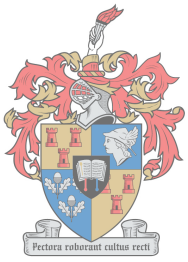
Optimality in Sewer Network Design

by

Nico-Ben de Villiers



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

Supervisor: Dr. G.C. Van Rooyen
Co-supervisor: Dr. A.N. Sinske

March 2018

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: ..... March 2018 .....

Abstract

Optimality in Sewer Network Design

N. de Villiers

Department of Civil Engineering,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Dissertation: PhD (Civil Engineering Informatics)

December 2017

Sewer networks form a vital part of urban infrastructure and represent a capital intensive expense in the public sector. In South Africa, a recent estimation stated that R44.75 billion is required to provide basic sanitation services to those in need, while the budget allocation for sanitation was approximately R3.2 billion nation wide. This state of affairs is not limited to South Africa alone. It is clear that funds allocated to sanitation have to be spent as effectively as possible.

It is within this space that this dissertation makes a contribution, by exploiting modern digital computing power and optimization techniques to find more cost effective solutions to sewer network design and analysis problems. This investigation proposes and evaluates new solutions that address critical shortcomings in existing sewer network optimization algorithms.

The distinct lack of adequate benchmark problems to test and compare the performance of new sewer network optimization algorithms is addressed and a thorough investigation into the nature of optimal solutions is made by capturing trial solutions of optimization algorithms at regular intervals. The characteristics of the solutions are quantified through proposed network characteristic parameters.The evolution of trial solution characteristics allow insight and understanding into how the network changed over the course of the algorithm's life cycle. This information is then exploited to propose heuristic influence factors which guide the optimization procedures toward decisions favouring characteristics which have

demonstrated good solution fitness.

In this way the proposed sewer network optimization algorithms, which are shown to perform better than the algorithms it took inspiration from, are further improved through the use of heuristics. The heuristics, and their formulation, reveal characteristics of optimal sewer network designs which can be used by future researchers as well as design engineers, to improve their solutions.

Uittreksel

Optimaliteit in Rioolnetwerk Ontwerp *("Optimality in Sewer Network Design")*

N. de Villiers

Departement Siviele Ingenieurswese,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.

Proefskrif: PhD(Siviele Ingenieurs Informatika)

Desember 2017

Rioolnetwerke vorm 'n belangrike deel van stedelike infrastruktuur en verteenwoordig 'n kapitale intensiewe koste in die openbare sektor. In Suid-Afrika, verklaar 'n onlangse beraming dat R44.75 miljard nodig is om basiese sanitasiedienste te verskaf aan diegene in nood, terwyl die begrotingstoewysing vir sanitasie nagenoeg R3.2 miljard landwyd was. Hierdie stand van sake is nie net beperk tot Suid-Afrika nie. Dit is duidelik dat fondse toegewys aan sanitasie so effektief as moontlik bestee moet word.

Dit is binne hierdie spasie dat hierdie proefskrif 'n bydrae lewer, deur die gebruik van moderne digitale rekenkrag en optimaliseringstegnieke om meer kosteeffektiewe oplossings vir rioolnetwerkontwerp en ontledingsprobleme te vind. Hierdie ondersoek stel en evalueer nuwe oplossings wat kritieke tekortkominge aanspreek in bestaande rioolnetwerk optimalisering algoritmes.

Die duidelike gebrek aan voldoende maatstafprobleme om die prestasie van nuwe rioolnetwerk optimalisering algoritmes te toets en te vergelyk word aangespreek. 'n Deeglike ondersoek na die aard van optimale oplossings word gemaak deur proefoplossings van optimaliseringsalgoritmes vas te lÃª met gereelde tussenposes. Die eienskappe van die oplossings word gekwantifiseer deur die voorgestelde netwerk eienskap parameters. Die evolusie van proefoplossing eienskappe laat insig en begrip toe in hoe die netwerk verander in die loop van die algoritme se lewensiklus. Hierdie inligting is dan uitgebuit om heuristiese invloedsfaktore voor te stel wat die optimalisering prosedures dryf na besluite wat gedemonstreerde

eienskappe bevoordeel.

Op hierdie manier word die voorgestelde rioolnetwerk optimalisering algoritmes, wat getoon word om beter te presteer as die algoritmes wat dit uit inspirasie geneem het, verder verbeter deur die gebruik van heuristiek. Die heuristiek, en hul formulering, onthul eienskappe van optimale rioolnetwerk ontwerpe wat deur toekomstige navorsers, sowel as ontwerp ingenieurs, gebruik kan word om hul oplossings te verbeter.

Acknowledgements

There are so many people who have contributed in some way toward this dissertation. To anyone who has ever leant an ear or said a word of encouragement: My deepest thank you, you have each in some way contributed to the success of the dissertation.

I must express my deepest gratitude and appreciation to my parents, Ben and Carika de Villiers. Human language does not offer adequate vocabulary to express to you how thankful I am for the life and opportunities you have given me. Without you none of this would be possible. Your words of encouragement, shared excitement and pride in me kept me motivated throughout my academic journey and I will forever be grateful to you. To my siblings, Carin, Pierre and Danie. Your words of encouragement, every time you asked how things were progressing, every time you listened to me complain was and is immensely appreciated.

To my academic mentor and thesis supervisor, Dr G.C. Van Rooyen. I can not imagine so much as undertaking this project without your guidance, let alone completing it. You have taught me more about engineering and informatics than I imagined possible. Your pragmatic approach to problem solving has impressed, inspired and at times out right amazed me. Your vast knowledge and collected thoughts are a constant source of inspiration. I am forever grateful to you.

To Prof M. Middendorf for affording me the opportunity to study abroad and for offering your own valuable time to help with my research.

To Dr A.N. Sinske and the people at GLS Consulting. For proposing the dissertation's topic. For the unwavering support you showed me throughout my years of postgraduate studies.

To Prof. P.J. Pahl for the time spent with me discussing optimization and recommending how to approach the problem.

To my business associates, Dr J. Potgieter and L. Theron, for having the patience and understanding when my attention needed to be diverted. For offering advice, words of encouragement and generally picking up my slack when the workload became too much.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The South African Human Rights Commission, in their report entitled "Water is Life. Sanitation is Dignity: Accountability to people who are Poor" (Ramkissoon, 2014) argues that access to decent sanitation services and infrastructure is a human right. They investigated the financial and technical requirements of delivering basic sanitation to an estimated 5.2 million people still lacking such services in South Africa alone. It is stated that an estimated R44.75 billion is required to provide basic services, while R31.25 billion is needed to refurbish and upgrade existing infrastructure. In stark contrast to these amounts, the South African budget allocation for sanitation was approximately R3.2 billion nation wide. This state of affairs is not limited to South Africa alone. In an article titled "A Place to Go", National Geographic Magazine (Royte, 2017) reports that an estimated 950 million people currently defecate in the open worldwide.

Towards alleviating this public health hazard, it is clear that money allocated to the provision of sanitation services has to be spent as effectively as possible. This statement is true for all civil engineering endeavours and engineers often pride themselves for being able to rise to this challenge. The collective efforts and experience of past and present engineers have lead to modern design principles and techniques which do, most often, provide very good designs and in some cases even optimal designs. However, some problems in the engineering domain are categorized as NP-Hard (Knuth, 1974; Leeuwen, 1998). NP-Hard problems are decision based and the number of possible solutions grow exponentially relative to the size of the problem. Even the most experienced engineers, relying on a combination of intuition, experience and experimentation, can only consider a trivially small subset of the possible solutions to such problems.

The rapid increase in digital computing power, paired with the development of techniques capable of exploiting that power to address NP-Hard problems, have in the last two decades opened up exciting possibilities in engineering design. It is within this domain that the research described here aims to make a contribution.

1.1 Overall research objective

Many techniques have been developed, both recently and in the past, to optimize the design of sewer networks. Sewer network optimization requires the solution of two problems. The first is to determine an optimal layout of the network components, referred to as layout optimization. The second is to determine the optimal hydraulic characteristics of any given layout, referred to as hydraulic optimization. Most existing optimization algorithms focus exclusively on the hydraulic optimization part, while the layout remains static (Lejano, 2006). Recently, however, promising algorithms have been developed which aim to solve both problems simultaneously, for example Moeini and Afshar (2012).

Given the enormous need for sanitation services, the financial constraints inherent to providing it, and inspired by technical advances like Moeini and Afshar (2012), the overall research objective is to develop new techniques to minimise the capital expenditure required for the installation of sewer networks. The aim is not only to improve on existing algorithms. While addressing their shortcomings, new approaches and test cases are developed which enable a thorough investigation into the nature of the optimality of sewer network designs. Changes in network layout parameters during optimization are monitored and their effects on the cost of the solution are determined. This knowledge is then exploited to modify the proposed algorithms in order to further improve their behaviour.

The broad scope of the stated objective is limited and focussed in the overview of the dissertation chapters presented below.

1.2 Document Structure

While it is common for theses and dissertations to present a complete overview of the relevant literature and problem aspects early on, this document does not. The research described here employs a broad set of techniques across many different fields of engineering, computer science and mathematics. To present the reader with all the relevant information of each facet, out of context of the flow of the investigation, is deemed detrimental to the readability of the document. The conclusions reached at each stage of the investigation often to lead to a natural next step in the process, which only becomes obvious at that stage. For example, a discussion of "simplex noise" and its role in the development of a problem library (cf. Chapter 6) does not offer the reader much insight at this stage. Its application only becomes clear once the need to generate specific topographies for sewer networks has been motivated, which in turn only becomes clear once the results of the simultaneous optimization algorithms (cf. Chapter 5) have been discussed. Consequently, comprehensive literature reviews, problem statements and discussions

of relevant techniques are included in each chapter, as described below.

1.2.1 Chapter 2: Sewer Network Design as an Optimization Problem

The reader is introduced to sewer network design from an optimization perspective. The focus is on presenting a clear understanding of the problem which is to be solved and the key aspects which will be focussed on by proposed solutions. Additionally, the objective function of the optimization is presented. Design constraints, such as minimum and maximum flow velocities, to which any solution will be subject are presented and their inclusion motivated. Analysis and design equations that incorporate the constraints are presented and their use motivated. Simplifying or limiting assumptions made during the design procedure are introduced, discussed and motivated. Significant shortcomings in existing optimization approaches are highlighted and the methodology to overcome them is described and motivated.

1.2.2 Chapter 3: Hydraulic Optimization

Chapter 3 focusses exclusively on the hydraulic optimization problem, i.e. determining the optimal set of hydraulic parameters for a given layout. A comprehensive overview of existing hydraulic optimization techniques is given and their shortcomings are noted and discussed. A new algorithm, which relies on minimum slope information to perform near optimal diameter selections with very little computational effort, is presented. The new algorithm, denoted Heuristic Optimization by Minimum Slopes (HOMS), is applied to two case studies and its results and computational performance are compared to current state of the art solutions.

1.2.3 Chapter 4: Layout Optimization

This chapter focusses exclusively on the layout optimization problem. A comprehensive review of existing sewer network layout creation strategies and optimization algorithms is presented. New layout creation strategies, with varied characteristics, are presented and Ant Colony Optimization (ACO) is employed to perform optimization of the layouts they produce. An overview of the basic ACO algorithm is given, and modifications made to it are discussed and motivated.

1.2.4 Chapter 5: Sewer Network Optimization

In this chapter the HOMS algorithm of Chapter 3 is combined with the layout optimization algorithms of Chapter 4 to form hybrid algorithms capable of solving

both sub-problems simultaneously. Three example problems are presented. The new algorithms are used to solve each of the example networks and the results are compared to references from literature. Key areas of interest are identified and motivated for further investigation.

1.2.5 Chapter 6: Standard Problem Library

The results of Chapter 5 indicate that a thorough study of algorithm behaviour for problems with specific characteristics will be beneficial. This chapter describes the development of software which is capable of generating sewer network instances with controllable characteristics, for example the number of manholes in the network, the average length of pipes, etc. Through combinations of network characteristics, 153 classes of sewer network problems are defined. The software is then used to generate 20 instances of each class, resulting in a total of 3060 sewer network instances. These instances can be used as a standard library to evaluate the performance of optimization algorithms when exposed to the different problem classes.

1.2.6 Chapter 7: Characteristics of Optimal Networks

The question why do particular algorithms perform better for certain problems than others, is investigated. The aim is not simply to determine which algorithm characteristics yielded better solutions, but rather to determine what characteristics were displayed by the best solutions.

Based on this, network parameters are proposed which offer insight into the characteristics of a specific layout. The algorithms developed in Chapter 5 are then used to solve each of the 3060 problems generated in Chapter 6 multiple times. During execution of the optimization algorithms the proposed network parameter values are captured at regular intervals. The associated characteristics are compared to the cost of the network at the respective intervals and used to establish correlations between network characteristics and cost.

Using the established correlations, heuristic influence factors are proposed which guide the ACO algorithms towards decisions which more strongly favour or avoid desired or undesired characteristics respectively. The effect of each heuristic influence on the network characteristics are carefully monitored to ensure they operate as far as possible on the desired characteristic alone. Due to the interdependent nature of sewer network characteristics some side effects are unavoidable.

1.2.7 Chapter 8: Augmenting Sewer Network Optimality

The heuristic influence factors, 7, and multiple combinations of them, are applied

to the algorithms of Chapter 5. The modified algorithms are used to solve each of problems from the standard problem library again. The results are presented compared to the unaltered state of the algorithm. Furthermore, the best performing heuristic influence factor or combination of factors at varying stages of the algorithm's execution are also presented.

1.2.8 Chapter 9: Conclusions and Recommendations

Chapter 9 summarises the important results and conclusions of each chapter. A critical evaluation of contribution of each chapter is included. Recommendations and guidelines for further research are discussed.

Chapter 2

Sewer Network Design as an Optimization Problem

In this chapter an overview of gravity sewer network design relevant to the optimization problem is presented. In this investigation the sewer networks under considerations are limited to only gravity sewer networks with no special structures, such as rising mains or pump stations, present and no divergent structures. These restrictions are motivated where relevant to the sewer network design aspect currently under discussion. The mathematical modelling of the two sub-problems of sewer networks as used in this investigation are introduced. The cost function used to determine a unit capital cost associated with a complete design throughout this investigation is introduced.

The design of sewer networks consists of two parts: firstly a suitable layout, taking into account existing or planned infrastructure, such as roads or buildings, has to be determined. Secondly the hydraulic design has to be performed to determine cumulative flow rates, pipe diameters, cover depths and flow velocities. The hydraulic analysis model used to calculate cumulative flow rates is introduced and its use motivated. The reader is introduced to the Fitness Warping phenomenon present in some simultaneous sewer network optimization algorithms.

2.1 Calculating Network Cost

The design is to be optimized, in this investigation, in terms of capital-investment cost. The necessary steps to calculate an accurate estimated construction cost for a complete sewer network is a strenuous and complex task. To calculate an accurate estimate in real currency of the capital-investment cost of a network requires an abundance of information, such as cost of excavation rates which take soil conditions into consideration, the cost of network elements such as pipes or

manholes, labour expenses, etc. The list is truly enormous. An accurate estimation of capital expenditure is the concern of the contractor once a detailed design has been completed. Furthermore, the cost-rates used for the different aspects of construction are often well guarded due to the competitive nature of the tendering process in construction. Consequently, automating an accurate real-currency cost estimation of a complete sewer network is beyond the scope of this investigation.

Instead, functions have been formulated in the literature which allow a unit cost value to be calculated. These functions approximate the expected real-currency cost, based on the most capital intensive variables i.e. soil excavation and pipe diameters. It is important to note that for the purpose of optimization accurate costs are not required. Rather, the calculated costs for a multitude of network designs should be correct relative to one another. That is to say, if the real-currency estimated cost of Network A is higher than that of Network B, then the approximate unit cost function should deliver the same result if not necessarily by the same ratio. If a cost function has this property it allows an optimization algorithm to select the correct network as the least or less expensive option correctly and consistently. The formulation and testing of such a cost-function is beyond the scope of this investigation. A Cost function, which uses interchangeable unit values, is taken from the literature (Moeini and Afshar, 2012) which allow accurate relative comparison between examples for the purpose of optimization.

2.1.1 Capital-Investment Cost Function

The function used to calculate the capital-investment is as used by Moeini and Afshar (2012):

$$C = \sum_{i=0}^{N} L_i K_i(d_i, E_i^{ave}) + \sum_{j=0}^{M} K_j(h_j) \tag{2.1}$$

Where:

C = Cost Function of Sewer Network
L_i = The length of pipe i, i $\in$ {1, ... , N}
K_i = Unit cost function of pipe i, defined in terms of its diameter (d_i) and average cover depth (E_i^{ave})
K_j = Unit cost function of manhole j, defined in terms of its height (h_j)
N = The number of pipes in the network
M = The number of manholes in the network

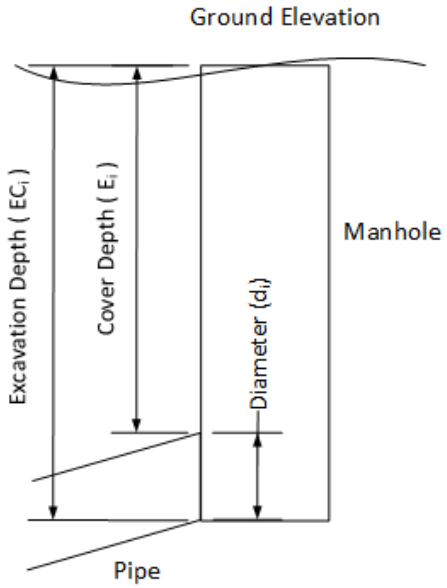


Figure 2.1: Definition of Depth Variables

Figure 2.1 shows the definition of depth variables used in the cost function, as well as throughout the entirety of this investigation.

This function is dependent on unit cost functions which may be modified for a specific problem to more accurately represent the expected cost for the network.

2.1.2 Unit Cost Functions

In this investigation the majority of problems under consideration are theoretical. Consequently, the unit cost functions do not require adjusting based on the problem. The only constraint is that the same unit cost functions be used for networks which require their costs to be compared, assuming the unit cost function produce accurate relative costs. Similarly to the cost function, the derivation of the unit cost functions is beyond the scope of this investigation. Two unit cost functions are used in this investigation. The first is used for all problems excluding a single benchmark problem where different unit cost functions have to be used to allow direct comparison of results from previous investigations. Both sets of unit cost functions are introduced here.

The first set of unit cost functions are as used by Afshar *et al.* (2011), as shown below.

$$K_i = 1.93e^{3.43d_i} + 0.812E_i^{1.53} + 0.437d_iE_i^{1.47} \tag{2.2}$$

$$K_j = 41.46 h_j \tag{2.3}$$

Where:

d_i = The diameter of pipe i $[m]$
E_i = The average cover depth of pipe i $[m]$
h_j = The height of manhole j $[m]$

These are exponential unit cost functions dependent on the diameter of a pipe, its excavation depth and the excavation depth of a manhole. Intuitively, the exponential growth in cost of a pipe as its diameter and excavation depth increases is sensible. Furthermore, the unit cost a pipe has a term which depends on both parameters simultaneously which takes into account the increased excavation width for a larger diameter. The unit cost function for a manhole is a linear function dependent only on the excavation depth. These unit cost functions are used throughout the investigation, excluding one benchmark problem in Chapter 3 where it is stated again.

The second set of unit costs functions are proposed by Meredith (1972), as shown below.

$$K_i = \begin{cases} 10.98d + 0.8E - 5.98 \\ \quad \text{if d } \leq 3ft \text{ and E} \leq 10ft \\ 5.94d + 1.166E + 0.504Ed - 9.64 \\ \quad \text{if d} \leq 3ft \text{ and E} > 10ft \\ 30.0d + 4.9E - 105.9 \\ \quad \text{if d} > 3ft \end{cases} \tag{2.4}$$

$$K_j = 250 + h_m^2 \tag{2.5}$$

Where:

d = The diameter of pipe i $[ft]$
E = The average cover depth of pipe i $[ft]$
h_m = The height of manhole m $[ft]$

The unit costs proposed by Meredith (1972) are calculated using imperial unit values. The unit cost of a pipe presented by Meredith (1972) takes into account the variation in expected cost as excavation depths and diameter increases occur and define a step-wise bi-linear function rather than an exponential function. Further, the unit cost of a manhole increases exponentially rather than linearly. These unit costs are only used during the evaluation of the proposed hydraulic optimization procedure of Chapter 3 to allow direct comparison of results.

2.1.3 Formulating the Objective Function

All optimization algorithms require an objective function by which to evaluate the fitness of a solution. In this investigation, as stated previously, the objective is to minimize capital-investment cost. Consequently, the objective function is:

$$Minimize\ C = \sum_{i=0}^{N} L_i K_i(d_i, E_i^{ave}) + \sum_{j=0}^{M} K_j(h_j)$$

where all variables are as defined for Equation 2.1. This objective function is subject to a multitude of constraints discussed later in this chapter, which may be violated by a given solution. It is common practice in the use of optimization algorithms to use a penalized version of the objective function to account for the violation of constraints:

$$Minimize\ P = C + \alpha \sum_{i} g_i \tag{2.6}$$

Where:

P = The Penalized Objective Function
C = Cost of the Sewer Network, Equation 2.1
g_i = Violation value of constraint i, 0 if unviolated
α = A sufficiently large constant so ensure feasible solutions have a better fitness than infeasible solutions, i.e. solutions that violate one or more constraints

It is important not to discard infeasible solutions from a meta-heuristic algorithm as, especially early on, the algorithm may struggle to find the feasible region of the search space and will be required to learn from infeasible regions. Furthermore, an infeasible solution may be right on the edge of the feasible search space which is adjacent to the optimal valley or peak of the search space. If this solution is discarded the algorithm cannot learn from this solution, despite how close to the optimal solution it is. The penalized objective function allow infeasible solutions to contribute to the progression of the meta-heuristic algorithm while ensuring it will never be considered a better solution than a feasible solution if α is sufficiently large. Note the contribution of the second term of Equation 2.6 is 0 if no constraint is violated. Equation 2.6 is used throughout this investigation as the objective function.

2.2 Layout Design

Layout design of sewer networks is again a two part problem. The first is to determine the spatial location of manholes and pipes. The second is to determine the direction of flow of each pipe. Only once both parts of the problem have been solved is a full network layout found.

2.2.1 Spatial Design of the Network

Referring to Figure 2.2. The nodes, representing manholes labelled 1, 2, 3 and 4 respectively, are connected by edges, representing pipes labelled 1, 2, 3, 4 and 5 respectively.

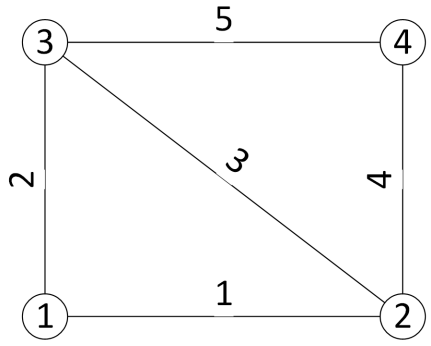


Figure 2.2: Layout Design Example

In this example the spatial position of the manholes and pipes have already been determined, i.e. the first part of the design is complete. However, it may have been equally feasible to place the manholes at different locations and to connect them differently, for example by placing pipe 3 between manholes 1 and 4 rather than 2 and 3. This part of the design, deciding on the spatial location of manholes and how to connect them with pipes, is most often governed by existing or planned infrastructure, such as roads or buildings, and topographical considerations, such as hills or steep inclines. In this investigation it is assumed that the positioning of manholes and pipes is completed prior to the optimization process aimed at minimizing the installed cost of the sewer network. The positioning of manholes and pipes is referred to as the base layout, or base graph, of the layout. All pipes and manholes included in the base graph must be present in the final solution. The base layout is modelled mathematically as an undirected graph where the vertices

represent manholes and the edges pipes:

$\mathbf{G}_{base} = (\mathbf{V}, \mathbf{E})$

Where:

$\mathbf{G}_{base}$ = The base graph

$\mathbf{V}$ = The vertex set, whose elements are the vertices of $\mathbf{G}_{base}$ which represent manholes of the sewer network

$\mathbf{E}$ = The edge set, whose elements are the edges of $\mathbf{G}_{base}$ which represent pipes of the sewer network. As the graph, $\mathbf{G}_{base}$, is undirected the individual edges are undoredered pairs ($\mathbf{u}$, $\mathbf{v}$) where $\mathbf{u}$ and $\mathbf{v}$ are vertices in $\mathbf{V}$

2.2.2 Directional Design of the Network

The second part of the layout design is to determine the direction of flow for each pipe. This part of the layout design is deceptively complex and the number of possible permutations grows exponentially as the number of vertices and edges present in the base graph increases. This part of the layout design is the concern of the optimization procedures which are discussed in detail in Chapter 4; A brief overview of the required decisions to complete the directional design is given here. Figure 2.3 shows two directional graphs, directions are indicated by arrows. Figure

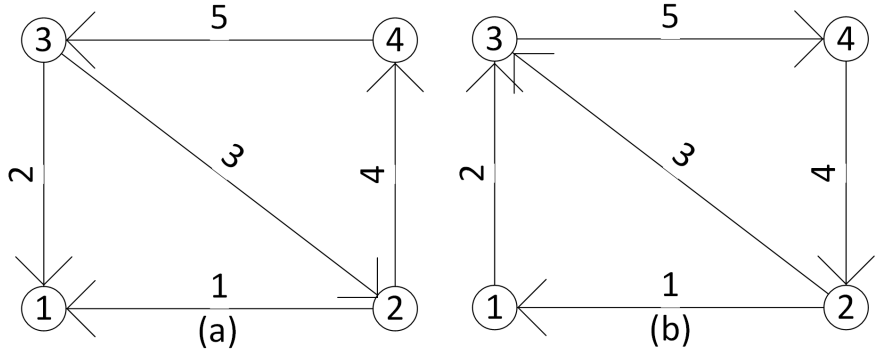


Figure 2.3: Directional Layout Design Examples

2.3 shows two feasible final layouts, of a possible $5^2 = 25$, of the base layout shown in Figure 2.2. The choice of flow directions can heavily influence the final capital investment cost of the completed sewer network, especially when adverse

topographical conditions are present. If, for example, manhole 2 has a much lower elevation than manhole 3 then using sound engineering intuition we can readily observe that the design of Figure 2.3 (a) requires less excavation than that of Figure 2.3 (b). This reduction in required excavation can be expected to lead to a reduction in capital expenditure. However, the problem becomes increasingly difficult as the size of the base graph increases, since the change in flow direction of a single pipe may have significant effects on the cumulative downstream flow rates within pipes and therefore their required diameters and slopes.

Notice that in both designs of Figure 2.3 cycles are present in the final layout designs. For Figure Figure 2.3 (a) the cycle 2-4-3 exists and for Figure 2.3 (b) the cycle 2-3-4 exists. As stated before, in this investigation only gravity sewer networks are considered with no special structures present.

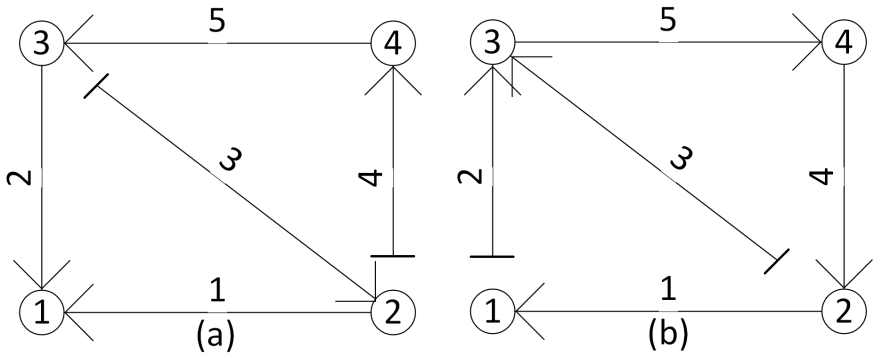


Figure 2.4: Directional Layout Design Examples With No Cycles

Moeini and Afshar (2012) propose disconnecting pipes from their upstream manholes and creating what they term adjacency nodes, which are created artificially for purpose of the optimization at the same location as the existing upstream manhole. The practical implication of this is that the pipe has no upstream inflow from the manhole, and cycles are removed from the network. Referring to Figure 2.4, the networks shown are similar to those in Figure 2.3. In this case, however, the pipes have been disconnected from their upstream manholes and adjacency nodes created, indicated by perpendicular lines on the upstream end of the pipe.

2.2.3 Layout Constraints on the Optimization Problem

As stated previously the layout optimization procedures in this investigation are only concerned with the directional design of the network, the spatial design is

completed a-priori. It is also assumed that all elements present in the base layout have to be included in the final layout design, with no cycles present. Cycles are avoided through the use of adjacency nodes. How the adjacency nodes are selected is the concern of the layout construction algorithm and described in Chapter 4 of this document. When constructing a network layout the single out-degree constraint is introduced. The single out-degree constraint limits the network such that all manholes may only have a single outlet pipe. This simultaneously avoids cycles and diversion structures within the network. The resulting layout produces a simple branched gravity sewer network with no special structures. Additionally, pumping stations are excluded from all networks. These constraints on the layout drastically simplifies the hydraulic analysis of the network. This reduction in hydraulic analysis complexity allows greater focus on the layout design problem, which is a key concern of this investigation. Furthermore the restriction on diversion structures and pumping stations is not considered to be severe. It is postulated that these structures decompose a network into a number of sub-networks, each of which can be designed using the techniques developed here.

2.2.3.1 Branched Network Layout

Only gravity sewer networks are considered and no divergence structures are allowed in the network. This implies that all manholes may only have a single outgoing pipe.

To ensure that any proposed layout adheres to these simplifications, only branched network layouts are selected from the power set of the base layout, which contains all possible looped and branched layouts. Mathematically the restrictions for a branched layout in a network with M manholes are (Moeini and Afshar, 2012):

$$\begin{aligned} X_{jl} + X_{lj} &= 1 \qquad \forall\, j, l \in \{1, ..., M\} \\ \sum_{l=0}^{M} X_{jl} &= 1 \qquad \forall\, j \in \{1, ..., M\} \end{aligned} \tag{2.7}$$

Where:

$$X_{jl} = \begin{cases} 1 & \text{if an edge with flow from } j \text{ to } l \text{ exists} \\ 0 & \text{otherwise} \end{cases}$$

This constraint is augmented with the continuity of flow requirement at each node

j:

$$\sum_{l=0}^{M} X_{lj} Q_i^{lj} - \sum_{l=0}^{M} X_{jl} Q_i^{jl} = 0 \quad \forall\, j \in \{1, ..., M\} \wedge j \neq s \tag{2.8}$$

Where:

Q_i^{lj} = flow rate in pipe i between nodes l and j with either node as source or target

s = the outlet node

All networks are defined with a single outlet in this investigation. The continuity equation is not enforced at the outlet since only the inflow is modelled for the outlet node. Note that this restriction does not affect the generality of the proposed layout creation algorithms as the same method may be applied for multiple outlets simultaneously. The exclusion of pumping stations is not modelled mathematically as this constraint is simply enforced by exclusion.

2.3 Hydraulic Design

The hydraulic design of a sewer network can only be completed once a complete spatial and directional layout design has been obtained. Furthermore, the suitability of any set of hydraulic parameters for a design is dependent on the layout as the layout influences the cumulative downstream flow rates which in turn affect other design variables such as pipe diameter and slope. The hydraulic design requires many variables to be solved simultaneously, such as cumulative downstream flow rates, diameters, slopes, excavation depths, manhole depths, and flow velocities. All of these variables are subject to design constraints, such as maximum flow velocities or minimum cover depths, which ensure the completed design adheres to sound engineering guidelines and regulation codes. Further, there are hydraulic design variables which have to be determined a-priori. This section is not intended to give a comprehensive overview of sewer network design, rather the relevant constraints to the optimization problem are presented.

2.3.1 A-Priori Design Parameters

There are two variables relevant to the hydraulic design which have to be determined a-priori to the design phase. The variables and some practical considerations are discussed here.

2.3.1.1 Ground Elevations

The ground elevation at manholes has to be determined a-priori by land surveyors or GIS systems. There is no alternative way to calculate elevations if an accurate representation is to be obtained. It is assumed that the slope of ground elevations between manholes is linear. While this is perhaps not strictly correct, it is reasonable to assume that the spatial design would not include a pipe which runs through a hill or other adverse topographic occurrence which would require a significant increase in excavation from the linear assumption.

2.3.1.2 Inflow Rates

The expected inflow rates into the sewer network due to service connections or flow contributions from an existing network have to be determined a-priori. Inflow rates are assumed to enter a pipe at its upstream manhole, consequently the inflow hydrographs are assigned at manholes. The expected inflow rate due to service or other connections is specified as 24-hour hydrographs.

2.3.2 Hydraulic Constraints on the Optimization Problem

Sewer network design is subject to a multitude of hydraulic constraints of varying complexity. These constraints ensure the completed design can meet the required design flow rates, protect the network from failure and address some obvious health and safety concerns associated with raw sewage contaminating the soil. The design constraints are motivated and formulated mathematically below as part of the optimization problem.

2.3.2.1 Cover Depth

Both minimum and maximum cover depth constraints are enforced. Minimum cover depths protect pipes from imposed loads, such as vehicle loads where sewer pipes pass under roads. The minimum cover depth also prevents cross contamination between water distribution networks by ensuring sewer pipes are placed below water mains. Furthermore the minimum cover depth ensures an adequate drop for house connections.
Similarly, the maximum cover depth prevents pipe failure under excessive soil and imposed loads. Maximum cover depth may also be enforced to avoid excessive excavation, specifically where soil conditions are adverse.

$$E_{min} \leq E_i \leq E_{max} \quad \forall\, i \in \{1, ..., N\}$$

Where:

E_i = The cover depth of any pipe i at its source or target node

E_{min} = The minimum allowable cover depth

E_{max} = The maximum allowable cover depth

2.3.2.2 Velocity

Both minimum and maximum velocity constraints are enforced at the peak design flow rate. The minimum velocity prevents the deposition of solid particles within pipes and manholes. The maximum flow velocity is enforced to prevent erosion of the pipe material.

$$v_{min} \leq v_i \leq v_{max} \quad \forall\, i \in \{1, ..., N\} \tag{2.9}$$

Where:

v_i = The velocity in pipe i at the peak design flow rate

v_{min} = The minimum allowable velocity

v_{max} = The maximum allowable velocity

2.3.2.3 Slope

A minimum slope is enforced on all pipes. This is to prevent inaccurate placement during construction or adverse slopes resulting from pipe settlement. The minimum slope requirement also ensures that during full flow conditions the minimum flow velocity is achieved.

$$S_{min} \leq S_i \quad \forall\, i \in \{1, ..., N\}$$

Where:

S_i = The slope of pipe i

S_{min} = The minimum allowable slope

2.3.2.4 Required Spare Capacity

A percentage spare capacity is enforced at peak flow conditions to ensure that pressurised flow does not occur. The constraint has the additional benefit of providing a margin of safety if storm water ingress is experienced during peak flow

times.
Some investigators (Moeini and Afshar, 2012; Haghighi and Bakhshipour, 2015), use a maximum relative flow depth constraint rather than percentage spare capacity. This is merely a different formulation of the same constraint. Moeini and Afshar (2012) enforce a minimum relative flow depth. In this implementation no such constraint, or an equivalent, is enforced. This constraint is not common engineering practice, furthermore near the sources of the network it becomes almost impossible to enforce this constraint due to unavoidable low flow rates.

$$SC = \frac{Q_{full} - Q_{peak}}{Q_{full}}$$

Where:

SC = Spare Capacity ratio
Q_{full} = The full flow rate $[m^3/s]$
Q_{peak} = The partially full flow rate at peak conditions $[m^3/s]$

This constraint results in partially full flow conditions in all pipes. To solve for the hydraulic parameters Manning's equation is used to estimate velocities throughout this investigation.

$$Q_i = \frac{1}{n_i} \frac{A_i^{\frac{5}{3}}}{P_i^{\frac{2}{3}}} \sqrt{S_i} \quad \forall\, i \in \{1, ..., N\} \tag{2.10}$$

Where:

Q_i = The flow rate of pipe i
A_i = The area of flow in pipe i
P_i = The wetted perimeter of pipe i
N = The number of pipes in the network
n_i = Manning's roughness coefficient

2.3.2.5 Commercially Available Diameters

Diameters may only be selected from a discrete set of commercially manufactured diameters.

$$d_i \in \{\mathbf{D}\} \quad \forall\, i \in \{1, ..., N\} \tag{2.11}$$

Where:

d_i = The diameter of pipe i
$\{\mathbf{D}\}$ = The set of commercially available pipe diameters

2.3.2.6 Progressive Pipe Diameters

The diameter of a pipe may only be equal to or larger than any of the pipes directly preceding itself. This is to prevent possible blockage, damming of waste water and sudden increase in flow velocities in the network.

$$d_i \geq \{\mathbf{D}\}_i \quad \forall\, i \in \{1, ..., N\} \tag{2.12}$$

Where:

$\{\mathbf{D}\}_i$ = The set of directly preceding diameters of pipe i

2.3.2.7 Progressive Pipe Depths

The outflow pipe of any manhole may not be placed above the deepest inflow pipe. This prevents permanent damming of waste water and solid deposition in the manhole.

$$EC_i \geq \{\mathbf{EC}\}_i \quad \forall\, i \in \{1, ..., N\} \tag{2.13}$$

Where:

EC_i = The excavation depth of the outgoing pipe i at the node

$\{\mathbf{EC}\}_i$ = The excavation depths of all incoming pipes at the source node of pipe i

If all of the above constraints are met a completed design is considered feasible.

