared to be able to learn the true value function of the DM to some extent, but struggled to do so effectively at a large degree of inconsistency. This approach seemed to perform worse when subjected to both a large degree and extent of inconsistency ($\overline{\sigma} = 0.9$ and $\epsilon = 4$), as shown in Figure 8.3(d).

To examine the learning ability of the various parameter combinations for each of the three sets of experiments in §8.2, the GLPMI values during the fourth iteration of the learning cycle in each of the eight sets of experiments are analysed in the three cases of incorporating the ratio-based AHP method with the power AHP judgement scale in conjunction with the LLSM and the EM of aggregation, as well as for the case of employing the difference-based MACBETH method.

The GLPMI values for the first case, where the power AHP judgement scale is combined with the LLSM, is shown in Table 8.5. The corresponding ANOVA and Levene p-values are also included.



(a) *AHP LLSM*

(b) *AHP EM*

(c) *MACBETH*

FIGURE 8.4: *GLPMI samples associated with the fourth iteration of the learning cycle of Figure 5.1 when populating the DSS with the ratio-based AHP in conjunction with the LLSM and EM of aggregation as well as the difference-based MACBETH method in the context of various inconsistency and update parameter value combinations.*

	GLPMI mean									
$\overline{\sigma}$:	0.5	0.5	0.9	0.9	0.5	0.5	0.9	0.9		
ϵ :	2	2	2	2	4	4	4	4	ANOVA	Levene
ζ :	0.7	1.0	0.7	1.0	0.7	1.0	0.7	1.0	test p-values	test p-values
AHP LLSM	0.097	0.100	0.157	0.167	0.118	0.126	0.223	0.228	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
AHP EM	0.102	0.108	0.127	0.129	0.118	0.112	0.159	0.146	3.88×10^{-11}	9.23×10^{-06}
MACBETH	0.098	0.099	0.131	0.139	0.126	0.125	0.183	0.157	4.41×10^{-09}	1.00×10^{-05}

TABLE 8.5: *ANOVA and Levene test p-values for the GLPMI values after 4 iterations of the learning cycle in Figure 5.1 when populating the DSS of §5 with the power AHP judgement scale in conjunction with the LLSM of aggregation and employing $\chi = 1$ anchor at extents of inconsistency $\epsilon = 2$ and 4, degrees of inconsistency $\sigma = 0.5$ and 0.9, and update weights $\zeta = 0.7$ and 1.0.*

Based on the ANOVA p-value in Table 8.5, the relevant null-hypothesis (stating that the GLPMI means are statistically indistinguishable across all parameter combinations) may be rejected at a significance level of $\alpha = 5\%$. It furthermore follows from the Levene test p-values that the relevant null-hypothesis (stating that the GLPMI samples are homoscedastic) may be rejected at a 5% level of significance. The Games-Howell *post hoc* test is therefore used to determine between which pairs of parameter combinations the GLPMI values are statistically distinguishable in the case of combining the power AHP judgement scale with the LLSM of aggregation.

The Games-Howell *post hoc* test p-values in Table 8.6 indicate at a 5% level of significance that the GLPMI sample means are statistically indistinguishable for both the update weights $\zeta = 0.7$ and $\zeta = 1.0$ when adopting the power judgement scale and LLSM of aggregation. Due to this similarity in performance, only one of the two update weights may be considered when analysing the remaining Games-Howell test results in Table 8.6 and Figure 8.4(a). The p-values indicate that an increase in degree of inconsistency from $\sigma = 0.5$ to $\sigma = 0.9$ yields a difference in the ability of the framework to learn the underlying true value function of the DM. The same claim may be made when the extent of inconsistency $\epsilon = 2$ is increased to $\epsilon = 4$ in combination with $\overline{\sigma} = 0.5$, but not when increasing the value of ϵ in conjunction with $\overline{\sigma} = 0.9$.

The GLPMI values for the second case where the power AHP judgement scale is combined with the EM is shown in Table 8.5. The corresponding ANOVA and Levene p-values are also included in Table 8.5. Based on the ANOVA p-value, the relevant null-hypothesis (stating that the GLPMI means are statistically indistinguishable across all parameter combinations) may be rejected at a significance level of $\alpha = 5\%$. As in the case of incorporating the LLSM, it furthermore follows from the Levene test p-values that the relevant null-hypothesis (stating that the GLPMI samples are homoscedastic) may be rejected at a 5% level of significance. The Games-Howell *post hoc* test is therefore used to determine between which pairs of parameter combinations the GLPMI values are statistically distinguishable when the power AHP judgement scale in conjunction with the EM of aggregation.

The Games-Howell *post hoc* test p-values in Table 8.7 indicate at a 5% level of significance that the GLPMI sample means are again statistically indistinguishable for both update weights $\zeta = 0.7$ and $\zeta = 1.0$ when adopting the power judgement scale and EM of aggregation. Due to this similarity in performance, only one of the two update weights may again be considered when analysing the remaining Games-Howell test results in Table 8.7 and Figure 8.4(b). The p-values indicate that the increase in inconsistency degree from $\overline{\sigma} = 0.5$ to $\overline{\sigma} = 0.9$ yields no statistical difference in the GLPMI values when combined with the inconsistency extent $\epsilon = 2$, although this is not the case when increasing the inconsistency degree to $\overline{\sigma} = 0.9$ in combination with a large extent of inconsistency $\epsilon = 4$. In the case where the extent of inconsistency is increased

	$\zeta = 1.0$ $\epsilon = 2$ $\overline{\sigma} = 0.5$	$\zeta = 0.7$ $\epsilon = 2$ $\overline{\sigma} = 0.9$	$\zeta = 1.0$ $\epsilon = 2$ $\overline{\sigma} = 0.9$	$\zeta = 0.7$ $\epsilon = 4$ $\overline{\sigma} = 0.5$	$\zeta = 1.0$ $\epsilon = 4$ $\overline{\sigma} = 0.5$	$\zeta = 0.7$ $\epsilon = 4$ $\overline{\sigma} = 0.9$	$\zeta = 1.0$ $\epsilon = 4$ $\overline{\sigma} = 0.9$
$\zeta = 0.7$, $\epsilon = 2$, $\overline{\sigma} = 0.5$	9.97×10^{-01}	5.00×10^{-03}	6.00×10^{-03}	2.42×10^{-01}	4.30×10^{-02}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
$\zeta = 1.0$, $\epsilon = 2$, $\overline{\sigma} = 0.5$	—	1.10×10^{-02}	1.20×10^{-02}	5.22×10^{-01}	1.38×10^{-01}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
$\zeta = 0.7$, $\epsilon = 2$, $\overline{\sigma} = 0.9$	—	—	9.99×10^{-01}	2.67×10^{-01}	5.42×10^{-01}	3.50×10^{-02}	1.80×10^{-02}
$\zeta = 1.0$, $\epsilon = 2$, $\overline{\sigma} = 0.9$	—	—	—	1.78×10^{-01}	3.69×10^{-01}	1.98×10^{-01}	1.25×10^{-01}
$\zeta = 0.7$, $\epsilon = 4$, $\overline{\sigma} = 0.5$	—	—	—	—	9.98×10^{-01}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
$\zeta = 1.0$, $\epsilon = 4$, $\overline{\sigma} = 0.5$	—	—	—	—	—	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
$\zeta = 0.7$, $\epsilon = 4$, $\overline{\sigma} = 0.9$	—	—	—	—	—	—	9.99×10^{-01}

TABLE 8.6: *Games-Howell post hoc p-values for the GLPMI means in Table 8.5 resulting from the AHP and aggregation by the LLSM.*

	$\zeta = 1.0$ $\epsilon = 2$ $\overline{\sigma} = 0.5$	$\zeta = 0.7$ $\epsilon = 2$ $\overline{\sigma} = 0.9$	$\zeta = 1.0$ $\epsilon = 2$ $\overline{\sigma} = 0.9$	$\zeta = 0.7$ $\epsilon = 4$ $\overline{\sigma} = 0.5$	$\zeta = 1.0$ $\epsilon = 4$ $\overline{\sigma} = 0.5$	$\zeta = 0.7$ $\epsilon = 4$ $\overline{\sigma} = 0.9$	$\zeta = 1.0$ $\epsilon = 4$ $\overline{\sigma} = 0.9$
$\zeta = 0.7$, $\epsilon = 2$, $\overline{\sigma} = 0.5$	9.87×10^{-01}	7.70×10^{-02}	8.70×10^{-02}	2.33×10^{-01}	8.11×10^{-01}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
$\zeta = 1.0$, $\epsilon = 2$, $\overline{\sigma} = 0.5$	—	3.19×10^{-01}	3.15×10^{-01}	7.67×10^{-01}	9.98×10^{-01}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
$\zeta = 0.7$, $\epsilon = 2$, $\overline{\sigma} = 0.9$	—	—	9.99×10^{-01}	9.62×10^{-01}	6.42×10^{-01}	5.20×10^{-02}	5.34×10^{-01}
$\zeta = 1.0$, $\epsilon = 2$, $\overline{\sigma} = 0.9$	—	—	—	9.38×10^{-01}	6.10×10^{-01}	1.15×10^{-01}	7.19×10^{-01}
$\zeta = 0.7$, $\epsilon = 4$, $\overline{\sigma} = 0.5$	—	—	—	—	9.84×10^{-01}	$< 1 \times 10^{-17}$	2.23×10^{-02}
$\zeta = 1.0$, $\epsilon = 4$, $\overline{\sigma} = 0.5$	—	—	—	—	—	$< 1 \times 10^{-17}$	3.00×10^{-03}
$\zeta = 0.7$, $\epsilon = 4$, $\overline{\sigma} = 0.9$	—	—	—	—	—	—	8.63×10^{-01}

TABLE 8.7: *Games-Howell post hoc p-values for the GLPMI means in Table 8.5 resulting from the AHP and aggregation by the EM.*

	$\zeta = 1.0$ $\epsilon = 2$ $\overline{\sigma} = 0.5$	$\zeta = 0.7$ $\epsilon = 2$ $\overline{\sigma} = 0.9$	$\zeta = 1.0$ $\epsilon = 2$ $\overline{\sigma} = 0.9$	$\zeta = 0.7$ $\epsilon = 4$ $\overline{\sigma} = 0.5$	$\zeta = 1.0$ $\epsilon = 4$ $\overline{\sigma} = 0.5$	$\zeta = 0.7$ $\epsilon = 4$ $\overline{\sigma} = 0.9$	$\zeta = 1.0$ $\epsilon = 4$ $\overline{\sigma} = 0.9$
$\zeta = 0.7$, $\epsilon = 2$, $\overline{\sigma} = 0.5$	9.99×10^{-01}	4.60×10^{-02}	9.60×10^{-02}	1.23×10^{-01}	3.50×10^{-02}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
$\zeta = 1.0$, $\epsilon = 2$, $\overline{\sigma} = 0.5$	—	4.10×10^{-02}	9.60×10^{-02}	1.13×10^{-01}	2.60×10^{-02}	$< 1 \times 10^{-17}$	$< 1 \times 10^{-17}$
$\zeta = 0.7$, $\epsilon = 2$, $\overline{\sigma} = 0.9$	—	—	9.99×10^{-01}	9.99×10^{-01}	9.99×10^{-01}	1.24×10^{-01}	4.39×10^{-01}
$\zeta = 1.0$, $\epsilon = 2$, $\overline{\sigma} = 0.9$	—	—	—	9.85×10^{-01}	9.75×10^{-01}	4.47×10^{-01}	9.47×10^{-01}
$\zeta = 0.7$, $\epsilon = 4$, $\overline{\sigma} = 0.5$	—	—	—	—	9.99×10^{-01}	6.40×10^{-02}	2.11×10^{-01}
$\zeta = 1.0$, $\epsilon = 4$, $\overline{\sigma} = 0.5$	—	—	—	—	—	4.70×10^{-02}	1.21×10^{-01}
$\zeta = 0.7$, $\epsilon = 4$, $\overline{\sigma} = 0.9$	—	—	—	—	—	—	8.92×10^{-01}

TABLE 8.8: *Games-Howell post hoc p-values for the GLPMI means in Table 8.5 resulting from the difference-based MACBETH method.*

to $\epsilon = 4$, the p-values are larger than 5% and it may not be claimed with 95% confidence that the sample means are statistically indistinguishable in combination when either $\overline{\sigma} = 0.5$ or $\overline{\sigma} = 0.9$. It appears that the AHP in conjunction with the EM of aggregation performs better when inconsistency is introduced into the framework of this dissertation. This combination of the ratio-based AHP method is, therefore, employed in the remainder of this chapter when its learning ability is compared with that of the difference-based MACBETH method.

The final set of fourth-iteration GLPMI experiments are analysed for the difference-based approach. The relevant GLPMI values are shown in Table 8.5 and these are accompanied by the relevant ANOVA and Levene p-values. The ANOVA p-value indicates at 95% confidence that the null-hypothesis (stating that the GLPMI means are statistically indistinguishable across all parameter combinations experiments) may be rejected and it furthermore follows from the Levene test p-values that the relevant null-hypothesis (stating that the GLPMI samples are homoscedastic) may be rejected at a 5% level of significance. The Games-Howell *post hoc* test is once again used for the difference-based MACBETH approach.

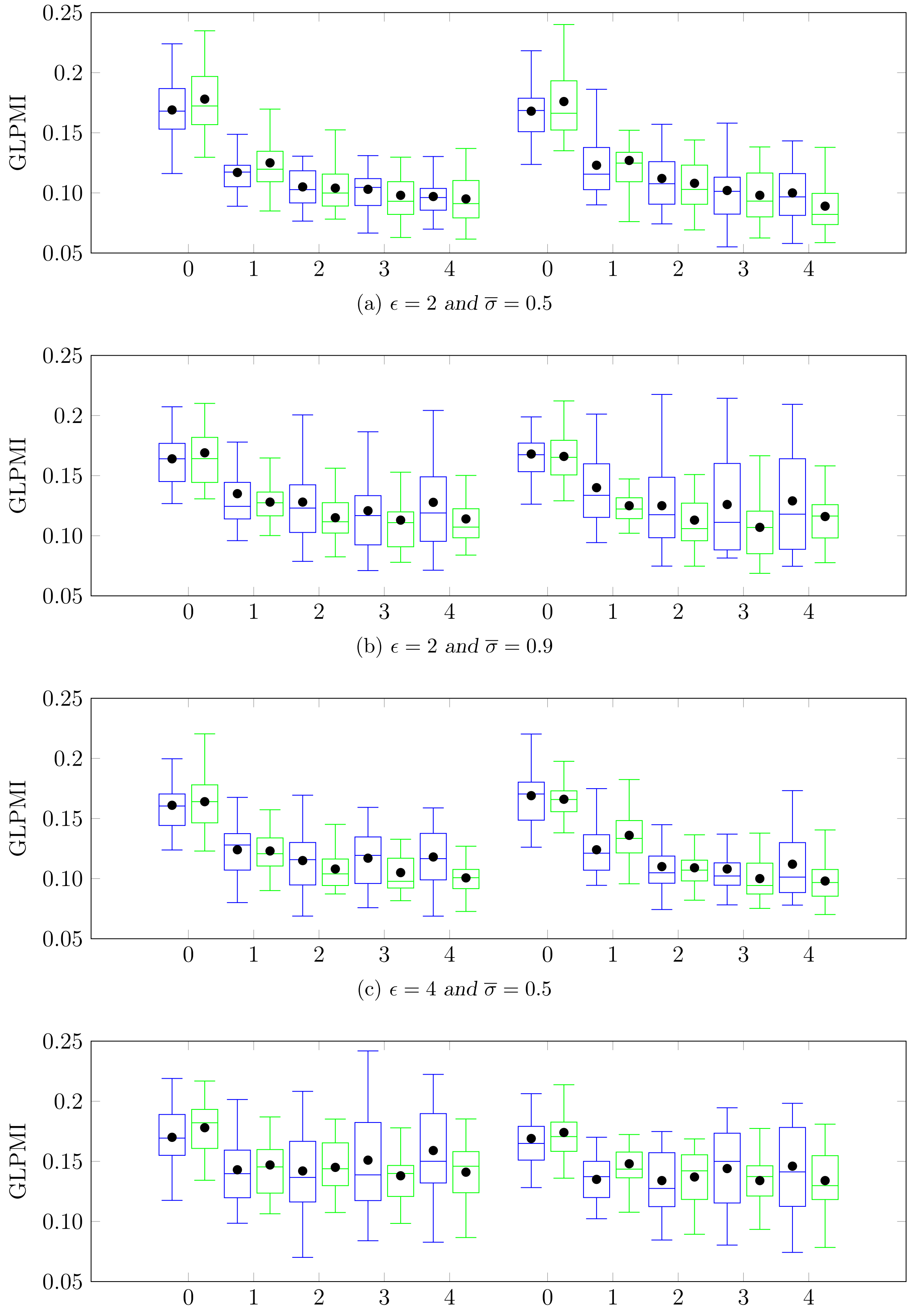
Similar to the result for the aforementioned AHP ratio-based approach in combination with the LLSM and EM of aggregation, the Games-Howell *post hoc* test p-values in Table 8.8 indicate at a 5% level of significance that the GLPMI sample means are statistically indistinguishable for both update weights $\zeta = 0.7$ and $\zeta = 1.0$. This corresponds to and confirms the results of the sensitivity analysis of §7.5 for the weight update parameter. In terms of the increase in inconsistency degree $\overline{\sigma}$ and extent of inconsistency ϵ, it appears that the only parameter increase that leads to GLPMI values which may be claimed to be statistically distinguishable is when the degree of inconsistency is increased from $\overline{\sigma} = 0.5$ to $\overline{\sigma} = 0.9$ in combination of the inconsistency extent of $\epsilon = 2$.

8.3 The effect of anchor set cardinality on DM inconsistency

The purpose of this section is to a assess the potential of increased learning effectiveness when accommodating DM inconsistency by increasing the number of anchors in the anchor set. More specifically, it is investigated whether there is any benefit in increasing the anchor set cardinality in terms of learning the DM's value function in the presence of various extents and degrees of DM inconsistency. This analysis is also carried out in respect of the GLPMI as learning performance indicator during the fourth iteration of the learning cycle in Figure 5.1. Only two of the three representative methods previously considered are considered in this section, namely the power AHP judgement scale in combination with the EM of aggregation when incorporating the ratio-based AHP, and the MACBETH method when incorporating the difference-based approach.

8.3.1 The effect of increasing anchor set cardinality on the AHP

The results of the set of experiments reported and interpreted in this section corresponds to the power AHP judgement scale and EM of aggregation combination mentioned in §8.1.2. In this set, the effectiveness with which the DSS of §5 can learn the DM's underlying value function is assessed in eight experiments, corresponding to the various combinations of employing $\chi = 1$ or $\chi = 2$ anchors at degrees of inconsistency of $\overline{\sigma} = 0.5$ or $\overline{\sigma} = 0.9$ and extents of inconsistency $\epsilon = 2$ or $\epsilon = 4$, for weight update parameter values $\zeta = 0.7$ or $\zeta = 1.0$. Learning effectiveness is measured in according to the GLPMI of (6.3) after four iterations of the learning cycle in Figure 5.1. Figure 8.5 contains box plots of the GLPMI means associated with four iterations



(a) $\epsilon = 2$ and $\overline{\sigma} = 0.5$

(b) $\epsilon = 2$ and $\overline{\sigma} = 0.9$

(c) $\epsilon = 4$ and $\overline{\sigma} = 0.5$

(d) $\epsilon = 4$ and $\overline{\sigma} = 0.9$

FIGURE 8.5: *GLPMI values associated with four iterations of the learning cycle in Figure 5.1 when populating the DSS of §5 with the power AHP judgement scale in conjunction with the EM of aggregation for eight combinations of inconsistency parameters when incorporating $\chi = 1$ anchor (blue plots) or $\chi = 2$ anchors (green plots).*

	GLPMI		ANOVA	Levene
	1 Anchor	2 Anchors	p-value	p-value
$\zeta = 0.7$, $\epsilon = 2$, $\overline{\sigma} = 0.5$	0.102	0.095	2.21×10^{-01}	5.34×10^{-01}
$\zeta = 1.0$, $\epsilon = 2$, $\overline{\sigma} = 0.5$	0.108	0.089	2.37×10^{-03}	1.45×10^{-01}
$\zeta = 0.7$, $\epsilon = 2$, $\overline{\sigma} = 0.9$	0.127	0.114	1.18×10^{-01}	3.92×10^{-03}
$\zeta = 1.0$, $\epsilon = 2$, $\overline{\sigma} = 0.9$	0.129	0.116	1.31×10^{-01}	8.09×10^{-07}
$\zeta = 0.7$, $\epsilon = 4$, $\overline{\sigma} = 0.5$	0.118	0.101	1.96×10^{-03}	6.13×10^{-03}
$\zeta = 1.0$, $\epsilon = 4$, $\overline{\sigma} = 0.5$	0.112	0.098	2.00×10^{-02}	1.35×10^{-03}
$\zeta = 0.7$, $\epsilon = 4$, $\overline{\sigma} = 0.9$	0.159	0.141	3.88×10^{-02}	6.87×10^{-03}
$\zeta = 1.0$, $\epsilon = 4$, $\overline{\sigma} = 0.9$	0.146	0.134	1.40×10^{-01}	1.03×10^{-02}

TABLE 8.9: *GLPMI means associated with the fourth iteration of the learning cycle when populating the DSS framework of §5 with the power AHP judgement scale in conjunction with the EM, together with the relevant p-values for the ANOVA and Levene statistical tests for $\chi = 1$ anchor and for $\chi = 2$ anchors. Table entries smaller than [larger than, respectively] 0.05 are typeset in black [red, respectively], show that the relevant probability of a Type I error is small [large, respectively] and hence that the null-hypothesis may be rejected [may not be rejected, respectively].*

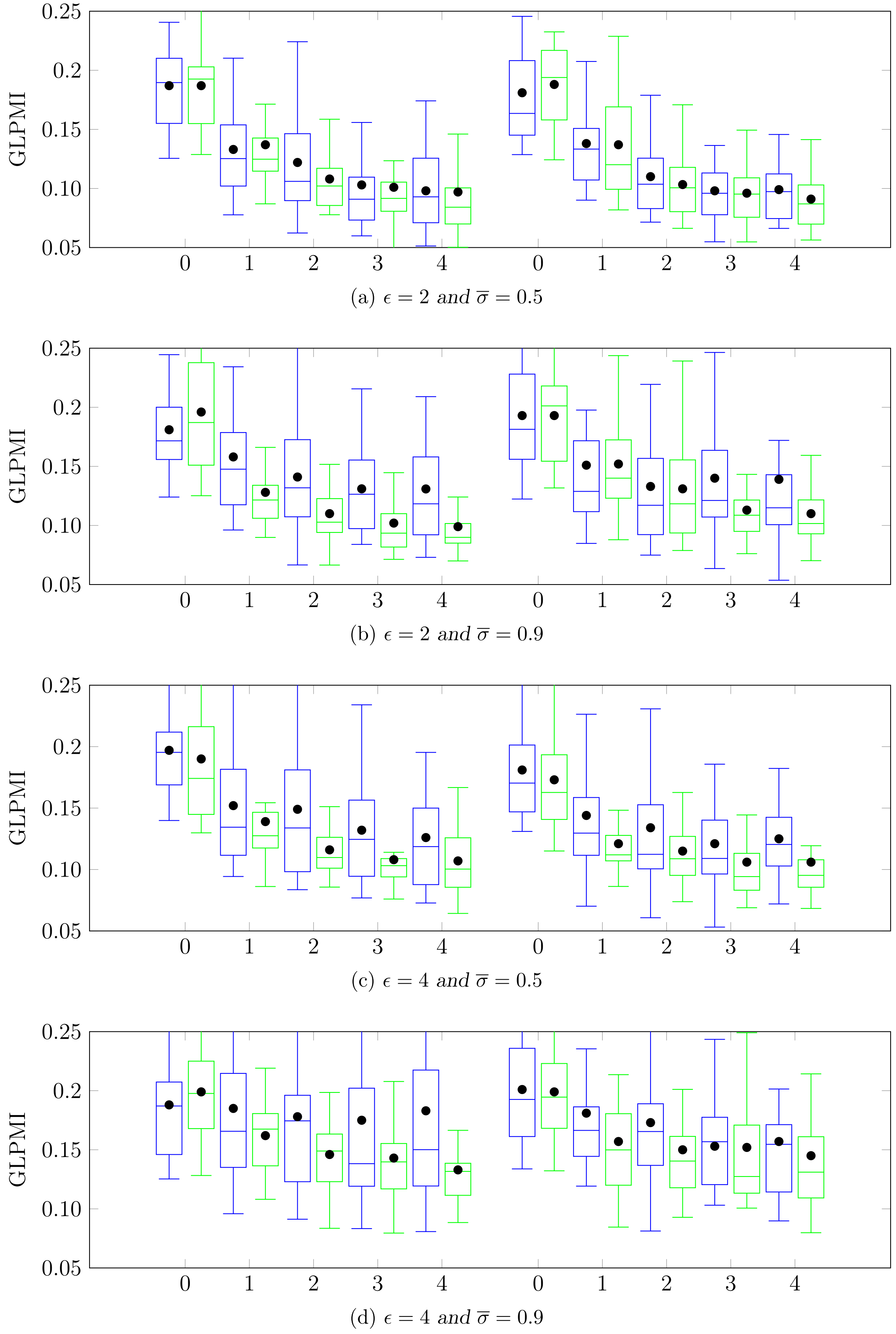
of the learning cycle of Figure 5.1 for the power judgement scale in conjunction with the EM when employing $\chi = 1$ or $\chi = 2$ anchors at a degree of DM inconsistency of $\overline{\sigma} = 0.5$ or $\overline{\sigma} = 0.9$ and extents of inconsistency $\epsilon = 2$ or $\epsilon = 4$ for update weight parameter values of $\zeta = 0.7$ or $\zeta = 1.0$. The ANOVA and Levene test p-values are also provided after the fourth iteration of each experiment for $\chi = 1$ and $\chi = 2$ in Table 8.9.

It follows from the ANOVA p-values in Table 8.9 that the relevant null-hypothesis (stating that the GLPMI means are statistically indistinguishable across anchor set cardinalities) may be rejected at a significance level of $\alpha = 5\%$ in the case of an inconsistency degree of $\overline{\sigma} = 0.5$ and an inconsistency extent $\epsilon = 2$ for a weight update value $\zeta = 1.0$, in the case of $\overline{\sigma} = 0.5$ and $\epsilon = 4$ (for both update weight values $\zeta = 0.7$ and 1.0) and in the case of $overline\sigma = 0.9$ and $\epsilon = 4$ for the weight update parameter $\zeta = 0.7$. It may therefore be claimed with 95% confidence that increasing the number of anchors employed does affect the ability of the DSS of §5 to learn the DM's value function in these cases. The same claim cannot, however, be made for the remaining extents and degrees of inconsistency.

It furthermore follows from the Levene test p-values that the relevant null-hypothesis (stating that the GLPMI samples are homoscedastic) may be rejected at a 5% level of significance in all cases, except for the smallest degree and extent of inconsistency combination $\overline{\sigma} = 0.5$ and $\epsilon = 2$. Based on a visual inspection of Figures 8.5(a)–(d), it may therefore be stated with 95% confidence that with an increase in anchor set cardinality, the GLPMI means perform better or remain statistically indistinguishable when populating the DSS of §5 with the power judgement scale in conjunction with the EM.

8.3.2 The effect of increasing anchor set cardinality on the MACBETH method

This section contains a presentation of the results of eight further experiments, this time corresponding to the difference-based method. In each set of experiments, the effectiveness with which the DSS of §5 can predict the DM's underlying true value function is assessed in these experiments after four iterations of the learning cycle in Figure 5.1, corresponding to the various combinations of employing $\chi = 1$ or $\chi = 2$ anchors at degrees of inconsistency of $\overline{\sigma} = 0.5$ or $\overline{\sigma} = 0.9$ and extents of inconsistency $\epsilon = 2$ or $\epsilon = 4$, for two values of update weight values $\zeta = 0.7$ or $\zeta = 1.0$. Figure 8.6 contains box plots of the GLPMI means associated with four iterations



(a) $\epsilon = 2$ *and* $\overline{\sigma} = 0.5$

(b) $\epsilon = 2$ *and* $\overline{\sigma} = 0.9$

(c) $\epsilon = 4$ *and* $\overline{\sigma} = 0.5$

(d) $\epsilon = 4$ *and* $\overline{\sigma} = 0.9$

FIGURE 8.6: *GLPMI values associated with four iterations of the learning cycle in Figure 5.1 when populating the DSS of §5 with the difference-based MACBETH method for eight combinations of inconsistency parameters when incorporating* $\chi = 1$ *anchor (blue plots) or* $\chi = 2$ *anchors (green plots).*

	GLPMI		ANOVA	Levene
	1 Anchor	2 Anchors	p-value	p-value
$\zeta = 0.7$, $\epsilon = 2$, $\overline{\sigma} = 0.5$	0.098	0.097	9.24×10^{-01}	8.06×10^{-01}
$\zeta = 1.0$, $\epsilon = 2$, $\overline{\sigma} = 0.5$	0.099	0.091	2.55×10^{-01}	4.96×10^{-01}
$\zeta = 0.7$, $\epsilon = 2$, $\overline{\sigma} = 0.9$	0.131	0.099	1.93×10^{-03}	2.51×10^{-03}
$\zeta = 1.0$, $\epsilon = 2$, $\overline{\sigma} = 0.9$	0.139	0.110	3.46×10^{-02}	2.83×10^{-03}
$\zeta = 0.7$, $\epsilon = 4$, $\overline{\sigma} = 0.5$	0.126	0.107	4.65×10^{-02}	2.35×10^{-02}
$\zeta = 1.0$, $\epsilon = 4$, $\overline{\sigma} = 0.5$	0.125	0.106	3.90×10^{-02}	9.77×10^{-01}
$\zeta = 0.7$, $\epsilon = 4$, $\overline{\sigma} = 0.9$	0.183	0.133	6.89×10^{-03}	4.11×10^{-04}
$\zeta = 1.0$, $\epsilon = 4$, $\overline{\sigma} = 0.9$	0.157	0.145	3.57×10^{-01}	8.31×10^{-01}

Table 8.10: *GLPMI means associated with the fourth iteration of the learning cycle when populating the DSS framework of §5 with the MACBETH method, together with the relevant p-values for the ANOVA and Levene statistical tests for $\chi = 1$ anchor and for $\chi = 2$ anchors. Table entries smaller than [larger than, respectively] 0.05 are typeset in black [red, respectively], show that the relevant probability of a Type I error is small [large, respectively] and hence that the null-hypothesis may be rejected [may not be rejected, respectively].*

of the learning cycle of Figure 5.1 for the difference-based MACBETH method when employing $\chi = 1$ or $\chi = 2$ anchors at various degrees and extents of DM inconsistency for two weight update values. The corresponding ANOVA and Levene test p-values are provided in Table 8.10.

It follows from the ANOVA p-values in Table 8.10 that the relevant null-hypothesis (stating that the GLPMI means are statistically indistinguishable across both anchor set cardinalities) may be rejected at a significance level of $\alpha = 5\%$ in the case of an inconsistency extent of $\epsilon = 2$ and inconsistency degree of $\overline{\sigma} = 0.9$ for both update weight values $\zeta = 0.7$ and $\zeta = 1.0$, as well as in the case of an inconsistency extent $\epsilon = 4$ and an inconsistency degree $\overline{\sigma} = 0.5$, also for both update weight values $\zeta = 0.7$ and $\zeta = 1.0$, but only for the weight update value of $\zeta = 0.7$ when the largest inconsistency degree $\sigma = 0.9$ and inconsistency extent $\epsilon = 4$ are incorporated. It may therefore be claimed with 95% confidence that an increase in the number of anchors employed does not affect the ability of the DSS of §5 to learn the DM's true value function (to some extent) in the case of increasing DM inconsistency. The same claim cannot, however, be made if the DM exhibits a very large extent and degree of inconsistency when the updating parameter is chosen as $\zeta = 1.0$ in the case of incorporating the difference-based MACBETH method.

It furthermore follows from the Levene test p-values that the relevant null-hypothesis (stating that the GLPMI samples are homoscedastic) may also be rejected at a 5% level of significance in the same cases, except in the case where $\overline{\sigma} = 0.5$, $\epsilon = 4$ and $\zeta = 1.0$. Based on a visual inspection of Figures 8.6(a)–(d), it may therefore be claimed with 95% confidence that with an increase in anchor set cardinality, the GLPMI means either perform better or remain statistically indistinguishable when populating the DSS of §5 with the difference-based MACBETH method.

8.3.3 Discussion of results

When considering the box plots in Figures 8.5 and 8.6, it would appear that the power AHP judgement scale in conjunction with the LLSM is capable of facilitating better learning of the DM's value function than the difference-based MACBETH method in the presence of DM inconsistency, although as more anchors are employed in the DSS of §5 it appears that the performance difference between employing $\chi = 1$ and $\chi = 2$ anchors is slightly more significant for the MACBETH method than for the ratio-based AHP approach. Although the incorporation of multiple anchors in the DM comparison sets seem to hold no benefit for a perfectly consistent DM, as expected, the GLMPI means in Tables 8.9 and 8.10 decrease as more anchors are employed (*i.e.*

as the value of χ increases). Whereas this phenomenon is encouraging, it seems that once a large extent and degree of inconsistency is reached ($\overline{\sigma} = 0.9$ and $\epsilon = 4$), the increased number of anchors seems to have less effect. This may be a result of the GLPMI formula in (6.3) not being sensitive enough to large deviations between value function estimates and the DM's true value function, as explained in §8.2.3.

8.4 Chapter summary

The purpose of this chapter was to assess the effects on the iterative learning ability of the DSS in §5 of introducing small to medium levels of DM inconsistency. After describing the method in which DM inconsistency was modelled and the experimental setup of the aforementioned assessment in §8.1, the assessment itself was carried out in §8.2.1 for the ratio-based AHP. Not all judgement scale and aggregation method combinations of §7 were, however, considered in the assessments of this chapter. In particular, the effects of DM inconsistency on the GLPMI in (6.3) were assessed in §8.2.1 for two representative combinations of AHP judgement scale and aggregation methods. The entire assessment was then repeated in §8.2.2 for the difference-based MACBETH method.

The focus next shifted in §8.3 to a study of the potential of increased effectiveness in accommodating DM inconsistency by increasing the number of anchors in the anchor set. More specifically, the investigation centred around whether there is any benefit in terms of learning the DM's value function when increasing the anchor set cardinality in the presence of various levels and amounts of DM inconsistency. This analysis was carried out in respect of both the best performing representative AHP judgement scale and aggregation method (in §8.3.1) and the difference-based MACBETH method (in §8.3.2).

The main findings resulting from the numerical experiments conducted in this chapter were the following:

- The DSS of §5 is still capable of learning the DM's value function globally, even if small or medium levels of DM inconsistency are introduced.
- As the level of DM inconsistency increases, the quality of value function estimation deteriorates marginally.
- The EM is more adept at handling DM inconsistency (when increasing the anchor set size) than the LLSM if the goal is to learn the DM's value function globally.
- The power AHP judgement scale in combination with the EM and the difference-based MACBETH method are capable of handling DM inconsistency (when increasing the anchor set size).

CHAPTER 9

Real Case Study

The purpose of this chapter is to document two case studies carried out by the author in conjunction with two human subjects as a more realistic validation of the DSS proposed in Chapter 5 than those carried out in Chapters 7 and 8. Both case studies are related to the CSP — the first (in §9.1) essentially being a collection of 2DCSP instances (similar to that considered in Chapters 6–8), and the second (in §9.2) being an instance of the full 3DCSP. The chapter finally closes with a brief summary of the case study findings in §9.3.

9.1 2D colour selection case study

A case study was conducted in respect of the 2DCSP introduced in §6, but involving a real DM instead of modelling the DM (as in Chapters 7 and 8). The case study was carried out on September 20th, 2017 with Mr JC van der Walt [215] as consensual subject (hereafter referred to as "the DM" in the remainder of this section). The experimental setup of and results obtained during this case study are presented in this section.

9.1.1 Experimental setup

The case study involved asking the DM to experiment unaided with the full three-dimensional RGB colour cube in order to identify what he perceived to be the most suitable colour for painting a bedroom. The DM was asked not to disclose this favourite colour to the author, but he was instructed to keep this colour in mind when comparing alternatives presented to him during the iterative learning process described in §5. He was furthermore asked to base his pairwise comparisons on the degree of similarity between the various alternatives presented to him and his (undisclosed) favourite colour in a bid to remain consistent. The DM was requested to stick to his favourite colour even if, during the learning process, he were by chance to be presented with more beautiful colours than he had considered during his original exploration of the decision space.

Although essentially an instance of the full 3DCSP, it was decided to conduct this case study in the form of four instances of the 2DCSP solved in parallel, each instance representing a slice of the three-dimensional RGB colour cube corresponding to a constant value of the blue RGB coordinate. This decision was taken so as to render the case study reminiscent of the 2DCSP instances considered in previous chapters (there a slice corresponding to the value red = 0 was considered throughout) and also to ensure that (slices of) the DM's value function estimate can be represented graphically. Four equidistant two-dimensional slices of the RGB cube were considered in this case study, corresponding to planes within the cube for which blue = 0, blue = 85, blue = 170 and blue = 255, respectively. These slices give rise to four independent instances of the 2DCSP which were solved concurrently in real-time consultation with the DM. The best colour alternatives thus found for each instance were finally combined by means of a final 4×4 matrix of pairwise comparisons performed by the DM in respect of representative colours from each 2DCSP (or RGB slice) instance in order to identify an overall most pleasing colour.

Only $\chi = 1$ anchor was employed during each of the four parallel 2DCSP instances, namely the colours $(126, 59, 0)$ (🟫), $(182, 232, 85)$ (🟩), $(211, 32, 170)$ (🟪) and $(211, 32, 255)$ (🟪), respectively. A total of $\omega = 2$ iterations of the learning cycle of Figure 5.1 were carried out in respect of the four 2DCSP instances, every time presenting the DM with a comparison set comprising $n = 8$ colours (of which seven were fresh each time, and one was the anchor). The GUI in Figure 9.1 was used to present colours to the DM, upon which he was asked to express $\binom{8}{2} = 28$ pairwise comparison preferences during each iteration of the respective learning cycles according to the power AHP judgement scale in Table 6.1. The EM was employed to compute value function scores for the alternatives. Value functions were then estimated from these scores by means of piecewise linear interpolation.

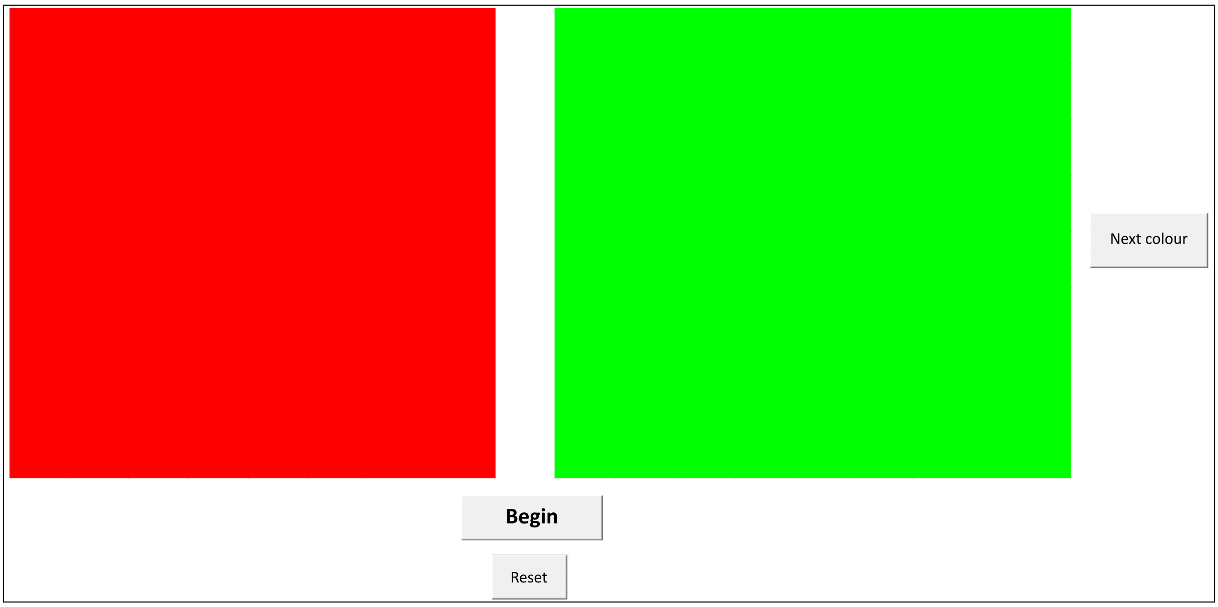


FIGURE 9.1: *Screen shot of the GUI developed by the author for the purpose of presenting pairs of colours to the DM for comparison.*

The *Exploit* method of exploitation/exploration, described in §5.3.1 and §7.4, was used for the generation of new alternatives during each of the learning iterations. As many of the exponentially decreasing exploitation radii $r_1 = 120$, $r_2 = 64$, $r_3 = 34$ and $r_4 = 18$ as required were employed in each of the parallel 2DCSP instances in order to favour exploitation of the decision space increasingly towards the end of the learning process after having focussed on exploration of the decision spaces earlier on during the learning process. Of the seven fresh colour alternatives generated during iteration $i \in \{1, 2\}$ of each of the four parallel learning cycles, four alternatives were *exploitation alternatives* (sampled from *within* a radius r_i around the global maximum of the current value function estimate), one was the relevant anchor and the remaining three were *exploration alternatives* (sampled from *outside* a radius r_i around the global maximum of the current value function estimate). All new alternatives were sampled according to a uniform distribution over their respective (exploration or exploitation) domains.

9.1.2 Case study progression and results

The DM was presented with the four comparison sets during Iteration 0 of the learning cycle, containing the colour alternatives in Table 9.1, eight for each of the four parallel (and independent) 2DCSP instances mentioned in §9.1.1. All of these alternatives were initially exploration points of each square decision space (for each respective blue RGB coordinate value), one of which also doubled as anchor. The roles of the alternatives are also shown in Table 9.1.

After considering the alternatives in each of these comparison sets separately, the DM returned the pairwise comparison matrices in Figures A.1–A.4 of Appendix A upon which the value function scores in Table 9.1 were computed by means of the EM. Linear interpolation of these score values yielded the four parallel value function slice estimates during Iteration 0 of the learning cycle shown in Figure 9.2.

From the four value functions in Figure 9.2 the four new sets of alternatives in Table 9.2 were generated by exploiting the value function area within a radius of $r_1 = 120$ of the local maxima of $(255, 255, 0)$ (🟨), $(0, 255, 85)$ (🟩), $(0, 0, 170)$ (🟦) and $(145, 255, 255)$ (🟦), respectively, and exploring the decision space by randomly sampling fresh colours outside of these exploitation radii. These four new sets of alternatives each consisted of four exploitation alternatives, three exploration alternatives and the relevant anchor, and are shown in Table 9.2.

After considering these new comparison sets, the DM returned the pairwise comparison matrices in Figures A.5–A.8 of Appendix A upon which the new value function scores in Table 9.2 were computed by means of the EM. The score values (in Table 9.1) of all alternatives previously considered during Iteration 0 were brought in line with the new score values (in Table 9.2) by means of the anchor colour's value score for Iteration 1 as explained in §5.4 using an update weight of $\zeta = 1.0$. Linear interpolation of the (updated) score values of the combined comparison sets of Iterations 0 and 1 then yielded the four value function slice estimates shown in Figure 9.3 after application of piecewise linear interpolation of these points.

The learning cycle was performed once more and the final sets of newly generated alternatives generated are shown in Table 9.3 together with each alternative's role. Upon considering these alternatives, the DM returned the final pairwise comparison matrices in Figures A.9–A.12 of Appendix A. The value function scores computed by means of the EM are again shown in Table 9.3. These values were yet again used to adjust all previous score values as described in §5.4 adopting an update weight of $\zeta = 1.0$ after which linear interpolation was applied to the (updated) score values of the entire archive of alternatives considered during all the iterations to yield the final value function slice estimates shown in Figure 9.4.