2.3.3 Calculating Cumulative Downstream Flow Rates

Inflow rates due to service and other connections are assigned at the upstream manhole of a pipe as a 24 hour hydrograph as stated in Section 2.3.1.2. The flow rates need to be routed downstream to the outlet manhole. There are many techniques to calculate the downstream cumulative flow rates. Depending on what is required, different techniques of varying complexity can be used to obtained the required level of accuracy. van Heerden (2014) compares three alternative hydraulic analysis models for sanitary sewer systems, namely the (i) contributor hydrograph, (ii) kinematic wave and (iii) fully dynamic wave models. In his investigation van Heerden (2014) found that the suitability of a model is dependent on the phase of the engineering process it is applied to. He refers to three phases of the engineering process: planning, design and evaluation. In this investigation the solutions fall within the planning phase. van Heerden (2014) found that during

the planning phase the level of accuracy required is not such that the shortcomings of the contributor hydrograph model severely impact on the relevant design variables obtained, most notably for this investigation the peak cumulative downstream flow rates. Consequently the contributor hydrograph model is used here as it is the least complex of the three models and has the lowest computational complexity, making it ideal for combination with optimization algorithms where computation time can be a limiting factor. A brief overview of the contributor hydrograph model is presented.

2.3.3.1 Contributor Hydrograph Model

The contributor hydrograph model was first introduced by Shaw (1963). Later, Stephenson and Hine (1982) found that ordinary time-lag routing, used in contributor hydrographs, is of sufficient accuracy for sewer network design purposes. van Heerden (2014) found that it was indeed sufficient if the design is in the planning phase, as is the case in this investigation. In the contributor hydrograph model inflow rates are defined as 24 hour hydrographs. The inflow hydrographs are defined by parameters associated with the land use of the service connections and the number of units serviced. In this implementation the local inflows are calculated from South African standard unit hydrographs associated with a specific landuse. The unit hydrographs are included in Appendix A. The local inflows for each of the landuses is summed to give the total local inflow to the pipe, which is assigned at the upstream manhole.

$$Q_t = \sum_{i=0}^{n} EE_{LUi} \times (UH_{LUi_t} \times P_{LUi} + L_{LUi}) \tag{2.14}$$

Where:

Q_t = Local inflow at time t
n = Number of considered land uses
EE_{LUi} = Number of land parcels with land use i
L_{LUi} = Leakage for land use i
P_{LUi} = Peak for land use i
UH_{LUi} = Unit Hydrograph for land use i

Peaks of hydrographs are shifted and the accumulated flow attenuates as the flow is routed downstream. This is due to the considered time delay of flow to reach downstream manholes. Time delay is calculated using Manning's equation at full flow conditions. Calculating time delay at full flow conditions produces a

conservative estimate of time delays while greatly reducing computational effort as determining the depth of flow in the pipe is not required.

$$\begin{aligned} \text{Time Delay} &= t &&= \frac{L}{v} \\ \text{Velocity} &= v &&= \frac{Q_f}{A_f} \\ \text{Full Flow Rate} &= Q_f &&= \frac{1}{n}\frac{A_f^{5/3}}{P_f^{2/3}}\sqrt{S} \end{aligned} \tag{2.15}$$

Because of time delay effects the accumulated flow rates cannot be directly summed. Once a hydrograph has been shifted due to time delay the flow values are no longer known at the downstream time steps of the hydrograph where flow rates are known. If, for example, after shifting a hydrograph's peak is at hour 4.5, and the inflow is required at hours 4 and 5 then the peak will be lost at the downstream summation. A conservative linear interpolation method is used for peak shifting to ensure maximum flows are always simulated. Figure 2.5 shows the hydrograph coordinates during interpolation.

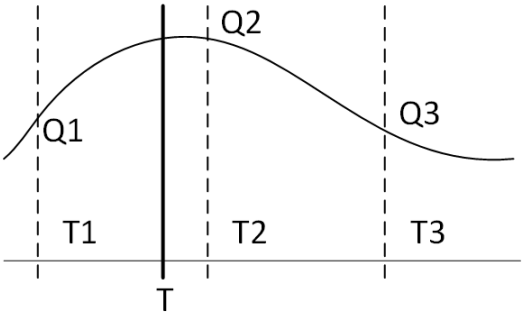


Figure 2.5: Hydrograph Interpolation Points

The procedure to calculate the shifted flow rate at time T is as follows:

1. The first point (T_1, Q_1), before time T is found
2. The second point (T_2, Q_2), following time T is found
3. The third point (T_3, Q_3), following time T_2 is found
4. if $Q_2 > Q_1$ and $Q_2 > Q_3$, the peak flow rate, Q_2, is shifted to time T.
5. else, the resultant flow at time T is equal to $Q_1 + t\,(Q_2 - Q_1)$.

Figure 2.6 shows the 24 hour composite hydrograph of a single very high income residential unit, as well as a shifted hydrograph. To calculate the time delay a diameter of $0.16m$, slope of $0.013m/m$, Manning roughness coefficient of 0.012 and a pipe length of $200m$ was used.

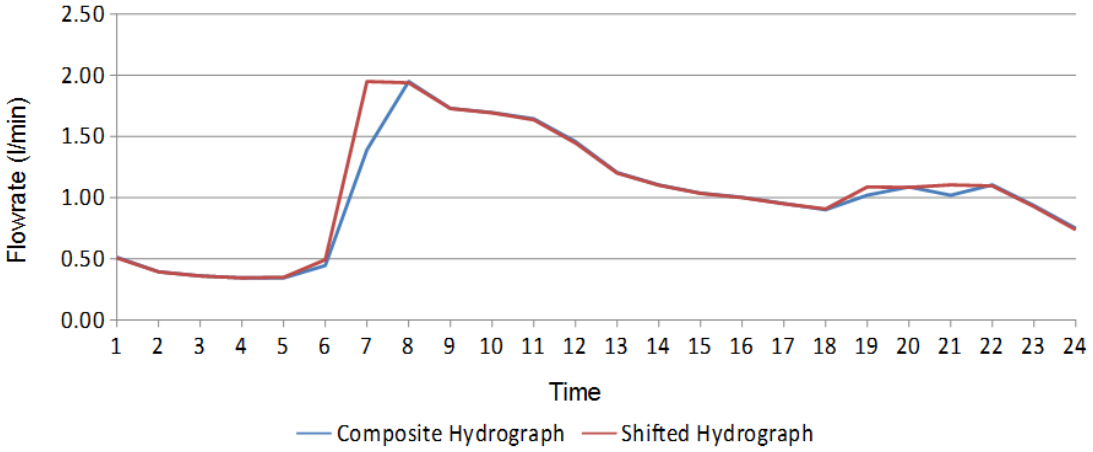


Figure 2.6: Time Shifted Hydrograph

To calculate the time delay the full flow rate is required. Using equations 2.15 the full flow rate may be calculated as $Q = 0.0223m^3/s$, the flow velocity, v, is then $1.1113m/s$ resulting in a time delay, t, of $180s$. Referring to Figure 2.6, the peak before shifting is at hour 8. Once the time delay of $180s$ is included the peak falls at hour 8.05, similarly all other known values are moved 0.05 forward. If this hydrograph is to be added to a downstream manhole the values at the original time intervals have to be calculated. The calculation of the shifted flow rate value at hour 7, the time T, is shown below following the method above:

1. Find the first point (T_1, Q_1), before time T. Select T_1 as hour 6.05 of the attenuated hydrograph, where $Q_1 = 1.39l/s$.

2. Find the second point (T_2, Q_2), after time T. Select T_2 as hour 7.05 of the attenuated hydrograph, where $Q_2 = 1.95l/s$.

3. Find the third point (T_3, Q_3), after time T_2. Select T_3 as hour 8.05 of the attenuated hydrograph, where $Q_3 = 1.73l/s$.

4. $Q_2 = 1.95l/s > Q_1 = 1.39l/s$ and $Q_2 = 1.95l/s > Q_3 = 1.73l/s$. The flow rate at time $T = Q_2 = 1.95l/s$.

This procedure is repeated for each time step of the hydrograph and the complete shifted hydrograph obtained which may be summed with the downstream flow rate to determine the cumulative flow rate. Note that the peak is shifted to an earlier rather than later time despite being delayed. van Heerden (2014) noted that the contributor hydrograph model does not accurately determine the expected time of a peak flow rates, though the value of the peak flow rate is determined accurately. For this investigation, the time of the peak flow rate is not relevant, the important factor is that the peak value is correct to allow accurate calculation of the other hydraulic design variables. If, once the algorithms described in this section has completed a preliminary design, more accurate hydraulic calculations are required the proposed design can be transferred to a hydraulic analysis package more suited to the requirements of the design.

2.4 Fitness Warping

As mentioned before sewer network optimization consists of two sub-problems which must be solved simultaneously. To this effect many algorithms have been developed, as will be discussed later in Chapters 3 and 4. Many of the algorithms approach this problem by determining both the layout and element size characteristics simultaneously, i.e. element sizes are selected before a complete layout has been found and hydraulic analysis is possible. While at first this approach appears to be sound there is a critical phenomenon present that warrants caution: Fitness Warping.

Fitness warping, defined here for the first time to the best of the author's knowledge, refers to the skewed fitness value associated with a good layout when accompanied by poor element sizes. Because the element sizes are far from optimal for the layout, the entire fitness of the solution will be deemed poor and the algorithm unable to recognize that a good layout has been found. This effect is most severe in the early stages of the optimization where layouts and element sizes are still selected at random as the driving mechanism of the algorithm has not had sufficient time to learn from previous solutions.

This phenomenon is not unique to sewer networks, but may be present in any optimization problem where two or more sub-problems are solved simultaneously. Researchers have applied optimization techniques and successfully avoided fitness warping to a variety of problems. Giustolisi and Savic (2006) developed a hybrid regression method that combines genetic algorithms with conventional numerical regression techniques. Haghighi and Bakhshipour (2015) use a complex Tabu-Search model combined with a loop-by-loop cutting algorithm (Haghighi, 2013) to

determine sewer network layouts and an adaptive genetic algorithm (Haghighi and Bakhshipour, 2012) to determine element sizes for each layout. However, none of the researchers draw specific attention to the phenomenon or attempt to define and characterise its effects. Here, the phenomenon is actively avoided and the effects demonstrated in Chapter 5 by means of comparison to an algorithm where the phenomenon is present.

To avoid fitness warping the optimal set of element sizes has to be determined for each layout. However, determining element sizes is an optimization problem in its own right. Consequently, this approach can become extremely computationally expensive if, for example, metaheuristic algorithms are employed for both sub-problems. For this reason a computationally efficient hydraulic optimization model is developed (cf. Chapter 3). It is combined with a metaheuristic layout optimization algorithm (cf. Chapter 4), so that each individual layout produced by the metaheuristic can be hydraulically optimized and the entire algorithm completed in reasonable time.

2.5 Summary and Conclusions

In this chapter the aspects of sewer network design relevant to the optimization problem were introduced. It was stated that in this investigation the network is to be optimized in terms of capital investment cost. The function used to calculate the investment cost as well as the unit cost functions associated with it were introduced. The use of a penalty function to avoid infeasible solutions was motivated.

The two parts of the layout design problem, spatial and directional layout design, were introduced. Spatial design of the layout is to be completed a-prior to the optimization procedures which are concerned with the directional design of the network. The concept of a base layout, which includes all required manholes and pipes of the final solution, was described. The design constraints relevant to the layout creation algorithms, namely that no cycles may be present in the final layout, was reviewed. The adjacency node implementation used by Moeini and Afshar (2012) to avoid cycles in the layout was described.

Constrains on the hydraulic design parameters relevant to the optimization was reviewed and their inclusion motivated. It was stated that the suitability of a given set of hydraulic parameters for a design is dependent on the network layout. The a-priori design variables, namely ground elevations at manholes and inflow rates due to service connections, were introduced and the required data format described. The calculation of the downstream cumulative flow rates using the contributor hydrograph model was determined to be sufficiently accurate for use in the design techniques described here. The method used to attenuate flow rates

with time delay and a conservative linear interpolation procedure to determine intermediate flow values of the hydrographs was introduced.

The fitness warping phenomenon of algorithms which select both layout and hydraulic parameters simultaneously was introduced. To avoid this phenomenon, a computationally efficient hydraulic optimization model (cf. Chapter 3) which can be used to optimize the hydraulic parameters of each layout produced by a metaheuristic layout optimization algorithm (cf. Chapter 4) is proposed by this investigation, thus avoiding fitness warping.

This chapter did not go into any detail on how the optimization algorithms incorporate the design constraints or how the layout creation algorithms use the concept of adjacency nodes. Rather, the focus was on formulating the problem of sewer network design as an optimization problem.

The process by which an optimized design can be obtained comprises layout optimization and hydraulic optimization for each candidate layout. In chapter 3 the hydraulic optimization procedure is described, followed by the layout optimization procedure in chapter 4. It should be noted that the hydraulic optimization procedure is useful for any given layout, i.e. it can be used even if no attempt is made to optimize the network layout.

Chapter 3

Hydraulic Optimization

Hydraulic optimization is the component of the sewer network optimization problem in which element sizes, installation depths and slopes are determined for a given layout. Due to the highly constrained nature of hydraulic optimization and complexity of simultaneous solution algorithms this part has seen considerably more work than the layout optimization problem (Lejano, 2006). In this chapter an overview of the hydraulic optimization problem is presented. The state of the art solutions to the problem are reviewed and their shortcomings, relevant to this investigation, identified. A computationally efficient heuristic optimization algorithm which relies on required slope information to systematically solve for all hydraulic parameters, is developed and applied to two case studies from the literature. It is shown to obtain near optimal solutions while requiring very little computation time.

3.1 Problem Statement

The hydraulic optimization problem can be seen as an element size selection, or pipe diameter selection, problem. While other approaches may be viable, this is certainly the best option as diameters can only be selected from a discrete set of commercially available diameters. The hydraulic optimization problem is categorized as a mixed integer linear programming optimization problem.

Referring to Equation 2.10, Manning's equation for open channel flow, it is clear that flow rate, Q, pipe slope, S and diameter, d are dependent variables. The flow area, A, and wetted perimeter, P, are dependent on the diameter as well as the flow depth, y, within a pipe. Full flow conditions are often assumed when performing hydraulic calculations for sewer networks as this greatly simplifies the required calculations. All three variables, namely the flow rate, slope and diameter,

have to be determined, either by some mechanism of the optimization algorithm, or by calculation. Most commonly diameters are selected by a mechanism of the optimization algorithm from the discrete set of available diameters. Cumulative flow rates for pipes are often predefined in optimization benchmarking problems, so no hydraulic analysis to determine flow rates is required. This only leaves the slope to be determined. As Manning's equation requires the flow area and wetted perimeter the slope can, at this stage, still not be calculated as the flow depth needs to be determined. However, as stated it is common practice to assume full flow conditions and calculate flow area and wetted perimeter based on this assumption. This allows calculation of the required slope at full flow conditions. If the assumption of full flow conditions is not made the flow depth within a pipe has to be determined. Calculating the flow depth requires a highly implicit equation in terms of the flow depth to be solved using some form of numerical analysis.

The problem is further complicated if flow rates are not predefined and need to be obtained using hydraulic analysis. In this case values for the slopes are often estimated and adjusted if need be after hydraulic analysis. However, it is possible in the case of a branched network layout to perform the hydraulic analysis without the need to estimate slopes as at the upper ends of the network the inflow into a pipe is known at the start of the analysis. The hydraulic analysis procedure can thus start at the upper pipes in the network and proceed downstream.

If the hydraulic optimization procedure is to be successful it must overcome all of these challenges simultaneously, with or without assumptions. In this investigation as few as possible assumptions are made during the analysis procedure. Existing hydraulic models are modified or new procedures developed to allow all variables to be calculated accurately from engineering theory and principles rather than to rely on estimations.

3.2 State of the Art

This section describes the most representative hydraulic optimization procedures in the literature and identifies shortcomings, within the context of this investigation, of these procedures. A motivation for the development of an algorithm which overcomes the limiting shortfalls of the existing state of the art optimization procedures is presented.

Many optimization techniques have been applied to the hydraulic optimization problem of sewer networks. In the past, dynamic programming methods or even manually manipulable spreadsheets have been developed which were able to find

very good solutions to problems subject to very strict conditions or limiting assumptions. These algorithms will be referred to as the classic algorithms. It should be noted that the review of the classic algorithms is relevant to this study as a similar approach is followed in this investigation albeit significantly more modern in its implementation. In more recent times, with the emergence and meteoric rise in popularity of metaheuristic algorithms, the hydraulic optimization problem has seen a resurgence in the literature as researchers develop metaheuristic algorithms to solve the problem.

3.2.1 Classic Algorithms

The classic optimization algorithms more often than not relied upon dynamic programming (DP) or some variation thereof. Dynamic programming optimizes each individual sub-problem of a larger problem and assumes that these local optima will result in a good, if not globally optimal, solution for the large scale problem. Furthermore, if sub-problems are prone to repetition within a problem the result of a previous solution is stored and called upon at a later time. In the case of sewer networks a typical sub-problem is to select the optimal diameter for an individual pipe, without taking into account the effect this diameter has on up-or downstream pipes. In sewer networks it is very unlikely that the conditions are replicated exactly for two or more pipes, so storing previous solutions may not offer as much value as for other problems.

Mays and Wenzel (1976) applied Discrete Differential Dynamic Programming (DDDP) to the design of gravity sewer networks with the assumption that the direction of flow is fixed, no cycles exist within the network and it operates solely under the effects of gravity. This is similar to the assumptions made in this investigation, and in fact the majority of investigations which came thereafter. DDDP is an iterative procedure which limits the search space to a discrete set. This has some significant drawbacks. Reducing the search space to discrete partitions reduces the likelihood of finding the global optimum, furthermore the discrete set for each variable has to be determined manually, thereby reducing its practicality. The algorithms developed by Mays and Wenzel (1976) divides the network into so-called stages using predefined isonodal lines between manholes. These isonodal lines are imaginary lines defined by the user which indicates stages. Each stage is solved individually using a recursive equation to determine the optimal drop in crown elevation of a pipe across the stage. At each stage a cost is determined which is analogous to the cost of installing each pipe and constructing the upstream manholes of the respective pipes for each previously completed stage as well as the current stage. This procedure is repeated until the final result is obtained. The main drawback of the DDDP method is the required user input in defining the

stages and isonodal lines, as well as limitations on the search space, which makes the likelihood that the global or even a good local optimum has been found very slim.

Robinson and Labadie (1981) used a generalized DP package, referred to as CSUDP, to develop an optimization algorithm. Their algorithm allowed a combined maximum of 50 manholes and pipes, severely restricting its practical use for modern day applications. Furthermore, their application limited manholes to a maximum of three incoming pipes and one outgoing pipe. Their algorithm divided the network into stages according to pipes. If, for example, two pipes enters a manhole there are two stages associated with the manhole. The state variables of their algorithm is the pipe crown elevation while the decision variable is the pipe diameter. Their algorithm starts at the upstream end of the network and systematically determines the least cost connection at each stage of the problem to eventually find the best solution.

Miles and Heany (1988) developed a stormwater drainage design method using a spreadsheet package, requiring manual input and manipulation from the user, which may be applied to static layouts. The shortcomings of this method are obvious. While their spreadsheet based algorithm did produce very good results the manual manipulation requirement makes its use extremely impractical if the aim, as is the case with this investigation, is to combine the algorithm with a layout optimization procedure.

Some classic hydraulic optimization algorithms have been developed and successfully combined with layout optimization algorithms. Walters (1985) used a DP algorithm for simultaneous layout and size optimization. The hydraulic optimization part of the DP algorithm (Walters, 1985) uses a discrete set of pipe soffit levels from which to select the up and downstream levels of pipes. Once the soffit levels are known the slope can readily be calculated. The next step is to determine the diameter, Walters (1985) selects the smallest diameter which meets the capacity requirement at the specified slope as the optimal. This process is repeated for all pipes in the completed layout, the construction of which is discussed in Chapter 4, until a final network cost is obtained. This approach limits the slope variable to a discrete set, severely restricting the likelihood of locating the global optimum.

Li and Matthew (1990) make use of DDDP to generate a network layout, and then to size the sewers and pumps while keeping the layout fixed. The hydraulic optimization algorithm of Li and Matthew (1990) is based on a modified version of the DDDP algorithm developed by Mays and Wenzel (1976). The modification

were made by Li (1986) and resulted in a reported 6% improvement in capital investment cost.

Diogo *et al.* (2000) developed a simultaneous layout and element sizing algorithm. DP was employed to solve two hydraulic models, the first assuming steady uniform flow and the second a one-dimensional, gradually varied, unsteady open channel flow. Their DP algorithm uses constraint information to determine the upper and lower bounding slopes for each diameter. The diameter and slope pair with the smallest feasible slope is selected as the optimal, i.e. the least buried depth solution for the given layout is obtained. Walters (1985) states that the least buried depth solution is between 5% and 15% more expensive than the optimal solution, making this implementation an unattractive option as it is known that better solutions exists.

The most severe restrictions of the classic algorithms are their required user interaction and limitations on the search space. The results of these techniques do, however, lend weight to the notion that such algorithms are capable of producing very good results if they are seeded with intelligent input. An obvious benefit of these algorithms is their relatively low computation requirement by today's standards, especially when compared to the computational requirements associated with metaheuristic algorithms which often require upward of 100 000 full network solutions and evaluations.

3.2.2 Metaheuristic Algorithms

Metaheuristic algorithms began to emerge within the field of optimization during the early 1990's. Since then, they have gained the reputation of the best optimization approach to solve combinatorial optimization problems. Metaheuristics are higher-level procedures or heuristics aimed at generating, selecting or finding a heuristic which may provide a sufficiently good solution to an optimization problem, especially when data is incomplete or imperfect (Leonora *et al.*, 2009). Furthermore, metaheuristics make very few assumptions about the optimization problem, enabling widespread application to a variety of optimization problems (Blum and Roli, 2001). The most consistent drawback of applying metaheuristics for hydraulic optimization in this investigation is the computational requirement. As stated previously, the intention in this investigation is to combine a hydraulic optimization procedure which can perform the hydraulic optimization of each trial layout generated by a metaheuristic layout optimization procedure. Consequently, the accompanying hydraulic optimization procedure of this investigation is required to be extremely computationally efficient, a characteristic often lacking in

metaheuristic algorithms.

Pan and Kao (2009) developed a hydraulic optimization algorithm which used a GA combined with Quadratic Programming (QP). In their approach the GA is responsible for determining the location of pumping stations and diameters of pipes, while QP was used to determine the fitness value of a chromosome. The decision variables of the QP model are the slopes and buried depths at the downstream ends of pipes. The majority of the constraints were incorporated into the QP model. Their algorithm required 300 minutes of computation time and did not improve in the final solution of their presented benchmark problem. Afshar *et al.* (2011) used Cellular Automata (CA) for hydraulic optimization. The nodal elevations of the network are used as the decision variables in the CA formulation. The algorithm produced results within of a relatively small number of function evaluation, but it has two drawbacks. Firstly, the formulation of the algorithm is, compared to other algorithms, very complex making its application difficult. Secondly, while the results produced were comparable to other approaches they were around 5% worse than the best solution presented by Afshar *et al.* (2011).

Afshar (2012) used a Rebirthing-Genetic Algorithm to solve the hydraulic optimization problem. The rebirthing heuristic used relies on the assumption that a GA's search is effectively limited to an already narrowed down search space around the optimal or near optimal solution at the later stages of the search (Afshar, 2012). The algorithm is restarted, or 'rebirthed', with an already near optimal set of solutions once the population has converged to allow the algorithm to converge again and in so doing hopefully improve the final solution. The rebirthing algorithm proved effective in finding near optimal solutions, and was less sensitive to substring lengths of decision variables and population size to its standard GA counterpart. This solution is computationally expensive limiting its use in this investigation where a primary concern of the hydraulic optimization procedure is computational efficiency.

Moeini and Afshar (2012) proposed a Max-Min Ant System (MMAS) for simultaneous optimization of the sewer network problem. The decision variables of the ants incorporate both the layout and diameter simultaneously by splitting each eligible diameter for current pipe into anther decision. A Tree-Growing (TG) algorithm is used to limit the eligible set at each decision point of the MMAS. The decision variables, layout and pipe diameter, are combined into a single decision by splitting each eligible diameter for the current pipe into another decision, creating decision pairs of pipe and diameter. In their approach slopes are determined by assuming each pipe flows at maximum allowable capacity. The major drawback

here is the increased number of decisions due to the decision pair formulation, which greatly increases the search space for the ants relative to only the layout being determined. Furthermore it is subject to the fitness warping phenomenon described in Section 2.4.

Rohani and Afshar (2015) developed a hybrid optimization model combining a GA with a general hybrid cellular automata algorithm. The hybrid algorithm is hereafter referred to as the GA-GHCA algorithm. Their algorithm proved efficient and effective at finding near optimal solutions of pumped sewer networks with a fixed layout. They proposed two alternative versions of the hybrid algorithm. In the first approach, pump locations and the corresponding pumping heads are decided by the GA, while the diameter and nodal cover depths of the pipes are determined by the GHCA taking the decisions of the GA into account (Rohani and Afshar, 2015). In the second approach only the pump locations are decided by the GA and all other parameters determined by the GHCA algorithm. The algorithm proved to be efficient and extremely effective in finding optimal solutions for pumped sewer networks, however the required computational effort again limits its application here.

3.3 Optimization by Minimum Slopes

In this section a new heuristic optimization algorithm, hereafter referred to as the HOMS algorithm, which relies on minimum slope information is presented. The algorithm is shown to perform well compared to metaheuristic alternatives, while requiring very little computational effort. Additionally, it is shown to find better solutions than the classic dynamic programming approaches for the benchmark problems considered.

3.3.1 Heuristic Optimization

Heuristic optimization is a term loosely used to refer to an optimization algorithm which does not neatly fit any another classification. Usually, a heuristic algorithm relies on an intuitive or a historically proven property of the problem to make decisions based upon the expected outcome. In essence, heuristic optimization employs prior knowledge of the problem in order to attempt to find the optimal or near optimal decision. This intuitive or historic property is, in this document, referred to as the 'Heuristic Hypothesis'. Additionally, some evaluation procedure should be put in place to allow to algorithm to evaluate its decisions. If no such evaluation mechanism is in place a heuristically-greedy solution is obtained which is known to often be significantly worse than the optimal solution.

3.3.2 Heuristic Hypothesis

Refer to the capital investment cost function, Equation 2.1. The capital expenditure is exponentially dependent on the pipe diameters and excavation depths, increasing with both variables. The aim, then, is to keep both the diameter and excavation depth to a relative minimum for each pipe. Li and Matthew (1990) found that the optimal solutions tend toward minimizing the total required excavation of a network, however not necessarily to the absolute minimum. As mentioned previously,Walters (1985) states that the least buried depth solution, i.e. the network with the least required excavation, is between 5% and 15% more expensive than the then optimal solutions. From this it can be concluded that a network where slopes are kept to a relative, rather than absolute, minimum, thereby keeping the required excavation low, leads to good, if not necessarily optimal, designs.

The heuristic hypothesis on which the algorithm of this investigation is constructed can be stated as: A reduction in required slope while adhering to all hydraulic constraints leads to economical decisions, if not necessarily the optimal decision. However, if any reduction in slope is considered beneficial, regardless of magnitude, then the minimum buried depth solution would be obtained, which is known to not be the optimal solution. Consequently, an evaluation factor should be introduced to evaluate the expected benefit of a reduction in slope and the decision reversed if the evaluation was not successful.

3.3.3 Requirements of the Algorithm

The hydraulic optimization of sewer networks requires three variables to be calculated or selected optimally, namely the diameter, d, cumulative flow rates, Q, and pipe slopes, S. Once these are known all other variables can be calculated from the engineering theory. As described previously, cumulative flow rate is calculated by the hydraulic analysis procedure and has to be determined through analysis rather than selected by a mechanism of the optimization procedure. Furthermore, the slope is a continuous variable making its selection impractical unless discrete partitions are enforced, which in turn limits the search space and makes finding the optimal solution unlikely. The diameters are already a discrete set, due to constraint 2.3.2.5, making this the ideal variable to be selected by the optimization algorithm. Consequently, the algorithm is to select diameters and calculate slopes. The diameter selection procedure is discussed in Section 3.3.7.1, while slope calculations are discussed in Section 3.3.4. The goal, then, is to determine how a change in diameter of a pipe affects the required slope, investigated in Section 3.3.5, and when is an increase or decrease in diameter beneficial, i.e. when does a change in

diameter lead to a reduction in capital expenditure.

The design constraints are to be analysed and slope information extracted from constraints where possible, as shown in Section 3.3.4. Additionally, constraints should be included in the calculation procedure as far as possible to avoid infeasible solutions, also included in Section 3.3.4. An evaluation factor should be introduced, discussed in Section 3.3.6, which is capable of evaluating the benefit of a reduction in slope and notify the algorithm to reverse the action which lead to the reduction if it is deemed unsatisfactory.

3.3.4 Formulating the Hydraulic Constraints

The hydraulic constraints introduced in Section 2.3 are required to all be simultaneously satisfied by the solution. This subsection systematically analyses the constraints and incorporates them into the hydraulic optimization procedure to ensure, in so far as possible, adherence during calculation. The minimum required slope expressions are derived from the constraints directly. Where relevant, derivations are shown for each constraint, otherwise, the mode of incorporation or lack thereof in the algorithm is described. First, the lower bounding slopes of a pipe are introduced, whereafter all other relevant constraints are addressed.

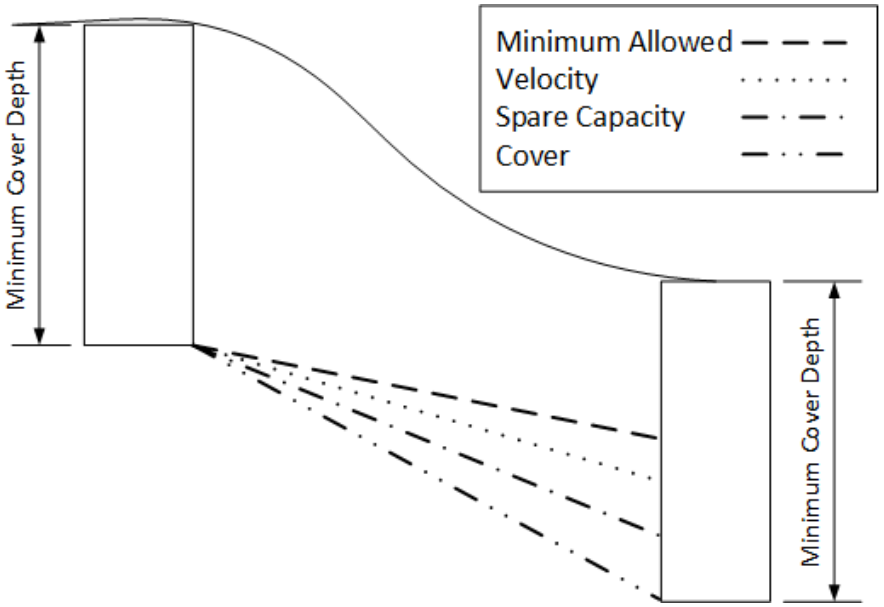


Figure 3.1: Minimum Required Slopes

The lower bounding slope of a pipe is influenced by a number of constraints, as indicated by fig 3.1.

3.3.4.1 Minimum Allowed Slope

The pipe is placed at the minimum allowable slope if this is compatible with the spare capacity, minimum velocity and cover depth constraints. If the maximum velocity constraint is exceeded, the diameter is modified to restore feasibility, as described in Section 3.3.5.

3.3.4.2 Minimum Required Cover Depth

Cover depth requirements may enforce a steep slope on pipes in order to achieve the necessary depth at the downstream manhole, as shown in Figure 3.1. In this case, if the slope at maximum velocity is exceeded the diameter can be modified to restore feasibility. Alternatively, the depths of the upstream end of the pipe as well as the connecting manhole can be increased to reduce the required slope. How the diameter is modified is, again, discussed later in Section 3.3.5.

3.3.4.3 Maximum Allowable Cover Depth

If a pipe is placed at its minimum feasible slope for a given diameter but then exceeds the maximum allowable cover depth, the diameter has to be modified to reduce the required minimum slope, and accompanying cover depth, to within the feasible range. However, since the algorithm attempts to place all pipes at their smallest allowable slope, the maximum allowable cover depth constraint is not enforced directly. If all pipes of a design are placed at their minimum allowable slopes and the maximum allowable cover depth is still exceeded, for example in the downstream areas of the network, the design is considered infeasible by the algorithm. This is the only constraint not directly enforced during calculations and, consequently, is the only potential cause of infeasible designs.

3.3.4.4 Commercially Available Diameters

The set of diameters available for selection, $\mathbf{D}$, is limited to only include commercially available pipe diameters.

3.3.4.5 Progressive Pipe Diameters

This set of eligible diameters for an individual pipe is limited to a subset of the set of all available diameters, $\mathbf{D}$, such that the smallest eligible pipe is larger than or equal to the largest immediate upstream pipe.

3.3.4.6 Progressive Pipe Depths

The outgoing pipe of a manhole is placed level to the lowest incoming pipe of the manhole.