9.1.3 Recommendation and discussion

The maxima of the four value function slice estimates in Figure 5.4 are $(52, 161, 0)$ (🟩), $(44, 130, 85)$ (🟩), $(88, 179, 170)$ (🟩) and $(153, 196, 255)$ (🟦), respectively. In order to decide which of these alternatives to present to the DM as final colour recommendation, the anchors

Alt	RGB Coordinates	Colour	Role	Score	Alt	RGB Coordinates	Colour	Role	Score
$\boldsymbol{x}_{0,1}^{0}$	$(248, 254, 0)$	🟨	Exploration	0.763	$\boldsymbol{x}_{0,2}^{0}$	$(23, 236, 0)$	🟩	Exploration	0.016
$\boldsymbol{x}_{0,3}^{0}$	$(8, 142, 0)$	🟩	Exploration	0.223	$\boldsymbol{x}_{0,4}^{0}$	$(178, 111, 0)$	🟧	Exploration	0.295
$\boldsymbol{x}_{0,5}^{0}$	$(78, 60, 0)$	🟫	Exploration	0.244	$\boldsymbol{x}_{0,6}^{0}$	$(52, 98, 0)$	🟩	Exploration	0.231
$\boldsymbol{x}_{0,7}^{0}$	$(54, 14, 0)$	🟫	Exploration	0.203	$\boldsymbol{a}^{0}$	$(126, 59, 0)$	🟫	Anchor	1.000

(a) Value function estimates for the comparison set $\mathcal{C}_0^{\text{blue}=0}$

Alt	RGB Coordinates	Colour	Role	Score	Alt	RGB Coordinates	Colour	Role	Score
$\boldsymbol{x}_{0,1}^{85}$	$(128, 255, 85)$	🟩	Exploration	1.590	$\boldsymbol{x}_{0,2}^{85}$	$(111, 27, 85)$	🟪	Exploration	3.329
$\boldsymbol{x}_{0,3}^{85}$	$(212, 37, 85)$	🟥	Exploration	0.392	$\boldsymbol{x}_{0,4}^{85}$	$(78, 95, 85)$	⬛	Exploration	76.677
$\boldsymbol{x}_{0,5}^{85}$	$(98, 129, 85)$	🟩	Exploration	23.611	$\boldsymbol{x}_{0,6}^{85}$	$(122, 234, 85)$	🟩	Exploration	1.269
$\boldsymbol{x}_{0,7}^{85}$	$(242, 35, 85)$	🟥	Exploration	0.272	$\boldsymbol{a}^{85}$	$(182, 232, 85)$	🟩	Anchor	1.000

(b) Value function estimates for the comparison set $\mathcal{C}_0^{\text{blue}=85}$

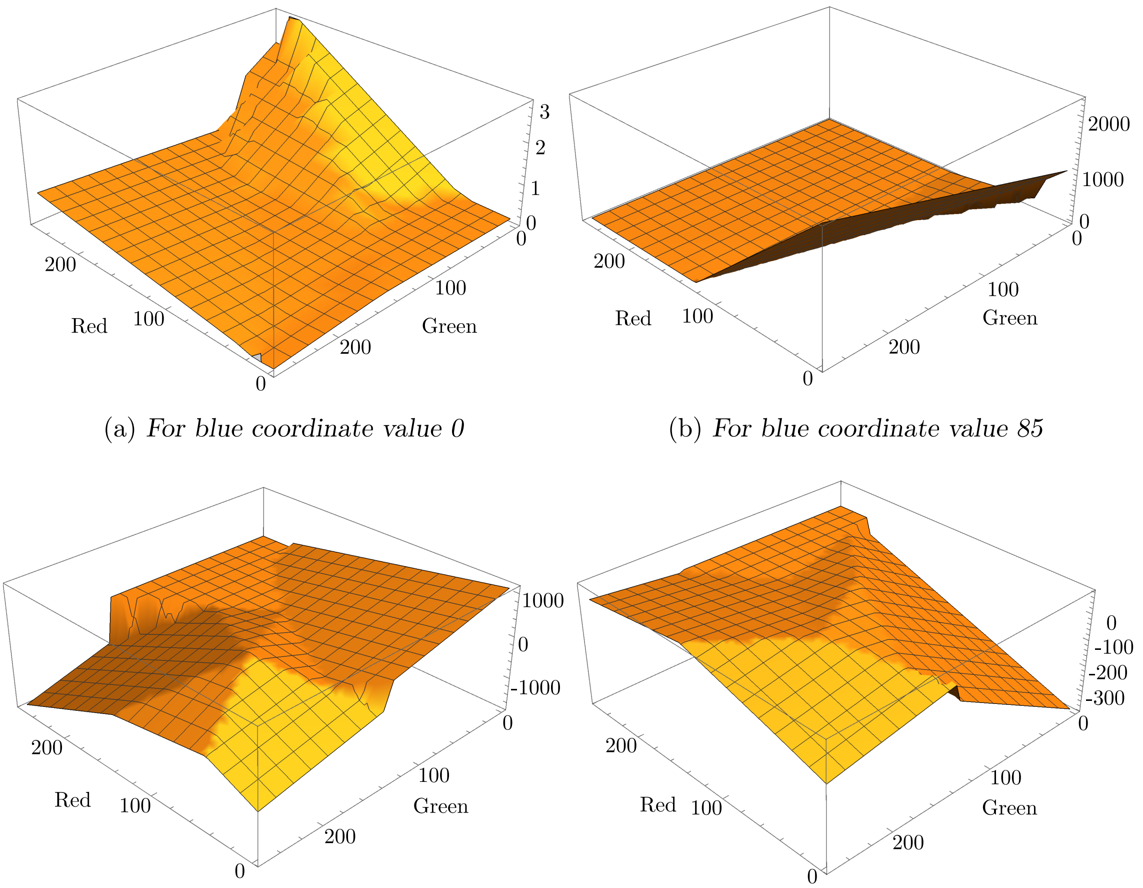
Alt	RGB Coordinates	Colour	Role	Score	Alt	RGB Coordinates	Colour	Role	Score
$\boldsymbol{x}_{0,1}^{170}$	$(60, 232, 170)$	🟩	Exploration	19.792	$\boldsymbol{x}_{0,2}^{170}$	$(115, 215, 170)$	🟩	Exploration	32.901
$\boldsymbol{x}_{0,3}^{170}$	$(88, 179, 170)$	🟦	Exploration	582.159	$\boldsymbol{x}_{0,4}^{170}$	$(35, 152, 170)$	🟦	Exploration	154.775
$\boldsymbol{x}_{0,5}^{170}$	$(185, 121, 170)$	🟪	Exploration	5.811	$\boldsymbol{x}_{0,6}^{170}$	$(143, 89, 170)$	🟪	Exploration	3.858
$\boldsymbol{x}_{0,7}^{170}$	$(232, 55, 170)$	🟪	Exploration	0.971	$\boldsymbol{a}^{170}$	$(211, 32, 170)$	🟪	Anchor	1.000

(c) Value function estimates for the comparison set $\mathcal{C}_0^{\text{blue}=170}$

Alt	RGB Coordinates	Colour	Role	Score	Alt	RGB Coordinates	Colour	Role	Score
$\boldsymbol{x}_{0,1}^{255}$	$(60, 232, 255)$	🟦	Exploration	1.8511	$\boldsymbol{x}_{0,2}^{255}$	$(115, 215, 255)$	🟦	Exploration	76.075
$\boldsymbol{x}_{0,3}^{255}$	$(88, 179, 255)$	🟦	Exploration	72.646	$\boldsymbol{x}_{0,4}^{255}$	$(35, 152, 255)$	🟦	Exploration	28.792
$\boldsymbol{x}_{0,5}^{255}$	$(185, 121, 255)$	🟪	Exploration	0.974	$\boldsymbol{x}_{0,6}^{255}$	$(143, 89, 255)$	🟪	Exploration	36.343
$\boldsymbol{x}_{0,7}^{255}$	$(232, 55, 255)$	🟪	Exploration	0.261	$\boldsymbol{a}^{255}$	$(211, 32, 255)$	🟪	Anchor	1.000

(d) Value function estimates for the comparison set $\mathcal{C}_0^{\text{blue}=255}$

Table 9.1: *Alternatives considered and (unscaled) value function scores computed during Iteration 0 of the learning cycle in the case study of §9.1. The four comparison matrices returned by the DM during this iteration may be found in Figures A.1–A.4 of Appendix A. Exploration alternatives were generated randomly within the relevant square RGB slice in each case.*



(a) *For blue coordinate value 0*

(b) *For blue coordinate value 85*

(c) *For blue coordinate value 170*

(d) *For blue coordinate value 255*

FIGURE 9.2: *Estimates of four representative value function slices corresponding to blue coordinates* 0, 85, 170 *and* 255*, respectively, in the RGB colour scheme for Iteration 0 of the learning cycle in the case study of §9.1. These estimates were found by applying linear interpolation to the value function scores in Table 9.1. The global maxima of these value function slice estimates occur at (a)* $(255, 255, 0)$ *(*🟨*), (b)* $(0, 255, 85)$ *(*🟩*), (c)* $(0, 0, 170)$ *(*🟦*) and (d)* $(145, 255, 255)$ *(*🟦*).*

Alt	RGB Coordinates	Colour	Role	Score	Alt	RGB Coordinates	Colour	Role	Score
$\boldsymbol{x}_{1,1}^{0}$	$(30, 112, 0)$	■	Exploitation	10.184	$\boldsymbol{x}_{1,2}^{0}$	$(142, 135, 0)$	■	Exploitation	1.338
$\boldsymbol{x}_{1,3}^{0}$	$(138, 44, 0)$	■	Exploitation	3.988	$\boldsymbol{x}_{1,4}^{0}$	$(168, 206, 0)$	■	Exploitation	1.240
$\boldsymbol{x}_{1,5}^{0}$	$(141, 219, 0)$	■	Exploration	0.204	$\boldsymbol{x}_{1,6}^{0}$	$(253, 195, 0)$	■	Exploration	0.307
$\boldsymbol{x}_{1,7}^{0}$	$(244, 185, 0)$	■	Exploration	0.307	$\boldsymbol{a}^{0}$	$(126, 59, 0)$	■	Anchor	1.000

(a) Value function estimates for the comparison set $\mathcal{C}_1^{\text{blue}=0}$

Alt	RGB Coordinates	Colour	Role	Score	Alt	RGB Coordinates	Colour	Role	Score
$\boldsymbol{x}_{1,1}^{85}$	$(254, 213, 85)$	■	Exploitation	2.973	$\boldsymbol{x}_{1,2}^{85}$	$(44, 130, 85)$	■	Exploitation	155.862
$\boldsymbol{x}_{1,3}^{85}$	$(192, 72, 85)$	■	Exploitation	1.782	$\boldsymbol{x}_{1,4}^{85}$	$(8, 255, 85)$	■	Exploitation	0.461
$\boldsymbol{x}_{1,5}^{85}$	$(78, 186, 85)$	■	Exploration	8.777	$\boldsymbol{x}_{1,6}^{85}$	$(0, 188, 85)$	■	Exploration	6.626
$\boldsymbol{x}_{1,7}^{85}$	$(39, 238, 85)$	■	Exploration	0.742	$\boldsymbol{a}^{85}$	$(182, 232, 85)$	■	Anchor	1.000

(b) Value function estimates for the comparison set $\mathcal{C}_1^{\text{blue}=85}$

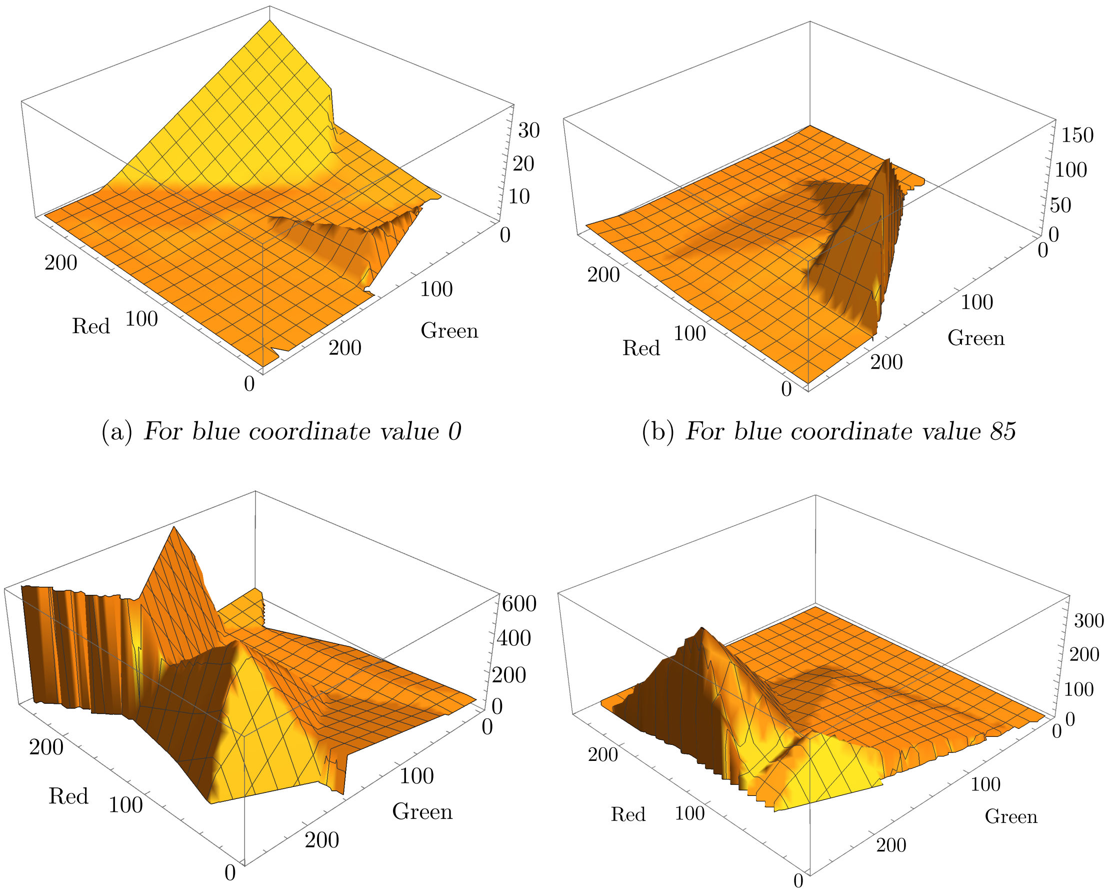
Alt	RGB Coordinates	Colour	Role	Score	Alt	RGB Coordinates	Colour	Role	Score
$\boldsymbol{x}_{1,1}^{170}$	$(183, 91, 170)$	■	Exploitation	4.209	$\boldsymbol{x}_{1,2}^{170}$	$(234, 57, 170)$	■	Exploitation	1.122
$\boldsymbol{x}_{1,3}^{170}$	$(157, 189, 170)$	■	Exploitation	296.443	$\boldsymbol{x}_{1,4}^{170}$	$(43, 27, 170)$	■	Exploitation	71.542
$\boldsymbol{x}_{1,5}^{170}$	$(11, 38, 170)$	■	Exploration	71.542	$\boldsymbol{x}_{1,6}^{170}$	$(11, 1, 170)$	■	Exploration	57.146
$\boldsymbol{x}_{1,7}^{170}$	$(94, 71, 170)$	■	Exploration	11.688	$\boldsymbol{a}^{170}$	$(211, 32, 170)$	■	Anchor	1.000

(c) Value function estimates for the comparison set $\mathcal{C}_1^{\text{blue}=170}$

Alt	RGB Coordinates	Colour	Role	Score	Alt	RGB Coordinates	Colour	Role	Score
$\boldsymbol{x}_{1,1}^{255}$	$(174, 65, 255)$	■	Exploitation	1.075	$\boldsymbol{x}_{1,2}^{255}$	$(6, 34, 255)$	■	Exploitation	2.886
$\boldsymbol{x}_{1,3}^{255}$	$(35, 145, 255)$	■	Exploitation	61.375	$\boldsymbol{x}_{1,4}^{255}$	$(72, 177, 255)$	■	Exploitation	128.831
$\boldsymbol{x}_{1,5}^{255}$	$(144, 221, 255)$	■	Exploration	354.635	$\boldsymbol{x}_{1,6}^{255}$	$(241, 238, 255)$	■	Exploration	12.831
$\boldsymbol{x}_{1,7}^{255}$	$(91, 233, 255)$	■	Exploration	14.738	$\boldsymbol{a}^{255}$	$(211, 32, 255)$	■	Anchor	1.000

(d) Value function estimates for the comparison set $\mathcal{C}_1^{\text{blue}=255}$

TABLE 9.2: *Alternatives considered and (unscaled) value function scores computed during Iteration 1 of the learning cycle in the case study of §9.1. The four comparison matrices returned by the DM during this iteration may be found in Figures A.5–A.8 of Appendix A. Exploitation alternatives were generated randomly within a circle with a radius of 120 around the global optima (a)* $(255, 255, 0)$ *(■), (b)* $(0, 255, 85)$ *(■), (c)* $(0, 0, 170)$ *(■) and (d)* $(145, 255, 255)$ *(■) of the iteration value function estimates in Figure 9.2 and within the relevant square RGB slice in each case. The exploration alternatives were generated randomly outside this radius but within the relevant square RGB slice in each case.*



(a) *For blue coordinate value 0*

(b) *For blue coordinate value 85*

(c) *For blue coordinate value 170*

(d) *For blue coordinate value 255*

FIGURE 9.3: *Estimates of four representative value function slices corresponding to blue coordinates* 0, 85, 170 *and* 255*, respectively, in the RGB colour scheme for Iteration 1 of the learning cycle in the case study of §9.1. These estimates were found by applying linear interpolation to the updated value function score in Table 9.1 together with the new value function scores in Table 9.2. The global maxima of these value function slice estimates occur at (a)* $(255, 0, 0)$ *(🟥), (b)* $(44, 130, 85)$ *(🟩), (c)* $(255, 246, 170)$ *(🟨) and (d)* $(144, 221, 255)$ *(🟦).*

Alt	RGB Coordinates	Colour	Role	Score	Alt	RGB Coordinates	Colour	Role	Score
$\boldsymbol{x}_{2,1}^{0}$	$(163, 64, 0)$	🟫	Exploitation	0.693	$\boldsymbol{x}_{2,2}^{0}$	$(62, 185, 0)$	🟩	Exploitation	0.333
$\boldsymbol{x}_{2,3}^{0}$	$(169, 100, 0)$	🟧	Exploitation	0.480	$\boldsymbol{x}_{2,4}^{0}$	$(202, 21, 0)$	🟥	Exploitation	0.101
$\boldsymbol{x}_{2,5}^{0}$	$(215, 45, 0)$	🟥	Exploration	0.160	$\boldsymbol{x}_{2,6}^{0}$	$(207, 31, 0)$	🟥	Exploration	0.127
$\boldsymbol{x}_{2,7}^{0}$	$(222, 46, 0)$	🟥	Exploration	0.160	$\boldsymbol{a}^{0}$	$(126, 59, 0)$	🟫	Anchor	1.000

(a) Value function estimates for the comparison set $\mathcal{C}_2^{\text{blue}=0}$

Alt	RGB Coordinates	Colour	Role	Score	Alt	RGB Coordinates	Colour	Role	Score
$\boldsymbol{x}_{2,1}^{85}$	$(46, 197, 85)$	🟩	Exploitation	2.894	$\boldsymbol{x}_{2,2}^{85}$	$(201, 249, 85)$	🟩	Exploitation	0.594
$\boldsymbol{x}_{2,3}^{85}$	$(232, 60, 85)$	🟥	Exploitation	0.168	$\boldsymbol{x}_{2,4}^{85}$	$(45, 148, 85)$	🟩	Exploitation	13.761
$\boldsymbol{x}_{2,5}^{85}$	$(99, 160, 85)$	🟩	Exploration	40.537	$\boldsymbol{x}_{2,6}^{85}$	$(13, 138, 85)$	🟩	Exploration	32.055
$\boldsymbol{x}_{2,7}^{85}$	$(88, 133, 85)$	🟩	Exploration	23.401	$\boldsymbol{a}^{85}$	$(182, 232, 85)$	🟩	Anchor	1.000

(b) Value function estimates for the comparison set $\mathcal{C}_2^{\text{blue}=85}$

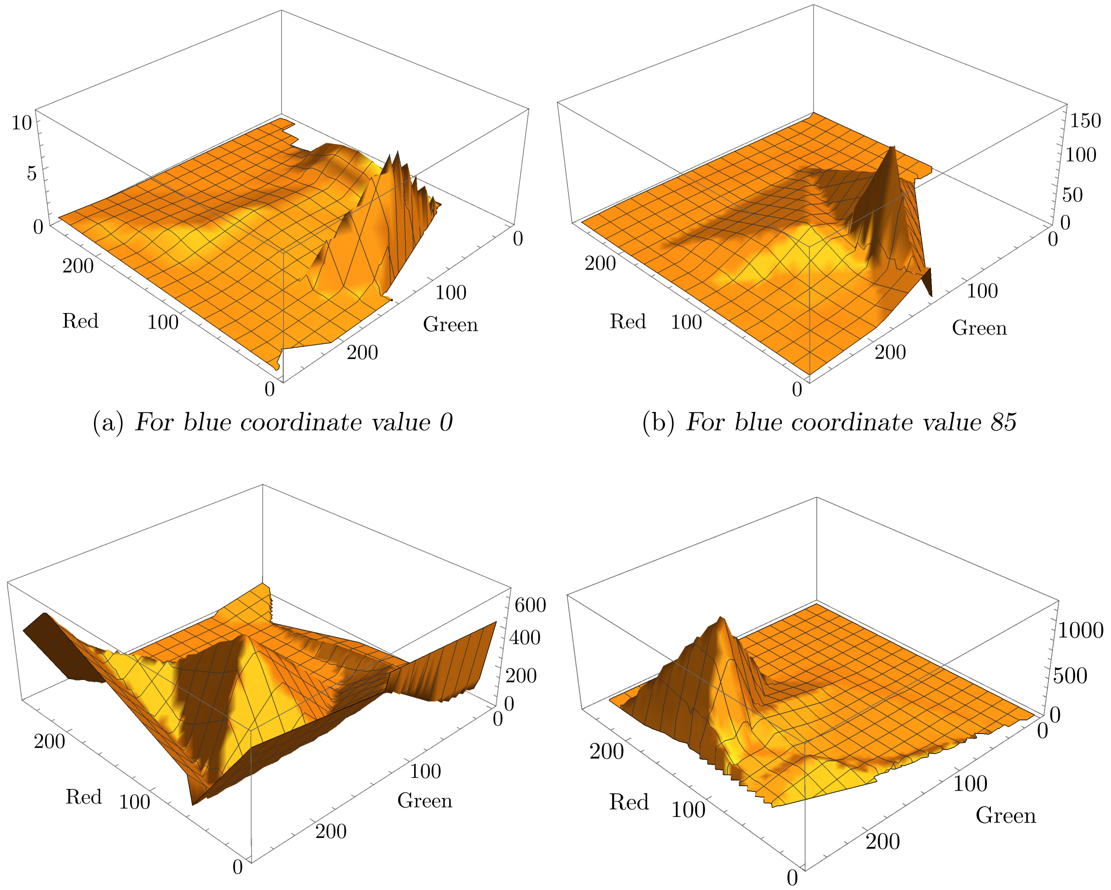
Alt	RGB Coordinates	Colour	Role	Score	Alt	RGB Coordinates	Colour	Role	Score
$\boldsymbol{x}_{2,1}^{170}$	$(65, 40, 170)$	🟦	Exploitation	155.718	$\boldsymbol{x}_{2,2}^{170}$	$(103, 54, 170)$	🟪	Exploitation	7.345
$\boldsymbol{x}_{2,3}^{170}$	$(3, 123, 170)$	🟦	Exploitation	380.261	$\boldsymbol{x}_{2,4}^{170}$	$(215, 199, 170)$	🟫	Exploitation	45.742
$\boldsymbol{x}_{2,5}^{170}$	$(248, 186, 170)$	🟧	Exploration	6.232	$\boldsymbol{x}_{2,6}^{170}$	$(248, 214, 170)$	🟨	Exploration	17.174
$\boldsymbol{x}_{2,7}^{170}$	$(211, 229, 170)$	🟩	Exploration	344.782	$\boldsymbol{a}^{170}$	$(211, 32, 170)$	🟪	Anchor	1.00

(c) Value function estimates for the comparison set $\mathcal{C}_2^{\text{blue}=170}$

Alt	RGB Coordinates	Colour	Role	Score	Alt	RGB Coordinates	Colour	Role	Score
$\boldsymbol{x}_{2,1}^{255}$	$(37, 80, 255)$	🟦	Exploitation	2.427	$\boldsymbol{x}_{2,2}^{255}$	$(68, 160, 255)$	🟦	Exploitation	20.672
$\boldsymbol{x}_{2,3}^{255}$	$(40, 124, 255)$	🟦	Exploitation	6.680	$\boldsymbol{x}_{2,4}^{255}$	$(153, 196, 255)$	🟦	Exploitation	1230.440
$\boldsymbol{x}_{2,5}^{255}$	$(134, 185, 255)$	🟦	Exploration	248.704	$\boldsymbol{x}_{2,6}^{255}$	$(110, 224, 255)$	🟦	Exploration	9.649
$\boldsymbol{x}_{2,7}^{255}$	$(120, 199, 255)$	🟦	Exploration	68.487	$\boldsymbol{a}^{255}$	$(211, 32, 255)$	🟪	Anchor	1.000

(d) Value function estimates for the comparison set $\mathcal{C}_2^{\text{blue}=255}$

TABLE 9.3: *Alternatives considered and (unscaled) value function scores computed during Iteration 2 of the learning cycle in the case study of §9.1. The four comparison matrices returned by the DM during this iteration may be found in Figures A.9–A.12 of Appendix A. Exploitation alternatives were generated randomly within a circle with a radius of 64 around the global optima (a)* $(255, 0, 0)$ (🟥)*, (b)* $(44, 130, 85)$ (🟩)*, (c)* $(255, 246, 170)$ (🟨) *and (d)* $(144, 221, 255)$ (🟦) *of the iteration value function estimates in Figure 9.3 and within the relevant square RGB slice in each case. The exploration alternatives were generated randomly outside this radius but within the relevant square RGB slice in each case.*



(a) *For blue coordinate value 0*

(b) *For blue coordinate value 85*

(c) *For blue coordinate value 170*

(d) *For blue coordinate value 255*

FIGURE 9.4: *Estimates of four representative value function slices corresponding to blue coordinates* 0, 85, 170 *and* 255, *respectively, in the RGB colour scheme for Iteration 2 of the learning cycle in the case study of §9.1. These estimates were found by applying linear interpolation to the updated value function scores in Tables 9.1 and 9.2 together with the new value function scores in Table 9.3. The global maxima of these value function slice estimates occur at (a)* $(52, 161, 0)$ *(🟩), (b)* $(44, 130, 85)$ *(🟩), (c)* $(88, 179, 170)$ *(🟦) and (d)* $(153, 196, 255)$ *(🟦).*

$(126, 55, 0)$ (🟫), $(182, 232, 85)$ (🟩), $(211, 32, 170)$ (🟪) and $(211, 32, 255)$ (🟪) of the respective slices were finally presented to the DM, who performed the final $\binom{4}{2} = 6$ pairwise comparisons encapsulated in the matrix

$$\boldsymbol{A}^{anchors} = \begin{array}{c} \\ \boldsymbol{a}^{0} \\ \boldsymbol{a}^{85} \\ \boldsymbol{a}^{170} \\ \boldsymbol{a}^{255} \end{array} \begin{array}{c} \begin{array}{cccc} \boldsymbol{a}^{0} & \boldsymbol{a}^{85} & \boldsymbol{a}^{170} & \boldsymbol{a}^{255} \end{array} \\ \left[\begin{array}{cccc} 1 & \frac{1}{25} & \frac{1}{64} & \frac{1}{81} \\ 25 & 1 & \frac{1}{64} & \frac{1}{81} \\ 64 & 64 & 1 & \frac{1}{36} \\ 81 & 81 & 36 & 1 \end{array}\right] \end{array}.$$

The normalised eigenvector $[0.0023, 0.0116, 0.1452, 1.0000]^T$ corresponding to the eigenvalue with largest modulus of this matrix prompted the author to present the colour $(153, 196, 255)$ (🟦) to the DM as final recommendation.

Upon receiving this recommendation, the DM revealed that he had originally decided on the alternative $(140, 190, 255)$ (🟦) as his "favourite" colour. The Euclidean distance between the recommended colour and that actually selected by him *a priori* is a mere 14.317. These colours are so close to each other within the RGB cube that it is difficult to distinguish between them with the human eye. The DM was pleased with the recommendation and, in fact, somewhat surprised that the author could get so close to his "favourite" colour. The DM, however, revealed that during the process of carrying out pairwise comparisons over the course of the experiment, a number of beautiful colours were presented to him that he had not considered during his initial private exploration of the RGB colour cube — hence the aforementioned use of quotation marks when referring to the DM's favourite colour.

The above experimental outcome led to the following conclusions:

1. By even employing only one anchor, the DSS proposed in this dissertation is able to uncover decision alternatives that are indeed pleasing to a human DM exhibiting some level of inconsistency in terms of subjective preference within the context of colour selection.

2. The DSS proposed in this dissertation is capable of leading a DM with relatively little effort on the part of the DM to regions of the decision space that would otherwise not have been considered, hence alleviating to some extent the detrimental effect of satisficing on the quality of decision making in the context of colour selection.

In closing, it is claimed that the pure coincidence of choosing a 2DCSP slice at the start of the experiment actually containing the DM's initially identified favourite colour exactly (*i.e.* selecting a slice for which blue = 255) does not take away from the experiment conducted in this case study. This claim is substantiated by the fact that "good" alternatives identified in the adjacent colour slice (for which blue = 177) were also close to the DM's originally identified colour from a visually discerning point of view. Had more than four slices been considered, alternatives that are visually indistinguishable from the DM's initial selection would almost certainly have been identified in colour slices neighbouring the slice for which blue = 255, although this would have come at an additional comparison burden on the part of the DM.

9.2 3D colour selection case study

A second case study was conducted on October 16th, 2018 in respect of the full 3DCSP, again involving a real DM. This time the subject was Mr M du Plessis [58] (hereafter referred to as "the DM" in the remainder of this section). In contrast to the previous case study where the ratio-based AHP was employed, the difference-based MACBETH method is employed here. The experimental setup of and results obtained during this case study are presented in this section.

9.2.1 Experimental setup

The DM was provided with the same full three-dimensional RGB colour cube as was provided to the DM in §9.1 and he was again requested to identify a preferred colour. The DM was asked not to disclose this favourite colour to the author. In contrast to the case study of §9.1, the DM was allowed to stray from his initial favourite colour if, per chance, presented with a more desirable alternative during the pairwise comparison process of some iteration of the learning cycle described in §5. In this way the DM was instructed to maintain a favourite colour throughout the experiment which could be updated (perhaps even multiple times), initially taking the colour identified in private *a priori*. The DM was instructed to aim to be consistent by basing his pairwise comparisons on the degree of similarities between the various alternatives presented to him and his current "favourite" colour. A single anchor was again employed in this case study, namely the colour $(0, 255, 0)$ (🟩) and ten alternatives were considered (including the anchor alternative) during each iteration. The 3D colour cube was considered in its entirety, containing all 16 777 216 colour alternatives. Consequently, the estimated value function curves constructed during each learning iteration cannot be represented graphically and therefore only the estimated value function values are provided for each alternative. A total of $\omega = 3$ iterations of the learning cycle in Figure 5.1 were performed, each time presenting the DM with a comparison comprising $n = 10$ alternatives (of which nine were fresh each time). Initially the DM was presented with ten colours and asked to rank them in reducing attractiveness. The same GUI shown in Figure 9.1 was then used to present alternatives to the DM, upon which he was asked to express $\binom{10}{2} = 45$ pairwise comparison preferences during each iteration of the learning cycle according to the difference-based MACBETH judgement scale. Value functions were estimated from these scores by means of piecewise linear interpolation, this time over the full three-dimensional decision space.

The *Exploit* method of exploitation/exploration, described in §6.2.1, was again employed for the generation of new alternatives, making use of the exponentially decreasing exploitation radii $r_1 = 120$, $r_2 = 64$ and $r_3 = 34$ at the start of iterations 1, 2, and 3, respectively, in order to favour exploitation of the decision space increasingly towards the end of the learning process. Of the nine fresh colour alternatives generated during each iteration of the learning cycle, four were *exploitation alternatives* (sampled from *within* a radius r_i around the global maximum of the current value function estimate during Iteration $i \in \{1, 2, 3\}$). The remaining alternatives were the anchor value and five *exploration alternatives* (sampled from *outside* a radius r_i around the global maximum of the current value function estimate). All new alternatives were sampled according to uniform distributions over their respective (exploration or exploitation) domain.

9.2.2 Case study progression and results

The DM was initially presented with the ten colour alternatives in Table 9.4, which he ranked according to decreasing aesthetic preference. Pairwise comparisons were then requested from him, upon which he returned the pairwise comparison matrix shown in Figure A.13. Of the first ten alternatives presented to the DM, eight were the corner points of the cubic decision space, while the remaining two were generating randomly according to a uniform distribution. In all cases, one of these corner point choices (the alternative $(0, 255, 0)$ (🟩)) doubled as anchor. The (unscaled) value function scores in Table 9.4 were computed according to the MACBETH method explained in §3.2.2. The *incoherency value* of the DM in (3.16) was calculated as $c = 0.005$. The DM's value function was thereafter estimated by piecewise linear interpolation of these score values in four-dimensional (decision, score)-space. The global maximum of this function estimate was achieved at the alternative $(208, 0, 245)$ (🟪).

The comparison sets presented to the DM during the first iteration are shown in Table 9.5. The table shows the alternatives ranked according to the DM's preference. For these alternatives the DM returned the pairwise comparison matrix in Figure A.14 of Appendix A. The value function scores shown in Table 9.5 were computed next, after which the scores of the alternatives previously considered during Iteration 0 were brought in line with the new score values considered during Iteration 1, as described in §5.4, adopting an update weight of $\zeta = 1.0$. The *incoherency value* of the DM in (3.16) during Iteration 1 was calculated as $c = 0.01$, showing a slight increase from the previous iteration. This was followed by another estimation of the DM's value function by piecewise linear interpolation of the (partially updated) score values of the entire archive in four-dimensional (decision, score)-space. The global maximum of these function estimates was achieved at the alternative $(87, 167, 216)$ (🟦).

Alt	RGB Coordinates	Colour	Role	Score
$\boldsymbol{x}_{0,1}$	$(0, 255, 255)$	🟦	Corner point	6.532
$\boldsymbol{x}_{0,2}$	$(29, 240, 223)$	🟦	Exploration	5.527
$\boldsymbol{x}_{0,3}$	$(0, 0, 255)$	🟦	Corner point	5.025
$\boldsymbol{a}_{1}$	$(0, 255, 0)$	🟩	Anchor	4.315
$\boldsymbol{x}_{0,4}$	$(53, 255, 16)$	🟩	Exploration	4.020
$\boldsymbol{x}_{0,5}$	$(255, 255, 0)$	🟨	Corner point	3.015
$\boldsymbol{x}_{0,6}$	$(255, 0, 0)$	🟥	Corner point	2.512
$\boldsymbol{x}_{0,7}$	$(255, 255, 255)$		Corner point	1.507
$\boldsymbol{x}_{0,8}$	$(0, 0, 0)$	⬛	Corner point	0.502
$\boldsymbol{x}_{0,9}$	$(255, 0, 255)$	🟪	Corner point	0.000

Table 9.4: Alternatives considered and (unscaled) value function scores computed during Iteration 0 of the learning cycle in the case study of §9.2. The comparison matrix returned by the DM during this iteration may be found in Figure A.13 of Appendix A. The exploration alternatives were generated randomly within the RGB cube.

Alt	RGB Coordinates	Colour	Role	Score
$\boldsymbol{x}_{1,1}$	$(83, 198, 245)$	🟦	Exploitation	10.010
$\boldsymbol{x}_{1,2}$	$(62, 163, 21)$	🟩	Exploitation	10.00
$\boldsymbol{x}_{1,3}$	$(82, 220, 236)$	🟦	Exploitation	8.010
$\boldsymbol{x}_{1,4}$	$(66, 232, 245)$	🟦	Exploitation	8.000
$\boldsymbol{x}_{1,5}$	$(85, 233, 166)$	🟩	Exploration	7.000
$\boldsymbol{x}_{1,6}$	$(218, 162, 214)$	🟪	Exploration	6.000
$\boldsymbol{x}_{1,7}$	$(244, 100, 105)$	🟥	Exploration	5.000
$\boldsymbol{x}_{1,8}$	$(223, 57, 114)$	🟥	Exploration	3.000
$\boldsymbol{a}_{1}$	$(0, 255, 0)$	🟩	Anchor	1.000
$\boldsymbol{x}_{1,9}$	$(93, 13, 183)$	🟪	Exploration	0.000

TABLE 9.5: *Alternatives considered and (unscaled) value function scores computed during Iteration* 1 *of the learning cycle in the case study of §9.2. The comparison matrix returned by the DM during this iteration may be found in Figure A.14 of Appendix A. The exploration alternatives were generated randomly within the RGB cube.*

Thereafter, another iteration of the learning process was carried out exactly as described above. The comparison set presented to the DM during this iteration is given in Table 9.6 (ranked according to the DM's preference), for which the DM returned the pairwise comparison matrix in Figure A.15 of Appendix A. After computing function score values from this matrix by means of the MACBETH method and updating the alternatives considered during previous iterations, as described in §5.4, again employing an update weight $\zeta = 1.0$, new estimates of the DM's value function was computed for the experiment. The global maximum of this function estimate was achieved at the alternative $(84, 194, 243)$ (🟦). The *incoherency value* of the DM in (3.16) during Iteration 2 was calculated as $c = 0.506$, showing a significant increase in inconsistency from the previous iteration.

A final iteration of the learning process was similarly carried out. The comparison set presented to the DM during this last iteration is given in Table 9.7 (again ranked according to the DM's preference). This time the DM returned the pairwise comparison matrix in Figure A.16 of Appendix A. After yet again computing function score values from this matrix by means of the MACBETH method and updating the alternatives considered during all previous iterations as described in §5.4, a final estimate of the DM's value function was computed for the final iteration.

Alt	RGB Coordinates	Colour	Role	Score
$\boldsymbol{x}_{2,1}$	$(54, 168, 237)$	🟦	Exploitation	8.528
$\boldsymbol{x}_{2,2}$	$(44, 142, 217)$	🟦	Exploitation	8.082
$\boldsymbol{x}_{2,3}$	$(116, 155, 236)$	🟦	Exploitation	7.798
$\boldsymbol{x}_{2,4}$	$(67, 205, 189)$	🟩	Exploitation	7.023
$\boldsymbol{x}_{2,5}$	$(12, 254, 178)$	🟩	Exploration	6.023
$\boldsymbol{x}_{2,6}$	$(239, 219, 20)$	🟨	Exploration	4.520
$\boldsymbol{x}_{2,7}$	$(57, 225, 37)$	🟩	Exploration	4.016
$\boldsymbol{x}_{2,8}$	$(108, 214, 59)$	🟩	Exploration	3.518
$\boldsymbol{a}_1$	$(0, 255, 0)$	🟩	Anchor	3.508
$\boldsymbol{x}_{2,9}$	$(0, 0, 10)$	⬛	Exploration	0.000

Table 9.6: *Alternatives considered and (unscaled) value function scores computed during Iteration* 2 *of the learning cycle in the case study of §9.2. The comparison matrix returned by the DM during this iteration may be found in Figure A.15 of Appendix A. The exploration alternatives were generated randomly within the RGB cube.*

Alt	RGB Coordinates	Colour	Role	Score
$\boldsymbol{x}_{3,1}$	$(110, 192, 250)$	🟦	Exploitation	9.257
$\boldsymbol{x}_{3,2}$	$(64, 185, 236)$	🟦	Exploitation	8.010
$\boldsymbol{x}_{3,3}$	$(99, 197, 229)$	🟦	Exploitation	8.000
$\boldsymbol{x}_{3,4}$	$(82, 209, 234)$	🟦	Exploitation	7.756
$\boldsymbol{x}_{3,5}$	$(142, 241, 243)$	🟦	Exploration	7.257
$\boldsymbol{a}_1$	$(0, 255, 0)$	🟩	Anchor	5.505
$\boldsymbol{x}_{3,6}$	$(243, 64, 137)$	🟥	Exploration	3.752
$\boldsymbol{x}_{3,7}$	$(233, 222, 162)$	🟨	Exploration	1.752
$\boldsymbol{x}_{3,8}$	$(193, 97, 48)$	🟧	Exploration	1.000
$\boldsymbol{x}_{3,9}$	$(136, 114, 6)$	🟫	Exploration	0.000

Table 9.7: *Alternatives considered and (unscaled) value function scores computed during Iteration* 3 *of the learning cycle in the case study of §9.2. The comparison matrix returned by the DM during this iteration may be found in Figure A.16 of Appendix A. The exploration alternatives were generated randomly within the RGB cube.*

The global maximum of this function estimate was achieved at the alternative $(112, 209, 184)$ (🟩). During the final iteration the *incoherency value* of the DM in (3.16) was calculated as $c = 0.257$, which is slightly less than that of the previous iteration, but still larger than those of the first two iterations.

9.2.3 Recommendation and discussion

The aforementioned global optima and the best performing colour alternative in the archive of the experiment was presented to the DM as recommended colours, namely $(83, 198, 245)$ (🟦) and $(112, 209, 184)$ (🟩). Upon receiving this recommendation, the DM revealed that his favourite colour was $(110, 200, 255)$ (🟦). It appears that the shortest Euclidean distance between the DM's "favourite" colour and the recommended colours are 28.86 and 71.59 for the experiment. The DM found the results very interesting, commented that one of the recommended colours appear to be similar to his favourite colour, but slightly darker. He, however, mentioned that he found the global optimum $(112, 209, 184)$ (🟩) not to look anything like his "favourite" colour. The results of this case study led to the conclusion that by employing one anchor, the DSS proposed in this dissertation is capable of uncovering decision alternatives that are pleasing to a human DM exhibiting some level of inconsistency in terms of subjective preference within the context of the full 3DCSP, but was not convincingly effective in optimisation of the estimated value function. It is interesting that the closest alternative was returned directly from the archive, before having interpolated the value function over the entire decision space. It could be argued that this result would seem to suggest that the piecewise linear interpolation is not the most effective interpolation method or that the number of iterations was not large enough for the DSS to reach that area of the decision space. It should be acknowledged that the recommendations yielded after each iteration verged closer and closer to the DM's "favourite" colour and that a mere 37 colour alternatives were considered during this experiment out of the possible 16 777 216.

9.3 Chapter summary

The goal of this chapter was to assess the effectiveness and demonstrate the working of the DSS proposed in this dissertation for aiding DMs in solving subjective preferential decision problems within the context of two case studies involving real human DMs. The demonstration was performed in the form of two case studies involving a series of instances of the 2DCSP and an instance of the 3DCSP, the distinguishing characteristics of which are outlined in Table 9.8.

Four 2DCSP instances in parallel	One 3DCSP instance
$\chi = 1$ anchor in each instance	$\chi = 1$ anchor
$\omega = 2$ iterations performed for each instance	$\omega = 3$ iterations performed
$n = 8$ alternatives presented to the DM during each iteration	$n = 10$ alternatives presented to the DM during each iteration
Alternatives recommended to the DM: $(153, 196, 255)$ (🟦)	Alternatives recommended to the DM: $(83, 198, 245)$ (🟦) $(112, 209, 184)$ (🟩)
DM's favourite colour privately selected *a priori:* $(140, 190, 255)$ (🟦)	DM's favourite colour privately selected *a priori:* $(110, 200, 255)$ (🟦)
Euclidean error distance: 14.317	Euclidean error distances: 28.86, 71.59

TABLE 9.8: *Characteristics of the CSP case studies performed in this chapter.*

In the first case, the DSS of Chapter 5 was able to recommend an alternative that is almost indistinguishable by the human eye from the DM's "optimal" colour alternative, but in the 3DCSP case the results were slightly less effective.