3.3.4.7 Spare Capacity

Referring to constraint 2.3.2.4: Manning's equation for partially full flow conditions is used. From Manning's equation it is clear that flow rate, diameter and slope are interdependent variables. As described before, the accumulated flow rates are dependent on network layout. The diameter selections are limited to a set of commercially available diameters while the slope is a continuous variable with an upper and lower bound. Once flow rates and diameters are known the required slope can be calculated directly to satisfy Manning's equation, i.e. partially full flow that provides the required spare capacity of constraint 2.3.2.4. Substituting Manning's equation into Equation 2.3.2.4 and some manipulation yields:

$$A_p^{5/3} = \frac{(1 - SC)A_f^{5/3} P_p^{2/3}}{P_f^{2/3}} \tag{3.1}$$

Where:

A_p = Partial flow area $[m^2]$
SC = Spare Capacity ratio
A_f = Full flow area $[m^2]$
P_p = Partial wetted perimeter $[m]$
P_f = Wetted perimeter at full flow $[m]$

Substituting this into Manning's Equation for partial flow conditions, Q_p, after some manipulation yields:

$$Q_p = \frac{1}{n}\frac{(1 - SC)A_f^{5/3}}{P_f^{2/3}}\sqrt{S} \tag{3.2}$$

Where:

Q_p = Flow rate at partial flow conditions $[m^3/s]$
SC = Spare Capacity ratio
A_f = Full flow area $[m^2]$
P_f = Wetted perimeter at full flow $[m]$
S = Slope of the pipe $[m/m]$
n = Manning Roughness Coefficient

Equation 3.2 ensures the conditions of constraint 2.3.2.4 are satisfied if used to calculate the lower bounding slope, while additionally allowing for the calculation of the lower bounding slope in terms of the full flow area and wetted perimeter. Increasing the slope above this minimum value increases the spare capacity. Thus, an increase in slope does not violate the minimum spare capacity constraint. Using Equation 3.2 we can calculate the required slope in terms of the flow area and wetted perimeter. The only design parameter required to calculate these values is the diameter, which at this stage is known. Using equations 2.3.2.4 and Manning's Equation an expression for the required slope is found:

$$S = \left(\frac{Q_p n P_f^{2/3}}{(1 - SC) A_f^{5/3}} \right)^2 \tag{3.3}$$

Where:

S = Required slope to achieve minimum spare capacity $[m/m]$
Q_p = Flow rate at parial flow conditions $\left[m^3/s\right]$
n = Manning roughness coefficient
$P_f^{2/3}$ = Wetter perimeter at full flow $[m]$
SC = Spare Capacity Ratio
$A_f^{5/3}$ = Full flow area $\left[m^2\right]$

3.3.4.8 Minimum Velocity

The minimum allowable velocity constraint, constraint 2.3.2.2, combined with Manning's equation, yields:

$$v_{min} = \frac{1}{n} \frac{A^{2/3}}{P^{2/3}} \sqrt{S} \tag{3.4}$$

As v_{min} is a pre-defined known constant, the flow area, A, and wetted perimeter, P, can be determined if the flow depth can be determined. Then the required slope, S, to achieve the required minimum velocity can be calculated.
The flow depth can be calculated using the hydraulic equation $Q = vA$. The flow rate is known if the calculation is carried out in a topological sort order of the layout-graph, as at the upper ends of the network, as mentioned in Section 3.1, the flow rates are already known. The network layouts considered in this investigation are all trees, consequently the topological sort order can always be found. The

only unknown remaining is the flow area. Rewriting the hydraulic equation to allow solving with a line search algorithm and substituting v_{min} yields:

$$(Q - v_{min}A)^2 = 0 \tag{3.5}$$

Equation 3.5 is obviously a minimum when $Q = v_{min}A$, in all other cases a value > 0 is obtained. It should also be noted that it is important to solve for the flow depth, y, and not just the area, A, as in Equation 3.4 the wetted perimeter is also required. The flow area and wetted perimeter my be written in terms of the flow depth, refer to Figure 3.2 for parameter definitions of a circular profile.

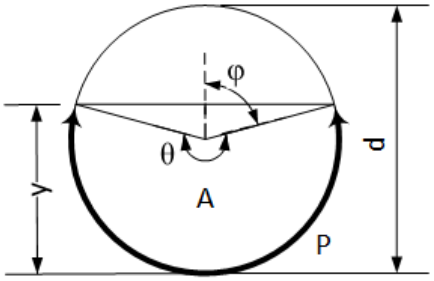


Figure 3.2: Parameters of a Circular Profile

$$\begin{aligned}
\phi &= \cos^{-1}\left(\frac{y - \frac{d}{2}}{\frac{d}{2}}\right) \\
A &= \frac{d^2}{8}(\theta - \sin(\theta)) \\
&\Rightarrow \\
A &= \frac{d^2}{8}\left(2\pi - 2\cos^{-1}\left(\frac{y - \frac{d}{2}}{\frac{d}{2}}\right)\right. \\
&\quad \left. - \sin\left(2\pi - 2\cos^{-1}\left(\frac{y - \frac{d}{2}}{\frac{d}{2}}\right)\right)\right) \\
P &= \frac{d}{2}\theta
\end{aligned}$$

Substituting the expressions for the area in terms of the flow depth into Equation 3.5 yields a highly implicit equation in terms of both the diameter and flow depth. Diameters are the selected variable of the hydraulic optimization procedure, so the flow depth can be solved for using a line search algorithm. In this implementation an interval halving algorithm (Rao, 2009) is used. Once the flow depth is known,

it is used in a Equation 3.6, a rearranged version of Equation 3.4, to solve for the minimum required slope to achieve the minimum allowed velocity. Increasing the slope increases the flow velocity, consequently an increase in slope does not compromise this constraint.

$$S = \left(\frac{v_{min} n P^{2/3}}{A^{2/3}}\right)^2 \tag{3.6}$$

3.3.4.9 Maximum Velocity

Maximum velocity is calculated exactly as for the minimum velocity, v_{min} is substituted with v_{max} in all equations. The maximum allowable velocity constraint is only relevant if the minimum allowable feasible slope of the pipe exceeds the maximum velocity slope, as described in sub sections 3.3.4.1 and 3.3.4.2.

3.3.5 Effect of a Diameter Change on Slopes

The optimization procedure developed here will increase or decrease diameters of pipes to reduce required slopes. In this section the effect that a change in diameter has on the relevant minimum allowable slope constraints, as derived in the previous section 3.3.4, is analysed. There are only two relevant minimum slope constraints, namely: Minimum Spare Capacity and Minimum Velocity. The objective, then, is to determine how a change in diameter effects the required slope to achieve minimum spare capacity and the required slope to achieve minimum velocity.

The effect of a change in diameter is evaluated by keeping the flow rate, manning roughness coefficient and constraint parameters constant while varying the diameter during required slope calculations. To determine the required slope at minimum spare capacity equation 3.3 is used, while to determine the required slope for minimum velocity equation 3.6 is used.

Using $Q = 15l/s$, $n = 0.015$, $v_{min} = 0.75m/s$ and $SC = 0.3(30\%)$, the required slopes for various diameters are shown in Figure 3.3. From this figure it is clear that increasing the diameter affects the required slopes differently. The required slope to maintain a minimum spare capacity decreases, while the required slope to achieve a specified velocity increases. This increase in slope for velocity is counter to what is common in engineering practice. When equation 3.4 is used to calculate the required slope the assumption of full flow conditions is often made in

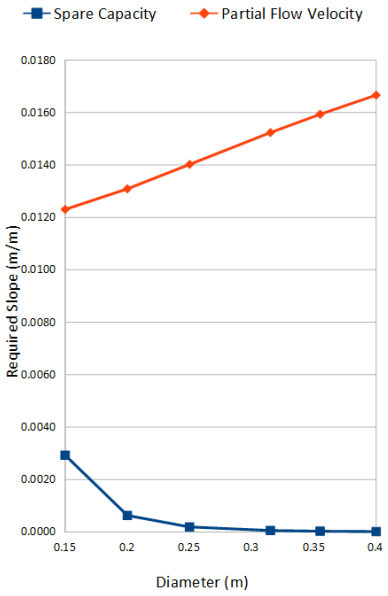


Figure 3.3: Effect of Diameter Increase on Slope

engineering practice. Under the assumption of full flow conditions increasing the diameter reduces the required slope to achieve the minimum velocity. To explain this discrepancy the hydraulic equation $Q = vA$, is used. If both the flow rate, Q, and the velocity, v, remain constant then the flow area, A, also remains constant. When assuming full flow conditions under the same minimum velocity, i.e. v remains constant, for a larger diameter the flow area increases and consequently so does the flow rate. It is due to this increase in flow rate and flow area that the required slope decreases. In the case of Figure 3.3, the flow rate and velocity is kept constant and consequently so is the flow area. When the diameter increases the flow area is maintained by a decrease in flow depth. This increase in diameter leads to an increase in wetted perimeter, despite the reduction in flow depth. Referring to equation 3.6, all the variables remain constant during a diameter increment apart from wetted perimeter, P, which increases. As the wetted perimter, P, is directly proportional to the slope, S, in Equation 3.6 this leads to an increase in the required slope.

This implies that increasing the diameter may be beneficial, i.e. lead to a reduction in required slope, in two cases: (i) when spare capacity is the active lower bounding slope constraint, and (ii) when the maximum velocity is the active upper bound constraint.

3.3.6 Relative Benefit of a Reduction in Slope

Refer to the formulation of the minimum allowable slope and minimum cover depth constraints in section 3.3.4. An increase in required slope to achieve a specified velocity can be used to restore feasibility in these cases.

It is important to note that if any decrease in slope is considered beneficial the minimum excavation depth solution is obtained. As noted before this solution has been shown to be between 5% and 15% worse than solutions found by other means. For this reason, the relative benefit of a decrease in slope due to a diameter change should be evaluated.

A factor, denoted with γ, is introduced which is used to evaluate the reduction in slope when a diameter is increased. The factor is calculated by:

$$\gamma = \frac{S_{i-1} - S_i}{S_{i-1}} \tag{3.7}$$

Where:

$$\begin{aligned} \gamma &= \text{Slope change factor} \\ S_i &= \text{Required slope for the current diameter } [m/m] \\ S_{i-1} &= \text{Required slope for the previous diameter } [m/m] \end{aligned}$$

If $\gamma \geq \gamma_b$, where $0.0 \leq \gamma_b \leq 1.0$, the increase in diameter is considered beneficial, i.e. the diameter increment leads to a sufficient reduction in slope that a reduction in capital expenditure is expected. Otherwise the previous diameter is accepted if it resulted in a feasible design. If the current diameter is the the smallest eligible diameter, this evaluation step is simply skipped.

γ_b, the beneficial slope change factor, is a predefined variable of the optimization procedure, similar to the evaporation rate of ACO or mutation rate of a GA. When the hydraulic optimization algorithm is applied to a single static layout the procedure should be run multiple times with different values of γ_b to determine the optimal solution. γ_b is best understood by evaluating the effects of its extreme values on the final solution. At $\gamma_b = 0$ any reduction in slope is always considered beneficial, i.e. the least buried depth solution is obtained. At $\gamma_b = 1$ the steepest feasible slope is used as no reduction in slope is ever considered beneficial, i.e. the smallest diameter which results in a feasible slope is used for each pipe. In most cases this results in capacity being the active minimum slope constraint. This is similar to the assumption Moeini and Afshar (2012) make in their hydraulic analysis procedure where the flow depth is assumed to always be the maximum

allowable, however their diameter is not necessarily the smallest feasible.

Figure 3.8 shows a sensitivity analysis of the γ_b parameter for the two case studies of Section 3.4.3. The results of Figure 3.8 are discussed in Section 3.4.3.

3.3.7 The Heuristic Algorithm

In this section the derivations, analyses and definitions of the previous sections are used to formalize a new heuristic hydraulic optimization algorithm. All hydraulic parameters need to be calculated for all network elements. Upon initialization of the algorithm the following properties of the network are known or specified:

- The spatial location of each manhole and their ground elevations.
- The direction of flow, as well as up and downstream connecting manholes or adjacency nodes, as described in Section 2.2.2.
- The expected inflow rates for each pipe represented as a 24-hour hydrograph at the upstream manhole of the pipe.
- The set of commercially available pipe diameters, $\mathbf{D}$, is defined.
- The upper and lower values to be enforced for each constraint are specified.
- The value of the beneficial slope change factor, γ_b, is specified.

The following hydraulic parameters are to be calculated or selected by some mechanism of the algorithm. Intermediate calculations, such as time delay during flow calculations, are not included here.

- The cumulative flow rates, Q, entering each manhole, represented as a 24-hour hydrograph.
- The diameter, d, for each pipe.
- The slope, S, for each pipe.
- The up and downstream cover depths, E, for each pipe.
- The flow velocity, v, for each pipe.
- The spare capacity, SC, for each pipe.
- The invert level, or height h, of each manhole.

A broad overview of the entire algorithm is presented first. Thereafter, the sub-algorithm for the diameter selection and slope calculation process is presented in detail.

1. The topological sort order of the layout is determined. This can always be found as layouts are restricted to branched-layouts, where no cycles are present. The calculation of hydraulic properties of pipes is to be completed in the topological sort order, starting from the upper ends of the network where no incoming pipes are present and thus the total flow rate in each single outgoing pipe is known.

2. The next manhole(s) are selected from the topological sort order. All hydraulic parameters are calculated or selected for the current manhole's outlet pipe. Flow rate is routed downstream, using contributor hydrograph theory, and added to the cumulative flow rate at the downstream manhole. This process is discussed in detail in Section 3.3.7.1.

3. The required invert level for the upstream manhole of the current pipe is calculated. The required invert level is, at a minimum, equal to the lowest, or deepest, invert level of all inflow pipes. If the diameter of outgoing pipe is so large that if placed at minimum required cover depth its invert level is lower than the lowest incoming pipe, the manhole's invert level is set to this invert level. As the calculations are completed in the topological sort order, all hydraulic parameters of all incoming pipes should have already been determined. If no incoming pipe is present, as is the case in the top ends of the network, the invert level is equal to the invert level required by the outgoing pipe.

4. Steps 2 and 3 are repeated, in the topological sort order, for each manhole and pipe pair until all hydraulic parameters have been calculated. Step 2 is omitted at the outlet node, as its outlet pipe is not modelled.

5. Once all parameters for all manholes and pipes have been calculated the network cost is determined, using Equation 2.1 and appropriate unit cost functions, such as Equation 2.2. It should be noted that the cost is only calculated once the entire solution has been completed: The algorithm relies entirely on the heuristic to make economical decisions during the procedure.

3.3.7.1 The Pipe Design Algorithm

The pipe design algorithm is used to determine the hydraulic parameters of each individual pipe of the network. Figure 3.4 shows a simplified flow chart of the

algorithm, which is described in detail below.
For Figure 3.4, variable definitions are as follows.

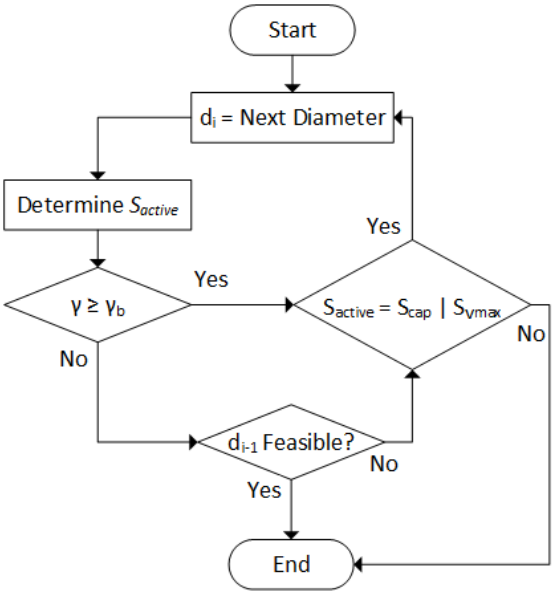


Figure 3.4: Diameter Selection Procedure

$\mathbf{D}_{el}$	=	The eligible set of diameters, satisfying constraints 2.3.2.5 and 2.3.2.6
d_i	=	The current diameter in $\mathbf{D}_{el}$
d_{i-1}	=	The previous diameter in $\mathbf{D}_{el}$
S_{active}	=	The active, or critical, minimum slope
S_{cap}	=	Required slope to achieve Spare Capacity
$S_{v_{max}}$	=	Required slope to achieve Maximum Velocity
γ	=	Slope change factor
γ_b	=	Beneficial slope change factor

The algorithm starts by selecting the smallest eligible diameter, d_i, and increments the diameter by one size each iteration. The minimum required slopes for cover, spare capacity, minimum and maximum velocity are calculated as described in Section 3.3.4. The largest of all the minimum slopes, including minimum allowed, is selected as the active, or critical, lower bounding slope. If the lower bounding slope exceeds the maximum allowable velocity slope, the latter is used as the active

slope, otherwise the active lower bounding slope is used as the active slope.

$$
\begin{aligned}
&S_{lbound} = MAX\left(S_{min}, S_{cover}, S_{v_{min}}, S_{cap}\right)\\
&if \ \left(S_{lbound} > S_{v_{max}}\right)\\
&\quad S_{active} = S_{v_{max}}\\
&else\\
&\quad S_{active} = S_{lbound}
\end{aligned}
$$

Where:

S_{active}	=	The active, or critical, slope for the pipe
S_{lbound}	=	The active, or critical, lower bounding slope for the pipe
S_{min}	=	The minimum allowable slope, as specified by Constraint 2.3.2.1
S_{cover}	=	The minimum required slope to achieve the minimum allowable cover depth at the downstream manhole
$S_{v_{min}}$	=	The minimum required slope to achieve the minimum allowable flow velocity, Equation 3.6
$S_{v_{max}}$	=	The maximum slope to not exceed the maximum allowable flow velocity, Equation 3.6
$S_{v_{cap}}$	=	The minimum required slope to achieve the required spare capacity, Equation 3.3

If the current diameter, d_i, is not the smallest diameter in $\mathbf{D}_{el}$ then previous slope information is available and γ is calculated using Equation 3.7. If d_i is the smallest diameter, this step is simply omitted for the iteration. Then, If $\gamma \geq \gamma_b$ the change in diameter is considered beneficial, i.e. the reduction in slope is expected to reduce the cost. If the change in diameter is considered beneficial, it may be possible that further increments will be beneficial. If the active slope is either the spare capacity, S_{cap}, or maximum velocity, $S_{v_{max}}$, slope the diameter is incremented and the procedure restarted to once again evaluate the benefit of a diameter increment. If neither spare capacity nor maximum velocity were the active slope the current diameter is accepted as a diameter increase will not reduce the required slope. If no diameter increment was possible the current diameter is accepted and the maximum velocity slope prioritised over spare capacity. Where the diameter increment is not considered beneficial,$\gamma < \gamma_b$, the previous diameter is checked for feasibility. If the previous diameter resulted in a feasible slope it is accepted as the diameter for the pipe. If it did not the active slope constraints are checked to determine if a diameter increment may be beneficial, if not the current diameter,

assuming it is feasible, is accepted, otherwise the diameter is incremented.
Once a pipe has its diameter and slope calculated, using the procedure above, the upstream hydrograph of the pipe is routed downstream, with time-delay, and added to the cumulative flow rate in the downstream manhole, as described in Section 2.3.3.1.

As mentioned in Section 3.3.4.2, if the case is encountered where maximum slope to adhere to maximum velocity, $S_{v_{max}}$, is exceeded by the lower bounding slope, S_{lbound}, which is equal to the minimum required slope to achieve minimum cover depth, S_{cover}, an alternative to a diameter increment is used.

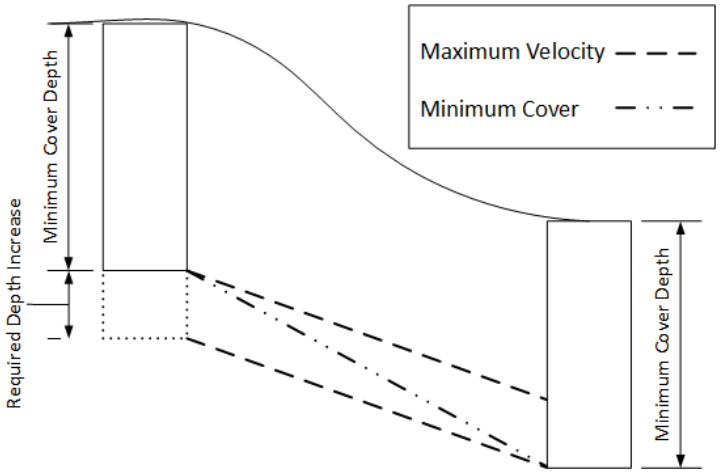


Figure 3.5: Cover Slope Exceeds Maximum Velocity Slope

In this case the pipe should placed a the maximum allowable slope for maximum velocity, $S_{v_{max}}$. The required invert level at the upstream manhole is calculated so as to reduce the required slope to adhere to both cover and maximum allowable velocity. The invert level of the upstream manhole is adjusted, and the pipe placed at the calculated depths and slope, as shown in Figure 3.5. If, however, this increase in invert level exceeds the maximum allowable cover depth, then the only possible course of action is to increase the diameter until feasibility is restored.

3.4 Case Studies

Two case study problems from the relevant literature are used to determine the effectiveness of the HOMS algorithm. In both cases a maximum relative flow depth and static flow values are specified. In order to ensure the proposed heuristic

algorithm is accurately compared to the case study algorithms the relative flow depth of the result cannot exceed the specified value. Since the maximum relative flow depth parameter is never directly incorporated, the spare capacity is set to a value which results in a maximum relative flow depth very close, but still below, the specified maximum. The static flow rates are incorporated by simply disabling the downstream routing of flow rates during the optimization of the benchmarks problems to allow direct comparison of results. For both case studies γ_b is varied over the interval $[0.0;\ 1.0]$ with a step of 0.05.

3.4.1 Case Study 1: Mays and Wenzel Network

The first example is a network originally designed by Mays and Wenzel (1976) which has since been solved by other investigators. The network, shown in Figure 3.6, has 21 manholes and 20 pipes.

The unit cost functions of Equation 2.1 proposed for this problem are (Meredith,

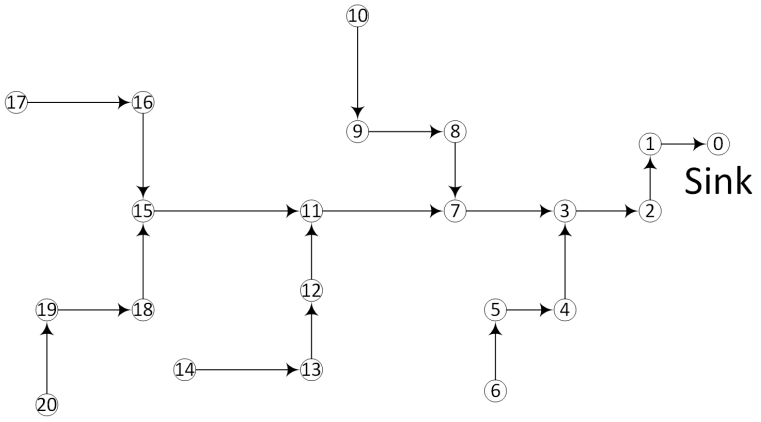


Figure 3.6: figure

Network Layout of Case Study 1

1972):

$$K_i = \begin{cases} 10.98d + 0.8E - 5.98 \\ \qquad \text{if d } \leq 3ft \text{ and E} \leq 10ft \\ 5.94d + 1.166E + 0.504Ed - 9.64 \\ \qquad \text{if d} \leq 3ft \text{ and E} > 10ft \\ 30.0d + 4.9E - 105.9 \\ \qquad \text{if d} > 3ft \end{cases}$$

$$K_j = 250 + h_m^2 \tag{3.8}$$

Where:

d = The diameter of pipe i $[ft]$
E = The average cover depth of pipe i $[ft]$
h_m = The height of manhole m $[ft]$

The pipe unit cost, K_i, is obtained in $\$/ft$ and the manhole unit cost, K_j, in \$. The Manning coefficient for all pipes is taken as 0.013. The maximum allowable relative flow depth, $\left(\frac{y}{d}\right)$, is taken as 0.82. The minimum allowable cover depth, E_{min}, is taken as 2.4m. The minimum velocity, v_{min}, and maximum velocity, v_{max}, are taken as 0.6m/s and 3.6m/s respectively. The minimum allowable slope is 0.001 (m/m). The set of commercially available pipes $\{\mathbf{D}\} = \{304,8$mm(12in), 381mm(15in), 457.2mm(18in), 533.4mm(21in), 762mm(30in), 914.4mm(36in), 1066.8mm(42in), 1219.2mm(48in)$\}$. Table 3.1 shows the necessary data for the benchmark problem.

Table 3.1: Data of Case Study 1

Pipe	Ground Elevation (m)		Length (m)	Design Discharge (m^3/s)
	Upstream	Downstream		
1-0	136.55	135.64	186.54	2.6617
2-1	137.46	136.55	152.40	2.5201
3-2	138.65	137.46	121.92	2.4635
4-3	140.21	138.65	105.23	0.2548
5-4	141.43	140.21	91.44	0.1699
6-5	142.65	141.43	121.92	0.1132
7-3	141.73	138.65	172.21	2.0104
8-7	143.26	141.73	106.68	0.5663
9-8	144.78	143.26	106.68	0.4530
10-9	147.83	144.78	152.40	0.2548
11-7	143.26	141.73	152.40	1.2459
12-11	144.78	143.26	106.68	0.4530
13-12	147.83	144.78	137.16	0.3398
14-13	149.35	147.83	147.64	0.2265
15-11	146.30	143.26	167.68	0.6229
16-15	147.83	146.30	131.08	0.2275
17-16	149.35	147.83	121.92	0.1132
18-15	148.49	146.30	106.68	0.2548
19-18	150.88	148.49	121.92	0.1982
20-19	152.40	150.88	106.68	0.1132

Table 3.2 shows the results obtained by various methods for the problem. The proposed heuristic method is able to achieve a result close to the non-classic op-

timization methods, while requiring 6 milliseconds of computation time using a personal computer with 3rd generation Intel Core i7-3630QM.

Table 3.2: Results of Case Study 1

Model	Cost (US$)	Function Evaluations
Mays and Wenzel (1976)	265 775	-
Robinson and Labadie (1981)	275 218	-
Miles and Heaney (1988)	245 874	-
Afshar (2006) ACO	241 496	29 900
Afshar et al. (2011) CA	253 483	50
Afshar (2012) RGA	241 896	100 000
Proposed Heuristic Method	**246 795**	-

Table 3.3 shows the detailed solution obtained by the proposed heuristic method. It indicates the method's ability to differentiate between the benefit of minimum possible slopes and diameter increments. Note that when capacity or maximum velocity is the active slope constraint a diameter increase may be beneficial, however using γ the algorithm was able to determine that the current diameter and slope pair is more economical than the diameter incremented pair. Note that for pipe 7-3 the special case where the cover slope exceeds the maximum allowable velocity slope is encountered. The cover depth of the pipe at the source end and the depth of its source manhole is increased to achieve the required slope.

3.4.2 Case Study 2: Kerman Network of Iran

The second problem, shown in Figure 3.7, is part of the "Kerman" network in Iran (Afshar et al., 2011).

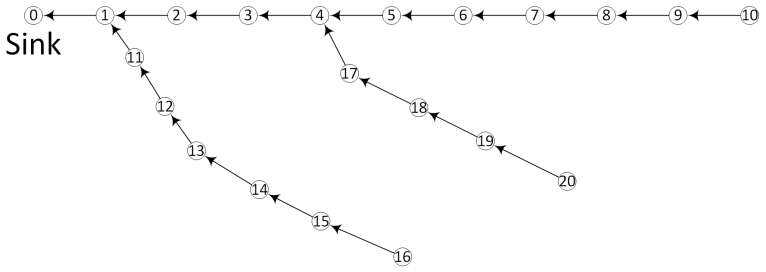


Figure 3.7: figure

Network Layout of Case Study 2

Table 3.3: Heuristic Solution of Case Study 1

Link	Slope (m/m)	Active Constraint	Diameter (mm)	Max Velocity (m/s)	Relative Flow Depth $(\frac{y}{d})$	Cover Depth (m) Source	Target
1-0	0.0043	Capacity	1219.2	2.6013	0.8188	2.5326	2.4231
2-1	0.0078	Capacity	1066.8	3.2168	0.8188	2.40	2.6850
3-2	0.0098	Cover	1066.8	3.5489	0.7251	2.4	2.4
4-3	0.0194	Capacity	381.0	2.5513	0.8184	2.4	2.8861
5-4	0.0107	Cover	381.0	1.8777	0.7402	2.6420	2.4
6-5	0.0126	Capacity	304.8	1.7710	0.8184	2.4	2.7182
7-3	0.0120	Max Vel	914.4	3.5914	0.7949	3.4133	2.4
8-7	0.0160	Capacity	533.4	2.8930	0.8184	2.4	2.5730
9-8	0.0142	Cover	533.4	2.6901	0.7051	2.4	2.4
10-9	0.0200	Cover	381.0	2.5857	0.8066	2.4	2.4
11-7	0.0115	Capacity	762.0	3.1188	0.8184	2.7016	2.9289
12-11	0.0142	Cover	533.4	2.6901	0.7051	2.4	2.4
13-12	0.0222	Cover	457.2	2.9854	0.6543	3.0723	2.4
14-13	0.0154	Capacity	381.0	2.2679	0.8184	2.4	3.1485
15-11	0.0193	Capacity	533.4	3.1822	0.8184	2.7316	2.9302
16-15	0.0154	Capacity	381.0	2.2679	0.8184	2.4	2.8840
17-16	0.0126	Capacity	304.8	1.7710	0.8184	2.4	2.4182
18-15	0.0205	Cover	381.0	2.6218	0.7949	2.4	2.4
19-18	0.0196	Cover	381.0	2.4822	0.6602	2.4	2.4
20-19	0.0142	Cover	304.8	1.8741	0.7715	2.4	2.4

For this example the unit cost functions given in Equation 2.2 (Afshar et al., 2011) are used in conjunction with Equation 2.1 for cost calculations.

$$\begin{aligned} K_i &= 1.93e^{3.43d_i} + 0.812E_i^{1.53} + 0.437d_iE_i^{1.47} \\ K_j &= 41.46h_j \end{aligned} \tag{3.9}$$

Where:

d_i = The diameter of pipe i $[m]$

E_i = The average cover depth of pipe i $[m]$

h_j = The height of manhole j $[m]$

The Manning coefficient for all pipes is considered as 0.013. The maximum allowable relative flow depth, $\left(\frac{y}{d}\right)$, is considered as 0.82. The minimum allowable cover depth, E_{min}, is considered as 2.45m. The minimum velocity, v_{min}, and maximum velocity, v_{max}, is considered as 0.3m/s and 3.0m/s respectively. The minimum allowable slope is 0.001 (m/m). The set of commercially available pipes, $\{\mathbf{D}\}$ = {150mm, 200mm, 250mm, 300mm, 400mm, 500mm, 600mm, 700mm, }.

The necessary data for the definition of benchmark problem 2 is shown in Table 3.4.

Table 3.4: Data of Case Study 2

Pipe	Ground Elevation (m)		Length (m)	Design Discharge (m^3/s)
	Source	Target		
1-0	65.42	64.5	320	0.1473
2-1	65.82	65.42	340	0.1047
3-2	66.22	65.82	350	0.1012
4-3	67.28	66.22	470	0.0967
5-4	68.24	67.28	400	0.0387
6-5	69.85	68.24	450	0.0366
7-6	71.19	69.85	300	0.0340
8-7	72.10	71.19	260	0.0324
9-8	73.66	72.10	460	0.0304
10-9	74.59	73.66	260	0.0279
11-1	66.10	65.42	590	0.0446
12-11	66.80	66.10	400	0.0403
13-12	68.60	66.80	500	0.0319
14-13	70.10	68.60	400	0.0300
15-14	71.50	70.10	400	0.0264
16-15	73.00	71.50	400	0.0211
17-4	68.40	67.28	440	0.0596
18-17	69.30	68.40	310	0.0580
19-18	69.90	69.30	260	0.0562
20-19	70.70	69.90	300	0.0549

The problem is solved using the proposed heuristic method. Table 3.5 shows the results obtained by various methods. All values shown in Table 3.5, except for the newly proposed heuristic method, were presented by Afshar et al. (2011). Table 3.5 shows the algorithm's ability to obtain a near optimal solution while also only requiring 6 milliseconds of computation time using a personal computer with 3rd generation Intel Core i7-3630QM.

Table 3.6 shows the detailed solution produced by the heuristic method for benchmark problem 2. This solution again demonstrates the algorithm's ability to differentiate between the relative benefit of an absolute minimum slope and a diameter increment.

3.4.3 Sensitivity Analysis of the Slope Change Factor, γ

Figure 3.8 shows the a sensitivity analysis of the γ_b parameter for both case studies.

Table 3.5: Results of Case Study 2

Model	Cost (US$)	Function Evaluations
Mansuri, Khanjani	83 116	-
BFGS	82 732	-
Fletcher-Reeves	81 553	-
GA	77 736	100 000
Cellular Automata	80 879	20
Proposed Heuristic Method	**78 779**	-

Table 3.6: Heuristic Solution of Case Study 2

Link	Slope (m/m)	Active Constraint	Diameter (mm)	Max Velocity (m/s)	Relative Flow Depth $(\frac{y}{d})$	Cover Depth (m) Source	Cover Depth (m) Target
1-0	0.0029	Cover	500	1.1286	0.6309	2.45	2.45
2-1	0.0012	Cover	500	0.7344	0.6816	2.45	2.45
3-2	0.0011	Cover	500	0.7214	0.6719	2.45	2.45
4-3	0.0023	Cover	400	0.8985	0.7988	2.45	2.45
5-4	0.0023	Cover	300	0.7354	0.6973	2.4932	2.45
6-5	0.0038	Capacity	250	0.8530	0.8164	2.45	2.5432
7-6	0.0045	Cover	250	0.9109	0.7109	2.45	2.45
8-7	0.0035	Cover	250	0.8161	0.7539	2.45	2.45
9-8	0.0034	Cover	250	0.7957	0.7266	2.45	2.45
10-9	0.0036	Cover	250	0.8008	0.6680	2.45	2.45
11-1	0.0012	Cover	400	0.5949	0.5762	2.45	2.45
12-11	0.0018	Cover	300	0.6537	0.8145	2.45	2.45
13-12	0.0036	Cover	250	0.8211	0.7383	2.45	2.45
14-13	0.0038	Cover	250	0.8286	0.6914	2.45	2.45
15-14	0.0032	Cover	250	0.7628	0.6641	2.5541	2.45
16-15	0.0041	Capacity	200	0.7684	0.8164	2.45	2.6041
17-4	0.0025	Cover	400	0.8628	0.5391	2.45	2.45
18-17	0.0029	Cover	400	0.9277	0.4980	2.8274	2.45
19-18	0.0034	Capacity	300	0.9076	0.8184	2.6162	2.9274
20-19	0.0032	Capacity	300	0.8866	0.8184	2.45	2.6162

From these figures it is clear that the final cost of a solution is not very sensitive to small changes of γ_b around the optimum. However, some improvement in the final cost is gained when running the algorithm for multiple values of γ_b with a small increment. As described before, as γ_b increases only severe reductions in slope are considered beneficial, resulting in very steep slopes and excessive excavation. The severe increase in cost seen in Figure 3.8 (a) and (b) around $\gamma_b = 0.5$ is due to this excessive excavation requirement at steep slopes.

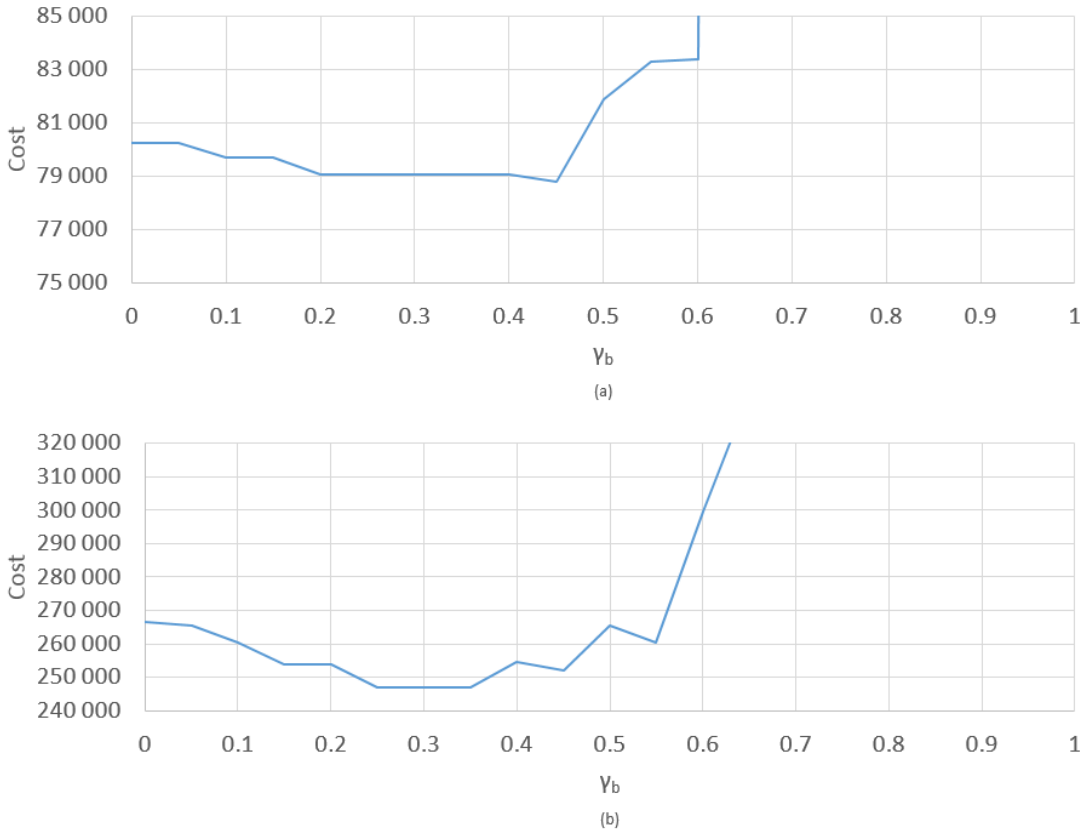


Figure 3.8: Sensitivity of γ_b for (a) Case Study 1 and (b) Case Study 2

As stated previously, to find the optimal value of γ_b for a static layout multiple runs of the heuristic algorithm is required, each with a different value of γ_b. Of course, this does make the algorithm slightly less computationally efficient, though with a computation time of 6ms on a personal computer the algorithm is certainly not expensive even with the multiple repetitions. However, from Figure 3.8 it is observed that the procedure may be interrupted once a significant increase in solution cost is observed for two or more consecutive increments of γ_b.