Part IV

Conclusion

CHAPTER 10

Conclusion

Contents

In closing, a summary of the contents of this dissertation is provided in §10.1, after which an appraisal of the contributions of the dissertation follows in §10.2.

10.1 A summary of the dissertation contribution

The introductory chapter of the dissertation, Chapter 1, opened in §1.1 with a general background in which the need for decision support in respect of subjective preference selection was motivated. The problem considered in the dissertation was formally described in §1.2 and the objectives pursued in the dissertation were outlined in §1.3. Thereafter, the dissertation scope was delimited to ratio-based and difference-based pairwise comparison elicitation from the DM as well as application of the DSS proposed later in the dissertation to instances of the CSP in §1.4. The final section of the chapter, §1.5, presented a detailed description of how the material contained in the remainder of the dissertation was organised into chapters and parts.

Apart from the introductory chapter, this dissertation comprised a further nine chapters (organised in four parts), a bibliography and an appendix. Part I was a literature review in fulfilment of Objective I in §1.3, and consisted of three chapters. The first chapter of Part I, Chapter 2, was devoted to a review of the academic literature on a number of mathematical prerequisites pertaining to the topic of this dissertation. The chapter opened in §2.1 with a discussion on basic notions related to and central results on eigenvalues and eigenvectors of square matrices, since these concepts formed one of the corner stones on which the elicitation of DM value function scores was based. The focus of the discussion then shifted to round-robin sports tournament modelling and the determination of winners in such tournaments by means of directed graphs in §2.2, since the theory of tournaments provides a very natural introduction to the ordering of SPBP alternatives according to a single selection criterion. Since clustering may profitably be applied to decision alternatives according to similarities based on DM value function score estimates between these alternatives, a brief overview of the large field of data clustering was presented in §2.3, singling out the k-means clustering technique which was applied in this dissertation. A brief discussion also followed in §2.4 on exact and approximate techniques for solving single-objective optimisation problems. Two techniques were singled out for special attention during this discussion. These were the PHC algorithm and the method of SA — two methods that were later employed in the dissertation within the interactive preference learning cycle. The DSS proposed in this dissertation for interactive preference learning involves stochastic elements, and so its validation necessarily had to be carried out within the realm of inferential statistics. A suite of well-known statistical tests was therefore reviewed in §2.5 for determining whether or not the means of a collection of approximately normally distributed data samples are distinguishable at a specified level of confidence. The chapter finally closed in §2.6 with a brief summary of the chapter content.

The second chapter of Part I, Chapter 3, followed on the discussion on sports tournament results ranking by providing a review of very basic concepts in subjective preference modelling in §3.1, as well as a thorough review in §3.2 of the well-known AHP, one of the early frameworks for solving multi-criteria instances of the SPBP according to a ratio-based approach. Another method of measuring subjective preference, the difference-based MACBETH method, was also discussed in §3.2. Thereafter, a review followed in §3.3 of various interactive preference learning models. The chapter finally closed with a brief summary of the chapter content in §3.4.

The final chapter of Part I, Chapter 4, was a review of general guidelines in the literature for the design, verification and validation of DSSs. The first section of Chapter 4 contained a discussion on various types of DSSs available in the literature. Although these DSSs typically contain different elements, the three most important elements were discussed in some detail in §4.2. Three design DSS methodologies were then reviewed in §4.3. The penultimate section, §4.4, contained an overview of the various verification and validation methods available for DSSs in the literature. A brief summary of the chapter contents was finally provided in §4.5.

Part II of the dissertation consisted of two chapters and were focused on the proposed DSS. Chapter 5 was the heart of the dissertation. In that chapter, the design of an interactive preference learning DSS was put forward in fulfilment of Objective II. After providing a high-level overview of the iterative working of the proposed DSS in §5.1, an architecture was proffered for the framework in §5.2. The two main components of the system (the *system configuration component* and the *data management component*) were then described in detail in §5.3, after which the discussion turned to the proposed working of a central mechanism of the framework in §5.4 — the method of rescaling of DM value function scores as the system progresses through successive iterations of the preference learning cycle. The purpose of this mechanism was to achieve a trade-off between reducing the DM's workload associated with pairwise comparisons of decision alternatives, on the one hand, and accommodating small to moderate levels of DM inconsistency when carrying out these comparisons, on the other. The mechanism involved strategic use of a number of pivotal decision alternatives, called *anchors*. The presentation of the interactive preference learning DSS was concluded in §5.5 with a detailed process description, paying special attention to the order of events and the flows of data within the framework.

The second chapter of Part II, Chapter 6, contained two detailed worked examples in which the framework was applied to an instance of the 2DCSP, after describing in §6.1 how the DM was modelled for system validation purposes and establishing two learning performance measure indicators according to which the framework effectiveness could be evaluated. One of these worked examples is presented in the context of a ratio-based approach (the AHP) in §6.2 and the other adopting a difference-based approach (the MACBETH method) in §6.3 towards measuring subjective preference in fulfilment of Objective III. The penultimate section of the chapter, §6.4, contained a discussion on the results obtained in the worked examples. A brief summary of the chapter content finally followed in §6.5.

In the first chapter of Part III, Chapter 7, the interactive preference learning DSS proposed in Chapter 5 was subjected to a thorough validation in the context of the 2DCSP. The intrinsic ability of the framework to learn was tested in §7.2 within the realm of inferential statistics at a 95% level of confidence for both a ratio-based and a difference-based pairwise comparison elicitation approach. The relative performances associated with adopting various judgement scales and value function aggregation methods were assessed for the ratio-based approach within the same inferential statistics paradigm in §7.3, after which the merits of exploration and exploitation of the 2DCSP decision space (in terms of preference learning effectiveness) were evaluated for both the ratio- and difference-based approaches in §7.4, adopting the same inferential statistical approach. A classical sensitivity analysis was finally performed in §7.5 with respect to a variety of parameters that appear in the decision support framework, before the chapter closed in §7.6 with a brief summary of its content.

Whereas the entire validation process of Chapter 7 was carried out under the strong assumption of perfect DM consistency when eliciting pairwise comparisons of decision alternatives, Chapter 8 was devoted to a study of the effects of DM inconsistency on the learning facilitation effectiveness of the DSS of Chapter 5 in fulfilment of Objective IV. The chapter opened in §8.1 with a description of how DM inconsistency was modelled for system evaluation purposes. The experimental setup of the aforementioned effectiveness evaluation analysis was also discussed. The first step in the analysis was to test in §8.2 how the the intrinsic ability of the DSS to learn DM preferences (for both ratio-based and difference-based approaches) is affected by the introduction of small to medium levels of DM inconsistency during the process of pairwise comparisons. This was followed in §8.3 by an assessment of the ability of the DSS to deal with DM inconsistency by increasing the number of anchors considered by the DM during each iteration. The chapter closed with a brief summary of the chapter content in §8.4.

Chapter 9 contained two realistic case studies based on the 2DCSP and involving real DMs in fulfilment of Objective V. These case studies were conducted as further validation of the practical applicability of the DSS proposed in Chapter 5. After describing the experimental setup of the first case study, in which a ratio-based approach was adopted, the progression of obtaining intermediate results from the DM when applying the system of Chapter 5 was presented in §9.1, culminating in a final decision recommendation to the DM and a discussion on the DM's response to this recommendation. The same format of reporting was followed in §9.2 for the second case study, in which a difference-based approach was adopted, before the chapter closed with a short summary in §9.3.

10.2 Appraisal of dissertation contributions

The main contribution of this dissertation was a design proposal for an interactive learning DSS aimed at facilitating the solution of SPBP instances. Although the working of the system was illustrated in the context of the ratio-based pairwise comparison paradigm of the AHP (described in §3.2.1) or the difference-based pairwise comparison paradigm of the MACBETH method (described in §3.2.2), the design of this system (presented in detail in Chapter 5) was generic in nature and driven by two main goals. These design drivers were:

1. To reduce the workload of the DM in terms of pairwise comparisons of alternatives that have to be carried out.

   (a) This workload reduction was achieved in part by reducing the actual number of pairwise comparisons required. If ω iterations of the learning cycle of Figure 5.1 are carried out according to the DSS framework proposed in this dissertation, with n alternatives being considered by the DM during each iteration (of which χ are anchors that are common to all iterations), then the number of pairwise comparisons required in total is

$$(\omega + 1)\binom{n}{2}.$$

The corresponding number of pairwise comparisons that would have been required for the same total number of $n + \omega(n - \chi)$ alternatives according to the classical AHP or MACBETH method is

$$\binom{n + \omega(n - \chi)}{2}.$$

The magnitude of the difference between these two quantities in (5.8) was illustrated in Table 5.1.

(b) The workload reduction was furthermore enhanced by pursuing a trade-off between *exploiting* well-performing areas of the decision space and *exploring* unknown regions of the entire decision space. The aim of this trade-off pursuit was to avoid wasting the DM's efforts in respect of considering decision alternatives that are known to be of poor relative quality, thus bringing about pairwise comparisons of alternatives that "matter" in the sense of enhancing exploitation or exploration. The success of the exploration aspect of this alternative generation process was confirmed by real DMs in case studies involving the CSP, who expressed satisfaction at previously unconsidered but "beautiful" colours presented to them by the DSS. The success of the exploitation aspect of this generation process was similarly confirmed by these DMs in the form of satisfaction expressed in respect of the final colour recommendations presented to them by the system.

2. To accommodate a limited level of DM inconsistency in the value function learning process. This was achieved by including a certain degree of redundancy in the generation of decision alternatives to be considered by the DM. This redundancy was implemented in the form of an *anchor set* embedded within the comparison set of each learning cycle iteration. In this way, the DM was required to compare the relative attractiveness of these anchors with other comparison set elements once and with each other multiple times. This redundancy was averaged in an attempt at smoothing out DM inconsistency, as was explained in §5.4. Although this attempt at mitigating DM inconsistency was only shown to be marginally successful, there were demonstrable, albeit slight, benefits associated with including the aforementioned redundancy as the level of DM inconsistency increased. This approach towards alleviating DM inconsistency certainly warrants further investigation, as is suggested in the following chapter.

The DSS proposed in Chapter 5 may be considered a valuable contribution from a practical operations research perspective, because its design was not limited to a conceptual level only. The working of the system was demonstrated in the context of a practical instance of the SPBP in Chapter 6. The system's performance and its sensitivity to parameter values embedded in it were also analysed thoroughly within the same context in Chapters 7 and 8. Two case studies involving human DMs were finally performed in Chapter 9 according the DSS framework proposed in Chapter 5.

The contribution of this dissertation may furthermore be considered innovative, because it provides a very simple mechanism for combining the efforts of a human DM and a machine during the process of alternative discovery when attempting to solve SPBP instances, as illustrated in Figure 5.1. In this way the classical methodology of the AHP or MACBETH method, in which there is no built-in mechanism for the explicit exploration of unchartered areas or the explicit exploitation of promising areas of the decision space, has been brought closer to the realm of (reinforcement) machine learning.

The contribution of this dissertation may also be considered significant, because the design driver discussed under point 1 above resulted in significant DM workload reduction. The reduction in the required number of alternative pairwise comparisons embodied in the expression in (5.8) alone translates into significant savings, as was demonstrated in Table 5.1, let alone the additional benefit brought about by transferring some of the DM workload (in terms of judicious alternative generation) to a computer.

It is finally noted that the contribution of this dissertation may be considered interesting from an academic point of view in the sense that it may lead to further research. Based on the discussion in this section alone, the two main areas for further research that come to mind are the pursuit of a more effective mechanism for mitigating DM inconsistency than that put forward in this dissertation, and explicit incorporation of reinforcement learning techniques, which have proved very successful in other areas of application, into the DSS framework proposed in this dissertation. A number of further ideas for possible future work are outlined in some detail in the closing chapter of the dissertation.

CHAPTER 11

Future work

Contents

A number of ideas for possible future follow-up work, building on the foundation laid in this dissertation, related to the scope of the preceding research, are provided in §11.1, while ideas for future work related to the working of the DSS proposed in Chapter 5 are proposed in §11.2.

11.1 Suggestions related to the preceding research scope

This section contains suggestions for three avenues of further investigation related specifically to the scope of the research carried out in the preceding chapters. In each case, the suggestion is stated formally and then elucidated and motivated briefly.

Suggestion 11.1 *Ascertain whether the superior performance of certain judgement scales in Chapter 7 is a result of the particular subjective preference selection problem considered, the measure of learning effectiveness adopted or the assumed true value function.*

In Chapter 7 it was found that a convex judgement scale (such as the power scale) seemed to outperform many other scales in terms of general learning ability (as measured by the GLPMI). A natural question that arises in this respect is whether this superior performance was a result of the particular subjective preference selection problem considered (*i.e.* the 2DCSP introduced in §6), the measure of learning effectiveness adopted (*i.e.* the GLPMI in (6.3)) or the shape of the assumed true value function of the DM (*i.e.* the Gaussian superposition in (6.1)). Perhaps the answer is that the superior performance of the convex judgement scale is a result of a combination of these factors. The author could not find any references in the literature to this phenomenon and so it would be interesting to attempt to get to the bottom of the reason(s) for convex judgement scales apparently being able to outperform other scales (by a significant margin in some cases, and at a 95% level of confidence). It would indeed be surprising if this were to be a universal phenomenon.

Suggestion 11.2 *Test the effectiveness of the DSS framework in respect of subjective preference selection case studies other than colour selection.*

Only one member of the class of SPBPs was considered as practical application in this dissertation — the CSP. It would, however, also be interesting to consider other practical case studies and to ascertain whether the findings of the DSS framework validation analysis performed in Chapters 7 and 8 were special cases limited to the application of colour selection in any way, or whether certain aspects of the analysis findings seem to hold more generally. Other examples of subjective preference selection applications may include coffee or wine blending.

Suggestion 11.3 *Broaden the scope of the effectiveness analysis of the DSS framework to multi-criteria SPBP instances.*

Although the DSS framework for SPBP put forward in Chapter 5 essentially accommodates multi-criteria decision problems, the validation analyses of Chapters 7 and 8, as well as the case studies of Chapter 9, were all carried out exclusively within the context of a single subjective preference criterion, namely *beauty* or *aesthetic preference*. It would be interesting to broaden the analyses so as to account for multiple criteria. A simple extension in this respect may be to consider both RGB combination suitability and roughness of surface finish as simultaneous selection criteria within the context of, say, selecting a suitable paint for painting a room.

11.2 Suggestions related to the working of the DSS

This final section contains suggestions for five avenues of further investigation as possible follow-up work on the contributions of this dissertation which are specifically related to the method of working of the DSS proposed in Chapter 5. In each case the suggestion is again stated formally and then elucidated and motivated briefly.

Suggestion 11.4 *Consider alternative AHP judgement scales for inclusion in the DSS framework.*

Six AHP judgement scales were considered and compared during the DSS framework validation experiments of Chapter 7 when viewing the framework within a ratio-based pairwise comparison paradigm. These were the linear, logarithmic, root, balanced, power and geometric judgement scales. This list of judgement scales is not, however, exhaustive. Examples of other judgement scales include the inverse linear judgement scale proposed by Ma and Zeng [142] in 1991 and the asymptotic judgement scale proposed by Dodd and Donegan [55] in 1995, to name but two. Since some AHP judgement scales were found to be more effective than others within the context of the 2DCSP introduced in §6, it may be interesting to include additional judgement scales in similar system validation analyses as those carried out in Chapters 7 and 8.

Suggestion 11.5 *Consider alternative AHP weight estimation methods for inclusion in the DSS framework.*

Two methods of aggregation were considered and compared in the DSS validation experiments of Chapter 7 when implementing the framework within a ratio-based pairwise comparison paradigm. These were the EM and the LLSM. Examples of other AHP aggregation methods include the least squares method proposed by Cogger and Yu in 1983 [41], and the method of renormalisation after the establishment of ratios proposed by Kamenetzky in 1982 [112], to name but two. Although the EM and LLSM were found to be of similar effectiveness when incorporated in a ratio-based approach within the DSS framework of Chapter 5 in the context of the 2DCSP, it would be interesting to include additional AHP methods of aggregation in similar system validation analyses as those carried out in Chapters 7 and 8 in order to ascertain whether the DSS framework proposed in Chapter 5 is indeed insensitive to the choice of aggregation method.

Suggestion 11.6 *Consider alternative generation paradigms for inclusion as exploration mechanisms in the DSS framework.*

All alternatives generated stochastically in this dissertation were generated according to a uniform distribution across the decision space of the 2D2SP. Other methods of stochastic alternative generation may, however, also be considered for inclusion in the DSS framework of Chapter 5. Examples of other stochastic alternative generation methods include *Latin hypercube sampling* [149] and alternative generation based on favourable *Kriging* results [153], to name but two. It would be interesting to test whether or not such more sophisticated alternative generation approaches are capable of leading to a markedly more effective exploration of the decision space.

Suggestion 11.7 *Consider alternative optimisation paradigms for inclusion as exploitation mechanisms in the DSS framework.*

Two different optimisation paradigms were considered and compared as exploitation drivers within the context of the DSS framework proposed in this dissertation. These optimisation paradigms were a direct exploitation method, called the Basic E&E method, based on a shrinking radius drawn around the global maximum of the current value function estimate as the learning process progresses and an implementation of the PHC algorithm described in §2.4.2. It is advocated that the relative effectiveness of other optimisation paradigms should also be tested as possible exploitation mechanisms within the DSS framework of Chapter 5. A population-based metaheuristic (such as a genetic algorithm with a limited mutation rate) may, for example, be incorporated into the framework for this purpose. Such a metaheuristic may function very effectively as an exploitation mechanism, while simultaneously promoting some level of exploration.

Suggestion 11.8 *Quantify the level of DM inconsistency and base the selection of the number of anchors included in each comparison set presented to the DM on this quantification.*

The purpose of including a number of anchors in each comparison set presented for pairwise comparison purposes to the DM was to accommodate some level of inconsistency on the part of the DM within the DSS framework of Chapter 5. It was shown in Chapter 8 that as more anchors are included in the comparison sets, DM inconsistency can be accommodated marginally more successfully, but that smaller savings result in terms of DM burden when adopting the approach of using anchors instead of requiring an exhaustive pairwise comparison of all alternatives by the DM in order to facilitate the value function construction process. This level of DM inconsistency was modelled by introducing small random changes to the comparisons that would have been returned by a perfectly consistent DM. It is, however, suggested that a measure of DM inconsistency, such as that in Saaty's AHP (in conjunction with a linear judgement scale) described in §3.2, should instead be employed to quantify the degree of DM inconsistency that can be accommodated when including a specific number of anchors in each comparison set presented to the DM. This quantification may then be used to guide the analyst in terms of how many anchors to include (perhaps in a dynamic fashion as the learning process progresses) when applying the DSS framework of Chapter 5 to SPBP instances.

References

[1] Aarts E, Korst J & Michiels W, 2005, *Simulated annealing*, pp. 187–210 in Burke EK & Kendall G (Eds), *Search methodologies*, Springer, New York (NY).

[2] Aho A, 1990, *Algorithms for finding patterns in strings*, pp. 255–300 in Leeuwen JV (Ed), *Handbook of theoretical computer science*, MIT Press, Cambridge (MA).

[3] Akrour R, Schoenauer M, Sebag M & Souplet JC, 2014, *Programming by feedback*, [Online], [Cited October 2017], Available from `https://hal.inria.fr/hal-00980839/PDF/programming_by_feedback.pdf`.

[4] Alrabea A, Senthilkumar A, Al-Shalabi H & Bader A, 2013, *Enhancing k-means algorithm with initial cluster centers derived from data partitioning along the data axis with PCA*, Journal of Advances in Computer Networks, **1(2)**, pp. 137–142.

[5] Alter S, 1980, *Decision support systems: Current practice and continuing challenges*, Systems Research and Behavioural Science, **27(1)**, pp. 91–92.

[6] Anderson R, 1986, *Introduction to linear algebra*, Holt, Rinehart and Winston, New York (NY).

[7] Anton H, 1987, *Elementary linear algebra*, 5th Edition, John Wiley & Sons, New York (NY).

[8] Aveson D & Fitzgerald G, 2006, *Methodologies for developing information systems: A historical perspective*, pp. 27–38 in Avison D, Elliot S, Krogstie J & Pries-Hije J (Eds), *The past and future of information systems: 1976–2006 and beyond*, Springer, Boston (MA).

[9] Avison DE & Fitzgerald G, 2003, *Where now for development methodologies?*, Communications of the Association for Computing Machinery, **46(1)**, pp. 78–82.

[10] Ayres Jr F, 1962, *Matrices*, Schaum Publishing Company, New York (NY).

[11] Babu GP & Murty MN, 1993, *A near-optimal initial seed value selection in k-means algorithm using a genetic algorithm*, Pattern Recognition Letters, **14(10)**, pp. 763–769.

[12] Baeza-Yates R & Frakes W, 1992, *Information retrieval: Data structures and algorithms*, Prentice-Hall, Upper Saddle Rivier (NJ).

[13] Bai L, Liang J, Sui C & Dang C, 2013, *Fast global k-means clustering based on local geometrical information*, Information Sciences, **245**, pp. 168–180.

[14] Baker FB & Hubert LJ, 1976, *A graph-theoretic approach to goodness-of-fit in complete-link hierarchical clustering*, Journal of the American Statistical Association, **71(356)**, pp. 870–878.

[15] Balakarishnann K, 2000, *A textbook of graph theory*, Springer-Verlag, New York (NY).

[16] Bang-Jensen J & Gutin G, 2009, *Digraphs: Theory, algorithms and applications*, 2nd Edition, Springer-Verlag, London.

[17] Banks J, Carson J & Nelson B, 2005, *Discrete-event system simulation*, 4th Edition, Pearson, New York (NY).

[18] Basak J & Krishnapuram R, 2005, *Interpretable hierarchical clustering by constructing an unsupervised decision tree*, IEEE Transactions on Knowledge and Data Engineering, **17(1)**, pp. 121–132.

[19] Belton V & Gear T, 1983, *On a short-coming of Saaty's method of analytic hierarchies*, Omega, **11(3)**, pp. 228–230.

[20] Berkhin P, 2006, *A survey of clustering data mining techniques*, Grouping Multidimensional Data, pp. 25–71.

[21] Bertsimas D & Tsitsiklis J, 1993, *Simulated annealing*, Statistical Science, **8(1)**, pp. 10–15.

[22] Bouyssou D, Dubois D, Prade H & Pirlot M, 2005, *Decision making process: Concepts and methods*, John Wiley & Sons, Hoboken (NJ).

[23] Bradley P, Bennett K & Demiriz A, 2000, *Constrained k-means clustering*, Unpublished Manuscript, [Online], [Cited November 2017], Available from `http://machinelearning102.pbworks.com/f/ConstrainedKMeanstr-2000-65.pdf`.

[24] Brand A, 1970, *Matrices: Pure and applied*, Edward Arnold, London.

[25] Branke J, Greco S, Słowiński R & Zielniewicz P, 2015, *Learning value functions in interactive evolutionary multiobjective optimization*, IEEE Transactions on Evolutionary Computation, **19(1)**, pp. 88–102.

[26] Brown DE & Huntley CL, 1992, *A practical application of simulated annealing to clustering*, Pattern Recognition, **25(4)**, pp. 401–412.

[27] Busetti F, 2003, *Simulated annealing overview*, [Online], [Cited August 2017], Available from `http://www.cs.ubbcluj.ro/~csatol/mestint/pdfs/Busetti_AnnealingIntro.pdf`.

[28] Busoniu L, Babuska R & De Schutter B, 2008, *A comprehensive survey of multi-agent reinforcement learning*, IEEE Transactions on Systems, Man, and Cybernetics — Part C: Applications and Reviews, **38(2)**, pp. 156–172.

[29] Cai X, Cui Z, Fang J & Zeng J, 2010, *A hybrid group search optimiser with Metropolis rule*, Proceedings of the 2010 International Conference on Modelling, Identification and Control, Okayama City, pp. 556–561.

[30] Cantor G, 1895, *Beiträge zur Begründung der transfiniten Mengenlehre*, Mathematische Annalen, **46(4)**, pp. 481–512.

[31] Castro RM, Coates MJ & Nowak RD, 2004, *Likelihood based hierarchical clustering*, IEEE Transactions on Signal Processing, **52(8)**, pp. 2308–2321.

[32] Celebi ME & Kingravi HA, 2015, *Linear, deterministic, and order-invariant initialization methods for the k-means clustering algorithm*, pp. 79–98 in Celebi ME (Ed), *Partitional clustering algorithms*, Springer, Cham.

[33] Chartrand G, 1977, *Introdutory graph theory*, Dover Publications, New York (NY).

[34] Chartrand G & Oellermann O, 1993, *Applied and algorithmic graph theory*, McGraw-Hill, Inc., New York (NY).

[35] Chartrand G & Zhang P, 2012, *A first course in graph theory*, Dover Publications, Inc., Mineola (NY).

[36] Cheng Y, 1995, *Mean shift, mode seeking, and clustering*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **17(8)**, pp. 790–799.

[37] Choo E & Wedley W, 2008, *Comparing fundamentals of additive and multiplicative aggregation in ratio scale multi-criteria decision making*, The Open Operational Research Journal, **2(1)**, pp. 1–7.

[38] Christiano P, Leike J, Brown TB, Martic M, Legg S & Amodei D, 2017, *Deep reinforcement learning from human preferences*, [Online], [Cited June 2017], Available from `arXiv%20preprint%20arXiv:1706.03741`.

[39] Christofides N, 1976, *Worst-case analysis of a new heuristic for the travelling salesman problem*, (Unpublished) Technical Report, Management Sciences Research Group, Carnegie-Mellon University, Pittsburgh (PA).

[40] Climer S & Zhang W, 2006, *Rearrangement clustering: Pitfalls, remedies, and applications*, Journal of Machine Learning Research, **7(1)**, pp. 919–943.

[41] Cogger KO & Yu PL, 1985, *Eigenweight vectors and least-distance approximation for revealed preference in pairwise weight ratios*, Journal of Optimization Theory and Applications, **46(4)**, pp. 483–491.

[42] Comaniciu D & Meer P, 1999, *Mean shift analysis and applications*, Proceedings of the Seventh IEEE International Conference on Computer Vision, Piscataway (NJ), pp. 1197–1203.

[43] Copeland BJ, 2004, *The essential Turing: The ideas that gave birth to the computer age*, The Clarendon Press, Oxford.

[44] Copeland M, 2016, *What's the difference between artificial intelligence, machine learning, and deep learning?*, [Online], [Cited March 2017], Available from `https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/`.

[45] Crawford G & Williams C, 1985, *A note on the analysis of subjective judgement matrices*, Journal of Mathematical Psychology, **29(4)**, pp. 387–405.

[46] Dahl GE, Yu D, Deng L & Acero A, 2012, *Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition*, IEEE Transactions on Audio, Speech, and Language Processing, **20(1)**, pp. 30–42.

[47] Dantzig G, 1963, *Linear programming and extensions*, Princeton University Press, Princeton (MA).

[48] Davidson J, Liebald B, Liu J, Nandy P, Van Vleet T, Gargi U, Gupta S, He Y, Lambert M & Livingston B, 2010, *The YouTube video recommendation system*, Proceedings of the 4th ACM Conference on Recommender Systems, Barcelona, pp. 293–296.

[49] Day W, 1992, *Complexity theory: An introduction for practitioners of classification*, pp. 199–233 in Arabie P, Hubert L & de Soete G (Eds), *Clustering and classification*, World Scientific, Singapore.

[50] De Amorim RC & Mirkin B, 2012, *Minkowski metric, feature weighting and anomalous cluster initializing in k-means clustering*, Pattern Recognition, **45(3)**, pp. 1061–1075.

[51] Deb K, 2001, *Multi-objective optimization using evolutionary algorithms*, John Wiley & Sons, New York, (NY).

[52] Dennis A, Wixom BH & Tegarden D, 2015, *Systems analysis and design: An object-oriented approach with UML*, John Wiley & Sons, Hoboken (NJ).

[53] Dewancker I, McCourt M & Ainsworth S, 2016, *Interactive preference learning of utility functions for multi-objective optimization*, [Online], [Cited December 2016], Available from `arXiv%20preprint%20arXiv:1612.04453`.

[54] Dixon W, 1960, *Simplified estimation from censored normal samples*, Annals of Mathematical Statistics, **31**, pp. 385–391.

[55] Dodd FJ & Donegan HA, 1995, *Comparison of prioritization techniques using inter-hierarchy mappings*, Journal of the Operational Research Society, **46(4)**, pp. 492–498.

[56] Dorigo M, Birattari M & Stützle T, 2006, *Ant colony optimization*, IEEE Computational Intelligence Magazine, **1(4)**, pp. 28–39.

[57] Dreo J, Petrowski A, Siarry P & Taillard E, 2006, *Metaheuristics for hard optimisation: Methods and case studies*, Springer, New York (NY).

[58] Du Plessis M, 2018, Masters student in the Department of Industrial Engineering at *Stellenbosch University*, [Personal Communication], Contactable at `18314937@sun.ac.za`.

[59] Dubes RC, 1987, *How many clusters are best? — An experiment*, Pattern Recognition, **20(6)**, pp. 645–663.

[60] Duda RO, Hart PE & Stork DG, 1973, *Pattern classification and scene analysis*, Wiley, New York (NY).

[61] Duin RP, Fred AL, Loog M & Pękalska E, 2012, *Mode seeking clustering by KNN and mean shift evaluated*, Proceedings of the Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), Berlin, pp. 51–59.

[62] E Costa & J-C Vansnick CB, 1994, *MACBETH — An interactive path towards the construction of cardinal value functions*, International Transactions in Operational Research, **1(4)**, pp. 489–500.

[63] Evangelos T, 2000, *Multi-criteria decision making methods: A comparative study*, Kluwer Academic Publications, Boston (MA).

[64] Fichtner J, 1984, *Some thoughts about the mathematics of the analytic hierarchy process*, Unpublished Manuscript, Hochschule der Bundeswehr, München.

[65] Finlay PN & Wilson JM, 1987, *The paucity of model validation in operational research projects*, Journal of the Operational Research Society, **38(4)**, pp. 303–308.

[66] Fishburn PC, 1970, *Utility theory for decision making*, John Wiley & Sons, New York (NY).

[67] Fukunaga K & Hostetler L, 1975, *The estimation of the gradient of a density function, with applications in pattern recognition*, IEEE Transactions on Information Theory, **21(1)**, pp. 32–40.

[68] Garnett R, Krishnamurthy Y, Xiong X, Schneider J & Mann R, 2012, *Bayesian optimal active search and surveying*, [Online], [Cited October 2017], Available from `https://arxiv.org/ftp/arxiv/papers/1206/1206.6406.pdf`.

[69] Geetha S, Poonthalir G & Vanathi P, 2009, *Improved k-means algorithm for capacitated clustering problem*, INFOCOMP Journal of Computer Sciences, **8(4)**, pp. 52–59.

[70] Geoffrion AM, Dyer JS & Feinberg A, 1972, *An interactive approach for multi-criterion optimization, with an application to the operation of an academic department*, Management Science, **19(4)**, pp. 357–368.

[71] George B & Thomas JR, 1961, *Calculus and analytic geometry*, 3rd Edition, Addison-Wesley Publishing Company, Inc., Boston (MA).

[72] Gkanogiannis A & Kalamboukis T, 2008, *A novel supervised learning algorithm and its use for spam detection in social bookmarking systems*, European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Antwerp, pp. 13–20.

[73] Glover F, 1989, *Tabu search — Part I*, ORSA Journal on Computing, **1(3)**, pp. 190–206.

[74] Goethals OM & Rokach L, 2005, *The data mining and knowledge discovery handbook*, Springer, New York (NY).

[75] Goldberg DE & Holland JH, 1988, *Genetic algorithms and machine learning*, Machine Learning, **3(2)**, pp. 95–99.

[76] Goodall DW, 1966, *A new similarity index based on probability*, Biometrics, **22(1)**, pp. 882–907.

[77] Google, 2017, *More than letters and words*, [Online], [Cited October 2017], Available from `https://www.google.com/docs/about/`.

[78] Gower JC & Ross G, 1969, *Minimum spanning trees and single linkage cluster analysis*, Applied Statistics, **18(1)**, pp. 54–64.

[79] Gray RM, 2011, *Entropy and information theory*, Springer Science & Business Media, Berlin.

[80] Groebner D & Shannon P, 1991, *Business statistics: A decision making approach*, 3rd Edition, Merrill Publishing Company, Singapore.

[81] Gulwani S, 2011, *Automating string processing in spreadsheets using input-output examples*, Proceedings of the 1th ACM SIGPLAN Conference, Redmond (WA), pp. 317–330.

[82] Haettenschwiler P, 1999, *Neues anwenderfreundliches Konzept der Entscheidungsunterstützung*, Gutes Entscheiden in Wirtschaft, Politik und Gesellschaft, Zurich, pp. 189–208.

[83] Hall LO, Ozyurt IB & Bezdek JC, 1999, *Clustering with a genetically optimized approach*, IEEE Transactions on Evolutionary Computation, **3(2)**, pp. 103–112.

[84] Hansen M & Yu B, 2000, *Wavelet thresholding via MDL for natural images*, IEEE Transactions on Information Theory, **46(5)**, pp. 1778–1788.

[85] Harker PT & Vargas LG, 1987, *The theory of ratio scale estimation: Saaty's analytic hierarchy process*, Management Science, **33(11)**, pp. 1383–1403.

[86] Hartuv E & Shamir R, 2000, *A clustering algorithm based on graph connectivity*, Information Processing Letters, **76(4-6)**, pp. 175–181.

[87] Hayter A, 1986, *The maximum familywise error rate of Fisher's least significant difference test*, Journal of the American Statistical Association, **81(396)**, pp. 1000–1004.

[88] Henning MA & Van Vuuren JH, In Prep, *Graph and network theory: An applied approach using Mathematica.*

[89] Hillier G, 2010, *Introduction to operations research*, 9th Edition, McGraw-Hill, Boston (MA).

[90] Hohn E, 1964, *Elementary matrix algebra*, MacMillan, New York (NY).

[91] Holder R, 1990, *Some comments on the analytic hierarchy process*, Journal of the Operational Research Society, **41(11)**, pp. 1073–1076.

[92] Holladay R, Javdani S, Dragan A & Srinivasa S, 2016, *Active comparison based learning incorporating user uncertainty and noise*, Proceedings of the RSS Workshop on Model Learning for Human-Robot Communication, Rome, No page numbers.

[93] Holzinger A, 2016, *Interactive machine learning for health informatics: When do we need the human-in-the-loop?*, Brain Informatics, **3(2)**, pp. 119–131.

[94] Hoos HH & Stützle T, 2005, *Stochastic local search: Foundations and applications*, Elsevier, New York (NY).

[95] Hormozi H, Hormozi E & Nohooji HR, 2012, *The classification of the applicable machine learning methods in robot manipulators*, International Journal of Machine Learning and Computing, **2(5)**, pp. 560–563.

[96] Howard A, 1976, *Elementary linear algebra*, 5th Edition, John Wiley & Sons, Toronto.

[97] Howard A & Rorres C, 1994, *Elementary linear algebra*, 7th Edition, John Wiley & Sons, Toronto.

[98] Howell J & Games P, 1974, *The effects of variance heterogeneity on simultaneous multiple-comparison procedures with equal sample size*, British Journal of Mathematical and Statistical Psychology, **27(1)**, pp. 33–37.

[99] Howell J & Games P, 1973, *The robustness of the analysis of variance and the Tukey WSD test under various patterns of heterogeneous variances*, Journal of Experimental Education, **41(4)**, pp. 72–81.

[100] Hromkovič J, 2013, *Algorithmics for hard problems: Introduction to combinatorial optimization, randomization, approximation, and heuristics*, Springer Science & Business Media, New York (NY).

[101] Huang MD, Romeo F & Sangiovani-Vincentelli AL, 1986, *An efficient general cooling schedule for simulated annealing*, Proceedings of the 1986 IEEE International Conference on Computer-Aided Design, Santa Carla (CA), pp. 381–384.

[102] Ishizaka A & Labib A, 2009, *Analytic hierarchy process and expert choice: Benefits and limitations*, OR Insight, **22(4)**, pp. 201–220.

[103] Jacobi J, Benson E & Linden G, 2001, *Use of electronic shopping carts to generate personal recommendations*, [Patent], US Patent No 6 317 722.

[104] Jain AK, 2010, *Data clustering: 50 years beyond k-means*, Pattern Recognition Letters, **31(8)**, pp. 651–666.

[105] Jain AK & Dubes RC, 1988, *Algorithms for clustering data*, Prentice-Hall, Inc., Upper Saddle River (NJ).

[106] Jain AK, Murty MN & Flynn PJ, 1999, *Data clustering: A review*, ACM Computing Surveys, **31(3)**, pp. 264–323.

[107] Jain A, Wojcik B, Joachims T & Saxena A, 2013, *Learning trajectory preferences for manipulators via iterative improvement*, Proceedings of the NIPS 2013 Workshop on Deep Learning, Advances in Neural Information Processing Systems, **26**, pp. 575–583.

[108] Joachims T, 1998, *Text categorization with support vector machines: Learning with many relevant features*, Poceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-98), Chemnitz, pp. 137–142.

[109] Johnson CR, Beine WB & Wang TJ, 1979, *Right-left asymmetry in an eigenvector ranking procedure*, Journal of Mathematical Psychology, **19(1)**, pp. 61–64.

[110] Jones CV, 1993, *User interface development and decision support systems*, pp. 181–209 in Holsapple C & Whinston A (Eds), *Recent developments in decision support systems*, Springer-Verlag, Berlin.

[111] Kaelbling LP, Littman ML & Moore AW, 1996, *Reinforcement learning: A survey*, Journal of Artificial Intelligence Research, **4**, pp. 237–285.

[112] Kamenetzky RD, 1982, *The relationship between the analytic hierarchy process and the additive value function*, Decision Sciences, **13(4)**, pp. 702–713.

[113] Kaufman L & Rousseeuw PJ, 2009, *Finding groups in data: An introduction to cluster analysis*, John Wiley & Sons, Hoboken (NJ).

[114] Kendall K & Kendall J, 2011, *Systems analysis and design*, 8th Edition, Pearson, Upper Saddle River (NJ).

[115] Kennedy J, 2011, *Particle swarm optimization*, pp. 760–766 in Claude S & Geoffrey IW (Eds), *Encyclopedia of machine learning*, Springer, New York (NY).

[116] Kidd M, 2016, Professor of Statistics and Actuarial Science at *Stellenbosch University*, [Personal Communication], Contactable at mkidd@sun.ac.za.

[117] King B, 1967, *Step-wise clustering procedures*, Journal of the American Statistical Association, **62(317)**, pp. 86–101.

[118] Kirkland S, Qiao P & Zhan X, 2016, *Algebraically positive matrices*, Linear Algebra and its Applications, **504**, pp. 14–26.

[119] Kirkpatrick S, Gelatt CD & Vecchi MP, 1983, *Optimisation by simulated annealing*, Research Report RC 9355, International Business Machines, Yorktown (NY).

[120] Kirkpatrick S, Gelatt C & Vecchi M, 1987, *Optimization by simulated annealing*, pp. 339–348 in Mezard M, Parisi G & Virasato M (Eds), *Spin glass theory and beyond: An introduction to the replica method and its applications*, World Scientific Press, Singapore.

[121] Kleijnen JP, 1995, *Verification and validation of simulation models*, European Journal of Operational Research, **82(1)**, pp. 145–162.

[122] Knox WB, Stone P & Breazeal C, 2013, *Training a robot via human feedback: A case study*, Proceedings of the International Conference on Social Robotics, New York (NY), pp. 460–470.

[123] Kober J, Bagnell JA & Peters J, 2013, *Reinforcement learning in robotics: A survey*, International Journal of Robotics Research, **32(11)**, pp. 1238–1274.

[124] Köksalan MM & Sagala PN, 1995, *Interactive approaches for discrete alternative multiple criteria decision making with monotone utility functions*, Management Science, **41(7)**, pp. 1158–1171.

[125] Krantz DH, Luce RD, Suppes P & Tversky A, 1971, *Foundations of Measurement: Volume 1 — Additive and polynomial representations*, Academic Press, New York (NY).

[126] Kumar V, Steinbach M & Tan P, 2017, *Cluster analysis: Basic concepts and algorithms*, [Online], [Cited November 2005], Available from http://www-users.cs.umn.edu/$%5Csim$kumar/dmbook/ch8.pdf.

[127] Land AH & Doig AG, 1960, *An automatic method of solving discrete programming problems*, Econometrica, **28(3)**, pp. 497–520.

[128] Lange T, Roth V, Braun ML & Buhmann JM, 2004, *Stability-based validation of clustering solutions*, Neural Computation, **16(6)**, pp. 1299–1323.

[129] Law A & Kelton WD, 2000, *Simulation modelling and analysis*, 3$^{\text{rd}}$ Edition, McGraw-Hill, Boston (MA).

[130] Law A & David K, 2000, *Simulation modelling and analysis*, 3$^{\text{rd}}$ Edition, McGraw-Hill, Boston (MA).

[131] Law AM, 2008, *How to build valid and credible simulation models*, Proceedings of the 40th Conference on Winter Simulation, Miami (FL), pp. 39–47.

[132] LeCun Y, Kavukcuoglu K & Farabet C, 2010, *Convolutional networks and applications in vision*, Proceedings of the 2010 IEEE International Symposium on Circuits and Systems, Paris, pp. 253–256.

[133] Lee SM, 1972, *Goal programming for decision analysis*, Auerbach, Philadelphia (PA).

[134] Leskinen P, 2000, *Measurement scales and scale independence in the analytic hierarchy process*, Journal of Multi-Criteria Decision Analysis, **9(4)**, pp. 163–174.

[135] Leung Y, Zhang J & Xu Z, 2000, *Clustering by scale-space filtering*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **22(12)**, pp. 1396–1410.

[136] Levene H, 1960, *Robust tests for equality of variances*, Contributions to Probability and Statistics, **1**, pp. 278–292.

[137] Li C & Biswas G, 2002, *Unsupervised learning with mixed numeric and nominal data*, IEEE Transactions on Knowledge and Data Engineering, **14(4)**, pp. 673–690.

[138] Liang P, Jordan MI & Klein D, 2010, *Learning programs: A hierarchical Bayesian approach*, Proceedings of the 27th International Conference on Machine Learning, Berkeley (CA), pp. 639–646.

[139] Linux.org, 2017, *More than letters and words*, [Online], [Cited October 2017], Available from `https://www.linux.org/`.

[140] Liu CL, 1968, *Introduction to combinatorial mathematics*, McGraw-Hill, New York (NY).

[141] Lootsma F, 2007, *Multi-criteria decision analysis via ratio and difference judgement*, Springer, Norwell (MA).

[142] Ma D & Zheng X, 1991, *9/9-9/1 scale method of AHP*, Proceedings of the 2nd International Symposium on the AHP, Pittsburg (PA), pp. 197–202.

[143] Mahdi IM & Alreshaid K, 2005, *Decision support system for selecting the proper project delivery method using analytical hierarchy process (AHP)*, International Journal of Project Management, **23(7)**, pp. 564–572.

[144] Management Study Guide, 2017, *Building a model-driven decision support system*, [Online], [Cited October 2017], Available from `https://managementstudyguide.com/building-model-driven-decision-support-system.htm`.

[145] Management Study Guide, 2017, *Designing a decision support system user interface*, [Online], [Cited October 2017], Available from `https://managementstudyguide.com/designing-decision-support-system-user-interface.htm`.

[146] Marakas GM, 1999, *Decision support system in the twenty-first century*, Prentice Hall, Inc., Upper Saddle River (NJ).

[147] Marinakis Y, Marinaki M & Dounias G, 2010, *A hybrid particle swarm optimization algorithm for the vehicle routing problem*, Engineering Applications of Artificial Intelligence, **23(4)**, pp. 463–472.