When the hydraulic optimization algorithm is combined with a layout optimization algorithm a calibration strategy may be adopted to avoid repeating the hydraulic optimization algorithm multiple times for each solution, assuming the layout optimization problem produces a large number of layouts. Figure 3.8 shows that solution fitness is not very sensitive to small changes of γ_b. Consequently, if at the start of the layout optimization procedure γ_b is calibrated it is reasonable to assume that the calibrated value will still be very near the eventual best solution's

optimum. This calibration procedure can be expanded to allow the layout optimization to repeat the calibration of γ_b as it progresses and improves its solution and at static, pre-defined, increments. It is recommended that the optimization algorithm is applied with multiple values of γ_b to the final layout found by the layout optimization algorithm to ensure the optimal value of γ_b is used in the final design.

3.5 Summary and Conclusions

In this chapter existing methods to solve the hydraulic optimization problem were reviewed. While many algorithms exists, they all have some characteristics restricting their usefulness for this investigation. The classic algorithms most often have restricting assumptions, severely limiting their field of application. Furthermore, the results obtained by the classic algorithms, for the two case studies, are significantly worse than those obtained by the meta heuristic alternatives, making them unattractive options. The metaheuristic algorithms, while able to find very good solutions, lack the required computational efficiency for application in this investigation. Consequently, a new algorithm was developed which aims to maintain solution quality similar to that of the metaheuristic options while requiring very little computation time.

The algorithm relies upon slope information derived directly from the constraints of the hydraulic optimization problem. The constraints are, where possible, manipulated to allow calculation of a minimum or maximum required slope to satisfy the specific constraint. In addition to the constraints incorporated directly through slope information, the incorporation of the other constraints, for which slope information cannot be derived, is briefly described. Then, using the derived slope information, an algorithm is proposed which iterates diameters and calculates the minimum required slope which adheres to all relevant constraints. The effect that a diameter change has on the required minimum slope was investigated. It was found that a diameter change reduces the required slope to adhere to the minimum spare capacity constraint, and increases the required slope to adhere to a velocity constraint. This information is then used to determine when a diameter increase is expected to lead to a reduction in slope.

Walters (1985) states that the least buried depth solution is not the true optimal solution. To avoid finding the least buried depth solution a new slope change factor, γ, is introduced. Using γ the change in a slope during a diameter increment is evaluated and compared to a pre-defined value γ_b, the beneficial slope change factor. This value is specified beforehand, and allows the algorithm to undo a di-

ameter increment and use a smaller diameter at a steeper slope if the slope change factor was not greater than the beneficial factor. A sensitivity analysis of γ revealed that changing the value does improve the final solution, and that it was not very sensitive to changes around the optimal value.

The algorithm was used to solve two case study problems found within the literature. In both cases the algorithm was able to discern when diameter increments are truly beneficial. Furthermore, the algorithm was able to find solutions within 2% of metaheuristic alternatives while requiring only 6ms of computation time on a personal computer. These characteristic make it ideally suited to be combined with a metaheuristic layout creation algorithm where for each layout the hydraulic optimization procedure must be repeated.

Chapter 4

Layout Optimization

Layout optimization of a sewer network is the part of the network optimization problem in which the position of pipes and manholes, as well as the flow direction in pipes, have to be determined. Sewer network layouts are often dictated by other infrastructure, such as roads or buildings, making random placement of manholes and pipes impractical. For this reason, rather than create an algorithm which determines all aspects of the network's layout, most layout optimization research has been concerned purely with determining the flow direction of each pipe for a given base layout. This part of the problem has been studied less than the hydraulic optimization problem. In this chapter an overview of the layout optimization problem is presented. State of the art layout optimization algorithms are reviewed. New Layout optimization algorithms, with varied characteristics, are proposed. In Chapter 5 the newly proposed algorithms are combined with the HOMS algorithm from Chapter 3. Consequently, this section includes no evaluation of algorithm performance as evaluating evaluating a layout requires the hydraulic analysis to be completed as well.

4.1 Problem Statement

Layout optimization of sewer networks is a two part problem. The first requires the exact location of manholes and pipes to determined. The second, how these pipes connect to manholes, as well as the flow direction within the pipe has to be determined, as described in Chapter 2. Theoretically, this problem has an infinite number of solutions for a given area. Practically, however, the surrounding infrastructure in the area will often dictate where manholes and sewer connections can be placed. For this reason, sewer network layout optimization in the literature is often only concerned with the second part of the problem. The first part is solved beforehand, presumably by the design engineer. Another common simplifying as-

sumption is to exclude cycles and divergence structures from the layout, i.e. the resulting network is branched, and all manholes only have a single outlet pipe.

With these assumptions and exclusions in place, the layout optimization problem is classified as a variant of the degree-constrained minimum spanning tree (d-MST) problem, a well known problem in the mathematical field of graph theory. In sewer network optimization, only the out-degree of each node, or manhole, is constrained and no clear definition of a minimum exists. It is possible to determine the number of spanning-trees of a graph in polynomial time, however determining what all of these spanning trees actually are is a much more strenuous and complex task. Then, in theory at least, it is possible to apply brute force and determine each spanning-tree of a graph and determine which is the minimum. However this is computationally too complex. Consequently, the problem is commonly solved using metaheuristic algorithms. Furthermore, in the case of sewer networks, the fitness of a layout is dependent on the hydraulic design. Consequently a brute force approach might not result in the true optimum, as there is no guarantee that the hydraulic design would be completed optimally for each layout.

It is important to note that the layout optimization problem can not be done without some form of hydraulic optimization in place for sewer networks. It would be extremely difficult, if not outright impossible, to accurately determine the capital investment cost for a given sewer network based purely on its layout. Consequently, the evaluation of a layout is dependent on the hydraulic design, as described in Chapter 3, which is an optimization problem in its own right. Consequently, layout optimization algorithms are almost always combined with some form of hydraulic optimization or heuristic design procedure resulting in a simultaneous sewer network optimization algorithm. Due to the complexity of such algorithms most research has been done on the hydraulic optimization problem, while the layout remains static (Lejano, 2006).

The layout optimization algorithm must be able to determine the spanning-tree, i.e. a layout, for a given base graph representing a sewer network, which results in the lowest possible construction capital expenditure for the complete design. In broad terms, there are three approaches to this:

1. **Complete enumeration**. In this approach all feasible layouts are generated and the hydraulic design of each is completed individually (Diogo *et al.*, 2000; Diogo and Graveto, 2006). While this approach is very useful for finding the best layout, its application is only practical for small problems.

2. **Separated design**. This approach separates the two design problems, ei-

ther through manual layout design or through simplified objective functions. Once the optimal layout is found, the optimal hydraulic parameters for this layout is determined by a separate algorithm (Pan and Kao, 2009; Haghighi, 2013). While this approach is useful for large scale problems, it is difficult to determine true optima (Haghighi and Bakhshipour, 2015).

3. **Simultaneous design**. The layout and element size problems are optimized simultaneously (Li and Matthew, 1990; Moeini and Afshar, 2012; Haghighi and Bakhshipour, 2015). While this approach is the most promising in finding true global optima for large solutions its implementation is the most complex and requires complex formulations and specific design algorithms (Haghighi and Bakhshipour, 2015). The fitness of a solution is calculated taking both layout and hydraulic parameters into account simultaneously.

4.2 State of the Art

This section describes some of the existing layout optimization procedures in the literature. A brief overview of some existing work regarding the d-MST problem is also presented. The aim is to give the reader an understanding of what has been done previously and what the shortfalls and strengths of the algorithms are.

Many optimization techniques have been applied to the layout optimization problem of sewer networks. In the past, dynamic programming methods were developed which could determine layouts under very strict, limiting conditions and assumptions. These algorithms will be referred to here, as with the hydraulic optimization algorithms, as classic algorithms. In more recent times, with the increase in computational power of computers and the arrival of efficient metaheuristic algorithms, algorithms which simultaneously solve both the layout and hydraulic optimization problems efficiently have been developed. The focus will be on the layout optimization procedures of the simultaneous algorithms.

4.2.1 Classic Algorithms

As with the hydraulic optimization problem almost all classic layout optimization algorithms relied on Dynamic Programming (DP) or some variation thereof. Section 3.2.1 includes a brief overview of DP.

Walters (1985) developed a DP algorithm capable of optimizing layout. In his approach the algorithm is responsible for stagewise selecting manhole positions from a predefined set. A stage is defined as placing a new manhole, as well as

connecting the manhole to predefined upstream manholes. The algorithm's first stage(s) is to determine the position, from a discrete grid, for those manholes which have no incoming connection. Thereafter, the algorithm proceeds downstream towards the outlet via its predefined stages placing manholes at the position which results in the lowest cumulative cost at the current stage. This algorithm has some obvious drawbacks. Inevitably, unless an infinite grid is used for each manhole, the final result will be sub-optimal. Furthermore, it is often the case that manholes can only be placed in a very select few locations due to existing infrastructure. Consequently, the algorithm's grids, when applied to practical problems, are very limited in size and often only include a single location for a manhole. In modern applications the positions of manholes and pipes are most often predetermined precisely for this reason.

Li and Matthew (1990) used a nonlinear programming model which incorporated both the topographic factors and hydraulic factors simultaneously. They used Dijkstra's algorithm, a well known algorithm in graph theory, to determine the shortest path spanning tree with weight being the product of pipe length and the average ground elevation. This spanning tree was used as the initial layout, thereafter after a Searching Direction method is deployed to cleverly alter the layout until another spanning tree is found and the hydraulic model can be applied. This process is repeated iteratively until the final solution cost does not decrease within a recommended three iterations.

As with the classic algorithms developed to solve the hydraulic optimisation problem the most severe restrictions of these algorithms are the limitations on the search space. With these approaches, a very small sample of the total possible spanning trees of a base graph are considered, making true optimality very unlikely.

4.2.2 Metaheuristic Algorithms

Diogo *et al.* (2000) used two layout optimization algorithms in their simultaneous layout and element size optimization algorithm. In both cases the algorithms starting points were created by using a Tree Growing (TG) algorithm to produce layouts at random starting form the sink or outlet nodes. They developed both a Simulated Annealing (SA) and Genetic Algorithm (GA), neither with any modifications, to optimize the layout. Their models incorporated multiple outlets, which could be processing plants or other connections, as well as pumping stations. Due to the multiple outlets a single solution to their problems results in a forest of layouts, rather than a singly tree, as each outlet caters only for a single tree. They executed several runs many times with small individual computing times, and then

adapting both algorithm and model parameters based on previous results to improve the final solutions. It is stated that the GA often performed better than the SA algorithm. While there are no obvious short comings to their approach, more modern applications of GA exists which have been shown to improve existing results.

Diogo and Graveto (2006) developed a comprehensive enumeration model which enumerates all the feasible forests from an undirected planar base graph, incorporating knowledge of the problem specific constraints to exclude some forests, and using the dynamic programming model of Diogo *et al.* (2000). This proved to be a very effective strategy in locating the optimal solution, however its practicality is limited to only small networks where the total number of spanning trees is relatively low, severely restricting its practicality. Comparing the results of the enumeration model to the SA of Diogo *et al.* (2000) revealed that for the small problem under consideration the SA algorithm was able to find the optimal solution, revealed by the complete enumeration of all layouts, in approximately 70% of its run while requiring only a fraction of the computation time and function evaluations full enumeration required. This further demonstrates that the practicality of full enumeration is not the best approach as its only really practical, with the available computation power, for small problems where the metaheuristic of general search algorithms are able to find the optimal solution more often than not anyway.

Moeini and Afshar (2012) proposed an Ant Colony Optimization (ACO) algorithm combined with a TG algorithm which performs both layout and diameter selections simultaneously. The TG algorithm is initialized from the outlet node, and then constructs the set of eligible pipes which, when added, result in a tree. The eligible set is then expanded to include pairs of pipes and feasible diameters. For example, if pipe A is eligible to be added, and the diameters 100mm, 200mm and 300mm are feasible for Pipe A, the decision pairs for pipe A are [A, 100mm], [A, 200mm] and [A, 300mm]. The ants are responsible for selecting such a pair from the set of eligible pairs. This approach significantly increases the number of decisions at each decision point within the algorithm, increasing algorithm complexity. They assume all pipes flow at capacity when performing the hydraulic analysis. Despite its increased complexity the algorithm is able to produce good solutions, largely due to the attached TG algorithm governing the construction of the layout. It is worth mentioning here that the TG algorithm used by Moeini and Afshar (2012) can be classified as the Growing Tree Algorithm. The Growing Tree Algorithm is very flexible in nature, and can, with minor tweaks, behave like other tree growing algorithms, such as Prim's or the Recursive Backtracker (Buck, 2015).

Haghighi and Bakhshipour (2015) combined previous works, namely the loop-

by-loop cutting algorithm (Haghighi, 2013) and an Adaptive GA (Haghighi and Bakhshipour, 2012), with a Tabu Search (TS) algorithm to create an effective hybrid algorithm for simultaneous layout and element size optimization. The loop-by-loop cutting algorithm populates a matrix based on the undirected base graph, or base layout, which models all loops within the network. The matrix is then modified iteratively to 'cut' loops from the layout one at a time. This process is repeated until a spanning tree is produced. Their Adaptive GA determines pipe diameters, pipe slopes and whether or not a pump station is present at the upstream end of a pipe. The algorithm adapts the parameter values of the GA to fall within feasible ranges of the corresponding design variable by linear interpolation. The TS algorithm governs how the loop-by-loop cutting algorithm performs its cuts to produce a spanning tree of the base graph.

4.2.3 Degree-Constrained Minimum Spanning Tree Algorithms

The degree-constrained spanning tree (d-ST) problem is a well known problem in graph theory. In essence, a spanning tree of a base graph has to be produced where the maximum vertex degree, or number of connections for a given vertex, is limited to a constant. The degree-constrained minimum spanning tree problem (d-MST) adds a layer of complexity on top of the d-ST problem. It assigns weights to edges, for example length, then the d-MST problem aims to find the d-ST where the sum of the weights of all edges is the minimum possible. The d-MST problem is known to be NP-hard (Bui and Zrncic, 2006). Many heuristic algorithms exists to solve the d-MST problem. Bui and Zrncic (2006) showed that ACO algorithms perform well for the solution of d-MST problems. Later, (Bau *et al.*, 2008) developed ACO algorithms to solve the d-MST problem using both Prim's and Kruskal's algorithms. Prim's algorithm operates by selecting nodes to add to the tree, while Kruskal's adds edges. In their study they found Kruskal's algorithm produced, on average, superior results for the d-MST problem. In their applications, the algorithms were used to populate a set of eligible decisions at a decision point of the ants, similar to how Moeini and Afshar (2012) use their TG algorithm. However, rather than rely on the greedy approach of these algorithms, which select the minimum currently eligible edge, the ants are responsible for performing the selections.

While these applications can not be directly applied to sewer networks, valuable insight is still gained. Bui and Zrncic (2006) found that ACO algorithms performed exceedingly well when solving the d-MST problem. However, despite this ACO is not often used, excluding Moeini and Afshar (2012), to solve sewer network layouts as researchers prefer other metaheuristic, heuristic or DP approaches.

Further, from Bau *et al.* (2008) it becomes clear that the performance of algorithms, even when coupled with a metaheuristic, depends heavily on how the eligible set of edges is compiled, and that minor differences or tweaks may greatly vary results based on the characteristics of a problem, even when only a single design variable exists. In the case of sewer networks there are multiple design variables which need to be solved simultaneously, increasing the complexity and thus the likelihood that different selection strategies will perform differently depending on the problem.

4.3 Layout optimization by Ant Colonies

Building on the findings of Bui and Zrncic (2006) that Ant Colony Optimization (ACO) algorithms are well suited to solving to the d-MST problem, ACO is used throughout this investigation for layout optimization. Furthermore, Bau *et al.* (2008) found that an algorithm's performance depends on its selection strategy. Consequently, multiple selection strategies are developed. The ACO algorithm of Moeini and Afshar (2012) mentioned in Section 4.2.2 used, perhaps unkowingly, the findings of both Bui and Zrncic (2006) and Bau *et al.* (2008) to produce a very good end result. Therefore the algorithm of Moeini and Afshar (2012) is used as a starting point from which to build alternative selection strategies for the layout optimization. An overview of their algorithm's selected strategy is included in Section 4.3.3.2.

4.3.1 Requirements of the Algorithm

An algorithm is to be developed capable of producing feasible layouts for a given base layout of a sewer network. The base layout includes all manholes, along with their ground elevations, and pipes required in the final design. Then, for a layout to be considered complete and feasible the algorithm is responsible for determining the flow direction of each pipe, i.e. the algorithm has to select for each pipe its up and downstream manholes. However, due to the constraint of Section 2.2.3.1, the final layout may not include any cycles. Consequently, the algorithm has to produce a spanning tree of the base layout. Since all pipes included in the base layout are required to form part of the final layout, a completion phase must be included. During which the pipes which are not contained in the spanning tree are reintroduced and their upstream manholes turned into adjacency nodes, as described in Section 2.2.2.

4.3.2 Ant Colony Optimization

For the convenience of the reader a brief overview of ACO is given. No attempt is made to provide a complete understanding of ACO Algorithms and the intricacies of the various mechanisms which propel the optimization. For a comprehensive understanding the reader is encouraged to the study the work of Marco Dorigo, the father of ACO (Dorigo, 1992; Dorigo *et al.*, 1996; Dorigo and Gambardella, 1997; Dorigo and Stützle, 2004).

ACO Algorithms have been successfully applied to various constrained optimization problems and achieve state-of-the-art results for several important problem classes, such as the Quadratic Assignment Problem (QAP), scheduling and routing (Dorigo and Stützle, 2004) and, most importantly for this application, the d-MST problem (Bui and Zrncic, 2006). The precursor algorithm to all ACO algorithms, Ant System (AS), was inspired by observing the pheromone-based trail-laying-trail-following of real ants (Dorigo *et al.*, 1996). Though modern ACO algorithms have come a long way from the initial AS model, the analogy of a colony of foraging ants is still useful in understanding the behaviour of the algorithm.

In ACO a number of individual ants each generates solutions independently and in parallel, over many iterations. The ants make decisions using a so-called 'pheromone-value', which models the fitness of an eligible decision at a decision point. The best solution in an iteration is used for trail-laying, i.e. the pheromone value is increased along the best trail, while some pheromone on all other trails are evaporated. Through this process of pheromone deposition and evaporation the search is intensified near good solutions over time, while initially maintaining diversity within the search space. The steps of a general ACO algorithm are as described by Dorigo *et al.* (1996)

1. Select a suitable size for the set of ants, $\mathbf{A}^k$, for each generation k and set initial pheromone values on all available selections to some suitable, but equal, value. Set generation count, $k = 0$

2. Starting from either a predetermined or randomly selected point, construct a solution for each individual ant, $a \in \mathbf{A}^k$ of the current generation, using the following transition rule to make a decision:

$$p_{ij}^k = \frac{\left[\tau_{ij}^k\right]^\alpha \left[\eta_{ij}\right]^\beta}{\sum_j \left(\left[\tau_{ij}^k\right]^\alpha \left[\eta_{ij}\right]^\beta\right)} \tag{4.1}$$

Where:

k = The generation number

p_{ij}^k = Probability of decision j at decision point i, hereafter "decision ij", in generation k

τ_{ij}^k = Pheromone value of decision ij in generation k

η_{ij} = Heuristic influence value at decision ij

α = Relative pheromone influence factor

β = Relative heuristic influence factor

3. Using a problem specific objective function, determine the fitness, $f(a)$, of each ant's solution, $a \in \mathbf{A}^k$.

4. Acquire the generation best solution, $f(\text{best})^k$. Compare acquired generation best solution to current global best solution, $f(\text{best})^{\text{global}}$, replacing the global best if the generation best solution is better.

5. Perform pheromone evaporation on all paths and increase the pheromone along the path selected by the generation best solution, using the following update rule.

$$\tau_{ij}^{k+1} = \tau_{ij}^k(1-\rho) + \triangle\tau_{ij} \tag{4.2}$$

with ρ the evaporation rate and $\triangle\tau_{ij}$ the pheromone increase of the generation best solution, defined as:

$$\triangle\,\tau_{ij} = \begin{cases} \dfrac{C}{f(\text{best})^k} & \text{if decision ij was made by the generation best solution} \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

where:

C = a constant real number

6. Check convergence of the algorithm. Usually convergence is accepted if a minimum number of generations has been completed and for a number of generations thereafter the global best solution has not improved. Alternatively, if all individuals of a generation produce the same solution the algorithm has converged.

If the algorithm has converged, accept current global best solution as final solution, otherwise return to step 2 and repeat the process.

4.3.2.1 Algorithm Modifications

Many modifications have been proposed in the literature to improve the behaviour of the ACO algorithm. The modifications used in this implementation are motivated and their effects are described:

- *Changing the Relative Heuristic Influence Factor.* Usually in an ACO implementation the values α, β and ρ are assigned fixed values throughout the entire run of the algorithm. Some authors have proposed varying these parameters to improve the behaviour of the algorithm. Merkle *et al.* (2002) proposed to vary the values of β. The relative heuristic influence factor, β, is varied to improve diversity of the search. Initially the heuristic values assist the first generations of ants to find good solutions but later on they might hinder the ants from following good pheromone trails and therefore steer the algorithm away from further improvement. This is especially true when static heuristic values are used, i.e., a heuristic where the values remain constant for the entire run of the algorithm. Therefore, they proposed to reduce the influence of the heuristic by decreasing the value of β from generation to generation until it becomes zero and only the pheromone values guide the decisions of the ants. The parameter is decreased linearly from a starting value β_0 to zero at a chosen point in the algorothm's life cycle. While this modification is not employed initially, as no heuristic influences are present, later in Chapter 6 it is introduced.

- *Changing the evaporation rate.* The evaporation rate ρ determines the convergence speed of the algorithm. In general, when a large search space is to be investigated a low value of ρ is beneficial since the algorithm will be allowed more time to explore the different regions of the search space before focusing on a small region. Merkle *et al.* (2002) found that, when the maximum number of iterations is restricted, a higher value of ρ usually performs better. Therefore they proposed that two different values of ρ be used during the run of an ACO algorithm. Initially, a low ρ is used which remains constant for the majority of the generations. For the last generations of the algorithm a high ρ value is used to perform a final intensive search near the best solution that has been found.

- *Modified elitist strategy.* Using an elitist strategy is a common modification to ACO algorithms. This entails using a pheromone update from both the generation best and current global best solution at the end of each generation. The pheromone update rule is modified to reflect this:

$$\tau_{ij}^{k+1} = \tau_{ij}^{k}(1-\rho) + \triangle\tau_{ij}^{k} + \triangle\tau_{ij}^{global} \tag{4.4}$$

  The elitist strategy has the advantage that the search is intensified around the current global best solution. However, if the global best solution remains unchanged for many generations it has a great influence on the pheromone values which may, during long runs, cause the algorithm to converge prematurely to the current global best solution. This is especially true if the current global best solution is a single good solution, with no other good solution in the neighbourhood. It is therefore proposed (Merkle *et al.*, 2002) to set a maximum number of generations, g_{max}, during which an elitist solution is allowed to remain unchanged. When the elitist solution has exceeded its maximum number of generations it is replaced by the current generation's best solution, even if this solution is worse than the current global best. The replacement is only in terms of pheromone updates, the solution is however retained as the current best solution of the optimization. When an elitist solution has good solutions in its neighbourhood it is likely the ants will discover it within reasonable time. Otherwise, it does not matter that the elitist solution has been discarded as no improved solutions are in its neighbourhood.

4.3.2.2 MMAS Algorithm

The Max-Min Ant System is a modified ACO algorithm derived from the AS algorithm (Stützle and Hoos, 2000). Investigations into improving the ACO algorithm's performance showed that improved exploitation of the best solutions found during the search was beneficial. However, using only the best solutions found during a search may aggravate premature stagnation of solution construction (Stützle and Hoos, 2000). The key, then, is to combine improved exploitation of the best solutions with an effective strategy to prevent early search stagnation (Stützle and Hoos, 2000). They developed the MMAS with specifically these characteristics in mind, varying it in three key aspects from the original AS:

- To exploit the best solutions found during an iteration or during the run of the algorithm, after each iteration only one single ant adds pheromone.
- To avoid stagnation of the search the range of possible pheromone trails on each solution component is limited to an interval $[\tau_{min}\,;\,\tau_{max}]$.

- Additionally, the initial pheromone trails are deliberately set to τ_{max}, achieving in this way a higher initial exploration of the search space.

Combination strategies to the pheromone updates have been proposed, in which dynamically varying updates from the iteration best solution to the best so far solution has been shown to improve performance of the algorithm (Stützle and Hoos, 2000). Further, they propose equations by which to limit the upper and lower pheromone bounds. These are:

$$\tau_{max} = \frac{1}{\rho\, f(best)} \tag{4.5}$$

$$\tau_{min} = \frac{\tau_{max}\left[1 - \sqrt[n]{p_{best}}\right]}{(avg - 1)\sqrt[n]{p_{best}}} \tag{4.6}$$

Where:

$f(best)$ = Fitness of best solution
n = Total number of decision points
p_{best} = The maximum probability of reproducing the best solution
avg = The average number of available decisions at each decision point

While the MMAS is not employed by any of the newly developed algorithms in this investigation, it was employed by Moeini and Afshar (2012) in their application and is reproduced here for comparison.

4.3.3 ACO of Sewer Networks

In this section the application of ACO to the sewer network problem is discussed in detail. As discussed previously, Bau *et al.* (2008) found that different selection strategies performed differently for the d-MST problem. Consequently, four alternative selection strategies are developed and compared in this investigation.

1. Edge-based selection which directly queries the base graph to construct a spanning-tree, similar to Moeini and Afshar (2012). Henceforth referred to as the "**direct-edge**" strategy.

2. Node-based selection which directly queries the base graph to construct a spanning-tree. Henceforth referred to as the "**direct-node**" strategy.

3. Constructing a spanning-tree using a permutation of unique edge identities. Henceforth referred to as the "**permutation-edge**" strategy.

4. Constructing a spanning-tree using a permutation of unique node identities. Henceforth referred to as the "**permutation-node**" strategy.

The layout creation procedure of each strategy is described in detail in Sections 4.3.3.2 through 4.3.3.5.For all the selection strategies above the hydraulic optimization is performed by the Heuristic Optimization Algorithm of Chapter 3. The hydraulic optimization algorithm is deployed for each individual layout created by the layout creation algorithm to determine the optimal set of hydraulic parameters. Once the layout creation and hydraulic design are complete the fitness of the solution may be calculated using Equation 2.6. In all cases no heuristic influence value, η_{ij}, is used since that would render direct comparison of the effectiveness of the strategies impossible. Figure 4.1 shows a small example network's base layout,

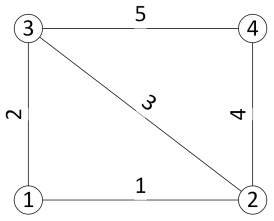


Figure 4.1: Example network

which will be used to describe the spanning-tree construction strategies. In this example node 1 is the outlet node. Starting from the outlet node, Figure 4.2 shows all the possible paths that can be followed to add the next node to the network. The nodes are shown in circles while the edges can be added after in the addition of the next node are shown in square brackets.
Figure 4.2, above, is not determined by using any of the selection strategies. It simply shows all the possible paths that can be followed to construct a spanning tree of the network. The selection strategies are employed to determine which decisions are eligible and decide how to present the eligible decisions at each point to the layout creation algorithm. For example, at the start the layout creation algorithm could be presented with either the eligible edges, 1, 2, or the eligible nodes, 2, 3. If edge 1, or node 2, was selected then the eligible set of edges and nodes at the following iteration are 2, 3, 4 and 3, 4 respectively.

4.3.3.1 Hop Rank Heuristic

In an ACO that uses a single pheromone matrix, the ants can only make one choice based on the pheromone. If another choice has to be made, some mechanism, usually a heuristic, is required to resolve it. For the sewer network layouts, a

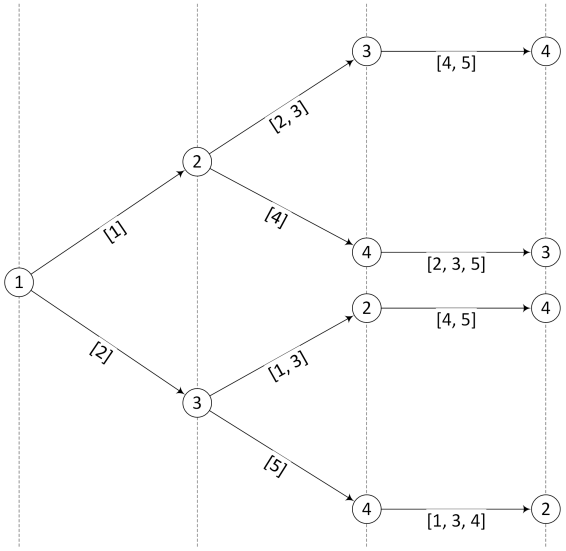


Figure 4.2: Example Network Iteration Selections

useful parameter proposed my Moeini and Afshar (2012) is the hop-rank. This parameter ranks nodes based on the number of preceding nodes in its branch. Referring to Figure 4.1, if a spanning tree consisting of edges 1, 3 and 5 is assumed, the hop-ranks of nodes 1 through 4 are respectively, 0, 1, 2 and 3. The hop-rank parameter can be used to favour selections which do not increase the length of already long branches. If required, the hop-rank parameter can be used to make a heuristic selection. In all procedures described below the use of this heuristic is stated clearly where relevant.

4.3.3.2 Direct-Edge Layout Creation

This strategy mimics the edge selection behaviour of the algorithm proposed by Moeini and Afshar (2012). The tree growing algorithm compiles a set of eligible edges, always starting from the static sink node. An edge is considered eligible for selection if only one of its vertices is already included in the growing spanning-tree. The edge selection procedure is shown below.

1. Define:

$$\begin{aligned} \mathbf{T} &:= \text{The spanning tree} \\ \mathbf{T}_N &:= \text{The set of nodes in the spanning tree} \\ \mathbf{E} &:= \text{Set of currently eligible edges} \\ n_t &:= \text{The target node of the new edge} \\ n_s &:= \text{The source node of the new edge} \end{aligned}$$

2. Initiate $\mathbf{T}$, $\mathbf{T}_N \leftarrow$ The sink node
3. Compile $\mathbf{E} = \{e \in \mathbf{E} \,|\, e \text{ has 1 node in } \mathbf{T}_N\}$
4. Select the next edge, e, of $\mathbf{T}$ from $\mathbf{E}$ using the transition rule described in Section 4.3.2
5. Identify n_t and n_s of e using $n_t \in \mathbf{T}_N$.
   Add n_s to $\mathbf{T}_N$.
6. Add e to $\mathbf{T}$
7. If $\mathbf{T}_N$ contains all nodes stop. Else return to 3.

Referring to Figure 4.2, if in the first iteration edge 1 was selected and in the second edge 2, then the set of eligible edges for iteration three would be $\mathbf{E} = \{4\,;\,5\}$. Edge 3 is not eligible since both its nodes are elements of $\mathbf{T}_N$ and it would introduce a cycle into the network. The way in which edge 3 will be added to the network is described in Section 4.3.3.6. Determining the source (upstream) and target (downstream) nodes of a selected edge is done simply by checking which node of the edge is already contained in $\mathbf{T}_N$ and assigning that as the target.

To model the pheromone matrix for the ACO algorithm it is assumed that at any iteration any edge could potentially be eligible. While this is, in reality, not the case, attempting to determine the possible eligible edges at each iteration is very complex and would offer very little benefit aside from reducing the size of the matrix. Assume $\mathbf{N}$ is the set of all nodes and $\mathbf{M}$ the set of all edges. Then it can be easily shown that the number of edges required to form a spanning tree of a graph is $|\mathbf{N}| - 1$. The pheromone matrix is then a $|\mathbf{N}| - 1 \times |\mathbf{M}|$ matrix, where the rows represent the iterations and the columns correspond to the pheromone value of an edge at the current iteration, i.e. the pheromone value of edge 4 at iteration 2 is stored in row 2, column 4 of the pheromone matrix.

4.3.3.3 Direct-Node Layout Creation

In this strategy the tree growing algorithm constructs sets of eligible source and target nodes. A node is considered to be an eligible source if any edge connected to it has a node which is already included in the growing spanning tree. This is best achieved by using the nodes already contained in the spanning tree as potential target nodes and finding their adjacency nodes, using the base graph, that can serve as potential source nodes. The direct-node strategy is formally described below.

1. Define:

$$\begin{aligned} \mathbf{N} &:= \text{Set of all nodes} \\ \mathbf{T} &:= \text{The spanning tree} \\ \mathbf{T}_N &:= \text{Set of nodes in the spanning tree} \\ e_i &:= \text{Eligible node, } i \\ \mathbf{E} &:= \text{Set of currently eligible nodes} \\ \mathbf{A}_i &:= \text{Set of nodes adjacent to node } i \\ n_i &:= \text{Node being added to the spanning tree at current iteration} \end{aligned}$$

2. Initiate $\mathbf{T}$, $\mathbf{T}_N \leftarrow \{\text{Sink node}\}$

3. Compile the set of eligible source nodes:
   $\mathbf{E} = \{e_i \in \mathbf{N} \backslash \mathbf{T}_N \,|\, \mathbf{A}_{e_i} \cap \mathbf{T}_N \neq \emptyset\}$.

4. Select the next node, n_i , of $\mathbf{T}$ from $\mathbf{E}$ using the transition rule described in Section 4.3.2.
   Add n_i to $\mathbf{T}_N$.

5. Find the eligible target nodes for the new edge:
   Compile $\mathbf{A}_{n_i}$. The set of eligible nodes for target selection is $\mathbf{E} = \mathbf{A}_{n_i} \cap \mathbf{T}_N$.

6. If $\mathbf{E}$ contains more than one element, select e_i with the lowest hop-rank as the target node. If nodes have equal hop-ranks, make a random choice between them. Alternatively take the single element of $\mathbf{E}$ as the target node.

7. Add a new edge from the source node to the target node to $\mathbf{T}$.

8. If $\mathbf{T}_N$ contains all nodes, stop. Else return to step 3

The direct node strategy places some limitations on the networks that can be produced by the algorithm. Referring to Figure 4.1, if only edges 2 and 5 have been included in the growing spanning tree, only node 2 is eligible as the next source node. The set of eligible target nodes of node 2 is $\mathbf{E} = \{1, 3, 4\}$. Due to the

hop-rank heuristic it would only ever be possible to select node 1 as the target node.

When modelling the pheromone matrix for the ACO algorithm it is assumed that at any iteration any node could potentially be eligible. As with the direct-edge strategy, this is not accurate. However, attempting to determine the possible eligible nodes at each iteration is very complex and would offer very little benefit aside from reducing the size of the matrix. Assume $\mathbf{N}$ is the set of all nodes. Since the procedure starts with the sink node already added to the spanning tree, the number of iterations required to add all other vertices is $|\mathbf{N}| - 1$. The pheromone matrix is then a $|\mathbf{N}| - 1 \times |\mathbf{N}|$ matrix, where the rows represent the iterations and the columns correspond to the pheromone value of a node at the current iteration, i.e. the pheromone value of node 4 at iteration 2 is stored in row 2, column 4 of the pheromone matrix.

4.3.3.4 Permutation-Edge Layout Creation

The permutation strategies are used as alternatives to the previous methods in which the base graph is queried directly. In this case an ant colony algorithm is used to construct a permutation of unique edge identities from which a spanning-tree of the base graph is eventually created. The steps that create the edge permutation are listed below.

1. Compile the set $\mathbf{N}$ of all base graph edges.
   Initialize permutation $\mathbf{P} := \emptyset$.

2. Compile set of eligible edges $\mathbf{E} = \mathbf{N} \backslash \mathbf{P}$.
   If $\mathbf{E} = \emptyset$ stop.

3. Using the transition rule described in Section 4.3.2, select the next edge $e_i \in \mathbf{E}$ to be added to the permutation. Add e_i to the end pf $\mathbf{P}$

4. Return to 2.

Once a permutation has been composed it can be used to construct a layout by simply using the order of the edges in the permutation as the order in which to add edges to a growing spanning tree. The permutation-edge layout construction, which is very similar to its direct-edge counterpart, is described below.

1. Define:
$$\begin{aligned} \mathbf{T} &:= \text{The spanning tree} \\ \mathbf{P} &:= \text{The permutation} \\ \mathbf{T}_N &:= \text{The set of nodes in the spanning tree} \\ \mathbf{E} &:= \text{Set of currently eligible edges} \\ n_t &:= \text{The target node of the new edge} \\ n_s &:= \text{The source node of the new edge} \end{aligned}$$
2. Initiate $\mathbf{T}$, $\mathbf{T}_N \leftarrow$ The sink node
3. Compile $\mathbf{E} = \{e \in \mathbf{E} \,|\, e \text{ has 1 node in } \mathbf{T}_N\}$
4. Iterate $\mathbf{P}$ to find the first $e \in \mathbf{P} \cap \mathbf{E}$.
   Edge e is the next edge in $\mathbf{T}$.
5. Identify n_t and n_s of e using $n_t \in \mathbf{T}_N e$
   $\mathbf{T}_N \leftarrow n_s$
6. Add e to $\mathbf{T}$.
7. If $\mathbf{T}_N$ contains all nodes stop. Else return to 3.

Assume $\mathbf{P} := \{12345\}$. Then, referring to Figure 4.1, after 2 iterations both edges 1 and 2 have been added and $\mathbf{E} = \{\,4\,;\,5\,\}$. Iterating through $\mathbf{P}$ encounters edge 4 prior to 5, so edge 4 is the next one to be added to the spanning tree.

The pheromone matrix of the ACO algorithm for the permutation cases operates on the permutation, rather than the network itself. Any edge can be placed at any point in the permutation, and all edges must be contained in the permutation. Assume $\mathbf{M}$ is the set of all edges. The pheromone matrix is then an $|\mathbf{M}| \times |\mathbf{M}|$ matrix, where the rows represent a point in the permutation and columns the pheromone value of an edge at that point, i.e. the pheromone value of edge 4 at position 2 in the permutation is stored in row 2 column 4 of the pheromone matrix.

4.3.3.5 Permutation-Node Layout Creation

In this case spanning tree layouts are constructed using permutations of unique node identities. An ant colony algorithm is used to create the node permutations as described below.

1. Compile the set $\mathbf{N}e$ of all base graph nodes.
   Initialize permutation $\mathbf{P} := \emptyset$.

2. Compile set of eligible nodes $\mathbf{E} = \mathbf{N} \backslash \mathbf{P}$. If $\mathbf{E} = \emptyset$ stop.

3. Using the transition rule described in Section 4.3.2, select the next node $e_i \in \mathbf{E}$ to be added to the permutation. Add e_i to the end of $\mathbf{P}$

4. Return to 2.

From the node permutation a spanning tree can be constructed by adding nodes to the spanning tree in the same order that they appear in the permutation, as described below.

1. Define:

$$\begin{array}{rcl} \mathbf{N} & := & \text{Set of all nodes} \\ \mathbf{T} & := & \text{The spanning tree} \\ \mathbf{P} & := & \text{The permutation} \\ \mathbf{T}_N & := & \text{Set of nodes in the spanning tree} \\ e_i & := & \text{Eligible node } i \\ \mathbf{E} & := & \text{Set of currently eligible nodes} \\ \mathbf{A}_i & := & \text{Set of nodes adjacent to node } i \\ n_i & := & \text{Node being added to the spanning} \\ & & \text{tree in the current iteration.} \end{array}$$

2. Initiate $\mathbf{T}$, $\mathbf{T}_N \leftarrow \{\text{Sink node}\}$

3. Compile the set of eligible source nodes, $\mathbf{E} = \{e_i \in \mathbf{N} \backslash \mathbf{T}_N \,|\, \mathbf{A}_{e_i} \cap \mathbf{T}_N \neq \emptyset\}$

4. Iterate through $\mathbf{P}$ to find the first $n_i \in \mathbf{P} \cap \mathbf{E}$. Node n_i is the next source node in the spanning tree. Add n_i to $\mathbf{T}_N$.

5. Determine the target node for the new edge. Compile $\mathbf{A}_{n_i}$. Compile eligible nodes for target selection $\mathbf{E} = \mathbf{A}_{n_i} \cap \mathbf{T}_N$

6. If $\mathbf{E}$ contains more than one element, iterate over $\mathbf{P}$. The first node n_j encountered in the iteration, such that $n_j \in \mathbf{E}$, is taken as the target for the new edge. Alternatively take the single element of $\mathbf{E}$ as the target node. Add n_j to $\mathbf{T}_N$.

7. Add a new edge from the source node n_i to the target node n_j to $\mathbf{T}$.

8. If $\mathbf{T}_N$ contains all nodes stop. Else return to step 3

Note that in the direct-node method, the hop-rank heuristic was used to select the target node, while in this case the permutation is used to choose the target node by selecting the first node encountered in $\mathbf{P}$ which is also in $\mathbf{E}$. This heuristic decision again places some restriction on the spanning trees that can be produced,

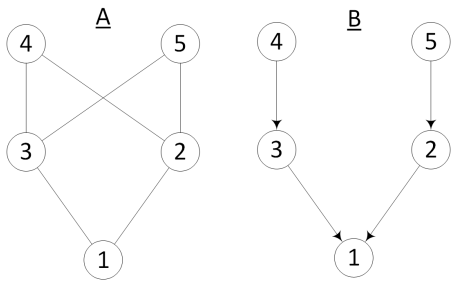


Figure 4.3: Node Permutation Restriction Example

as demonstrated using Figure 4.3. Figure 4.3A shows the base layout of a network, while Figure 4.3B shows a spanning tree of this network which cannot be created by the permutation-node approach. This is due to the fact that for node 4 to connect to node 3, rather than node 2, node 3 has to appear before node 2 in the permutation. Then, however, node 5 will also connect to node 3. The opposite is also true: if node 2 appeared before node 3 in the permutation then both nodes 4 and 5 will connect to node 2.

As with the permutation-edge, the ACO algorithm's pheromones operates on the permutation rather than on the network itself. Any node can be placed at any point in the permutation, and all nodes must be contained within the permutation. Assume $\mathbf{N}$ is the set of all nodes. The pheromone matrix is then an $|\mathbf{N}| \times |\mathbf{N}|$ matrix, where the rows represent a position in the permutation and columns the pheromone value of a node at that position, i.e. the pheromone value of node 4 at position 2 in the permutation is stored in row 2 column 4 of the pheromone matrix.

4.3.3.6 Layout Completion

Once a spanning tree has been created, all edges of the base graph that are not included in the spanning tree have to be added to complete the network. These edges have to be reintroduced in such a way that cycles are not formed. The adjacency node technique described in Section 2.2.2 is used to avoid cycle formation. The source and target node selection of an edge is performed using the hop-rank heuristic, choosing the target node as the one with the lowest hop-rank. If the hop-ranks of the nodes are equal, the direction of the edge is determined randomly. This technique is used for all the strategies investigated in this paper.

4.3.3.7 Dynamic Outlet Placement

In all the algorithms described above the outlet node is static. However, the algorithms can be modified to allow for dynamic outlet node placement with relative ease. The required modifications are summarized below:

- *Direct-edge*: Modify the spanning tree construction algorithm to have the ants initially select any edge and assume its end with the lowest ground elevation is the sink node.
- *Direct-node*: Add an additional initial decision for the ants where a sink node has to be selected from the list of all nodes.
- *Permutation-edge*: Similar to the direct-edge case, modify the spanning tree construction algorithm to use the first edge in the permutation as the starting edge, again using its lowest end as the sink node.
- *Permutation-node*: Use the first node in the permutation as the sink node.

4.4 Summary and Conclusions

In this chapter the layout optimization problem of sewer networks was reviewed. It was motivated that the problem is a variant of the d-MST problem, a well known NP-Hard problem in graph theory. Both classic algorithms, mainly DP implementations, and more modern algorithm, mainly metaheuristics, which solve the layout optimization problem were discussed. Drawing on the findings of researchers such as Bui and Zrncic (2006) and Bau *et al.* (2008) it was decided to employ ACO algorithms for layout optimization in this implementation. Moeini and Afshar (2012) developed an ACO algorithm which used a TG algorithm to determine how the ACO algorithm was able to construct the layouts. This proved to be an effective strategy and was selected as a starting point for the algorithms of this investigation.

For the convenience of the reader the steps of ACO, as well as the MMAS variant, were presented and modifications to the algorithm used in this investigation were introduced and their use motivated.

Four different selection strategies which rely on both node and edge information were developed and combined with an ACO algorithm to govern how the ants were able to construct layouts. Of the four strategies, two relied on querying the base layout directly. One queries and uses edge information, referred to a the direct-edge strategy, the other queries and uses node information, referred to as

the direct-node strategy. The other two algorithms used the ACO algorithm to construct a permutation of edge and node identities, respectively. Then, using the permutation of identities, a layout is constructed from the base layout. This chapter did not include any form of algorithm evaluation, as they have not yet been combined with a hydraulic design algorithm. In the following chapter the four layout strategies are combined with the HOMS algorithm to form simultaneous layout and design algorithms and applied to example problems to evaluate their effectiveness.

Chapter 5

Sewer Network Optimization

In this chapter the HOMS algorithm of Chapter 3 is combined with the four proposed layout optimization strategies from Chapter 4. The resulting simultaneous layout and hydraulic optimization algorithms are applied to three example problems and their performance compared to the original ACO-TGA of Moeini and Afshar (2012). Furthermore, the effect of fitness warping, introduced in Section 2.4, is demonstrated and discussed.

5.1 Simultaneous Layout and Hydraulic Optimization

In Chapters 3 and 4, the HOMS algorithm and multiple ACO algorithms with varying selection strategies have been proposed, respectively. In this section, the algorithms are combined to create a robust simultaneous sewer network optimization algorithm. The resulting algorithm is discussed and some concepts summarized for the reader's convenience. Figure 5.1 shows a simplified flow chart of the combined algorithm's steps, below follows a brief summary and discussion of each phase.

5.1.1 Defining the Problem

Prior to initialization of the optimization algorithm the design of the network's base layout, or, as described in Section 2.2.1, spatial design, has to be completed. Spatial design design of the network is the concern of the design engineer and it should incorporate external design factors, such as existing infrastructure, the optimization algorithm is unaware of. Additionally, the base layout has to include the ground elevation at each manhole as well as the expected inflow rates due to service connections defined as a 24 hour hydrograph at each manhole.

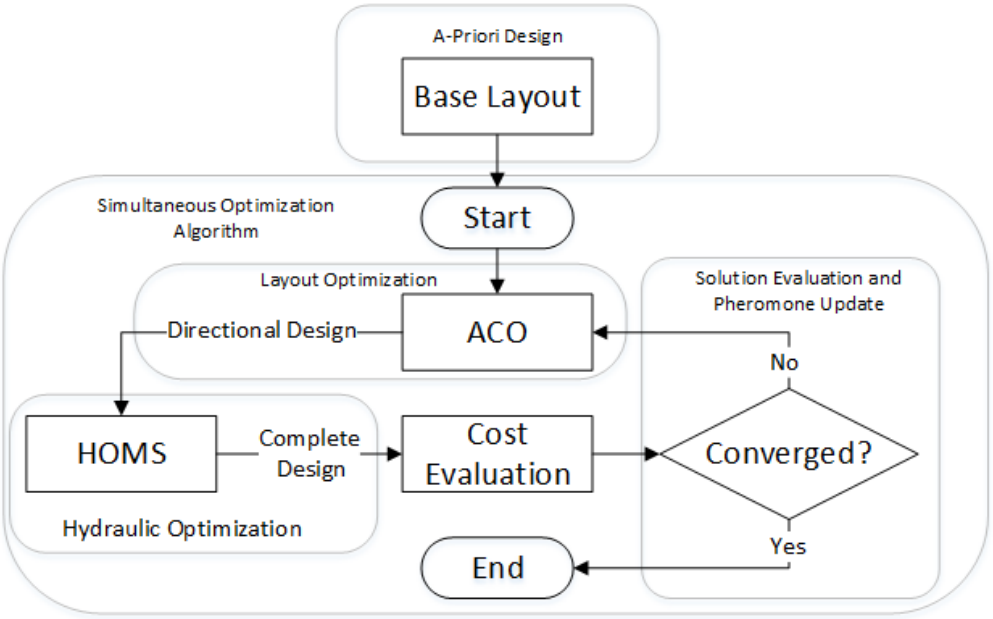


Figure 5.1: Simultaneous Optimization Model

Before initialization the design engineer has to define the constraint boundaries for each constraint of Sections 2.2.3.1 and 2.3.2. These constraints should take into account any and all design relevant design guidelines as well as any considerations unique to the proposed construction site of the network, such as adverse soil conditions making deep excavation difficult. In addition to the design constraints, the ACO algorithms responsible for layout creation as well as the HOMS algorithm have many required parameters, such the evaporation rate, ρ, and beneficial slope change factor, γ_b, which must all be selected before the optimization can commence.

Once all these factors have been defined the problem can be optimized using the algorithms proposed in this dissertation.

5.1.2 Layout Optimization

The layout optimization is carried out by the ACO algorithm presented in Chapter 4 with any one of the proposed layout creation strategies active. The ACO completes the directional design, discussed in Section 2.2.2, of thousands of individual layouts, learning and improving after each. Each of these now complete layouts are sent to the HOMS algorithm to have its hydraulic parameters determined.

The ACO algorithm is responsible for convergence checking. Many strategies exist for accepting algorithm convergence and are readily available in ACO sources, all of which could be implemented. However in this implementation is was decided that the algorithm would only be terminated once the specified maximum number of generations has been reached.

5.1.3 Hydraulic Optimization

The hydraulic optimization is carried out by the HOMS algorithm from Chapter 3 for each individual layout it receives from the ACO algorithm. It determines all required hydraulic characteristics of the network elements, such as pipe diameters, slope and flow rates and manhole invert levels. The HOMS algorithm then sends its entire solution with complete layout and hydraulic parameters to the cost analysis package for evaluation. Notice that the HOMS algorithm relies purely on its heuristic hypothesis to complete the design, the cost evaluation only takes place once the entire hydraulic design has been completed.

5.1.4 Cost Evaluation

Once both optimization processes has completed the entire solution's cost must be determined. Using Equation 2.1 the network's cost is determined. The cost evaluation algorithm, in addition to determining network element costs, takes all network constraints into considerations. Referring to Equation 2.6, any violations in constraints are added to their respective violation value functions, g_i, i.e. if the maximum allowable velocity constraint is exceeded by $0.2m/s$ then $g_{v_{max}}$ is incremented by 0.2. However, the layout creation algorithms and HOMS algorithm ensures adherence to all constraints other than the maximum cover depth constraints. Consequently, it is the only constraint where violations are expected. However, as a measure of safety the cost analysis package checks all constraints as well. The now complete solution is sent back to the ACO algorithm for evaluation.

5.1.5 ACO Solution Evaluation and Pheromone Update

The ACO algorithm identifies the generation best solution, i.e. the least cost solution of the generation. This solution is then compared to the current global best solution, if it is a better solution, i.e. the cost is lower, it replaces the current global best solution. However, if the current global best solution has remained for more than a predefined number of generations, the current generation's best solution replaces it as the global best solution regardless of associated costs. However, the previous global best solution is not discarded as it may remain the overall best solution found by the algorithm, it simply no longer contributes to pheromone

depositions. The overall best solution is updated if and only if a solution with a lower cost than its own has been found.

Then, the algorithm's current generation is checked to see if the maximum number of generations has been reached or any other convergence criteria have been met. If it has, the algorithm terminates and the overall best solution is presented as the final solution. If not, the algorithm updates the pheromones using the modified elitist strategy, discussed in Section 4.3.2.1. The ACO now starts its next generation and the process is repeated until either the maximum number of generations has been reached or convergence criteria have been met.

5.2 Example Problems and Results

Three example networks with varying size and topology characteristics were created to compare the effectiveness of the proposed layout optimization strategies of Section 4.3.3. The ACO-TG algorithm of Moeini and Afshar (2012) is used to provide a benchmark for each example network. It is the intention of this study to investigate the effects different topology and network sizes have on the performance of different layout creation strategies, which the examples provided in Moeini and Afshar (2012) do not allow.

Algorithm parameters for the ACO-TG algorithm are as used by Moeini and Afshar (2012) for examples of a similar size. The proposed ACO algorithms were calibrated using 500, 1000, 2500, 5000 and 10000 for potential generation limits with population sizes of 10, 20, 50 and 100 and the values which resulted in the best average fitness selected. Evaporation rates were chosen which resulted in a gradual convergence of the optimization, avoiding rapid convergence to local optima. The proposed example networks are all grids of varying size and topological characteristics. The sink node is static and marked with a dark fill in Figures 5.2, 5.6 and 5.9. Relevant elevations are shown and all slopes are assumed to be linear. In all three cases inflow hydrographs are defined at each manhole as if serving 250 very high income residential units, refer to Appendix A. The peak value is used to scale the unit values listed in the table and leakage values are added to provide a base flow rate.
In all cases the evaporation rate, ρ, is changed after 80% of the generation limit is reached. The current best solution is allowed to persist for a maximum of 25% of the generation limit, after which it is no longer eligible for pheromone deposit and instead the current generation's best solution, regardless of its fitness, is used for pheromone deposit. The initial pheromone is always taken as 5.0. Computations were performed using the University of Stellenbosch's Rhasatsha HPC: http://www.sun.ac.za/hpc. The results are averaged over 20 randomly initialized

runs for each case. The HOMS algorithm of Chapter 3 is seeded with the following constraint values: Minimum allowed cover depth $E_{min} = 1.2m$. Maximum allowed cover depth $E_{max} = 10m$. Minimum allowed slope $S_{min} = 0.01$. Minimum allowed velocity $v_{min} = 0.7m/s$. Maximum allowed velocity $v_{max} = 5.0m/s$. Minimum required spare capacity $SC_{min} = 30\%$. The set of commercially available pipe diameters $\{\mathbf{D}\}$ = {150mm, 200mm, 250mm, 315mm, 355mm, 400mm, 450mm, 525mm, 600mm, 675mm, 750mm, 825mm, 900mm, 1050mm, 1200mm, 1350mm, 1500mm, 1650mm, 1800mm}. PVC is used for pipes with diameters below 450mm, with a Manning coefficient $n = 0.009$. For all other pipes concrete is used with a Manning coefficient $n = 0.02$. The value of γ_b, described in Section 3.3.6, was calibrated beforehand, and α, in Equation 2.6, is taken as $1e8$. The layout of the eventual best solution obtained for each example problem is listed. Flow directions of the pipes are shown with arrows and adjacency nodes are indicated by perpendicular strokes at the end of a pipe. The pipe numbers are identifiers assigned during the optimization. Elevation contours are shown for each solution.

5.2.1 Example Network 1

The first example network, shown in Figure 5.2, is a small network on a flat surface.

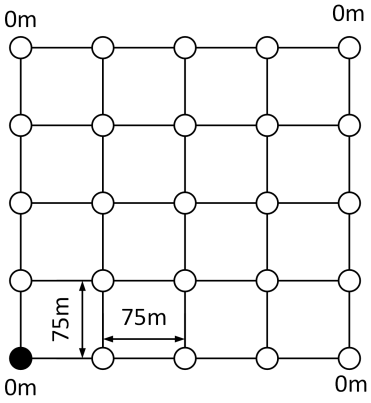


Figure 5.2: Example Network 1

The main purpose of this example problem is to demonstrate that all algorithms are performing correctly and comparably when the space for heuristic influences is minimal. Table 5.1 shows the algorithm parameters used during analysis and averaged results for this network. Figure 5.3 shows the fitness progress with function evaluations of the best result produced by each of the 5 algorithms. The node strategies require more computation time than their edge counter parts, as is expected due to the additional target-node decision that they have to make. The ACO-TGA has the slowest computation time of all, if only slightly worse than the node algorithms for such a small problem.

On average the algorithms perform very similarly, as can be expected for such a small problem and simple topography. The node strategies and ACO-TGA are less consistent in their final results. The permutation edge approach had the worst eventual best solution of all the algorithms. While the ACO-TGA did find the overall best solution, it is only 0.05% better than its nearest competitor. It also yielded the worst overall final solution.

Table 5.1: Example Network 1 Parameters and Results

Algorithm Parameters						
Algorithm	Generation Limit	Population Size	$\rho_{initial}$	ρ_{final}	C	γ
ACO – TGA	1000	100	0.05	-	1000	-
Dir – Node	1000	20	0.01	0.02	25	0.15
Dir – Edge	1000	20	0.0125	0.025	25	0.35
Perm – Node	1000	20	0.01	0.02	25	0.35
Perm – Edge	1000	20	0.0125	0.025	100	0.15

Algorithm Results						
Algorithm	Average Final Cost	Standard Deviation	Best Final Cost	Worst Final Cost	Average Comp. Time(s)	Average Infeasible Sol (%)
ACO – TGA	18 880.67	569.86	18 662.45	19 362.00	-	0.32
Dir – Node	18 829.87	172.44	18 758.40	18 955.44	18	0
Dir – Edge	18 707.55	74.23	18 673.90	18 756.22	11	0
Perm – Node	18 715.46	115.83	18 695.22	18 810.33	16	0
Perm – Edge	18 842.32	70.91	18 835.59	18 937.63	12	0

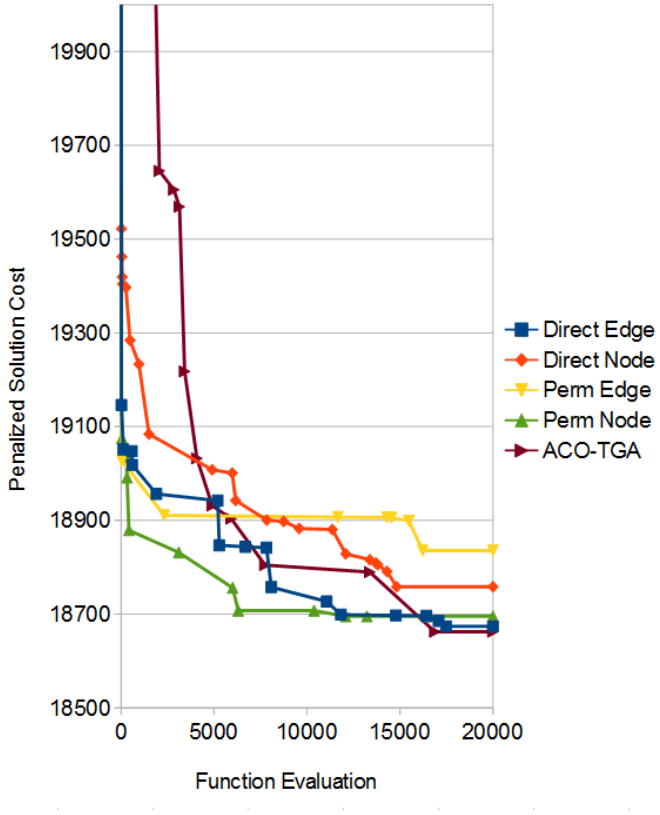


Figure 5.3: Example 1: Fitness progression of the best solution

Even for such a small example fitness warping is present in the ACO-TGA as on average 0.32% of its solutions were infeasible designs due to poor diameter selections. Figure 5.4 shows the best solution of the ACO-TGA for Example 1.

To further demonstrate the effect of fitness warping ten trial solutions from the early stages of the ACO-TGA were randomly selected from Example Problem 1. The HOMS algorithm was then used to perform the optimization of the layouts of these ten trial solutions. Figure 5.5 shows the fitness of the ten solutions, on the left obtained by the HOMS algorithm and on the right the fitness obtained by the ACO-TGA. The severity of the fitness warping is observed.A trial solution of the ACO-TGA with a cost of 71 303 024 produce a cost of 31 260 when the HOMS algorithm of Chapter 3 is used to perform the hydraulic optimization of its layout. This extreme warping of the layout's fitness due to poor diameter selections is seen in all 10 cases.

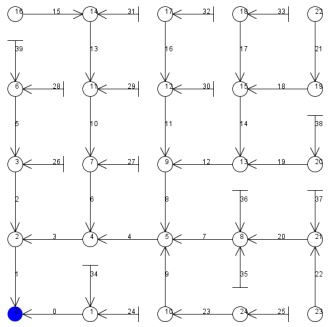


Figure 5.4: ACO-TGA Solution of Example 1

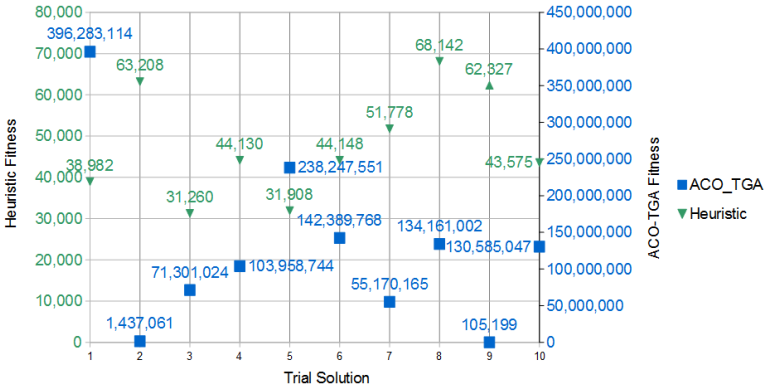


Figure 5.5: Effect of Fitness Warping

5.2.2 Example Network 2

The second example, shown in Figure 5.6, is a medium size network.

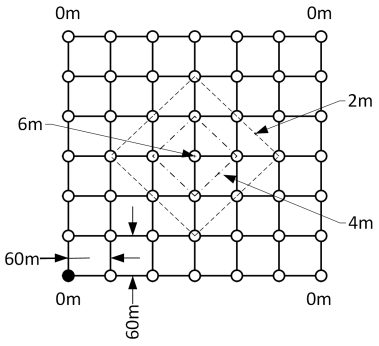


Figure 5.6: Example Network 2

The elevation topology mimics a hill with a flat surrounding area. The algorithm parameters during analysis and averaged results for example problem 2 are shown in Table 5.2. The ACO-TGA performs worse on average, and in its best run, than the other algorithms while having 4 times as many function evaluations. It is also far less consistent.

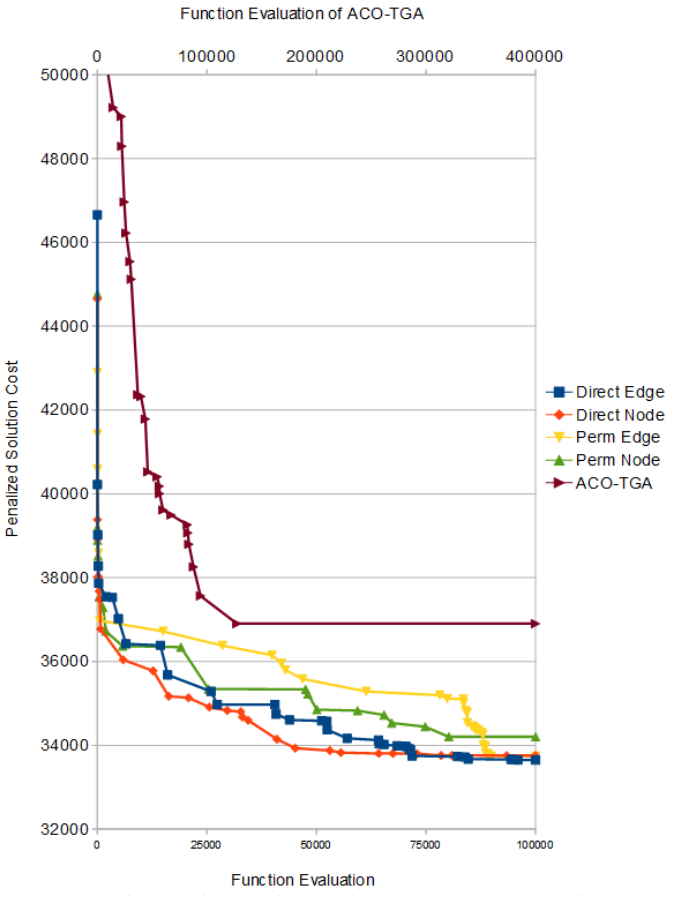


Figure 5.7: Example 2: Fitness progression of the best solution

Table 5.2: Example Network 2 Parameters and Results

Algorithm Parameters						
Algorithm	Generation Limit	Population Size	$\rho_{initial}$	ρ_{final}	C	γ
ACO – TGA	2000	200	0.05	-	1000	-
Dir – Node	5000	20	0.003	0.01	25	0.0
Dir – Edge	5000	20	0.003	0.02	25	0.1
Perm – Node	5000	20	0.003	0.02	25	0.05
Perm – Edge	5000	20	0.003	0.01	25	0.15

Algorithm Results						
Algorithm	Average Final Cost	Standard Deviation	Best Final Cost	Worst Final Cost	Average Comp. Time	Average Infeasible Sol (%)
ACO – TGA	39 054.99	4 009.44	36 902.78	42 730.31	-	0.07
Dir – Node	34 033.83	679.15	33 756.53	34 432.89	3m 34s	0
Dir – Edge	33 853.14	347.12	33 648.58	34 157.08	2m 26s	0
Perm – Node	34 739.67	800.67	34 203.41	35 130.76	3m 7s	0
Perm – Edge	34 106.28	611.90	33 738.79	34 504.63	2m 32s	0

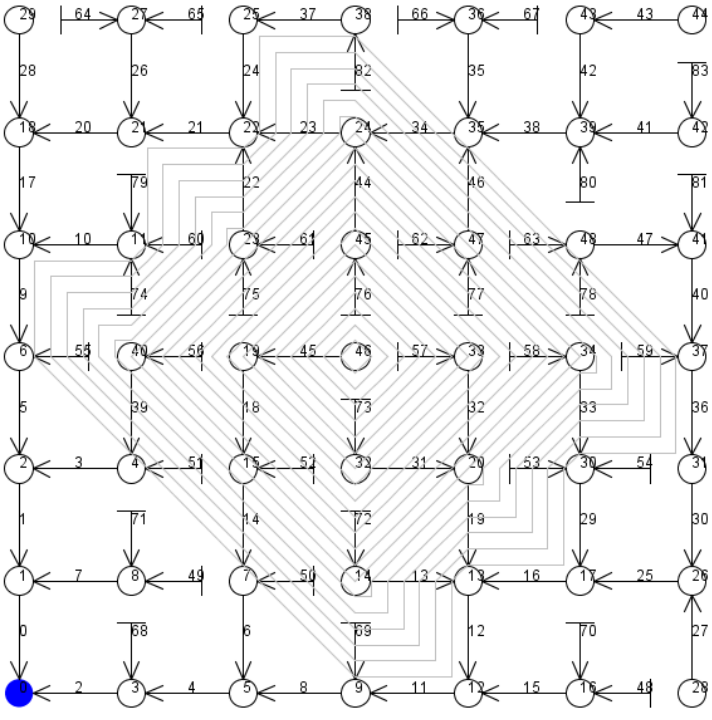


Figure 5.8: Direct-Edge Solution of Example 2

The ACO-TGA is again the only algorithm to produce infeasible solutions. On average the Direct-Edge strategy performs the best, while the three other newly proposed algorithms have comparable average performance. The overall best solution was found by the direct edge strategy, while the best solution obtained by the ACO-TGA is worse than the worst solution obtained by any of the algorithms. Again, the ACO-TGA requires the most computation and the node strategies are slightly slower than their edge counter parts. The progress of the best solutions is shown in Figure 5.7. The secondary X-Axis shows the function evaluations for the ACO-TGA.

The effect of fitness warping can be observed on the ACO-TGA as for approximately the first 50 000 function evaluations its current solution is worse than the solutions obtained by the other algorithms within 100 function evaluations. The convergence of the ACO-TGA stagnates in this example. This is similar to the results presented by Moeini and Afshar (2012).

5.2.3 Example Network 3

The final example, shown in Figure 5.9, is a large network with a gradual 1% slope. Table 5.3 shows the algorithm parameters and results for this example.

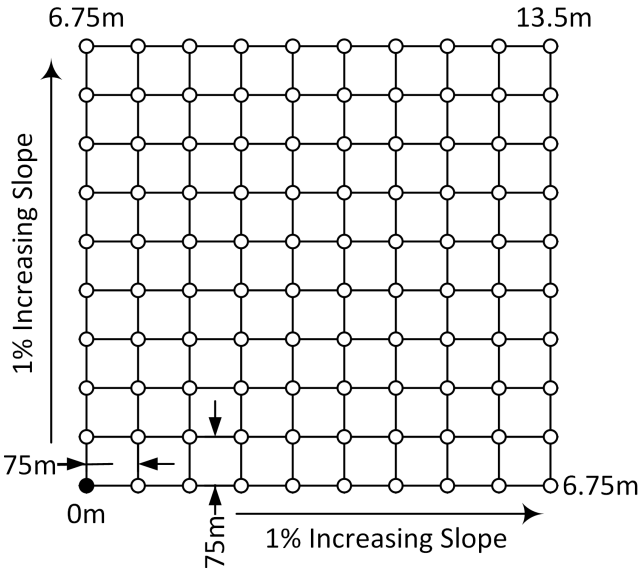


Figure 5.9: Example Network 3

The Perm-Edge algorithm performs the best of all algorithms. It obtained the best overall solution, has the best average final solution cost and its worst solution obtained across the 20 runs is better than the best solution obtained by algorithms other than the Direct-Node. Fitness warping is again observed in the ACO-TGA during its early trials. The ACO-TGA performs worse than the other algorithms on average and it is the most inconsistent. The computation time of the ACO-TGA is again the highest, while the other algorithms have comparable computation times. The rapid improvement in the ACO-TGA whereafter it plateaus is consistent with the results of Moeini and Afshar (2012)).