[148] Martinez-Cantin R, De Freitas N, Brochu E, Castellanos J & Doucet A, 2009, *A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot*, Autonomous Robots, **27(2)**, pp. 93–103.

[149] McKay MD, Beckman RJ & Conover WJ, 1979, *Comparison of three methods for selecting values of input variables in the analysis of output from a computer code*, Technometrics, **21(2)**, pp. 239–245.

[150] Microsoft, 2017, *MS-DOS overview*, [Online], [Cited October 2017], Available from `https://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/windows%5Cunderline%7B%5C%20%7Ddos%5Cunderline%7B%5C%20%7Doverview.mspx?mfr=true`.

[151] Mitchell TM, 1997, *Machine learning*, McGraw Hill, Burr Ridge (IL).

[152] Modha J, Gwinnett A & Bruce M, 1990, *A review of information systems development methodology (ISDM) selection techniques*, Omega, **18(5)**, pp. 473–490.

[153] Montero J, Fernández-Avilés G & Mateu J, 2015, *Spatial and spatio-temporal geostatistical modeling and Kriging*, John Wiley & Sons, Ltd., Chichester.

[154] Montgomery DC & Runger GC, 2010, *Applied statistics and probability for engineers*, John Wiley & Sons, Hoboken (NJ).

[155] Morton M, 1971, *Management decision systems*, Harvard University Press, Boston (MA).

[156] Munoz A, 2016, *Machine learning and optimization*, [Online], [Cited March 2017], Available from `https://www.cims.nyu.edu/_munoz/files/ml_optimization.pdf`.

[157] Nagy G, 1968, *State of the art in pattern recognition*, Proceedings of the IEEE, **56(5)**, pp. 836–863.

[158] Nahmias S, 1997, *Production and operations analysis*, Irwin, Homewood (IL).

[159] Narang R, 2011, *Database management systems*, 2nd Edition, PHI Learning Pvt. Ltd., New Delhi.

[160] Nel J, 2013, *Shoprite's storage volume problem for fresh fruit and vegetables*, Honours Year Project, Departement of Logistics, Stellenbosch University, Stellenbosch.

[161] *Oxford dictionary of English*, 2010, 3rd Edition, Oxford University Press, Oxford.

[162] Paley P, 1972, *Elements of abstract and linear algebra*, Holt, Rinehart and Winston, Inc., New York (NY).

[163] Papadimitriou CH & Steiglitz K, 1998, *Combinatorial optimization: Algorithms and complexity*, Dover Publications, Inc., New York (NY).

[164] Patil PS, Rao S & Patil SB, 2011, *Data integration problem of structural and semantic heterogenity: Data warehousing framework models for the optimization of the ETL process*, Proceedings of the International Conference and Workshop on Emerging Trends in Technology, Mumbai, pp. 500–504.

[165] Pease A & Colton S, 2011, *On impact and evaluation in computational creativity: A discussion of the Turing test and an alternative proposal*, Proceedings of the AISB Symposium on Artificial Intelligence and Philosophy, York, No page numbers.

[166] Pérez JL & Mokotoff E, 2006, *Another potential shortcoming of AHP*, Top, **14(1)**, pp. 99–111.

[167] Pillai SU, Suel T & Cha S, 2005, *The Perron-Frobenius theorem: Some of its applications*, IEEE Signal Processing Magazine, **22(2)**, pp. 62–75.

[168] Power D, 1996, *What is a DSS?*, On-Line Executive Journal for Data-Intensive Decision Support, **1(3)**, pp. 223–232.

[169] Power D, 2002, *Decision support systems: Concepts and resources for managers*, Quorum Books, Westport (CT).

[170] Quinlan JR, 2014, *C4.5: programs for machine learning*, Elsevier, Burlington (MA).

[171] Qureshi ME & Harrison SR, 2003, *Application of the analytic hierarchy process to riparian revegetation policy options*, Small-scale Forest Economics, Management and Policy, **2(3)**, pp. 441–458.

[172] Rabinowitz G, 1976, *Some comments on measuring world influence*, Journal of Peace Science, **2(1)**, pp. 49–55.

[173] Radford K, 1990, *Information systems for strategic decisions*, Prentice-Hall, Reston (VA).

[174] Rardin RL, 1998, *Optimization in operations research*, Prentice Hall, Upper Saddle River (NJ).

[175] RinnooyKan A & Telgen J, 1981, *The complexity of linear programming*, Statistica Neerlandica, **35(2)**, pp. 91–107.

[176] Rob R & Mohammad A, 2006, *Dilemma between the structured and object-oriented approaches to systems analysis and design*, Journal of Computer Information Systems, **46(3)**, pp. 32–42.

[177] Romeo F & Sangiovanni-Vincentelli A, 1991, *A theoretical framework for simulated annealing*, Algorithmica, **6(1)**, pp. 302–345.

[178] Ross S, 2006, *A first course in probability*, 7th Edition, Pearson / Prentice Hall, Upper Saddle River (NJ).

[179] Roubens M & Vincke P, 1985, *Preference modelling*, Springer Verlag, Berlin.

[180] Royce W, 1970, *Managing the development of large software systems: Concepts and techniques*, Proceedings of the 26th IEEE WESCON Conference, Los Angeles (CA), pp. 1–9.

[181] Russell S & Norvig P, 2009, *Artificial intelligence: A modern approach*, 3rd Edition, Pearson Education, Upper Saddle River (NJ).

[182] Saaty TL, 1977, *A scaling method for priorities in hierarchical structures*, Journal of Mathematical Psychology, **15(3)**, pp. 234–281.

[183] Saaty TL, 1980, *The analytic hierarchy process*, McGraw-Hill, New York (NY).

[184] Saaty TL, 1990, *Eigenvector and logarithmic least squares*, European Journal of Operational Research, **48(1)**, pp. 156–160.

[185] Saaty TL, 2008, *Decision making with the analytic hierarchy process*, International Journal of Services Sciences, **1(1)**, pp. 83–98.

[186] Saaty TL & Hu G, 1998, *Ranking by eigenvector versus other methods in the analytic hierarchy process*, Applied Mathematics Letters, **11(4)**, pp. 121–125.

[187] Sakamoto Y, Ishiguro M & Kitagawa G, 1986, *Akaike information criterion statistics*, Mathematics and its Application, **83(403)**, pp. 902–926.

[188] Salo AA & Hämäläinen RP, 1997, *On the measurement of preferences in the analytic hierarchy process*, Journal of Multi-Criteria Decision Analysis, **6(6)**, pp. 309–319.

[189] Salo AA & Hämäläinen RP, 2001, *Preference ratios in multiattribute evaluation (PRIME)-elicitation and decision procedures under incomplete information*, IEEE Transactions on Systems, Man, and Cybernetics — Part A: Systems and Humans, **31(6)**, pp. 533–545.

[190] Sargent RG, 2005, *Verification and validation of simulation models*, Proceedings of the 37th Conference on Winter Simulation, Orlando (FL), pp. 130–143.

[191] SAS, 2010, *Machine learning: What it is & why it matters*, [Online], [Cited April 2017], Available from `https://www.sas.com/en_za/insights/analytics/machine-learning.html`.

[192] Sauter V, 1997, *Decision support systems*, John Wiley & Sons, Inc., New York (NY).

[193] Schmoldt D, Kangas J, Mendoza G & Pesonen M, 2013, *The analytic hierarchy process in natural resource and environmental decision making*, Springer, Dordrecht.

[194] Schwarz G, 1978, *Estimating the dimension of a model*, Annals of Statistics, **6(2)**, pp. 461–464.

[195] Sebag M, 2015, *Learning with the human in the loop*, [Online], [Cited October 2017], Available from `http://www.lorentzcenter.nl/lc/web/2015/702/presentations/Michele%20Sebag%20-%20Learning%20with%20the%20human%20in%20the%20loop.pdf`.

[196] Sebastiani F, 2002, *Machine learning in automated text categorization*, ACM Computing Surveys, **34(1)**, pp. 1–47.

[197] Sedgewick R & Wayne K, 2011, *Algorithms*, 4^{th} Edition, Addison-Wesley, Westford (MA).

[198] Shamir O & Tishby N, 2007, *Cluster stability for finite samples*, Proceedings of the 21st Annual Conference on Neural Information Processing Systems, Vancouver, pp. 150–158.

[199] Sharma A, 2004, *Understanding color management*, Delmar Learning, Clifton Park (NY).

[200] Shivaswamy P & Joachims T, 2012, *Online structured prediction via coactive learning*, [Online], [Cited July 2017], Available from `https://arxiv.org/abs/1205.4213`.

[201] Sneath PH & Sokal RR, 1973, *Numerical taxonomy — The principles and practice of numerical classification.*, WH Freeman and Company, San Francisco (CA).

[202] Sprague Jr RH & Carlson ED, 1982, *Building effective decision support systems*, Prentice-Hall, Englewood Cliffs (NJ).

[203] Stanhope P, 2002, *Get in the Groove: Building tools and peer-to-peer solutions with the Groove platform*, Hungry Minds, New York (NY).

[204] Steuer RE, 1986, *Multiple criteria optimization: Theory, computation, and applications*, Wiley, New York (NY).

[205] Stonehocker T, Leblang J, Smart J, Ortega R, Manber U & Amacker M, 2006, *Methods and apparatus for automatic generation of recommended links*, [Patent], US Patent 11/096 719, URL: `https://www.google.com/patents/US20060059225`.

[206] Strang H, 1993, *Introduction to linear algebra*, Wellesley-Cambridge Press, Wellesley (MA).

[207] Al-Sultan KS, 1995, *A tabu search approach to the clustering problem*, Pattern Recognition, **28(9)**, pp. 1443–1451.

[208] Sutton RS, 1984, *Temporal credit assignment in reinforcement learning*, PhD Dissertation, University of Massachusetts, Amherst (MA).

[209] Sutton RS & Barto AG, 1998, *Reinforcement learning: An introduction*, MIT Press, Cambridge (MA).

[210] The Open Group, 2017, *What is UNIX?*, [Online], [Cited October 2015], Available from `http://www.unix.org/what%5Cunderline%7B%5C%20%7Dis%5Cunderline%7B%5C%20%7Dunix.html`.

[211] Tibshirani R, Walther G & Hastie T, 2001, *Estimating the number of clusters in a data set via the gap statistic*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), **63(2)**, pp. 411–423.

[212] Tseng LY & Yang SB, 2001, *A genetic approach to the automatic clustering problem*, Pattern Recognition, **34(2)**, pp. 415–424.

[213] Tung AK, Han J, Lakshmanan LV & Ng RT, 2001, *Constraint-based clustering in large databases*, Proceedings of the International Conference on Database Theory, London, pp. 405–419.

[214] Tversky A, 1969, *Intransitivity of preferences*, Psychological Review, **76**, pp. 31–48.

[215] Van der Walt JC, 2017, Masters student in the Department of Industrial Engineering at *Stellenbosch University*, [Personal Communication], Contactable at `johannchristiaanvanderwalt@sun.ac.za`.

[216] Viappiani P & Boutilier C, 2010, *Optimal Bayesian recommendation sets and myopically optimal choice query sets*, pp. 2352–2360 in Lafferty J, Williams CKI, Shawe-Taylor J, Zemel R & Culotta A (Eds), *Advances in neural information processing systems,* **23**, Curran Associates, Inc., New York (NY).

[217] Vigeh A, 2011, *Investigation of a simulated annealing cooling schedule used to optimise the estimation of the fiber diameter distribution in a peripheral nerve trunk*, MSc Thesis, California Polytechnic State University, San Luis Obispo (CA).

[218] Vincke BRP̃, 1987, *Pseudo-orders: Definition, proporties and numerical representation*, Mathematical Social Sciences, **14**, pp. 263–274.

[219] Wallace CS & Freeman PR, 1987, *Estimation and inference by compact coding*, Journal of the Royal Statistical Society: Series B (Statistical Methodological), **49(2)**, pp. 240–265.

[220] Wallace CS & Boulton DM, 1968, *An information measure for classification*, The Computer Journal, **11(2)**, pp. 185–194.

[221] Walsh G, 1975, *Methods of optimization*, John Wiley & Sons, Chichister.

[222] Walters S, Broady J & Hartley R, 1994, *A review of information systems development methodologies*, Library Management, **15(6)**, pp. 5–19.

[223] Ward Jr JH, 1963, *Hierarchical grouping to optimize an objective function*, Journal of the American Statistical Association, **58(301)**, pp. 236–244.

[224] Winston WL, 2004, *Operations research: Applications and algorithms*, 4th Edition, Brooks/Cole Cengage Learning, Belmont (CA).

[225] Wirth C, Furnkranz J & Neumann G, 2016, *Model-free preference-based reinforcement learning*, Proceedings of the 30th AAAI Conference on Artificial Intelligence, Phoenix (AZ), pp. 2222–2228.

[226] Wolfram Research, 2017, *Mathematica, Version 11.2*, [Online], [Cited April 2016], Available from `http://www.wolfram.com/?source=nav`.

[227] Xu R & Wunsch D, 2005, *Survey of clustering algorithms*, IEEE Transactions on Neural Networks, **16(3)**, pp. 645–678.

[228] Zhao Y, Kosorok MR & Zeng D, 2009, *Reinforcement learning design for cancer clinical trials*, Statistics in Medicine, **28(26)**, pp. 3294–3315.

APPENDIX A

DSS Validation Data

This appendix contains the pairwise comparison matrices returned by the DMs in the case studies conducted in Chapter 9. Figures A.1–A.12 contain the pairwise comparison matrices supplied by Mr van der Walt for the case study of §9.1, while the pairwise comparison matrices returned by Mr du Plessis for the case study of §9.2 may be found in Figures A.13–A.16.

$$\boldsymbol{A}_0^0 = \begin{array}{c|cccccccc} & \boldsymbol{x}^0_{0,1} & \boldsymbol{x}^0_{0,2} & \boldsymbol{x}^0_{0,3} & \boldsymbol{x}^0_{0,4} & \boldsymbol{x}^0_{0,5} & \boldsymbol{x}^0_{0,6} & \boldsymbol{x}^0_{0,7} & \boldsymbol{a}^0 \\ \hline \boldsymbol{x}^0_{0,1} & 1 & 4 & \frac{1}{4} & 16 & 64 & \frac{1}{4} & \frac{1}{4} & \frac{1}{16} \\ \boldsymbol{x}^0_{0,2} & \frac{1}{4} & 1 & \frac{1}{9} & \frac{1}{4} & \frac{1}{4} & \frac{1}{9} & \frac{1}{9} & \frac{1}{25} \\ \boldsymbol{x}^0_{0,3} & 4 & 9 & 1 & 9 & \frac{1}{9} & \frac{1}{4} & \frac{1}{4} & \frac{1}{16} \\ \boldsymbol{x}^0_{0,4} & \frac{1}{16} & 4 & \frac{1}{9} & 1 & 9 & 16 & 9 & \frac{1}{9} \\ \boldsymbol{x}^0_{0,5} & \frac{1}{64} & 4 & 9 & 9 & 1 & 4 & 4 & \frac{1}{9} \\ \boldsymbol{x}^0_{0,6} & 4 & 9 & 4 & 4 & \frac{1}{4} & 1 & 4 & \frac{1}{9} \\ \boldsymbol{x}^0_{0,7} & 4 & 9 & 4 & 4 & \frac{1}{4} & \frac{1}{4} & 1 & \frac{1}{9} \\ \boldsymbol{a}^0 & 16 & 25 & 16 & 16 & 9 & 9 & 9 & 1 \end{array}$$

FIGURE A.1: *Pairwise comparison matrix provided by Mr van der Walt during Iteration 0 of the learning cycle for the case where blue = 0 in the case study of §9.1.*

$$\boldsymbol{A}_0^{85} = \begin{array}{c|cccccccc} & \boldsymbol{x}^{85}_{0,1} & \boldsymbol{x}^{85}_{0,2} & \boldsymbol{x}^{85}_{0,3} & \boldsymbol{x}^{85}_{0,4} & \boldsymbol{x}^{85}_{0,5} & \boldsymbol{x}^{85}_{0,6} & \boldsymbol{x}^{85}_{0,7} & \boldsymbol{a}^{85} \\ \hline \boldsymbol{x}^{85}_{0,1} & 1 & \frac{1}{9} & 9 & \frac{1}{36} & \frac{1}{16} & 1 & 9 & 4 \\ \boldsymbol{x}^{85}_{0,2} & 9 & 1 & 9 & \frac{1}{49} & \frac{1}{25} & 4 & 9 & 4 \\ \boldsymbol{x}^{85}_{0,3} & \frac{1}{9} & \frac{1}{9} & 1 & \frac{1}{49} & \frac{1}{25} & \frac{1}{4} & 1 & \frac{1}{9} \\ \boldsymbol{x}^{85}_{0,4} & 36 & 49 & 49 & 1 & 16 & 36 & 81 & 49 \\ \boldsymbol{x}^{85}_{0,5} & 16 & 25 & 25 & \frac{1}{16} & 1 & 25 & 81 & 49 \\ \boldsymbol{x}^{85}_{0,6} & 1 & \frac{1}{4} & 4 & \frac{1}{36} & \frac{1}{25} & 1 & 4 & 4 \\ \boldsymbol{x}^{85}_{0,7} & \frac{1}{9} & \frac{1}{9} & 1 & \frac{1}{81} & \frac{1}{81} & \frac{1}{4} & 1 & \frac{1}{9} \\ \boldsymbol{a}^{85} & \frac{1}{4} & \frac{1}{4} & 9 & \frac{1}{49} & \frac{1}{49} & \frac{1}{4} & 9 & 1 \end{array}$$

FIGURE A.2: *Pairwise comparison matrix provided by Mr van der Walt during Iteration 0 of the learning cycle for the case where blue = 85 in the case study of §9.1.*

$$\boldsymbol{A}_0^{170} = \begin{array}{c} \\ \boldsymbol{x}_{0,1}^{170} \\ \boldsymbol{x}_{0,2}^{170} \\ \boldsymbol{x}_{0,3}^{170} \\ \boldsymbol{x}_{0,4}^{170} \\ \boldsymbol{x}_{0,5}^{170} \\ \boldsymbol{x}_{0,6}^{170} \\ \boldsymbol{x}_{0,7}^{170} \\ \boldsymbol{a}^{170} \end{array} \begin{array}{c} \begin{array}{cccccccc} \boldsymbol{x}_{0,1}^{170} & \boldsymbol{x}_{0,2}^{170} & \boldsymbol{x}_{0,3}^{170} & \boldsymbol{x}_{0,4}^{170} & \boldsymbol{x}_{0,5}^{170} & \boldsymbol{x}_{0,6}^{170} & \boldsymbol{x}_{0,7}^{170} & \boldsymbol{a}^{170} \end{array} \\ \left[\begin{array}{cccccccc} 1 & \frac{1}{9} & 49 & \frac{1}{36} & 9 & 16 & 64 & 49 \\ 9 & 1 & \frac{1}{49} & \frac{1}{25} & 4 & 16 & 64 & 64 \\ 49 & 49 & 1 & 25 & 49 & 81 & 81 & 81 \\ 36 & 25 & \frac{1}{25} & 1 & 16 & 36 & 64 & 64 \\ \frac{1}{9} & \frac{1}{4} & \frac{1}{49} & \frac{1}{16} & 1 & 4 & 16 & 9 \\ \frac{1}{16} & \frac{1}{16} & \frac{1}{81} & \frac{1}{36} & \frac{1}{4} & 1 & 16 & 16 \\ \frac{1}{64} & \frac{1}{64} & \frac{1}{81} & \frac{1}{64} & \frac{1}{16} & \frac{1}{16} & 1 & 1 \\ \frac{1}{49} & \frac{1}{64} & \frac{1}{81} & \frac{1}{64} & \frac{1}{9} & \frac{1}{16} & 1 & 1 \end{array}\right] \end{array}$$

FIGURE A.3: *Pairwise comparison matrix provided by Mr van der Walt during Iteration 0 of the learning cycle for the case where blue = 170 in the case study of §9.1.*

$$\boldsymbol{A}_0^{255} = \begin{array}{c} \\ \boldsymbol{x}_{0,1}^{255} \\ \boldsymbol{x}_{0,2}^{255} \\ \boldsymbol{x}_{0,3}^{255} \\ \boldsymbol{x}_{0,4}^{255} \\ \boldsymbol{x}_{0,5}^{255} \\ \boldsymbol{x}_{0,6}^{255} \\ \boldsymbol{x}_{0,7}^{255} \\ \boldsymbol{a}^{255} \end{array} \begin{array}{c} \begin{array}{cccccccc} \boldsymbol{x}_{0,1}^{255} & \boldsymbol{x}_{0,2}^{255} & \boldsymbol{x}_{0,3}^{255}55 & \boldsymbol{x}_{0,4}^{255} & \boldsymbol{x}_{0,5}^{255} & \boldsymbol{x}_{0,6}^{255} & \boldsymbol{x}_{0,7}^{255} & \boldsymbol{a}^{255} \end{array} \\ \left[\begin{array}{cccccccc} 1 & \frac{1}{4} & \frac{1}{16} & \frac{1}{9} & 9 & \frac{1}{9} & 16 & 9 \\ 4 & 1 & 9 & 16 & 36 & 25 & 81 & 81 \\ 16 & 16 & 1 & 4 & 25 & 16 & 81 & 81 \\ 9 & 9 & \frac{1}{4} & 1 & 36 & \frac{1}{9} & 49 & 49 \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{25} & \frac{1}{36} & 1 & \frac{1}{9} & 9 & 9 \\ 9 & 9 & \frac{1}{16} & 9 & 9 & 1 & 49 & 49 \\ \frac{1}{16} & \frac{1}{16} & \frac{1}{81} & \frac{1}{49} & \frac{1}{9} & \frac{1}{49} & 1 & \frac{1}{4} \\ 9 & \frac{1}{9} & \frac{1}{81} & \frac{1}{49} & \frac{1}{9} & \frac{1}{49} & 4 & 1 \end{array}\right] \end{array}$$