Table 5.3: Example Network 3 Parameters and Results

Algorithm Parameters						
Algorithm	Generation Limit	Population Size	$\rho_{initial}$	ρ_{final}	C	γ
$ACO-TGA$	2000	200	0.05	-	1000	-
$Dir-Node$	5000	50	0.004	0.01	100	0.3
$Dir-Edge$	5000	50	0.005	0.02	100	0.25
$Perm-Node$	5000	50	0.003	0.01	100	0.1
$Perm-Edge$	5000	50	0.004	0.02	100	0.35

Algorithm Results						
Algorithm	Average Final Cost	Standard Deviation	Best Final Cost	Worst Final Cost	Average Comp. Time	Average Infeasible Sol (%)
$ACO-TGA$	110 660.85	69 775.71	86 981.20	180 721.21	-	1.576
$Dir-Node$	81 998.65	2 849.34	80 819.48	84 292.63	10m 23s	0.002
$Dir-Edge$	85 418.28	3 064.40	83 365.34	88 133.87	9m 45s	0.011
$Perm-Node$	87 047.27	9 560.85	83 828.32	96 908.87	11m 8s	0.002
$Perm-Edge$	80 690.40	3 914.31	78 349.31	82 616.73	11m 11s	0.031

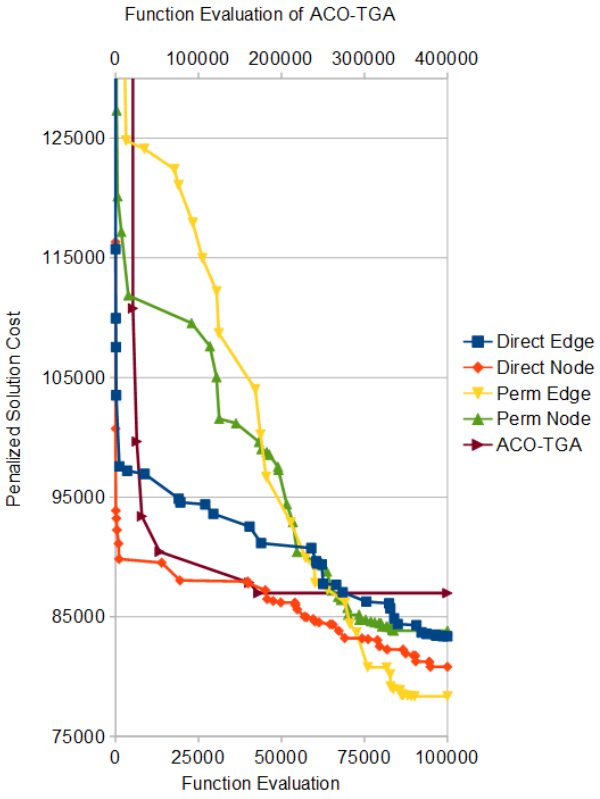


Figure 5.10: Example 3: Fitness progression of the best solution

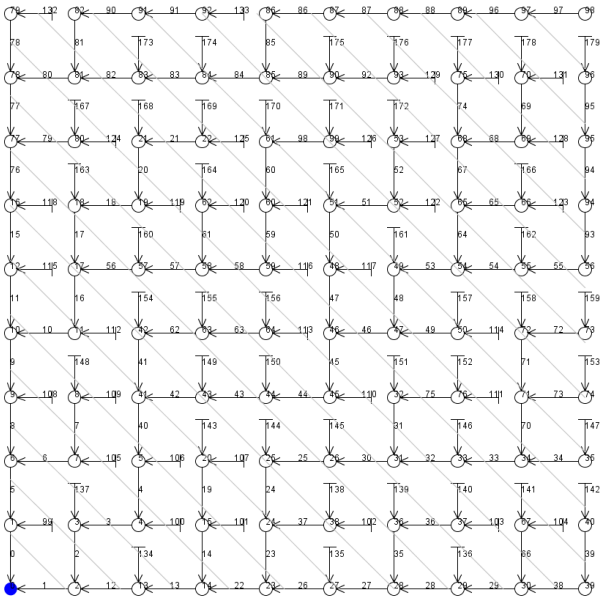


Figure 5.11: Perm-Edge Solution of Example 3

5.3 Summary and Conclusion

A sewer network optimization model was created using the layout optimization techniques described in Chapter 4 with the HOMS algorithm described in Chapter 4. This supports the use of any of the four layout technique

In order to benchmark the performance of the proposed layout algorithms, the ACO-TG algorithm of Moeini and Afshar (2012) was reproduced. The four newly developed algorithms, a long with the MMAS of Moeini and Afshar (2012) were used to solve each of the three example problems. Three example problems were analysed and the results of the various techniques compared.The ACO-TGA was able to produce the best answer for the smallest example problem, while performing worse than the other algorithms for the two more complex examples. The ACO-TGA performed, on average, the worst of all the algorithms. The proposed layout algorithms performed differently for the examples that were investigated. For example, the permutation-edge strategy performed the worst of all for the first example, on par with the others for the second example, and the best for the third example. Similar variation in performance is observed for the other algorithms and no clear winner emerged. The edge-strategies, while not always the best, consistently produced results at least comparable to the best strategy for all the examples making them an attractive option. However, most importantly, the effect of minor changes to the selection strategies in the final result are observed. This can only be ascribed to the heuristic differences in the selection algorithms and makes a strong case for further investigation.

In the case of the ACO-TGA both the hydraulic design and layout design are improving with each iteration leading to the observed reduction in cost. However, the reduction in cost seen during execution of the newly proposed algorithms is due purely to improvement in the layout. The HOMS algorithm produces a static, near optimal, result for each layout of the ACO, i.e. repeating the HOMS algorithm multiple times will not change the final cost of a layout. From this it is readily concluded that layout optimization leads to a significant reduction in capital investment cost. Lejano (2006) states that the majority of optimization algorithms usually operate only on the hydraulics of the problem due to the complexity of simultaneous algorithms. It is recommended that in future this approach, in so far as possible, be avoided as the importance of layout optimization is demonstrated in the results in this chapter.

The fitness warping phenomenon, introduced in Chapter 2, was seen to effect the ACO-TGA of Moeini and Afshar (2012). It was observed that for all problems the ACO-TGA produced very poor results early on, in most cases a significant portion of which were infeasible solutions. The proposed hybrid algorithms did not suffer from this drawback, exhibiting only expected behaviour of the algorithm very early on where layouts are created entirely at random. Furthermore, the severity of the fitness warping was demonstrated by selecting 10 early solutions of the ACO-TGA and using the HOMS algorithm to optimize the hydraulics of each solution. The difference in calculated capital costs were substantial, often improving the network's final cost by multiple orders of magnitude. It was recommended that in future investigations, where possible, fitness warping be avoided as it is certainly an undesired characteristic for an optimization algorithm.

Chapter 6

Standard Problem Library

In Chapter 5 it was observed that the different selection strategies' performance was influenced by the problem under consideration. In this chapter a library of sewer network optimization problems, with controlled characteristics, is developed to facilitate further investigation of the varied algorithm performance. Various parameters are proposed by which to control the characteristics of sewer network optimization problems. A library is created by combining different values for each parameter and generating multiple problem instances within each combination.

6.1 Introduction

Benchmark problems by which to compare the performance of algorithms is standard practice in optimization research. New algorithms are applied to problems which have previously been solved by multiple researchers in order to compare their effectiveness. Consequently in many fields of optimization research standardized problem libraries are used. Examples are the Resource-Constrained Project Scheduling Problem library called PSPLIB (Kolisch *et al.*, 1992), SNDlib (Orlowski *et al.*, 2009) and SimOpt (Pasupathy and Henderson, 2011). Libraries comprise multiple optimization problem instances within a specific field of research.

The software packages mentioned above are capable of generating problem instances with varied sizes and properties. Taking a lead from this approach, software was developed to generate sewer network base layouts with controlled characteristics and terrain topographies. The aim was not to provide a definitive library of sewer network optimization instances, since that would fall outside the scope of this dissertation. Instead, the aim was to generate a large enough number of networks with different properties to be able to study the nature of optimality in sewer network design.

After completion of this investigation several existing software packages from the literate capable of generating water distribution system and sewer network problem instances with controlled characteristics were brought to the author's attention. Sitzenfrei *et al.* (2010*a*) produced a complex cellular automata algorithm capable of producing virtual case studies with controllable and varied characteristics, named VIBe. Sitzenfrei *et al.* (2010*b*) used VIBe to produce 75 000 water distribution problem instances and applied a simple pipe sizing algorithm to each as an example. VIBe offers a platform which can be specialised to create problem instances for a variety of urban network optimization problems. Trifunovic *et al.* (2013) developed a network generation tool developed on the principles of graph theory which connects a pre-defined set of nodes, referred to as a seed of junctions. Their tool produces multiple networks from the same seed of junctions, resulting in layouts which vary only in edge connections. This approach readily produces a set of problems with very similar characteristics. These examples all generate water distribution systems, though the techniques used to generate network layouts can be easily applied to sewer networks.

Moderl *et al.* (2008) developed a Matlab-tool which generates a virtual sewer network problem instances in a stochastic manner with some controllable characteristics, such as pipe length, catchment slope, etc. Their package produces branched network layouts. Due to the branched nature of their networks the problems generated are only suitable for hydraulic optimization. Their techniques could, with relative ease, be expanded to introduce cycles and allow layout optimization. Blumensaat *et al.* (2012) created a novel approach to develop a hydraulic sewer model constrained by a minimum amount of data. Using a surface flow accumulation algorithm combined with a digital elevation model the algorithm generates close-to-reality sewer networks ready to be implemented in a hydraulic analysis or optimization package. The networks produced by their technique are demonstrated to closely match real world data, making this an attractive option.

6.2 Network Characteristics

The aim of the software model is to generate networks with enough information to apply optimization algorithms. This requires that a base layout graph be produced, ground elevations are assigned at manholes and flow rates are distributed throughout the network. Candidate networks have to adhere to certain practical constraints to ensure they mimic realistic engineering problems. Towards that the following requirements must be met:

1. Manholes and pipes should be placed irregularly, but not so irregularly as to produce networks which do not closely correlate to real networks.
2. The base layout graph may not have overlapping pipes or manholes, i.e. a planar graph has to be produced.
3. The manhole elevations should follow a relatively smooth curve, while including some irregularity to better match real terrain.
4. Flow rates should originate throughout the network to accurately model house and commercial connections.

In addition to these constraints some degrees of freedom are introduced to allow problems with varying characteristics to be generated. These are:

1. Number of Manholes, m: The number of manholes can be any positive integer value. The pipe count varies with this value as the base-graph building algorithm connects the manholes.
2. Average Pipe Length, l_a: a that value dictates the average length of pipes in the network.
3. Pipe Length Standard Deviation, l_{sd}: This value introduces irregularity in the pipe lengths in a consistent manner.
4. Sink Manhole Percentile, p: This value is used to determine where to place the sink, or outlet, manhole.
5. Average Slope, s: The average slope of the terrain across the problem domain.
6. Terrain Variance, t: A percentage value used to introduce irregularity into the terrain generation.
7. Total Cumulative Flow Rate: This is the total cumulative flow rate which will leave the network at the sink manhole.

6.3 Network Generation Algorithm

An algorithm that takes all of the constraints and degrees of freedom into consideration when generating problem instances is described below. Random numbers are often used in this implementation. R is used to indicate a random number with distribution $[0.0\,;\,1.0)$. The process is started by generating a symmetrical, uniformly distributed grid of vertices l_a apart. More vertices than the required

number of manholes, m, are created to allow greater variation in the final base-graph. Once this grid has been created each vertex is shifted in a random direction around its origin, as shown below: Referring to Figure 6.1:

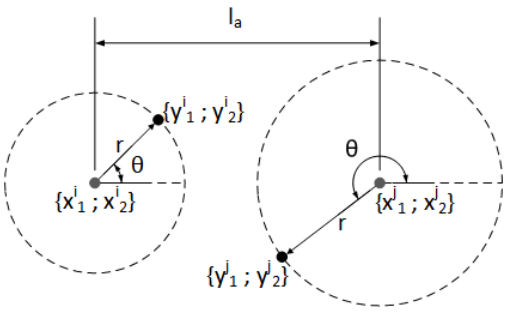


Figure 6.1: Irregular Manhole Shifting

$$\begin{aligned} \theta &= R2\pi \\ r &= \frac{l_{sd}}{2} + Rl_{sd} \\ y_1 &= x_1 + r\cos\theta \\ y_2 &= x_2 + r\sin\theta \end{aligned}$$

This shift is repeated for all vertices to produce an irregular grid. Edges are then introduced to the base graph. Referring to constraint 2, the resulting graph must be planar. It is common practice when creating planar graphs to use Delaunay Triangulation to introduce edges. In this implementation the Voronoi Diagram (de Berg *et al.*, 1997) is used to determine the Delaunay Triangulation (de Berg *et al.*, 1997) of the vertex grid. The resulting graph is planar and used as a starting point from which to create the sewer network's base-graph. Similar to the layout creation algorithm used in the ACO implementations of Chapter 4, a tree growing algorithm, starting from a random vertex, is used to produce an undirected base-graph for the sewer network. Figure 6.2 shows two typical base graphs produced with the same input values, $m = 50$, $l_a = 100m$ and $l_{sd} = 30m$. The resulting graphs have pipes with varying lengths and an irregular distribution of manholes, closely correlating to practical engineering models.

Introducing ground elevation levels over the terrain presents some unique challenges. The main concerns are controlling the slope over the domain and ensuring the terrain has predictable and consistent shape. To overcome these challenges, methods of interpolation in two dimensions (2D) are used to provide topography functions. In 2D interpolation, domains are normalized and shape functions are

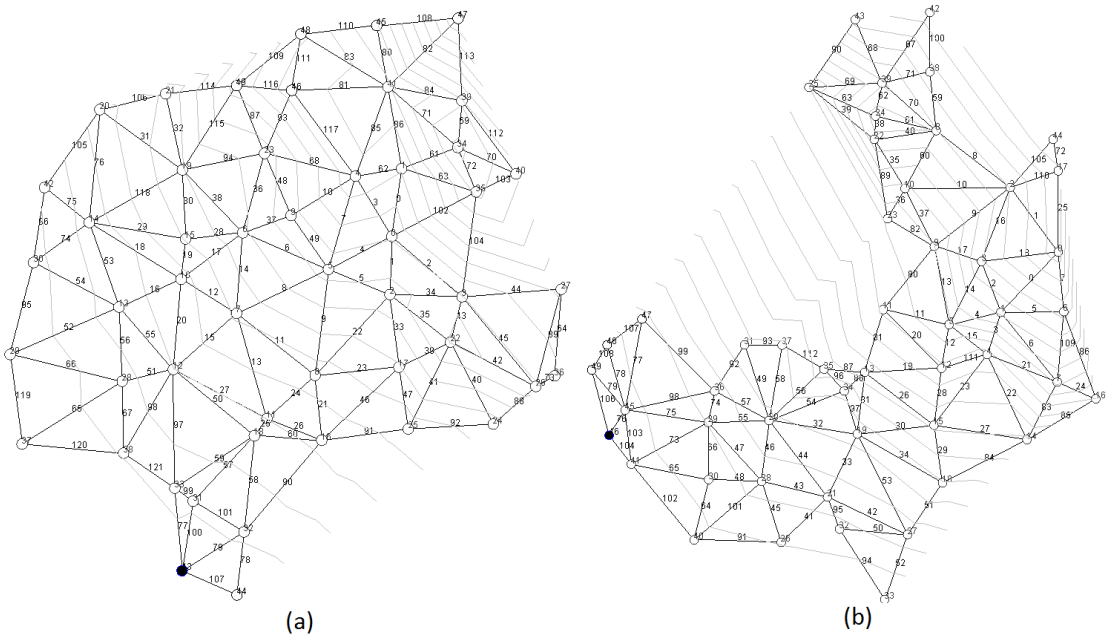


Figure 6.2: Example Base Graphs

used to determine intermediate values. In this implementation, interpolated values are not required, but the shape functions are used to create various terrain types with predictable slopes across a normalized domain which may be scaled to any size domain, maintaining shape and slopes. Figure 6.3 shows three shape functions used for terrain types.

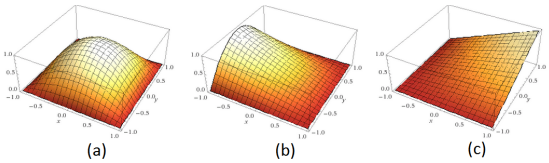


Figure 6.3: Shape Functions used for Terrain Generation

In addition to these three shape functions the inverse of Figure 6.3 (a) and (b) are used to produce two additional terrain types. A completely flat terrain as well as a random terrain, based on Ken Perlin's Simplex Noise (Perlin, 2001), is used for a total of 7 topography types. All of the terrain functions are used to calculate elevations on a normalized interval, which are then scaled to the physical domain. The terrain variance factor, t, is used to to vary the scale factor to produce an

irregular curve, rather than a completely smooth curve as defined by the shape functions, more closely modelling real terrain.

The 7 topography types are given names loosely based on the type of terrain they represent:

- *Bilinear Topography.* The bilinear topography uses the shape function shown in Figure 6.3 (c). It represents a basic, gradually varied topography, similar to the topography of Example Problem 3 in Chapter 5.
- *Hill Topography.* The hill topography is created by using the shape function shown in Figure 6.3 (a). It represents a uniform hill topography, similar to the topography of Example Problem 2 in Chapter 5.
- *Bowl Topography.* A bowl topography is achieved by using the inverse of the hill topography's shape function.
- *Convex Topography.* The convex topography is created using the shape function shown in Figure 6.3 (b). It represents the kind of topography one might expect to find in an area with rolling hills where towns are built in between the valleys and peaks of the hills. This represents an area where the network is built on such a peak.
- *Concave Topography.* The concave topography is created using the inverse of the shape function shown in Figure 6.3 (b). This kind of topography might be found in areas similar to the convex topography, however it would represent an area built in a valley of rolling hills.
- *Flat Topography.* Simply represents areas of flat topography.
- *Simplex Noise Topography.* This topography is created using simplex noise (Perlin, 2001). It creates a uniform randomized terrain with some degree of predictability, i.e. it is very unlikely that two coordinates adjacent to one another will have significantly varied elevations, which would often be the case if elevations were truly random.

Flow rate within the network is defined as a single 24 hour hydrograph expected at the sink node, i.e. the total cumulative flow within the network. This hydrograph is distributed evenly to all manholes of the base graph. Lastly, the sink node is located by specifying a percentile value, p. If $p = 10\%$ is specified the sink node is selected at random from 10% of the total manholes with the lowest elevation. This ensures the outlet is placed reasonably, i.e. not at the highest point of the network.

6.4 Creating a Problem Library

Problem instances, which closely mimic realistic engineering problems, with controllable characteristics and sufficient information to be optimized may be generated using the techniques described above. Due to the inherent randomness associated with generating a problem instance it is very difficult to predict whether or not it will have a feasible solution within a set of constraints, as these are not considered during problem generation. Most notably, if a maximum allowable cover depth for pipes is specified it is entirely possible, given other constraints such as minimum velocity and minimum allowable slope, that a solution does not exist if pumping stations are excluded from the optimization procedure.

The goal of the problem library is to create a set of standardized problem instances for which feasible solutions exist. Furthermore, the problems need to differ enough so that the performance of an algorithm for each problem is an indication of its ability, or lack thereof, to overcome specific challenges. In this study the network generation algorithm described in Section 6.3 was used to generate 153 classes of networks with varying properties. The way this particular number was arrived at is explained below. Within each of the problem classes 20 network instances were created. Consequently the problem library used here comprises 3060 network instances, all are available in Appendix B.

The variation of the different parameters, and their combination to form the 153 problem classes are described below.

6.4.1 Number of Manholes

The number of manholes is taken as either 50 or 100. Smaller problem instances do not provide much insight into an algorithm's overall performance, but can be used as an indication whether or not an algorithm is functioning as intended. Larger problem instances are problematic due to practical constraints, most notably maximum cover depths in the absence of pumping stations and rising mains. The proposed 50 or 100 manholes provides reasonable network sizes by which to compare an algorithm's ability to solve problems of varying scale within a given set of parameters.

6.4.2 Flow Rate

The total cumulative flow rate at the outlet node is specified during problem generation, once routed downstream through the eventual network the same flow rate should be obtained at the outlet. In this implementation flow is spread evenly

across the entire network. This has interesting effects when considering various levels of flow. For this reason, three cases are used for the problem set:

Low Flow. Low flow is loosely defined as the case where the vast majority of the network requires only the smallest available pipe diameter while at the downstream end only a few, if any, diameter increases are present to maintain capacity constraints.

High Flow. High flow is again loosely defined as the case where diameter increments are required in the upper ends of the network to maintain capacity considerations. Furthermore, large diameters are often present in the downstream ends of the network.

Medium Flow. A third flow condition is specified somewhere between the two extremes mentioned above. In this condition it is expected that a significant portion of the network requires some diameter increments due to capacity considerations.

The flow rate is specified as a 24 hour hydrograph. To produce the three flow conditions for the library a base hydrograph was created, shown in Table 6.1, which is used to create the flow rates of all generated networks. To vary the flow rates as

Table 6.1: Base Hydrograph

Hour	1	2	3	4	5	6	7	8	9	10	11	12
Flow Rate (l/s)	18.81	19.79	19.45	19.36	19.31	19.31	31.27	28.88	41.59	46.49	46.35	57.48
Hour	13	14	15	16	17	18	19	20	21	22	23	24
Flow Rate (l/s)	58.43	54.81	52.95	46.19	44.37	44.23	43.09	33.09	21.47	19.81	19.59	19.33

per the three definitions above, scale factors are applied to the base hydrograph to create the desired flow conditions within the network. However, what is considered low flow in a small network results in trivial flow for a large network, especially in the higher branches of the network. Consequently, different scale factors are used depending on the number of manholes for each desired flow condition.

6.4.3 Average Pipe Length and Standard Deviation

Average pipe length and standard deviation. Pipes lengths vary randomly within the system, however care is taken to ensure the average pipe length and standard deviation is consistent between problem instances where the same values are

Table 6.2: Flow Rate Scale Factors

Manholes	Flow Scale Factor		
	Low Flow	Medium Flow	High Flow
50	4.0	20.0	75.0
100	15.0	50.0	125.0

specified. For this set, problems are generated with two combinations of average pipe length and standard deviation. The combinations are shown in Table 6.3. This provides pipes with considerable variation in lengths to be produced while

Table 6.3: Pipe Variations

Combination	Average Length (m)	Standard Deviation (m)
1	100	25
2	250	75

maintaining specified averages across the system.

6.4.4 Elevations

Elevations are assigned at manholes to closely mimic real topographies with varying characteristics. The shape a specific topography takes is governed, as described in Section 6.3, by five different shape functions, a completely flat area and the simplex noise area. This results in a total of 7 different topographies. Furthermore, the steepness of slopes associated with the various topographies is governed by an average slope of 1%, 3% or 5%.

6.4.5 Necessary Exclusions

Combining the parameters above in some way creates a specific class of problems and multiple networks within the same class are equivalent to each other. Some exclusions are necessary to avoid redundancy. The most obvious redundancy is varying the slope of the simplex noise topography and the flat area. The inherent random nature of the noise function makes maintaining averages impossible, however maximum and minimum values are easily influenced through a scale factor. In this implementation only a single scale factor is used, resulting in maximum and minimum peaks of 4 meters and -4 meters from the baseline, respectively.

There is an obvious difficulty with cover constraints associated with problems in a set of 100 manholes and an average pipe length of 250 meters if only gravity sewer networks are investigated. Furthermore, increasing the pipe length increases the importance of the slope parameter. An algorithm's ability to overcome this characteristic is already evaluated when 50 manholes are present. Consequently, the case where a manhole count of 100 is combined with an average pipe length of 250 meters is excluded from the set of problem instances.

6.4.6 Problem Classes

For each parameters there now exists a few variations which are to be combined with each variation of the other parameters. There are 2 manhole counts, 3 flow rate configurations, 2 pipe length and standard deviation combinations, 3 average slope configurations and 7 topography functions of which 2 are not combined with an average slope configuration, namely the flat area and the simplex noise. The combination of these parameters is multiplicative, i.e. 2[Manhole Count] $\times$ 3[Flow Rate] $\times$ 2[Pipe Length] $\times$ 3[Slope] $=$ 36 combinations for each topography function, excluding the flat area and simplex noise where the 3 average slope variations are ignored so that their total combinations equal 12. The case where the manhole count is 100 and the average pipe length is 250 meters is excluded, this accounts for 1[Manhole Count] $\times$ 1[Pipe Length] $\times$ 3[Flow Rate] $\times$ 3[Slope] $=$ 9 cases in each of the first 5 topography functions and a total of 3 cases for the flat area and the simplex noise topography functions. Then, the total number of possible combinations is:

$$5.(36 - 9) + 2.(12 - 3) = 153$$

6.5 Summary and Conclusions

From Chapter 5 it was concluded that the observed varied algorithm performance depending on problem characteristics warranted further investigation. In this chapter techniques to generate sewer network optimization problems with controlled characteristics was described. The aim was not to create a definitive problem library for sewer network optimization, but rather to generate a sufficiently large number of problems instances to facilitate further investigation into the nature of optimality for various problems. The techniques developed here were used to define 153 so called problem classes and generate 20 problems within each class, resulting in 3060 instances of sewer network optimization problems.

Chapter 7

Characteristics of Optimal Networks

In this chapter the algorithms described in Chapter 5 are applied to each of the 3060 problem instances of Chapter 6. The performance of each algorithm is monitored and averaged over multiple runs. However, more insight into why algorithms perform differently is desired. If each algorithm was simply run and averaged the only possible conclusion would be that certain algorithms performed better for certain problems due to differences in layout construction, which has already been concluded from the results obtained in Chapter 5. Consequently, layout parameters by which insight into the nature of a layout may be gained, are defined. These parameters are then monitored as the optimization process proceeds. In doing so, an understanding of how the layouts changed from the early trials to the eventual solution is gained. In addition to monitoring the parameters, the algorithm which performed best, on average, is also saved for each problem.

The data captured during optimization monitoring is used to determine if statistical correlations between network parameters and network cost exist. Once correlations are established, heuristic influence factors, η_{ij} in equation 4.1, are defined which encourage decisions that augment an increase or decrease of the correlating parameter during layout creation, depending on the correlation. The heuristic influence factors, and various combinations of thereof, are then applied to the algorithms of Chapter 5, which are used again to solve each of the 3060 problem instances from the library. The change in the modified algorithms' performance is monitored and the best combination of heuristic factors for the different problem classes are presented.

7.1 Network Layout Parameters

A total of 13 parameters relating to network layout are proposed. Where possible, graphical aids are provided to visualize them. The proposed parameters can be applied as heuristic influence factors during layout creation by exploiting whatever information is present at that stage of the optimization process.

Figure 7.1 shows an example network from the problem set which is used throughout this section to display graphical representations of the proposed parameters. It has bi-linear elevation distribution with a terrain variance factor of 0.1. The average slope is $0.3m/m$ and the outlet manhole, marked in blue, is placed in the lowest 10% of elevation. There are 50 manholes and 112 pipes, with an average length of $100m$ and standard deviation of $25m$. The flow rate is defined as *high*.

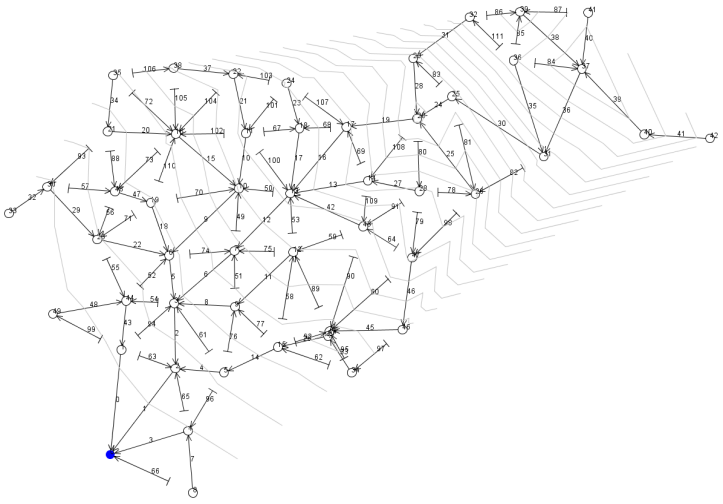


Figure 7.1: Example Network

All graphical representations are plotted on the best layout obtained by a single run of the *direct-edge* selection strategy from Chapter 5. The ACO algorithm was scheduled to run for 2500 generations with a population size of 20, an initial evaporation rate of 0.003 and an evaporation rate of 0.02 after 85% of the maximum generation limit has been reached. All hydraulic constraints are as defined in Section 3.4 with a γ_b value of 0.25. The proposed parameters are described in the subsections below. Their correlation with the eventual network cost is calculated using Spearman's correlation coefficient (Daniel, 1990) and discussed below. Note that the discussion of correlation refers to the complete result set obtained from

monitoring the parameters during optimization of the problem library, and the figures shown here are only to provide context. Furthermore, the correlations were calculated using the averaged results of the algorithm which performed the best, on average, for each problem of the library. Detailed results of each parameter's correlation for all problem classes can be found in Appendix B.

7.1.1 Elevation Rank Prominence

The Elevation Rank of a node is determined by simply ranking all nodes from lowest elevation, which has the lowest rank, to highest elevation, which has the highest rank. Elevation rank prominence is a measure of how often the pipes run from nodes with higher to ones with lower elevation, or vice versa. The parameter incorporates both conditions simultaneously and its value is a measure of balance between the two conditions. Elevation Rank Prominence is defined as:

$$\text{ERP} = \sum_{i}^{N} x_i$$

in which:

$$\begin{aligned} \text{ERP} &= \text{Elevation Rank Prominence} \\ \text{N} &= \text{Number of pipes, and} \\ x_i &= \begin{cases} 1 & \text{From high to low} \\ 0 & \text{Equal ranks} \\ -1 & \text{From low to high} \end{cases} \end{aligned}$$

The maximum and minimum value this parameter can achieve is N, if all pipes run from high to low ground, or -N if all pipes run from low to high. Figure 7.2 shows a graphical representation of elevation rank prominence. The bold pipes run from high to low ground, while the others run from low to high. Note that most of pipes run from high to low ground. The vast majority of pipes which run from low to high ground are pipes reintroduced during the completion phase where distance rank, as proposed by Moeni and Afshar (2012), rather than elevation, is used to determine the target node.

This parameter showed a strong negative correlation with the solution cost, i.e. the more pipes ran from high to low ground, the lower the network cost. This result is expected as this allows reasonable cover depths to be maintained. The negative correlation is observed more strongly for low flow rates, while for high flow rates negative correlation is only observed when the average ground slope is steep.

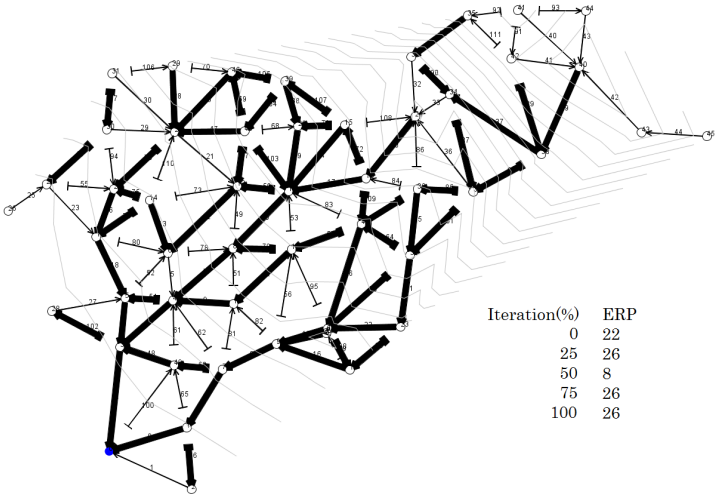


Figure 7.2: Elevation Rank

The network of Figure 7.2 has a high flow rate, yet clearly has the majority of its pipes running from high to low ground while those which are not were heuristically introduced after the layout construction procedure's conclusion. This is explained by observing the change in value of the elevation rank prominence during the optimization. For low flow rates the parameter often starts low and increases significantly during the optimization leading to the strong negative correlation. For networks with high flow the parameter starts high and only increases slightly over the course of the optimization and consequently a low negative correlation is observed. The correlation pattern for medium flow rates is, as might be expected, a balance between the pattern of low and high flow rates. Consequently it is important not to solely rely on the eventual correlations alone.

Increasing the average pipe length has no significant impact on the correlation for any case, however when increasing the number of manholes there is a distinct reduction in correlation for all cases. This is again due to the network starting with high elevation rank prominence to avoid excessive cover depths or even infeasibility.

7.1.2 Average Elevation Rank Difference

This is the average difference between ranks of connected manholes. Fundamentally it is a variation of the elevation rank prominence parameter and consequently its correlation patterns match that of the latter almost exactly. However, this parameter offers some insight into an ideal difference between elevation ranks when coupled with its standard deviation, defined below.

7.1.3 Elevation Rank Difference Standard Deviation

This is a measure of consistency between the differences in elevation ranks. This parameter shows the same correlation pattern as the previous two elevation rank parameters, however the values are inverted. The positive correlation of the standard deviation with cost implies that as the standard deviation of elevation rank difference decreases so too does the cost, and vice versa. This means that networks which have more evenly distributed and consistent elevation rank differences result in a lower cost.

7.1.4 Average Distance Rank

The distance rank of a manhole is defined as the number of manholes succeeding it in its branch down to the outlet sink. Figure 7.3 shows the distribution of distance ranks throughout the layout, where the drawn manholes sizes indicate distance rank. It is hard to observe any useful information from the distance rank distribution initially. However, from the correlations a pattern emerges. A

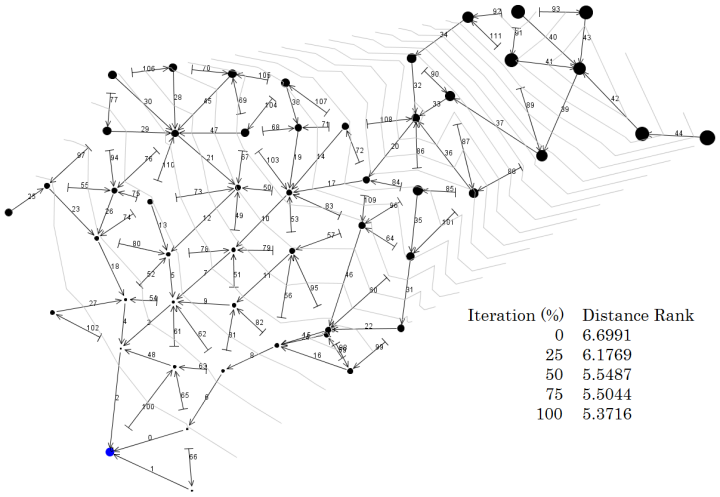


Figure 7.3: Distance Ranks

moderate positive correlation increases with flow rate and problem scale, i.e. higher average distance rank increases the cost of networks if the flow rates or number of manholes are higher. This seems logical since for high flow rates, longer pipes lead to substantial increases in accumulated flow rates downstream. Furthermore, the length of branches can be significantly higher in larger networks, increasing both the flow rate and required downstream cover depths. Referring now to Figure 7.3, and remembering that it is a best solution, it can be observed that no single

distance rank outlier is present but rather a reasonable spread of distance rank exists, which is consistent with the correlations. It is also worthwhile to note that in the Direct Node algorithm of Chapter 7, distance rank is used in the decision making process of the layout creation algorithm. This algorithm often outperformed its edge counterpart, where distance rank is not used during layout creation, for large scale problems and when flow rates were high. This further demonstrates the importance of lowering average distance rank for large networks with high flow rates.

7.1.5 Distance Rank Standard Deviation

This represents a measure of consistency between distance ranks of manholes. This parameter shows the same correlation pattern as the average, again reinforcing the hypothesis that networks with consistent branch lengths have lower costs than networks with outliers in terms of length.

7.1.6 Maximum Distance Rank

This is the maximum distance rank achieved by any manhole in the network. Again, the same pattern is observed as with the average and standard deviation. The conclusion is the same, namely that reducing the maximum length of a single branch, in favour of a more even distance rank distribution, increases fitness and thus lowers cost.

7.1.7 Average Slope

A consistently strong negative correlation between the average slope of all pipes and the network's fitness is observed. This correlation reduces significantly as flow rates and network sizes increase. This implies that for small networks with low flow rates, increasing the average slope in the network is a beneficial decision. Increasing the average slope keeps diameters smaller, which reduces network cost. This is consistent with the intent of the γ_b parameter of the HOMS algorithm, which is used to increase slopes and keep diameters smaller. For high values of γ_b a significant reduction in slope is required before a diameter increment will be accepted. Consequently, for small networks with low flow rates, high values of γ_b should yield lower eventual network cost.

Figure 7.4 shows the distribution of slopes. It is impossible to deduce useful information by visual inspection of the distribution of slopes. Side by side comparison with other distributions reveals that pipes with larger diameters often have less steep slopes. However, this is not always true, as in some cases the slope has

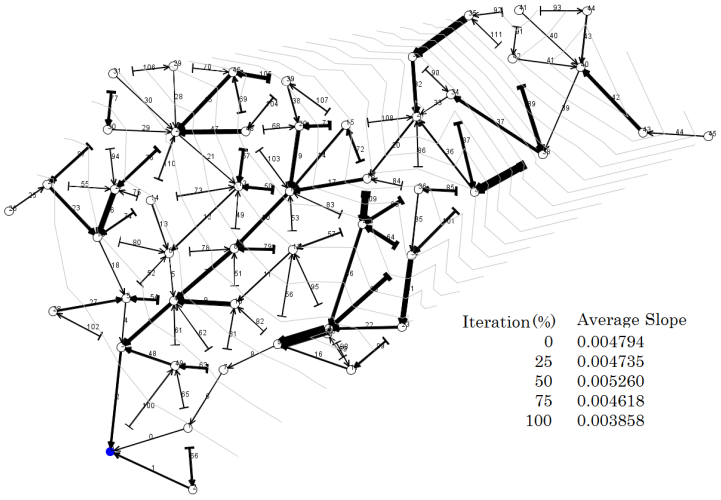


Figure 7.4: Distribution of Slopes

to be increased to maintain minimum cover depths at the downstream end, or a diameter increment was rejected in favour of a larger slope by the γ_b parameter of the HOMS algorithm.

While the average slope does offer some insight, it is impossible to use this parameter to influence the tree growing algorithm in any way, other than through the use of ground elevations, since no hydraulic analysis takes place before a network layout has been completed.

7.1.8 Slope Standard Deviation

This is a measure of the consistency of slopes across the network. A moderate negative correlation is observed which decreases with increasing flow rate and problem scale. This implies that for small networks with low flow rates, less consistent slopes are beneficial. For large problems with high flow rates there is very little to no correlation with fitness, i.e. the consistency of slopes gives no indication of fitness.

7.1.9 Maximum Pipe Slope

No meaningful correlation between network fitness and the maximum pipe slope in the network is observed for any of the tested networks.

7.1.10 Average Degree Centrality

In graph theory, degree centrality of a node is a measure of the contextual impor-

tance of the node relative to others in the network. Degree centrality is a local measure at each node, or manhole, of how many in and out edges, or pipes, it has. In this case, due to the single outlet constraint, only incoming pipes are considered. If high average degree centrality occurs often, this could easily be used to influence the decisions of the layout creation algorithm. However, this parameter demonstrated no significant correlation with network fitness for any problem.

7.1.11 Average Betweenness Centrality

Betweenness centrality of a node is a measure of how many shortest paths between other nodes pass through that specific node (Freeman, 1978). The more shortest paths that pass through the node, the higher its betweenness centrality. This is

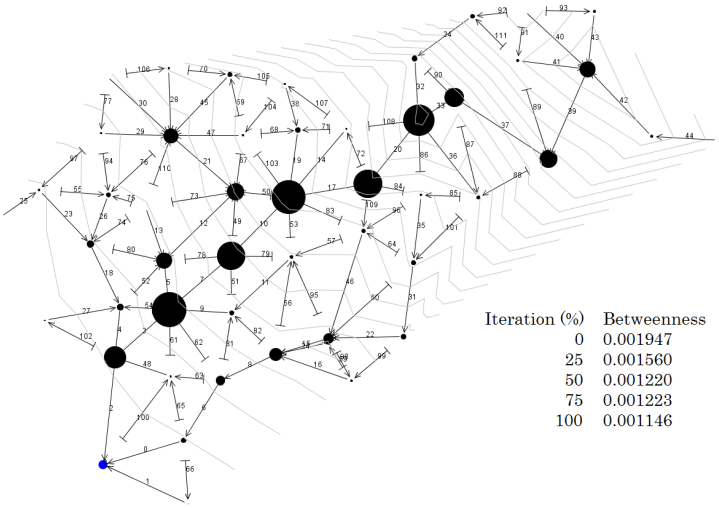


Figure 7.5: Betweenness Centrality Distribution

a particularly interesting parameter. It demonstrates moderate correlation with network fitness for many problem cases. However, it varies between positive and, though seldom, negative correlation and is heavily influenced by flow rate. When flow rates are low there is almost no correlation, but it increases with flow rate. Furthermore, when increasing the scale of the network the parameter correlates more strongly for low flow rates and less so for high flow rates.

Referring to Figures 7.5 and 7.6, a very clear correlation between betweenness and pipe diameter is observed. From that we can conclude that pipes between nodes with high betweenness will have high flow rates relative to the rest of the network. It is important to note that a positive correlation implies that a layout with multiple branches is more economical, while a negative correlation implies

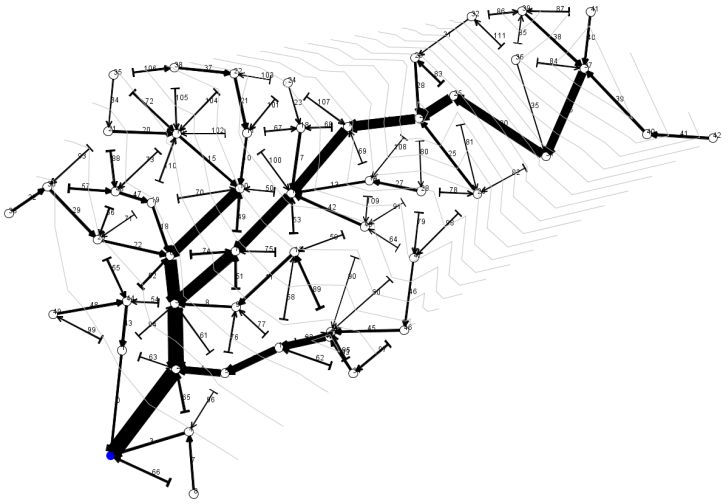


Figure 7.6: Diameter Distribution

that a single main branch to which all others flow is the more economical layout. Since positive correlations occurred more often than negative ones, indications are that multiple branched layouts should be more economical. This observation is reinforced by the next two parameters.

7.1.12 Betweenness Centrality Standard Deviation

This gives insight into the consistency of betweenness centrality across all nodes. This parameter is as sporadic as the average. It does show moderate positive correlation for medium flow rates. This implies that a more even distribution of branches, i.e. a more even spread of flow, leads to more economical layouts.

7.1.13 Maximum Betweenness Centrality

The maximum betweenness centrality aims to identify the presence of manholes of significant importance, i.e. the presence of a main branch into which all others feed. There is again a moderate positive correlation between this parameter and cost. From this it may be concluded that a reduction in maximum betweenness, i.e. avoiding a single branch which transports the majority of flow downstream, leads to more economical layouts.

7.2 Algorithm Performance

In addition to monitoring the parameters to calculate correlations, algorithm per-

formance was also monitored and averaged for each problem. Similar to the results of Chapter 5, the direct-edge strategy performed most consistently and produced, for the vast majority of the problem classes, the best average result. However, with increasing network size, the direct-node alternative sometimes performed better than its edge counterpart. This is consistent with the correlation pattern of distance rank, where a positive correlation increasing with problem size and flow rate was observed, since the direct-node strategy is already heuristically influenced by distance rank during its decision making. The permutation approaches only performed the best on average for two of the 153 problem classes, and then only by 0.1% and 0.3% respectively. These two classes, in contrast to the results of Chapter 5, were for two small problems with low and medium flow rates respectively. The only conclusion to made here is that while the permutation approach performed well for the idealistic example problem of Chapter 5, it struggled when exposed to more realistic problems with varying topology, scale and flow rates.

7.3 Heuristic Influence Factors

In Section 7.1 the correlation between network fitness and thirteen proposed layout parameters were discussed. Three parameters showed promising statistical correlation with solution fitness, namely (i) Distance Rank (ii) Elevation Rank and (iii) Betweenness Centrality. They are now used to propose heuristic influence factors, η_{ij} in Equation 4.1, which can be applied to the layout creation procedures of the ACO algorithms described in Chapter 5. The proposed heuristic influence factors are formulated such that they attempt to influence the desired parameter only. However, due to the interdependent nature of the problem variables, when one is changed others are unavoidably changed as well. The concomitant influences are kept to a minimum. Furthermore, the results of Section 7.2 indicated that the permutation approaches almost never performed the best on average for any of the problem classes. Consequently it was decided to not include these approaches in the heuristic influence factor study and to focus only on the two most promising layout creation strategies, namely direct-edge and direct-node.

As a visual assistance tool in the development of the heuristic influence factors, the Reingold Tilford Algorithm (Reingold and Tilford, 1981) is used to produce tree images of created layouts, as shown in Figure 7.7. The Reingold Tilford tree does not directly tie in with any of the thirteen parameters of Section 7.1, or with their correlations, but it does offer valuable insight when studying the networks and how the proposed heuristics influenced the layout. For example, when networks are studied which display a favourable value of a correlating parameter, such

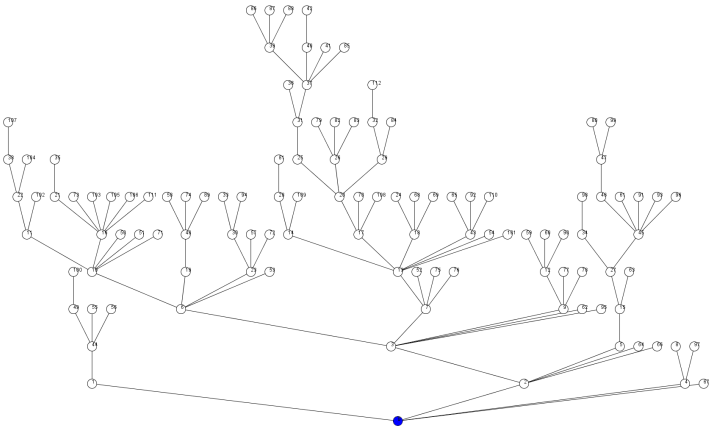


Figure 7.7: Reingold Tilford Tree Layout

a low average distance rank, it is helpful to compare the resulting Reingold Tilford tree with that of a solution where the average distance rank is high. Differences in branch structure and lengths can more readily be observed in the Reingold Tilford layout than in the original.

The formulation of the heuristic factors of the three parameters is described below.

7.3.1 Distance Rank

Distance rank is already present in the algorithms during the completion phase. However, it can be introduced during the tree growing phase with the aid of a heuristic influence factor. The ants select elements out of an eligible set as described see Sections 4.3.3.2 through 4.3.3.3. By defining the heuristic factor of the transition rule as Equation 7.1, the probability of an element being selected is significantly influenced by distance rank.

$$\eta_{ij}^{dr} = \left(\frac{1}{1+dr_j}\right)^5 \tag{7.1}$$

Where:

η_{ij}^{dr} = Distance rank heuristic influence factor

dr_j = Distance rank of node j

Since distance rank starts at 0, at the outlet manhole, the denominator is incremented by 1 to avoid division by zero. The degree of influence is controlled by the power factor, 5, in Equation 7.1. To arrive at this factor a random set of 20 problems were selected from the 3060 problems of the library developed in Chapter

6. The direct-edge algorithm was applied to each of the selected problems while using Equation 7.1 and varying the power factor from 1 through 10. The results were averaged over 20 runs for each problem. The power factor, 5, performed the best on average.

7.3.2 Elevation Rank

Elevation rank can be introduced as an influencing factor in two ways. The first is to replace distance rank as the deciding factor during the completion phase, see section 4.3.3.6 of the layout creation algorithm. The direction of the edge being reintroduced during completion is simply chosen from high to low ground if possible. If the elevation is flat the direction is chosen at random. The second is to introduce a heuristic influence factor, η_i^{er}, during the tree growing phase. The data of Section 7.1 shows that introducing edges such that they are directed from high to low ground is beneficial. However, it also suggests there may be a problem dependent optimal difference in elevation ranks. Consequently, the heuristic influence factor, η_i^{er}, is proposed such that it is dependent on the difference in elevation rank, as shown in Figure 7.8, rather than simply whether or not the edge runs from higher to lower elevation. Referring to Figure 7.8: C is the maximum influence

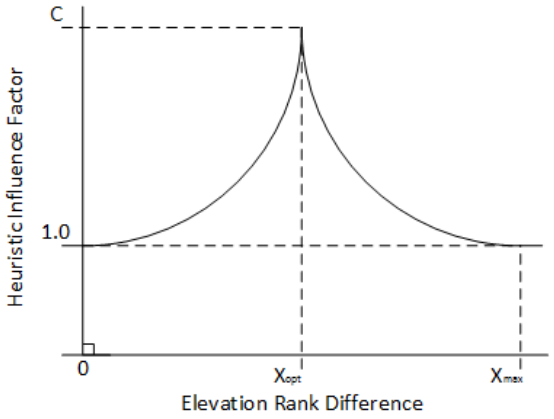


Figure 7.8: Elevation Rank Heuristic Distribution

factor at the optimal elevation rank difference, x_{opt} and x_{max} are chosen a-priori to govern when and how strongly the elevation rank heuristic influence factor, η_{ij}^{er}, is applied based on the difference between elevation ranks, resulting in Equation

7.2.

$$\eta_{ij}^{er}, = \begin{cases} 1.0 & x \leq 0 \\ \frac{C-1.0}{x-m^2}x^2 + 1 & 0 < x < x_{opt} \\ y & x_{opt} \leq x < x_{max} \\ 1.0 & x \geq x_{max} \end{cases} \tag{7.2}$$

Where:

$$x = \text{The difference in elevation rank of nodes } i \text{ and } j$$

$$y = ax^2 + bx + k \tag{7.3}$$

Where:

$$\begin{aligned} a &= \frac{C-1}{(x_{opt} - x_{max})^2} \\ b &= -2 * x_{max} * a \\ k &= 1 - ax_{max}^2 - bx_{max} \end{aligned}$$

7.3.3 Betweenness Centrality

Betweenness centrality demonstrated promising statistical correlation. Calculating the betweenness centrality for each eligible decision, at each decision point, is computationally expensive and does not reliably predict the eventual betweenness centrality of the completed layout. Consequently, the behaviour of the tree growing algorithm should rather be modified to produce layouts with either fewer, longer branches, or more, shorter branches thereby influencing the betweenness centrality in a more predictable manner.

The tree growing algorithm can be modified in different ways to influence the branching. Prioritizing the newest entry to the growing tree changes it to behave like a recursive backtracker algorithm, which produces long and windy branches. Prioritizing the oldest eligible edges in the tree produces many shorter branches. The unaltered state, i.e. random prioritization of edges, behaves very similar to Prim's algorithm, which creates both long and short branches (Buck, 2015). With a view to influence eventual betweenness centrality, four changes to the tree growing algorithms are proposed: (i) Prioritize the newest edge in the tree to produce long branches and often a central branch to which shorter branches connect. (ii) Avoid the newest edge in the tree to create many branches of varying length. (iii) Prioritize the oldest edges in the tree to create many short branches, and (iv) Avoid the oldest edges in the tree to slightly favour a few longer branches.

High average betweenness centrality implies there exists a single important branch in the network to which most smaller branches route, while low betweenness centrality implies that many, shorter branches, route down to the outlet. Consequently, the first proposed modification is expected to yield networks with high average betweenness centrality. The second and third modifications should yield networks with low average betweenness centrality, while the fourth modification is expected to yield a balanced average betweenness centrality.

The proposed changes to the heuristic factors are applied to both the direct-edge and direct-node algorithms.

7.4 Evaluating the Heuristic Influence Factors

In this section the heuristic modifications to the tree growing algorithm proposed in the previous section are applied to the direct-edge algorithm of Chapter 5. The algorithm is then used, with one heuristic active at a time, to optimize the problem shown in Figure 7.1 while monitoring each of the 13 parameters from Section 7.1. The intent here is not to evaluate the performance of each heuristic itself, but rather to observe whether its desired effect has been achieved. Furthermore the concomitant effect of each heuristic factor on the 13 proposed parameters is monitored as well. In Section 8 the heuristic influence factors, and combinations thereof, are applied to each of the 3060 library problems and their performance is evaluated.

Figure 7.9 shows the progress of solution cost over the course of the optimization for each proposed heuristic influence factor. The horizontal axis shows the generations of the ACO algorithm. The unaltered state as in Chapter 5, where

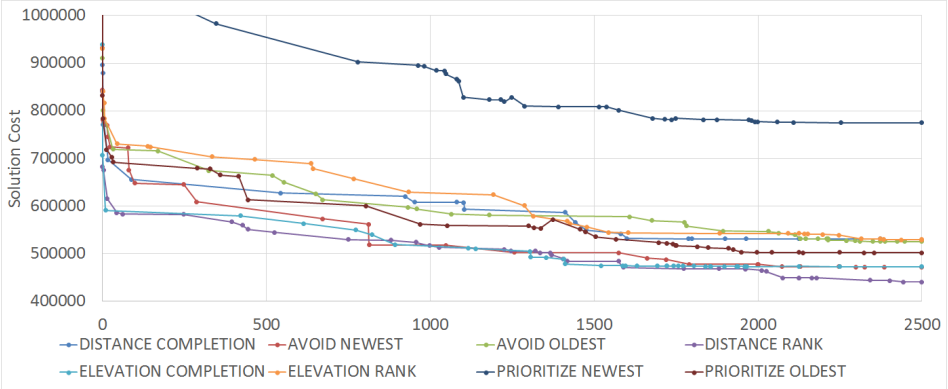


Figure 7.9: Progression of Best Solution Cost

distance rank is used during the completion phase and no other heuristic influence is present, is used as the benchmark. It is denoted DISTANCE COMPLETION. Prioritizing the newest edge resulted in a significant increase in solution cost. Avoiding the oldest edges and applying the elevation rank heuristic during the tree growing phase had very little to no effect. Prioritizing the oldest edges showed a slight improvement, while avoiding the newest edge showed the most significant improvement of all the modifications to the tree growing algorithm. Introducing elevation rank as the deciding factor during the layout completion phase showed significant improvement in fitness, while the best modification for this problem is using distance rank during the tree growing phase to influence decisions.

From these results it can be concluded that combining certain heuristic influence factors, e.g. using elevation rank during completion and distance rank to influence the tree growing algorithm, would be beneficial.

Figures 7.10 through 7.22 show the concomitant effect of the heuristic influence factors on the parameters of Section 7.1. The effect on each of the 13 parameters is discussed in the subsections below.

7.4.1 Elevation Rank

The three elevation rank parameters, which all showed strong correlation with the fitness, and how they are influenced by the various heuristics are shown below in Figures 7.10, 7.11 and 7.12. In all three cases, the elevation completion heuris-

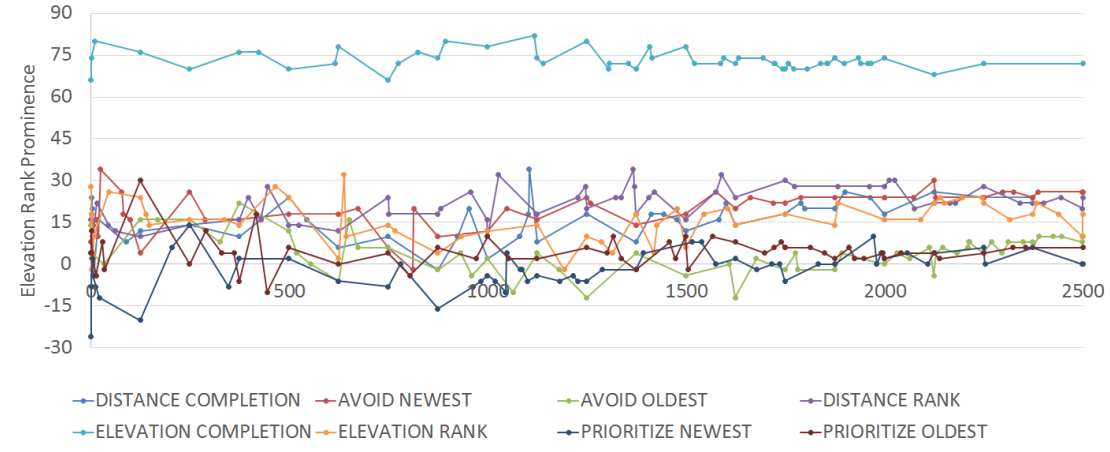


Figure 7.10: Progression of Elevation Rank Prominence

tic stands out, while the other heuristics have little to no effect on the elevation parameters. This is the desired result. Notice the slight upward trend for all

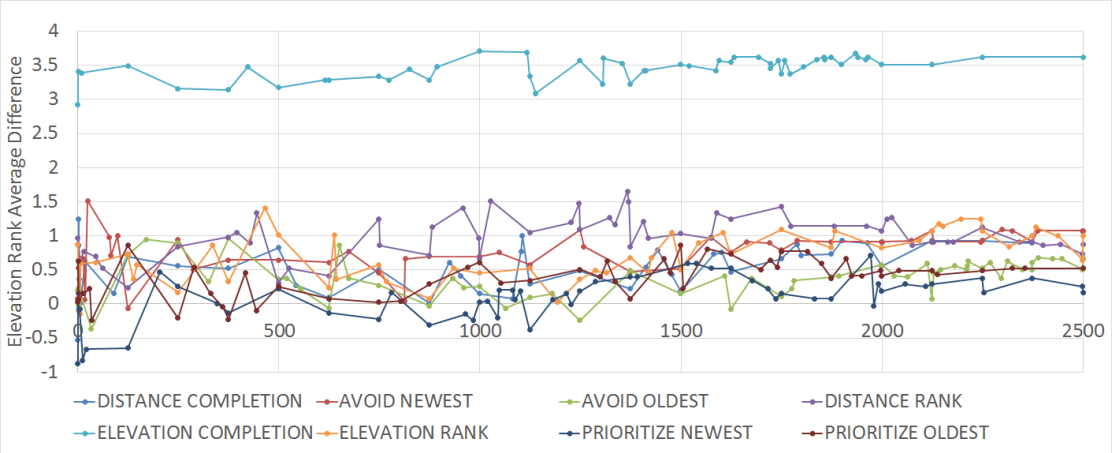


Figure 7.11: Progression of Elevation Rank Average Difference

parameters in both the elevation rank prominence and elevation rank average difference. This observation is consistent with the correlations of Section 7.1 where elevation rank prominence and elevation rank average difference displayed moderate negative correlations for networks with high flow rates, as is the case for the network under consideration.

Perhaps unexpectedly, the use of the elevation rank during the tree growing phase has very little to no effect on the elevation parameters. Referring to Figure 7.2, however, it is clear that the majority of edges are already running from high to low ground, while those which are not were introduced during the completion phase where distance rank is the deciding factor. Consequently, influencing the tree growing phase to select edges running from high to low ground is simply steering the algorithm toward decisions which were naturally made during the tree growing phase without any heuristic influence.

The reduction in standard deviation observed for the elevation completion is consistent with the positive correlation with solution cost observed in Section 7.1. Of all the heuristic factors the elevation completion performed the second best, after distance rank. The improved performance can be attributed here to the increase in elevation rank prominence and elevation rank average difference, along with the reduction in standard deviation. They all displayed corresponding correlation to the solution cost, namely negative correlation to parameter increase and positive correlation to parameter decrease.

7.4.2 Distance Rank

The three distance rank parameters experience more subtle changes than the elevation parameters, as seen in Figures 7.13 through 7.15. The distance rank heuristic

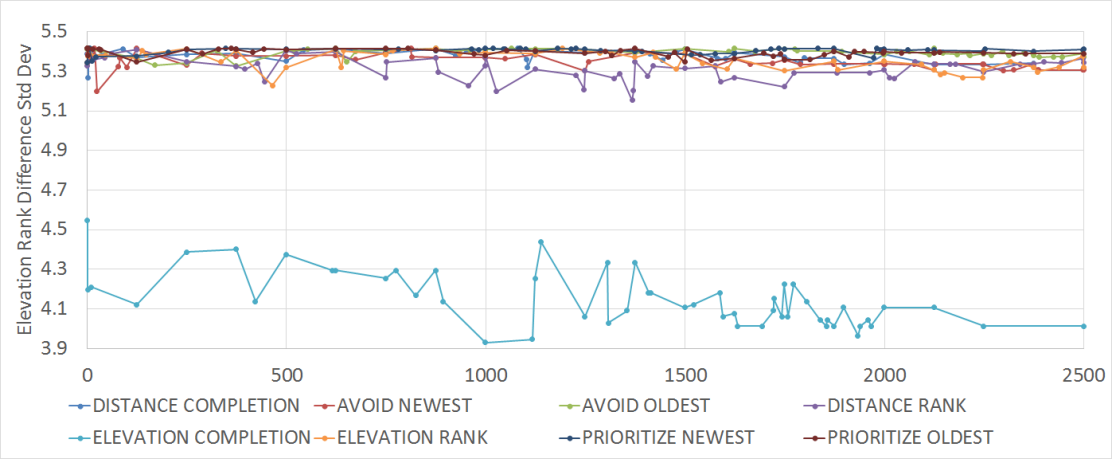


Figure 7.12: Progression of Elevation Rank Difference Standard Deviation

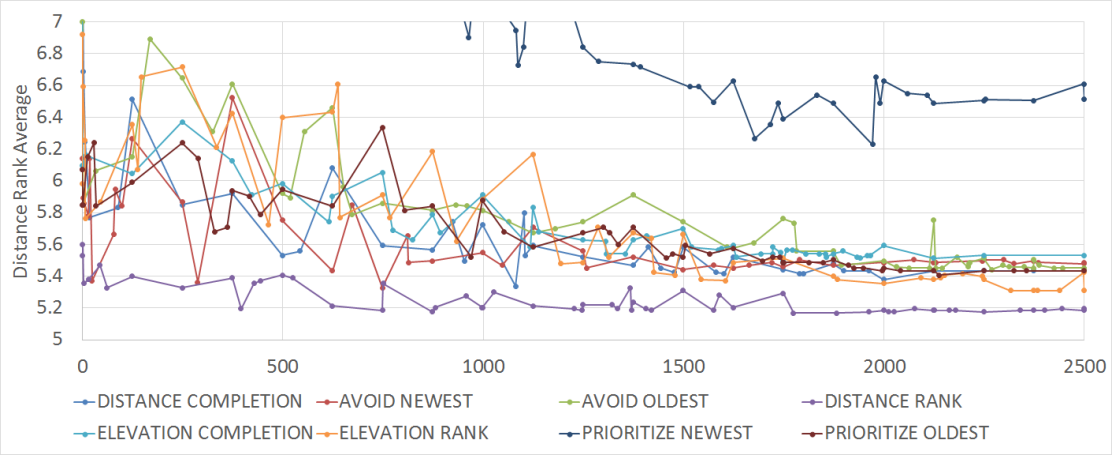


Figure 7.13: Progression of Distance Rank Average

reduces the average distance rank by a significant margin, as could be expected. A significant increase in average distance rank is observed when newest edge is prioritized. This can be attributed to the fact that prioritizing the newest edge will produce long branches. No other influence factor had a significant impact on the average distance rank. Notice the consistent downward trend for all factors, which is consistent with the positive correlation observed in Section 7.1. The distance rank standard deviation does not show any significant change apart from prioritizing the newest edge, which has a much higher standard deviation than the other parameters. There is again a consistent downward trend present in the standard deviation which is consistent with the parameter's correlation. The maximum distance rank does not reveal any discernible pattern with all heuristics tending

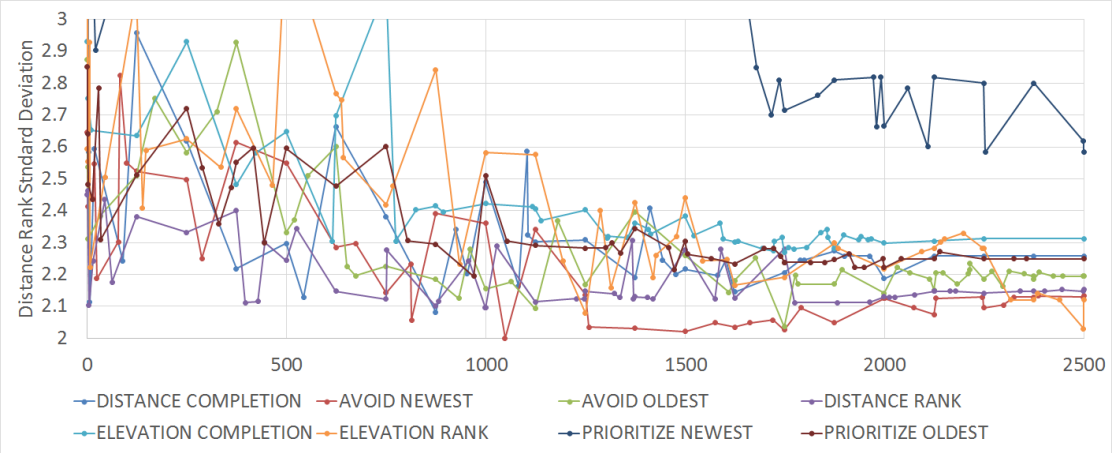


Figure 7.14: Progression of Distance Rank Standard Deviation

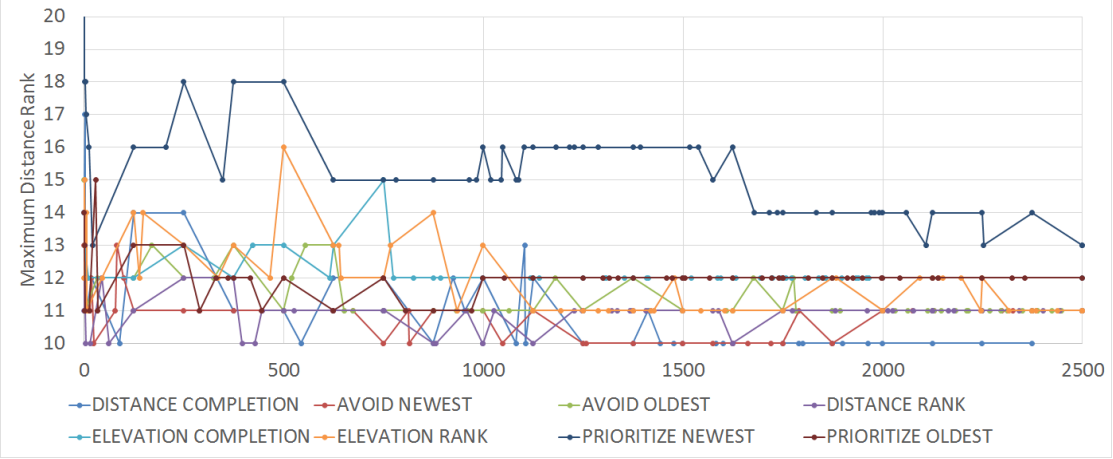


Figure 7.15: Progression of Maximum Distance Rank

towards similar values, aside from prioritising the newest edge which leads to a significant increase in maximum distance rank during the early trials. The only conclusion to be made here is that a low maximum distance rank, relative to the network under consideration, is beneficial.

In Figure 7.9 it is observed that prioritizing the newest edge performs significantly worse than the other heuristics. The increase in solution cost observed there can be attributed, at least in part, to the accompanying increase it causes in all three of the distance rank parameters. The reduction in solution cost observed for the distance rank heuristic, however, is compatible with the change in parameters seen here.

7.4.3 Slope

While none of the heuristics directly use slope information to alter the layout creation, the slope is nonetheless influenced by some of the factors. The progression of the three slope parameters are shown in Figures 7.16, 7.17 and 7.18. The

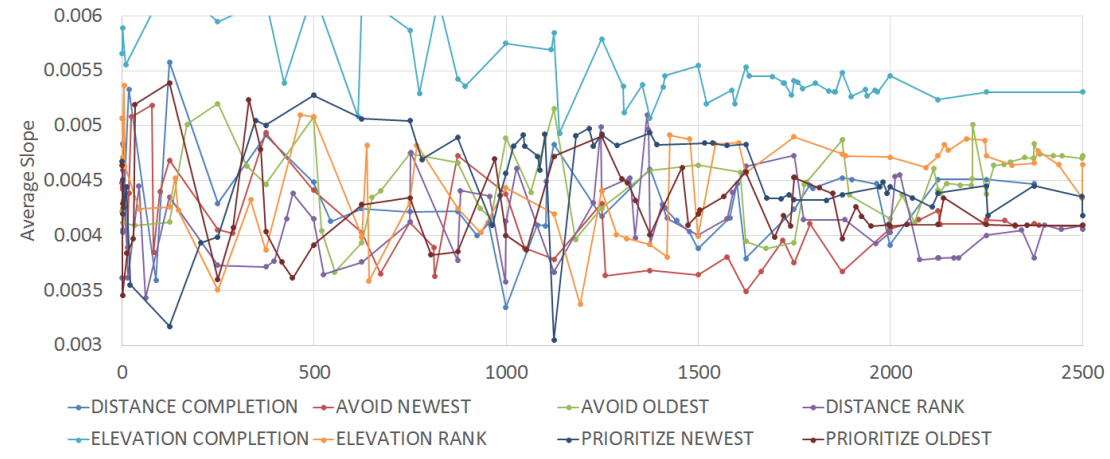


Figure 7.16: Progression of Average Slope

average slope showed a negative correlation with solution cost, reducing as the flow rate increased. The problem under consideration has high flow rate, and its class showed a very moderate negative correlation with average slope. This implies that increasing the average slope will still be mildly beneficial. This is seen in the slight upward trend of all factors in Figure 7.16. The elevation rank completion heuristic has lead to a significant increase, relative to the other parameters, in the average slope. This is due to the increased number of pipes added during the completion phase which flow from high to low ground. Pipes which are added during the completion phase have low flow rates in them, since they are disconnected from their upstream manholes. Consequently, the slope is most often determined by ground elevation and cover considerations, which in general result in more steep slopes than the hydraulic requirements at low flow rates. All other heuristics ended with very minor changes to the average slope relative to the unaltered state. The slope standard deviation does not offer any valuable insight as no heuristic influence factor caused a significant change. This is consistent with the observed correlations, where only for small problems with low flow rates a small negative correlation is observed. For the problem under consideration, which has a high flow rate, the standard deviation of slope offers no indication of the eventual solution cost. The maximum slopes, which showed very little to no correlation, also offers no valuable insight. Avoiding the newest edges during layout creation did result in

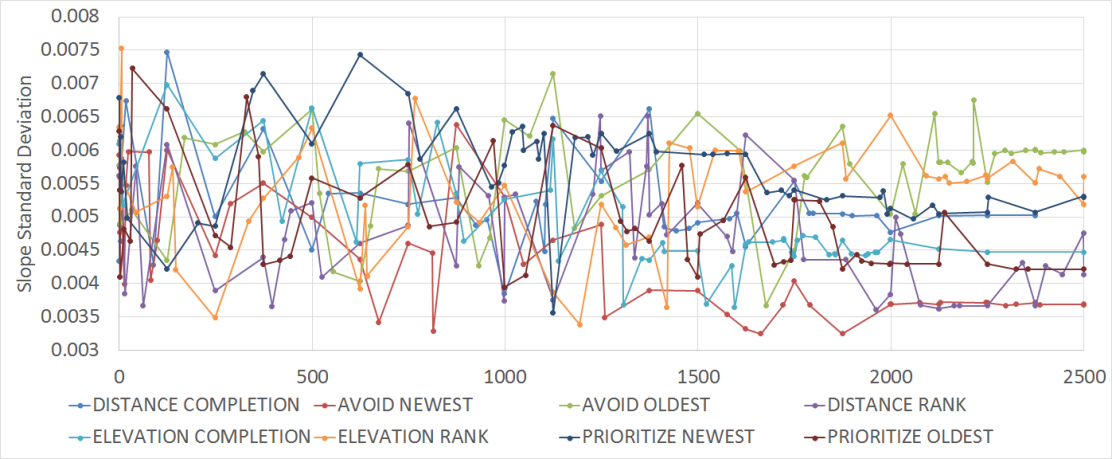


Figure 7.17: Progression of Slope Standard Deviation

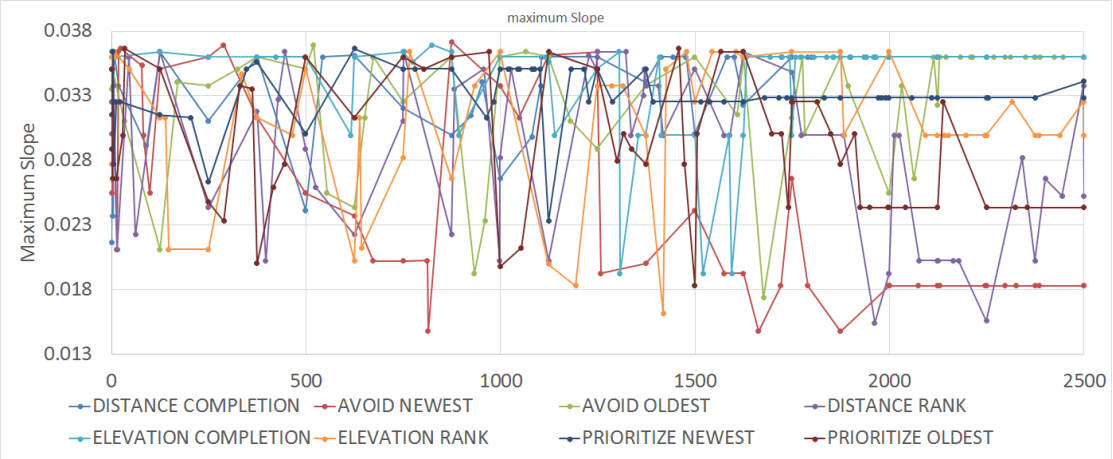


Figure 7.18: Progression of Maximum Slope

a significant reduction in maximum slope. However, due to its lack of correlation with fitness, this offers no insight into the impact it has on solution fitness.