FIGURE A.4: *Pairwise comparison matrix provided by Mr van der Walt during Iteration 0 of the learning cycle for the case where blue = 255 in the case study of §9.1.*

$$\boldsymbol{A}_1^{0} = \begin{array}{c} \\ \boldsymbol{x}_{1,1}^{0} \\ \boldsymbol{x}_{1,2}^{0} \\ \boldsymbol{x}_{1,3}^{0} \\ \boldsymbol{x}_{1,4}^{0} \\ \boldsymbol{x}_{1,5}^{0} \\ \boldsymbol{x}_{1,6}^{0} \\ \boldsymbol{x}_{1,7}^{0} \\ \boldsymbol{a}^{0} \end{array} \begin{array}{c} \begin{array}{cccccccc} \boldsymbol{x}_{1,1}^{0} & \boldsymbol{x}_{1,2}^{0} & \boldsymbol{x}_{1,3}^{0} & \boldsymbol{x}_{1,4}^{0} & \boldsymbol{x}_{1,5}^{0} & \boldsymbol{x}_{1,6}^{0} & \boldsymbol{x}_{1,7}^{0} & \boldsymbol{a}^{0} \end{array} \\ \left[\begin{array}{cccccccc} 1 & 4 & 4 & 9 & 25 & 36 & 36 & 25 \\ \frac{1}{4} & 1 & \frac{1}{4} & \frac{1}{4} & 4 & 4 & 4 & 4 \\ \frac{1}{4} & 4 & 1 & 9 & 16 & 16 & 16 & 1 \\ \frac{1}{9} & 4 & \frac{1}{9} & 1 & 4 & 4 & 4 & \frac{1}{9} \\ \frac{1}{25} & \frac{1}{4} & \frac{1}{16} & \frac{1}{4} & 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{36} & \frac{1}{4} & \frac{1}{16} & \frac{1}{4} & 4 & 1 & 1 & \frac{1}{4} \\ \frac{1}{36} & \frac{1}{4} & \frac{1}{16} & \frac{1}{4} & 4 & 1 & 1 & \frac{1}{4} \\ \frac{1}{25} & \frac{1}{4} & 1 & \frac{1}{4} & 4 & 4 & 4 & 1 \end{array}\right] \end{array}$$

FIGURE A.5: *Pairwise comparison matrix provided by Mr van der Walt during Iteration 1 of the learning cycle for the case where blue = 0 in the case study of §9.1.*

$$\boldsymbol{A}_1^{85} = \begin{array}{c|cccccccc|}
 & \boldsymbol{x}_{1,1}^{85} & \boldsymbol{x}_{1,2}^{85} & \boldsymbol{x}_{1,3}^{85} & \boldsymbol{x}_{1,4}^{85} & \boldsymbol{x}_{1,5}^{85} & \boldsymbol{x}_{1,6}^{85} & \boldsymbol{x}_{1,7}^{85} & \boldsymbol{a}^{85} \\
\boldsymbol{x}_{1,1}^{85} & 1 & \frac{1}{36} & 4 & 9 & \frac{1}{4} & \frac{1}{4} & 4 & 4 \\
\boldsymbol{x}_{1,2}^{85} & 36 & 1 & 64 & 81 & 64 & 64 & 81 & 81 \\
\boldsymbol{x}_{1,3}^{85} & \frac{1}{4} & \frac{1}{64} & 1 & 4 & \frac{1}{4} & \frac{1}{4} & 4 & 4 \\
\boldsymbol{x}_{1,4}^{85} & \frac{1}{9} & \frac{1}{81} & \frac{1}{4} & 1 & \frac{1}{16} & \frac{1}{16} & \frac{1}{4} & \frac{1}{4} \\
\boldsymbol{x}_{1,5}^{85} & 4 & \frac{1}{64} & 4 & 16 & 1 & 4 & 9 & 16 \\
\boldsymbol{x}_{1,6}^{85} & 4 & \frac{1}{64} & 4 & \frac{1}{16} & \frac{1}{4} & 1 & 16 & 16 \\
\boldsymbol{x}_{1,7}^{85} & \frac{1}{4} & \frac{1}{81} & \frac{1}{4} & 4 & \frac{1}{9} & \frac{1}{16} & 1 & \frac{1}{4} \\
\boldsymbol{a}^{85} & \frac{1}{4} & \frac{1}{81} & \frac{1}{4} & 4 & \frac{1}{16} & \frac{1}{16} & 4 & 1
\end{array}$$

FIGURE A.6: *Pairwise comparison matrix provided by Mr van der Walt during Iteration 1 of the learning cycle for the case where blue = 85 in the case study of §9.1.*

$$\boldsymbol{A}_1^{170} = \begin{array}{c|cccccccc|}
 & \boldsymbol{x}_{1,1}^{170} & \boldsymbol{x}_{1,2}^{170} & \boldsymbol{x}_{1,3}^{170} & \boldsymbol{x}_{1,4}^{170} & \boldsymbol{x}_{1,5}^{170} & \boldsymbol{x}_{1,6}^{170} & \boldsymbol{x}_{1,7}^{170} & \boldsymbol{a}^{170} \\
\boldsymbol{x}_{1,1}^{170} & 1 & 4 & \frac{1}{49} & \frac{1}{9} & \frac{1}{9} & \frac{1}{16} & \frac{1}{4} & 4 \\
\boldsymbol{x}_{1,2}^{170} & \frac{1}{4} & 1 & \frac{1}{81} & \frac{1}{49} & \frac{1}{49} & \frac{1}{64} & \frac{1}{36} & 1 \\
\boldsymbol{x}_{1,3}^{170} & 49 & 81 & 1 & 9 & 9 & 9 & 36 & 81 \\
\boldsymbol{x}_{1,4}^{170} & 9 & 49 & \frac{1}{9} & 1 & 1 & 1 & 25 & 81 \\
\boldsymbol{x}_{1,5}^{170} & 9 & 49 & \frac{1}{9} & 1 & 1 & 1 & 25 & 81 \\
\boldsymbol{x}_{1,6}^{170} & 16 & 64 & \frac{1}{9} & 1 & 1 & 1 & 9 & 81 \\
\boldsymbol{x}_{1,7}^{170} & 4 & 36 & \frac{1}{36} & \frac{1}{25} & \frac{1}{25} & \frac{1}{9} & 1 & 25 \\
\boldsymbol{a}^{170} & \frac{1}{4} & 1 & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{25} & 1
\end{array}$$

FIGURE A.7: *Pairwise comparison matrix provided by Mr van der Walt during Iteration 1 of the learning cycle for the case where blue = 170 in the case study of §9.1.*

$$\boldsymbol{A}_1^{255} = \begin{array}{c|cccccccc|}
 & \boldsymbol{x}_{1,1}^{255} & \boldsymbol{x}_{1,2}^{255} & \boldsymbol{x}_{1,3}^{255} & \boldsymbol{x}_{1,4}^{255} & \boldsymbol{x}_{1,5}^{255} & \boldsymbol{x}_{1,6}^{255} & \boldsymbol{x}_{1,7}^{255} & \boldsymbol{a}^{255} \\
\boldsymbol{x}_{1,1}^{255} & 1 & \frac{1}{4} & \frac{1}{36} & \frac{1}{81} & \frac{1}{81} & \frac{1}{25} & \frac{1}{49} & 9 \\
\boldsymbol{x}_{1,2}^{255} & 4 & 1 & \frac{1}{49} & \frac{1}{64} & \frac{1}{81} & \frac{1}{25} & \frac{1}{36} & 36 \\
\boldsymbol{x}_{1,3}^{255} & 36 & 49 & 1 & \frac{1}{4} & \frac{1}{36} & 16 & 36 & 81 \\
\boldsymbol{x}_{1,4}^{255} & 81 & 64 & 4 & 1 & \frac{1}{9} & 64 & 49 & 81 \\
\boldsymbol{x}_{1,5}^{255} & 81 & 81 & 36 & 9 & 1 & 81 & 81 & 81 \\
\boldsymbol{x}_{1,6}^{255} & 25 & 25 & \frac{1}{16} & \frac{1}{64} & \frac{1}{81} & 1 & 4 & 49 \\
\boldsymbol{x}_{1,7}^{255} & 49 & 36 & \frac{1}{36} & \frac{1}{49} & \frac{1}{81} & \frac{1}{4} & 1 & 81 \\
\boldsymbol{a}^{255} & 9 & \frac{1}{36} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{49} & \frac{1}{81} & 1
\end{array}$$

FIGURE A.8: *Pairwise comparison matrix provided by Mr van der Walt during Iteration 1 of the learning cycle for the case where blue = 255 in the case study of §9.1.*

$$\boldsymbol{A}_2^0 = \begin{array}{c|cccccccc} & \boldsymbol{x}_{2,1}^0 & \boldsymbol{x}_{2,2}^0 & \boldsymbol{x}_{2,3}^0 & \boldsymbol{x}_{2,4}^0 & \boldsymbol{x}_{2,5}^0 & \boldsymbol{x}_{2,6}^0 & \boldsymbol{x}_{2,7}^0 & \boldsymbol{a}^0 \\ \hline \boldsymbol{x}_{2,1}^0 & 1 & 4 & 4 & 4 & 4 & 4 & 4 & \frac{1}{4} \\ \boldsymbol{x}_{2,2}^0 & \frac{1}{4} & 1 & \frac{1}{4} & 4 & 4 & 4 & 4 & \frac{1}{4} \\ \boldsymbol{x}_{2,3}^0 & \frac{1}{4} & 4 & 1 & 4 & 4 & 4 & 4 & \frac{1}{4} \\ \boldsymbol{x}_{2,4}^0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 1 & \frac{1}{4} & 1 & \frac{1}{4} & \frac{1}{4} \\ \boldsymbol{x}_{2,5}^0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 4 & 1 & 1 & 1 & \frac{1}{4} \\ \boldsymbol{x}_{2,6}^0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 1 & 1 & 1 & 1 & \frac{1}{4} \\ \boldsymbol{x}_{2,7}^0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 4 & 1 & 1 & 1 & \frac{1}{4} \\ \boldsymbol{a}^0 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 1 \end{array}$$

FIGURE A.9: *Pairwise comparison matrix provided by Mr van der Walt during Iteration 2 of the learning cycle for the case where blue = 0 in the case study of §9.1.*

$$\boldsymbol{A}_2^{85} = \begin{array}{c|cccccccc} & \boldsymbol{x}_{2,1}^{85} & \boldsymbol{x}_{2,2}^{85} & \boldsymbol{x}_{2,3}^{85} & \boldsymbol{x}_{2,4}^{85} & \boldsymbol{x}_{2,5}^{85} & \boldsymbol{x}_{2,6}^{85} & \boldsymbol{x}_{2,7}^{85} & \boldsymbol{a}^{85} \\ \hline \boldsymbol{x}_{2,1}^{85} & 1 & 9 & 36 & \frac{1}{25} & \frac{1}{25} & \frac{1}{16} & \frac{1}{9} & 16 \\ \boldsymbol{x}_{2,2}^{85} & \frac{1}{9} & 1 & 4 & \frac{1}{25} & \frac{1}{16} & \frac{1}{25} & \frac{1}{36} & 1 \\ \boldsymbol{x}_{2,3}^{85} & \frac{1}{36} & \frac{1}{4} & 1 & \frac{1}{64} & \frac{1}{64} & \frac{1}{64} & \frac{1}{64} & \frac{1}{49} \\ \boldsymbol{x}_{2,4}^{85} & 25 & 25 & 64 & 1 & \frac{1}{16} & 1 & \frac{1}{4} & 25 \\ \boldsymbol{x}_{2,5}^{85} & 25 & 16 & 64 & 16 & 1 & 4 & \frac{1}{4} & 36 \\ \boldsymbol{x}_{2,6}^{85} & 16 & 25 & 64 & 1 & \frac{1}{4} & 1 & 9 & 49 \\ \boldsymbol{x}_{2,7}^{85} & 9 & 36 & 64 & 4 & 4 & \frac{1}{9} & 1 & 36 \\ \boldsymbol{a}^{85} & \frac{1}{16} & 1 & 49 & \frac{1}{36} & \frac{1}{36} & \frac{1}{49} & \frac{1}{36} & 1 \end{array}$$

FIGURE A.10: *Pairwise comparison matrix provided by Mr van der Walt during Iteration 2 of the learning cycle for the case where blue = 85 in the case study of §9.1.*

$$\boldsymbol{A}_2^{170} = \begin{array}{c|cccccccc} & \boldsymbol{x}_{2,1}^{170} & \boldsymbol{x}_{2,2}^{170} & \boldsymbol{x}_{2,3}^{170} & \boldsymbol{x}_{2,4}^{170} & \boldsymbol{x}_{2,5}^{170} & \boldsymbol{x}_{2,6}^{170} & \boldsymbol{x}_{2,7}^{170} & \boldsymbol{a}^{170} \\ \hline \boldsymbol{x}_{2,1}^{170} & 1 & 25 & \frac{1}{9} & 16 & 36 & 36 & \frac{1}{9} & 81 \\ \boldsymbol{x}_{2,2}^{170} & \frac{1}{25} & 1 & \frac{1}{25} & \frac{1}{16} & \frac{1}{4} & \frac{1}{9} & \frac{1}{25} & 49 \\ \boldsymbol{x}_{2,3}^{170} & 9 & 25 & 1 & 25 & 36 & 16 & 4 & 81 \\ \boldsymbol{x}_{2,4}^{170} & \frac{1}{16} & 16 & \frac{1}{25} & 1 & 9 & 16 & \frac{1}{36} & 81 \\ \boldsymbol{x}_{2,5}^{170} & \frac{1}{36} & 4 & \frac{1}{36} & \frac{1}{9} & 1 & \frac{1}{9} & \frac{1}{36} & 16 \\ \boldsymbol{x}_{2,6}^{170} & \frac{1}{36} & 9 & \frac{1}{16} & \frac{1}{16} & 9 & 1 & \frac{1}{36} & 49 \\ \boldsymbol{x}_{2,7}^{170} & 9 & 25 & \frac{1}{4} & 36 & 36 & 36 & 1 & 81 \\ \boldsymbol{a}^{170} & \frac{1}{81} & \frac{1}{49} & \frac{1}{81} & \frac{1}{81} & \frac{1}{16} & \frac{1}{49} & \frac{1}{81} & 1 \end{array}$$

FIGURE A.11: *Pairwise comparison matrix provided by Mr van der Walt during Iteration 2 of the learning cycle for the case where blue = 170 in the case study of §9.1.*

$$\boldsymbol{A}_2^{255} = \begin{array}{c|cccccccc}
 & \boldsymbol{x}_{2,1}^{255} & \boldsymbol{x}_{2,2}^{255} & \boldsymbol{x}_{2,3}^{255} & \boldsymbol{x}_{2,4}^{255} & \boldsymbol{x}_{2,5}^{255} & \boldsymbol{x}_{2,6}^{255} & \boldsymbol{x}_{2,7}^{255} & \boldsymbol{a}^{255} \\
\hline
\boldsymbol{x}_{2,1}^{255} & 1 & \frac{1}{16} & \frac{1}{9} & \frac{1}{81} & \frac{1}{64} & \frac{1}{25} & \frac{1}{49} & 25 \\
\boldsymbol{x}_{2,2}^{255} & 16 & 1 & 25 & \frac{1}{64} & \frac{1}{9} & 9 & \frac{1}{16} & 64 \\
\boldsymbol{x}_{2,3}^{255} & 9 & \frac{1}{25} & 1 & \frac{1}{64} & \frac{1}{49} & \frac{1}{4} & \frac{1}{49} & 81 \\
\boldsymbol{x}_{2,4}^{255} & 81 & 64 & 64 & 1 & 64 & 81 & 81 & 81 \\
\boldsymbol{x}_{2,5}^{255} & 64 & 9 & 49 & \frac{1}{64} & 1 & 81 & 49 & 81 \\
\boldsymbol{x}_{2,6}^{255} & 25 & \frac{1}{9} & 4 & \frac{1}{81} & \frac{1}{81} & 1 & \frac{1}{49} & 81 \\
\boldsymbol{x}_{2,7}^{255} & 49 & 16 & 49 & \frac{1}{81} & \frac{1}{49} & 49 & 1 & 81 \\
\boldsymbol{a}^{255} & \frac{1}{25} & \frac{1}{64} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & 1
\end{array}$$

FIGURE A.12: *Pairwise comparison matrix provided by Mr van der Walt during Iteration 2 of the learning cycle for the case where blue = 255 in the case study of §9.1.*

$$\boldsymbol{A}_0 = \begin{array}{c|cccccccccc}
 & \boldsymbol{x}_{0,1} & \boldsymbol{x}_{0,2} & \boldsymbol{x}_{0,3} & \boldsymbol{a}_1 & \boldsymbol{x}_{0,4} & \boldsymbol{x}_{0,5} & \boldsymbol{x}_{0,6} & \boldsymbol{x}_{0,7} & \boldsymbol{x}_{0,8} & \boldsymbol{x}_{0,9} \\
\hline
\boldsymbol{x}_{0,1} & 0 & 1 & 2 & 3 & 3 & 4 & 5 & 5 & 6 & 6 \\
\boldsymbol{x}_{0,2} & 0 & 0 & 1 & 2 & 2 & 3 & 4 & 4 & 5 & 5 \\
\boldsymbol{x}_{0,3} & 0 & 0 & 0 & 1 & 2 & 3 & 3 & 4 & 5 & 5 \\
\boldsymbol{a}_1 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 3 & 4 & 5 \\
\boldsymbol{x}_{0,4} & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 4 & 5 \\
\boldsymbol{x}_{0,5} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 4 \\
\boldsymbol{x}_{0,6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 3 \\
\boldsymbol{x}_{0,7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 \\
\boldsymbol{x}_{0,8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\boldsymbol{x}_{0,9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

FIGURE A.13: *Pairwise comparison matrix provided by Mr du Plessis during Iteration 0 of the learning cycle in the case study of §9.2.*

$$\boldsymbol{A}_1 = \begin{array}{c|cccccccccc}
 & \boldsymbol{x}_{1,1} & \boldsymbol{x}_{1,2} & \boldsymbol{x}_{1,3} & \boldsymbol{x}_{1,4} & \boldsymbol{x}_{1,5} & \boldsymbol{x}_{1,6} & \boldsymbol{x}_{1,7} & \boldsymbol{x}_{1,8} & \boldsymbol{a}_1 & \boldsymbol{x}_{1,9} \\
\hline
\boldsymbol{x}_{1,1} & 0 & 1 & 2 & 3 & 3 & 4 & 4 & 5 & 6 & 6 \\
\boldsymbol{x}_{1,2} & 0 & 0 & 2 & 3 & 3 & 4 & 4 & 5 & 6 & 6 \\
\boldsymbol{x}_{1,3} & 0 & 0 & 0 & 1 & 2 & 3 & 3 & 4 & 5 & 5 \\
\boldsymbol{x}_{1,4} & 0 & 0 & 0 & 0 & 2 & 3 & 3 & 5 & 5 & 5 \\
\boldsymbol{x}_{1,5} & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 3 & 5 & 5 \\
\boldsymbol{x}_{1,6} & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 5 & 5 \\
\boldsymbol{x}_{1,7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 4 \\
\boldsymbol{x}_{1,8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 \\
\boldsymbol{a}_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\
\boldsymbol{x}_{1,9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

FIGURE A.14: *Pairwise comparison matrix provided by Mr du Plessis during Iteration 1 of the learning cycle in the case study of §9.2.*

$$\boldsymbol{A}_2 = \begin{array}{c|cccccccccc|} & \boldsymbol{x}_{2,1} & \boldsymbol{x}_{2,2} & \boldsymbol{x}_{2,3} & \boldsymbol{x}_{2,4} & \boldsymbol{x}_{2,5} & \boldsymbol{x}_{2,6} & \boldsymbol{x}_{2,7} & \boldsymbol{x}_{2,8} & \boldsymbol{a}_1 & \boldsymbol{x}_{2,9} \\ \boldsymbol{x}_{2,1} & 0 & 1 & 2 & 3 & 3 & 4 & 4 & 5 & 6 & 6 \\ \boldsymbol{x}_{2,2} & 0 & 0 & 1 & 2 & 2 & 4 & 4 & 5 & 5 & 6 \\ \boldsymbol{x}_{2,3} & 0 & 0 & 0 & 2 & 2 & 4 & 4 & 5 & 5 & 6 \\ \boldsymbol{x}_{2,4} & 0 & 0 & 0 & 0 & 1 & 4 & 4 & 4 & 5 & 6 \\ \boldsymbol{x}_{2,5} & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 3 & 5 \\ \boldsymbol{x}_{2,6} & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 5 \\ \boldsymbol{x}_{2,7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 5 \\ \boldsymbol{x}_{2,8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 5 \\ \boldsymbol{a}_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \\ \boldsymbol{x}_{2,9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

FIGURE A.15: *Pairwise comparison matrix provided by Mr du Plessis during Iteration 2 of the learning cycle in the case study of §9.2.*

$$\boldsymbol{A}_3 = \begin{array}{c|cccccccccc|} & \boldsymbol{x}_{3,1} & \boldsymbol{x}_{3,2} & \boldsymbol{x}_{3,3} & \boldsymbol{x}_{3,4} & \boldsymbol{x}_{3,5} & \boldsymbol{a}_1 & \boldsymbol{x}_{3,6} & \boldsymbol{x}_{3,7} & \boldsymbol{x}_{3,8} & \boldsymbol{x}_{3,9} \\ \boldsymbol{x}_{3,1} & 0 & 1 & 1 & 2 & 2 & 3 & 5 & 6 & 6 & 6 \\ \boldsymbol{x}_{3,2} & 0 & 0 & 1 & 1 & 2 & 3 & 3 & 5 & 5 & 5 \\ \boldsymbol{x}_{3,3} & 0 & 0 & 0 & 1 & 1 & 3 & 4 & 5 & 5 & 5 \\ \boldsymbol{x}_{3,4} & 0 & 0 & 0 & 0 & 1 & 3 & 4 & 5 & 5 & 5 \\ \boldsymbol{x}_{3,5} & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 5 & 5 & 5 \\ \boldsymbol{a}_1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 4 & 4 & 5 \\ \boldsymbol{x}_{3,6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 4 \\ \boldsymbol{x}_{3,7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 3 \\ \boldsymbol{x}_{3,8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ \boldsymbol{x}_{3,9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

FIGURE A.16: *Pairwise comparison matrix provided by Mr du Plessis during Iteration 3 of the learning cycle in the case study of §9.2.*