7.4.4 Degree Centrality

Degree centrality did not show any significant correlation with solution cost, nor is any real pattern revealed by observing its progression, as seen in Figure 7.19.

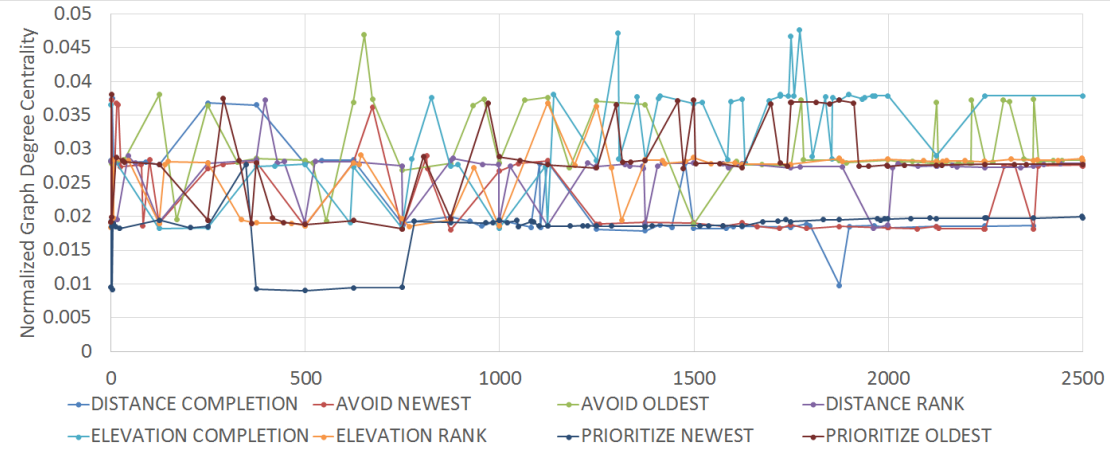


Figure 7.19: Progression of Graph Degree Centrality

7.4.5 Betweenness Centrality

The progression of the three betweenness centrality parameters are shown in Figures 7.20, 7.21 and 7.22. The distance rank heuristic has the most significant

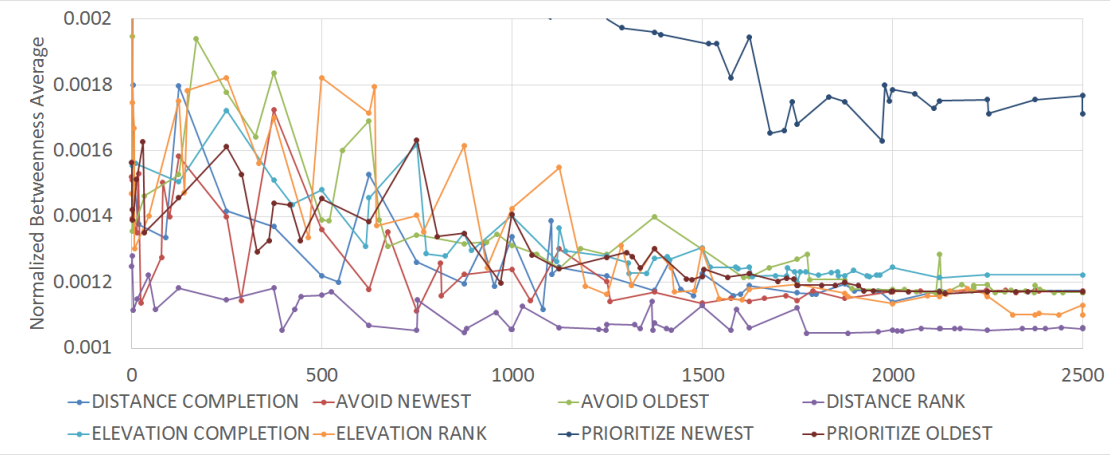


Figure 7.20: Progression of Average Betweenness Centrality

impact on the average betweenness centrality. This is not unexpected since betweenness centrality is heavily influenced by the structure of the network, which is itself heavily influenced by the length of branches. The only other parameter which showed significant deviation from the unaltered state is prioritizing the newest edge for selection, which showed a significant increase in average betweenness centrality. This is the expected result since a long central branch in the network will have,

relative to the rest of the network, high betweenness. The other parameters do influence betweenness centrality early on. However as the optimization progresses there is a definite downward trend towards, presumably, the optimal for this problem, which is consistent with the moderate positive correlation observed for this problem class.

Since this parameter is strongly coupled with diameters, as shown in Section 7.1.11, it offers valuable insight into the diameter distribution as well. Here a definite downward trend in average betweenness centrality is observed, i.e. the network has a balance of branches feeding into the outlet rather than a central branch into which others feed. Referring to Equation 2.2, the unit cost a pipe is exponentially dependent on pipe diameter. In a network with high flow rate where a central branch is present, large diameters will be required at the downstream end to accommodate the flow rate, increasing cost significantly. For a more balanced branch system the diameters across all branches will be smaller, leading to lower network cost. The betweenness centrality standard deviation is shown

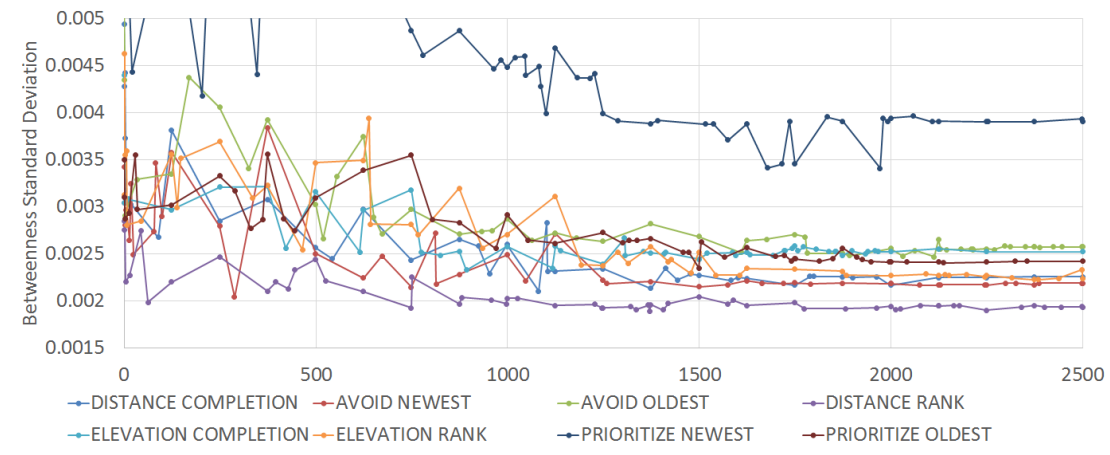


Figure 7.21: Progression of Betweenness Centrality Standard Deviation

in Figure 7.21. Again the distance rank heuristic has the greatest reducing effect and prioritizing the newest edge increases the standard deviation significantly. A reduction in standard deviation implies a well balanced branched network where no single branch of high importance is present, which has the same effect as a reduction in the average betweenness centrality as described above. The maximum betweenness centrality value for a single node is shown in Figure 7.22. The unaltered state has the same final value as that of distance rank, while all the others have increased the maximum betweenness by varying degrees. Since all three the betweenness centrality parameters demonstrated positive correlation for the

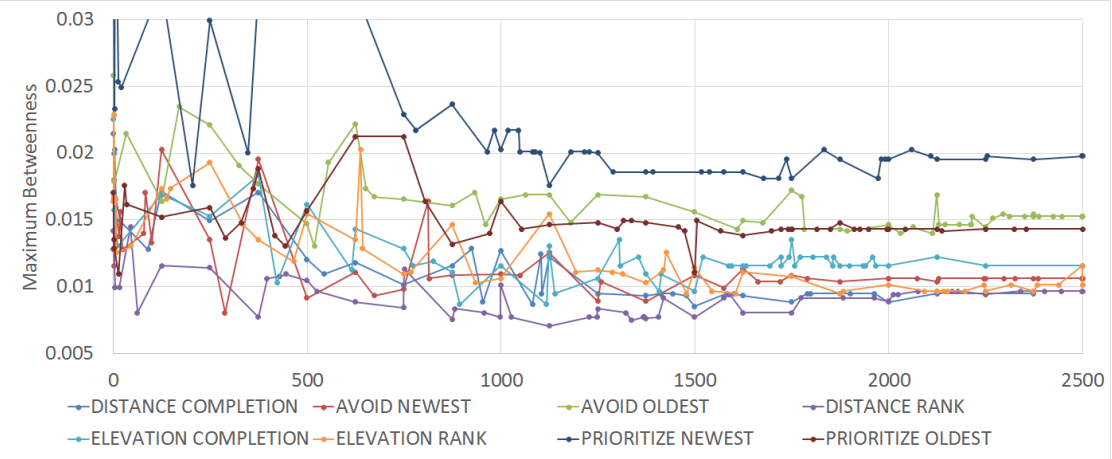


Figure 7.22: Progression of Maximum Betweenness Centrality

problem class under consideration, see Figure 7.1, a lower maximum value is beneficial. The maximum betweenness centrality of a single node is an indication of a central node, presumably toward the downstream ends of the network, through which most flow is routed before reaching the outlet. It appears therefore that the distance rank heuristic and the unaltered tree growing algorithm are more successful in avoiding such a central node. In contrast, the the relatively high values that prioritizing newest edge had on all the betweenness centrality parameters, combined with the effect it had on distance rank, account for its significant increase in eventual solution cost.

In summary, the most significant effects produced by the proposed heuristic factors were that of the distance rank heuristic and prioritizing the newest edge for selection during layout construction. The proposed factors have been shown to influence their respective layout parameters to the desired degree and the effect that the change in parameter has on the final layout could be explained.

7.5 Summary and Conclusions

The key focus of this chapter was to investigate the changes in sewer network layout parameters as the solution develops during execution of the optimization algorithm. The aim is to identify development patterns and to use this knowledge to positively influence layout creation decisions. This implied that the layout parameters must be enforceable during layout construction with what little information is available at the time, considering no hydraulic analysis or design has taken place and the layout is only partially constructed.

To this end 13 network layout parameters were proposed in Section 7.1. Graphic

representations of the parameter distributions were developed, where relevant and possible, as visual aids. The new algorithms of Chapter 5 were then used to solve each of the 3060 problems in the 156 problem classes of the problem library 20 times. The 13 proposed characteristic parameters were then monitored during each algorithm's execution and the changes captured. Spearman's correlation coefficient was used to establish correlation between the parameters and solution fitness, or cost. Many of the correlations showed promising results and three stood out as the best candidates for further investigation, namely distance rank, elevation rank and betweenness centrality.

It was observed that distance rank demonstrated a positive correlation with solution cost, i.e. as the distance rank decreases so does the solution cost. A heuristic influence factor was proposed to favour decisions that would reduce the distance rank in the network. Elevation rank also showed a positive correlation and again a heuristic influence factor was proposed which would favour decisions that would keep elevation rank to a minimum. For betweenness centrality, however, it is difficult to directly influence decisions which would have an impact on the parameter's value. Consequently, indirect heuristics were introduced which influence layout creation towards networks with the desired characteristics. That is, if the betweenness centrality should be increased, networks with a long main branches should be produced, while if it should be decreased many short branches must be favoured.

The proposed heuristic influence factors were applied to an example problem and their effects on each of the 13 proposed layout parameters were monitored. It was shown that the heuristic influence factors enhanced the desired characteristics. However, due to the interdependent nature of these problems, minor side-effects were present but deemed acceptable. Furthermore, it was shown the heuristic influence factors, when applied to the optimization algorithms, significantly improved the eventual solutions and the effect it had on its operating characteristic and consequent effect on solution cost were in line with the correlations.

Chapter 8

Augmenting Sewer Network Optimality

In section 7.1 certain network layout parameters were proposed and their correlations with solution fitness were observed. Heuristic influence factors that should enhance the construction of networks with advantageous layout characteristics were proposed in section 7.3. Their influence on the layout parameters was studied in section 7.4. In this section the proposed heuristic influence factors, and combinations thereof, are applied in the optimization of the 3060 library networks developed in Chapter 6 and their performance is evaluated.

8.1 Heuristic Combinations

The heuristic influence factors are to be applied individually as well as combined with others so as to influence multiple parameters simultaneously. A shorthand for each heuristic influence factor is introduced to avoid excessively long descriptions:

- Distance rank completion : C_d
- Elevation rank completion : C_e
- Distance rank during tree growing phase : T_d
- Elevation rank during tree growing phase : T_e
- Prioritize oldest element during tree growing phase : T_{po}
- Prioritize newest element during tree growing phase : T_{pn}
- Avoid oldest element during tree growing phase : T_{ao}

- Avoid newest element during tree growing phase : T_{an}

Notice that all heuristics which operate on the tree growing phase of the ACO algorithm, described in sections 4.3.3.2 through 4.3.3.5, start with T, while those that operate on the completion phase, described in section 4.3.3.6, start with C.

In addition to combing different heuristic influence factors, the modification of the ACO algorithm with respect to the contribution of heuristics, described in section 4.3.2.1, is also employed. Three states of heuristic influence are employed, these are: (i) The heuristic influences are applied throughout the entire lifetime of the algorithm with no reduction in impact, termed the no-fade scenario and denoted by NF. (ii) The heuristic influences are applied for a short duration, 40% of the maximum generations, and decreased linearly over the remaining application period until it reaches zero influence, termed the short-fade and denoted by SF. (iii) The heuristic influence is applied for the majority of the algorithm's life span, 80% of the maximum generations, and decreased linearly over the remaining application period until it reaches zero influence, termed the long-fade and denoted by LF.

The abbreviations defined above are employed throughout this section to describe the heuristic combinations and the ACO modification used in the optimization process. For example, $T_{ao}T_dC_e\_LF$ translates to: During the tree growing phase two heuristic influences are combined, namely avoid oldest element and use distance rank, while elevation rank is used during the completion phase and the long-fade modification to the ACO is used.

Heuristics are combined sensibly, e.g. heuristics that influence the way in which elements are prioritized for selection, namely T_{ao}, T_{an}, T_{po} and T_{pn}, would directly compete with one another if combined. Similarly, the two completion heuristics, C_d and C_e are never combined. However, during the tree growing phase the combination T_dT_e is compatible. Note that when no heuristic is employed, heuristic fading falls away. All rational combinations are included in the evaluation.

8.2 Results

The ACO algorithms were set up to run for 2500 generations with a population size of 20, an initial evaporation rate of 0.003 and an evaporation rate of 0.02 after 85% of the generation limit has been reached. All hydraulic constraints are as defined in section 3.4, and a γ_b value of 0.25 is used. Both algorithms, direct-edge and direct-node, were used to solve each of the 3060 problems 5 times for each of the heuristic combinations and results were averaged. Since it turned out that the direct-node algorithm only performed better than its edge counterpart for less

than 5% of problem classes, the averaged results presented below are only for the direct-edge algorithm. The complete set of results are included in Appendix B.

In Tables 8.1 through 8.8 the averaged results are shown for various network characteristics and heuristic combinations. There is a table for each terrain topograhy. In the "By Problem Scale" part of the table, values for the listed scales are averaged over flow rates. Similarly, in the "By Flow Rate" part, values for the listed flow rates are averaged over scales. In the "By Problem Scale and Flow Rate" values for the listed scale and flow rate are given. The tables contain five data columns:

Base Fitness shows the average fitness for the corresponding characteristics with no heuristic modifiers present, i.e. the fitness obtained by the unaltered algorithms of Chapter 5.

Best Ave Fitness % is shown as the average percentage decrease relative to the base fitness yielded by the best performing heuristic combination for the corresponding problem class.

Best Average Heuristic shows the heuristic combination which most often resulted in the best average fitness for problems of the shown class. For a heuristic combination to be considered it must occur more than once and cannot be tied for occurrences with more than one other heuristic combination. In the event of a tie the least complex heuristic combination is selected, i.e. $T_dC_e\_NF$ would be selected over $T_{ao}T_dT_eC_e\_LF$. If no such heuristic exists, i.e. there is no clear winning heuristic combination for problems of this class, "-" is shown.

Best Fitness % shows the average percentage decrease relative to the base fitness of the best solutions for each problem class.

Best Heuristic shows the heuristic combination which was most often present in the individual best performing solution, subject to the same constraints as the Best Average Heuristic column.

8.2.1 Bilinear Topography

Table 8.1 shows the summarized average results for all problem classes with a bilinear topography. The results of the bilinear topography by problem scale show that applying the $T_{an}T_eT_dC_d\_NF$ heuristic combination leads to approximately a 4% reduction in solution cost, on average, while the individual best solution obtained by all combinations of heuristics reduced the solution cost by 5% to 6.7%. Interestingly the impact of the heuristic combinations does not significantly

Table 8.1: Results of Bilinear Topography

Vertices	Edge Length	Flow Rate	Base Fitness	Best Ave Fitness (%)	Best Ave Heuristic	Best Fitness (%)	Best Heuristic
			By Problem Scale				
50	100		25 092 387.48	4.55	$T_{an}T_eT_dC_d\_NF$	5.94	$T_{ao}T_eT_dC_e\_LF$
50	250		80 482 172.49	3.32	$T_{an}T_eT_dC_d\_NF$	5.05	$T_dC_e\_NF$
100	100		81 537 616.45	4.07	$T_{an}T_eT_dC_d\_NF$	6.69	$T_{ao}T_eT_dC_e\_LF$
		Flow Rate			By Flow Rate		
		Low	15 790 348.89	1.37	$T_{an}T_eT_dC_d\_NF$	2.42	$T_dC_e\_NF$
		Medium	61 185 137.25	3.90	$T_{an}T_eT_dC_d\_NF$	4.82	$T_dC_e\_NF$
		High	113 711 574.51	7.00	$T_{an}T_eT_dC_d\_NF$	10.91	$T_dC_e\_NF$
Vertices	Edge Length	Flow Rate			By Problem Scale and Flow Rate		
50	100	Low	6 848 862.16	0.10	$T_{an}T_eT_dC_d\_NF$	0.14	–
50	250	Low	18 959 808.66	0.45	$T_{an}T_eT_dC_d\_NF$	0.91	–
100	100	Low	21 562 375.84	3.55	–	6.22	–
50	100	Medium	18 938 143.53	4.62	$T_{an}T_eT_dC_d\_NF$	4.81	–
50	250	Medium	54 826 312.34	3.66	–	4.94	–
100	100	Medium	109 790 955.88	3.41	–	4.70	–
50	100	High	49 490 156.74	8.92	–	12.88	–
50	250	High	167 660 396.48	5.85	–	9.32	–
100	100	High	129 120 468.20	5.84	–	10.36	–

change with problem scale. The heuristic combination which performed the best on average is very similar to that of the heuristic combination which performed the best for the two problem classes with an average edge length of 100m. However, the average best performing algorithm avoids the newest edge of the solution, while the individual best solutions avoid the oldest and the heuristic influence is reduced with the long fade strategy. Furthermore, the algorithms which perform the best on average use the distance rank completion strategy, while those with the individual best results use the elevation rank completion strategy.

When observing the effect of heuristic influence factors by flow rate it is clear that the heuristic combinations have more impact as the flow rate increases. In contrast to the scale averages there are no similarities between the best performing heuristic combination on average and the individual best solution. Furthermore, the same combination of heuristics performed the best for each of the flow rates. The difference in completion phase strategy of the average best, namely distance rank, and the individual best, namely elevation rank, is again observed.

When observing problem classes split across both scale and flow rate the increased effectiveness of the heuristic combinations with flow rate is again observed, but no clearly winning heuristic combination emerges. However, there is significant motivation for applying heuristic combinations since up to a 12.88% improvement from the base cost is seen.

It should be noted that the disparity between the individual best solution's heuristic combinations within a problem class and the heuristic combination which performed the best on average can be attributed to the fact that the algorithms were not carefully calibrated for each individual problem. Instead, a combination of ACO algorithm parameters emerged which performed well, although not optimally, for all problems. With more careful calibration during application of the best performing heuristics it should be possible to improve the consistency of the algorithm to more often result in the percentage decreases seen for the individual best solutions. This is true for all topographies and problem classes which were investigated.

8.2.2 Hill Topography

Table 8.2 shows the averaged results for all problem classes with the hill topography.

It can immediately be observed that for this topography a clearly winning heuristic combination emerges when observing the results more fine grained by scale and flow rate. Here the same combination of heuristics performed the best on average and single best. However, this is not the case in the scale alone and flow rate alone evaluations.

For both problem scale and flow rate the effect of the heuristics tends to increase as the scale/flow rate increases. The results by problem scale show that elevation completion was used by both the average and the best performing combination. However, for flow rates the individual best solutions all used distance rank completion and applied the distance rank heuristic for the entire duration of the algorithm. This makes sense since distance rank yields more, shorter branches, resulting in less cumulative flow in the downstream ends, in turn resulting in smaller diameters.

When observing the data more finely grained it is clear that distance ranking during the tree growing phase has a positive impact on the algorithm's performance. Interestingly, in all cases the heuristic influences are reduced over time with the long fade strategy. This implies that the algorithm's tree growing phase finds substantial benefit from the heuristic influence factors, but as it learns the pheromones eventually take over and find better solutions in the absence of heuristic influences.

Table 8.2: Results of Hill Topography

		Base Fitness	Best Ave Fitness (%)	Best Ave Heuristic	Best Fitness (%)	Best Heuristic
Vertices	Edge Length	By Problem Scale				
50	100	37 743 433.34	2.29	$T_dC_e\_LF$	3.43	$T_dC_e\_LF$
50	250	108 118 151.97	2.02	$T_{an}T_eT_dC_e\_NF$	4.99	$T_dC_e\_LF$
100	100	114 509 249.45	4.39	$T_{an}T_eT_dC_e\_NF$	6.70	$T_dC_e\_LF$
Flow Rate		By Flow Rate				
Low		20 130 595.17	2.23	$T_{an}T_eT_dC_e\_NF$	3.30	$T_dC_d\_NF$
Medium		61 418 795.28	2.43	$T_{an}T_eT_dC_e\_NF$	5.04	$T_dC_d\_NF$
High		186 860 468.66	3.99	$T_{an}T_eT_dC_e\_NF$	6.78	$T_dC_d\_NF$
Vertices	Edge Length Flow Rate	By Problem Scale and Flow Rate				
50	100 Low	8 331 132.59	0.29	$T_dC_e\_LF$	0.60	$T_dC_e\_LF$
50	250 Low	23 813 560.76	0.56	$T_dC_e\_LF$	1.15	$T_dC_e\_LF$
100	100 Low	28 247 092.16	5.83	$T_dC_e\_LF$	8.16	$T_dC_e\_LF$
50	100 Medium	18 509 402.15	4.56	$T_dC_e\_LF$	6.41	$T_dC_e\_LF$
50	250 Medium	53 199 674.63	1.13	$T_dC_e\_LF$	5.33	$T_dC_e\_LF$
100	100 Medium	112 547 309.07	1.59	$T_dC_e\_LF$	3.37	$T_dC_e\_LF$
50	100 High	86 389 765.29	2.01	$T_dC_e\_LF$	3.27	$T_dC_e\_LF$
50	250 High	247 341 220.52	4.35	$T_dC_e\_LF$	8.48	$T_dC_e\_LF$
100	100 High	246 845 395.94	6.42	$T_dC_e\_LF$	9.50	$T_dC_e\_LF$

8.2.3 Bowl Topography

Table 8.3 shows the averaged results for all problem classes with a bowl topography. For this topography the heuristics yielded a reasonable reduction in solution cost, with increased effectiveness for large problems or when flow rates are high. However, no winning heuristic combination emerged. The only observation which can be made is that for small problems with low flow rates the best choice is to enforce distance rank with elevation completion. This combination also performed the best for problems with high flow rates when they are observed in isolation, making it an attractive choice for problems of this class.

8.2.4 Concave Topography

Table 8.4 shows the averaged results for all problem classes with a concave topography. The concave topography is interesting in that no heuristic performed

Table 8.3: Results of Bowl Topography

			Base Fitness	Best Ave Fitness (%)	Best Ave Heuristic	Best Fitness (%)	Best Heuristic
Vertices	Edge Length		By Problem Scale				
50	100		17 603 535.78	1.57	–	2.55	$T_{po}C_d\_NF$
50	250		45 575 080.83	1.95	$T_dC_e\_NF$	3.11	$T_eC_d\_SF$
100	100		40 195 372.06	2.53	$T_dC_e\_NF$	4.27	$T_{po}C_d\_NF$
Flow Rate			By Flow Rate				
	Low		12 454 102.88	0.17	$T_dC_e\_NF$	0.32	–
	Medium		27 462 776.66	2.52	–	3.98	–
	High		66 364 826.26	3.46	$T_dC_e\_NF$	5.79	$T_dC_e\_NF$
Vertices	Edge Length	Flow Rate	By Problem Scale and Flow Rate				
50	100	Low	5 954 330.46	0.11	$T_dC_e\_NF$	0.14	–
50	250	Low	17 212 573.76	0.22	$T_dC_e\_NF$	0.44	–
100	100	Low	14 195 404.42	0.19	$T_dC_e\_NF$	0.38	–
50	100	Medium	11 284 847.27	0.73	$T_dC_e\_NF$	1.13	–
50	250	Medium	31 125 901.14	1.57	–	2.92	–
100	100	Medium	39 977 581.56	5.26	–	7.89	–
50	100	High	35 571 429.60	3.87	–	6.39	–
50	250	High	88 386 767.59	4.05	–	5.96	–
100	100	High	79 522 009.25	1.97	–	4.65	–

the best for problems when observed finely grained by scale and flow rate. This implies that the best heuristic combination is highly dependent on the specific characteristics of the problem. However, the positive impact of the heuristic combinations increases both with problem scale and flow rate. When observing the results by problem scale only, or by flow rate only, good heuristic combinations emerge. By problem scale it is clear that elevation rank is an important factor, although the short-fade strategy means that the heuristics are not active for a long time. By flow rate shows that distance rank plays a more important role than the elevation rank. It is interesting to note that this is the only topography where avoiding the newest edge is prominent, except for the simplex noise topography where it is present by problem scale. Avoiding the newest edge or node during the tree growing phase results in more, shorter branches being produced. Referring to Section 7.4, avoiding the newest edge resulted in values roughly in the middle of the upper and lower extremes for all thirteen layout parameters. Perhaps this characteristic of not favouring any specific layout parameter is important in the

Table 8.4: Results of Concave Topography

Vertices	Edge Length	Flow Rate	Base Fitness	Best Ave Fitness (%)	Best Ave Heuristic	Best Fitness (%)	Best Heuristic
			By Problem Scale				
50	100		34 444 030.79	1.28	$T_{an}C_e\_SF$	2.13	$T_{an}T_eC_e\_SF$
50	250		94 603 162.25	2.73	–	3.48	$T_{an}T_eC_e\_SF$
100	100		72 156 812.56	3.71	$T_{an}C_e\_SF$	7.25	$T_{an}T_eC_e\_SF$
Flow Rate					By Flow Rate		
		Low	17 593 963.72	1.72	$T_{an}T_eT_dC_d\_NF$	2.93	–
		Medium	41 915 800.66	2.78	$T_{an}T_eT_dC_d\_NF$	4.78	–
		High	150 386 419.79	3.16	$T_{an}T_eT_dC_d\_NF$	4.88	–
Vertices	Edge Length	Flow Rate			By Problem Scale and Flow Rate		
50	100	Low	7 269 871.07	0.42	–	0.68	–
50	250	Low	23 029 923.85	0.79	–	1.19	–
100	100	Low	22 482 096.24	3.95	–	6.91	–
50	100	Medium	16 654 460.89	2.41	–	3.75	–
50	250	Medium	43 605 429.80	1.87	–	2.61	–
100	100	Medium	65 487 511.30	4.08	–	8.00	–
50	100	High	79 407 760.40	1.02	–	1.97	–
50	250	High	217 174 133.09	5.54	–	6.63	–
100	100	High	156 672 838.94	2.81	–	6.64	–

concave topography problems.

8.2.5 Convex Topography

Table 8.5 shows the averaged results for all problem classes with a convex topography. The results reveal observable patterns. Firstly, the effect of the heuristics increases both with problem scale and flow rate. Secondly, on average, the $T_dC_e\_NF$ heuristic combination performs the best for all cases when both problem class and flow rate are considered. However, the individual best performers are not obvious and clearly outliers in the solution space were found by other combinations. Again, it may be possible to identify the individual best performers and carefully calibrate the algorithm to perform more consistently. When observing the results by problem scale it is apparent that distance rank during the tree growing phase combined with elevation rank completion is the winner again.

Table 8.5: Results of Convex Topography

			Base Fitness	Best Ave Fitness (%)	Best Ave Heuristic	Best Fitness (%)	Best Heuristic
Vertices	Edge Length			By Problem Scale			
50	100		23 040 213.24	0.90	$T_dC_e\_NF$	2.28	$T_dC_e\_LF$
50	250		59 284 064.14	4.11	–	6.55	$T_dC_e\_LF$
100	100		65 013 150.50	4.02	$T_dC_e\_NF$	7.29	$T_dC_e\_LF$
Flow Rate				By Flow Rate			
		Low	12 547 917.08	0.29	–	0.47	$T_dC_d\_LF$
		Medium	41 134 683.03	3.73	$T_{an}T_eT_dC_d\_LF$	5.77	$T_dC_d\_LF$
		High	97 235 037.43	5.14	$T_dC_e\_NF$	10.20	$T_dC_d\_LF$
Vertices	Edge Length	Flow Rate		By Problem Scale and Flow Rate			
50	100	Low	6 171 073.16	0.16	$T_dC_e\_NF$	0.18	$T_dC_e\_LF$
50	250	Low	16 813 306.28	0.24	$T_dC_e\_NF$	0.39	–
100	100	Low	14 659 371.80	0.47	$T_dC_e\_NF$	0.85	–
50	100	Medium	10 410 711.78	0.80	$T_dC_e\_NF$	1.61	–
50	250	Medium	37 222 320.60	3.91	$T_dC_e\_NF$	4.97	–
100	100	Medium	75 771 016.70	6.49	$T_dC_e\_NF$	10.73	–
50	100	High	52 538 854.79	1.76	$T_dC_e\_NF$	5.06	–
50	250	High	123 816 565.54	8.17	$T_dC_e\_NF$	14.29	–
100	100	High	124 407 019.24	5.65	$T_dC_e\_NF$	11.78	–

8.2.6 Flat Topography

Table 8.6 shows the averaged results for problems with a flat topography. For the flat topography, heuristic influences show better performance by increasing problem scale. However, by flow rates, the medium flow rate shows the best improvement and less of an improvement is observed for high flow rates. This is also seen in the results by both problem scale and flow rate. Note that the combination of 100 vertices and high flow rates were excluded from this topography, as stated in Chapter 6, since feasible solutions could not be obtained. Interestingly the best performing heuristic combination for the individual best solutions by problem scale, as well as the low and high flow rate problems, is the unaltered state of the algorithms. This implies that for a flat topology very little is gained, and in fact some lost, when applying the heuristic combinations to flat topography problems.

Table 8.6: Results of Flat Topography

			Base Fitness	Best Ave Fitness (%)	Best Ave Heuristic	Best Fitness (%)	Best Heuristic
Vertices	Edge Length		By Problem Scale				
50	100		29 376 716.04	1.58	–	2.13	C_d_NF
50	250		67 566 089.90	1.09	–	3.87	C_d_NF
100	100		49 437 235.68	2.89	–	4.48	C_d_NF
Flow Rate			By Flow Rate				
	Low		14 514 994.59	0.27	–	0.94	C_d_NF
	Medium		43 563 120.32	3.10	–	5.79	–
	High		107 734 272.24	1.82	–	3.38	C_d_NF
Vertices	Edge Length	Flow Rate	By Problem Scale and Flow Rate				
50	100	Low	6 925 089.61	0.00	–	0.07	–
50	250	Low	19 704 467.38	0.30	–	0.73	–
100	100	Low	16 915 426.76	0.52	–	2.03	–
50	100	Medium	10 712 776.74	2.44	–	3.64	–
50	250	Medium	38 017 539.62	1.61	–	6.79	–
100	100	Medium	81 959 044.59	5.25	–	6.93	–
50	100	High	70 492 281.77	2.29	–	2.68	–
50	250	High	144 976 262.70	1.36	–	4.08	–

8.2.7 Simplex Noise Topography

Table 8.7 shows the averaged results for the problem classes with topographies generated using simplex noise (Perlin, 2001). The simplex noise topography has no clear best performing heuristic combination on average, which may be expected due to its random nature. The only heuristic combination that stands out is the individual best solutions by problem scale where $T_{an}C_d\_LF$ is present for all three cases. It can be observed that the effect the heuristic combinations has an increasingly positive effect by both increasing problem scale and flow rate. However, which heuristic combination to apply for the best results depends heavily on the characteristics of the specific problem, which has been seen in most other topographies too. Consequently for a random topography such as this it is very difficult to predict which combination will be best in general.

Table 8.7: Results of Simplex Noise Topography

Vertices	Edge Length	Flow Rate	Base Fitness	Best Ave Fitness (%)	Best Ave Heuristic	Best Fitness (%)	Best Heuristic
			By Problem Scale				
50	100		19 764 723.62	2.51	–	2.68	$T_{an}C_d\_LF$
50	250		55 027 956.25	5.14	–	7.52	$T_{an}C_d\_LF$
100	100		38 438 746.03	1.66	–	5.61	$T_{an}C_d\_LF$
	Flow Rate		By Flow Rate				
		Low	16 194 303.75	0.46	–	1.15	–
		Medium	32 890 556.77	2.81	–	5.69	–
		High	77 000 475.06	8.24	–	10.65	–
Vertices	Edge Length	Flow Rate	By Problem Scale and Flow Rate				
50	100	Low	6 988 760.35	0.10	–	0.10	–
50	250	Low	20 731 868.44	0.43	–	0.60	–
100	100	Low	20 862 282.46	0.85	–	2.76	–
50	100	Medium	9 954 626.34	0.57	–	0.60	–
50	250	Medium	32 701 834.37	5.40	–	8.00	–
100	100	Medium	56 015 209.60	2.47	–	8.46	–
50	100	High	42 350 784.17	6.88	–	7.35	–
50	250	High	111 650 165.95	9.60	–	13.96	–

8.2.8 All Topographies

Table 8.8 shows the combined averaged results for all topography functions. Interesting patterns emerge when the results are viewed at this level. It is observed, as can be expected from the results of individual topographies, that the heuristic combinations have an increasingly positive effect with increasing problem scale and flow rates. Most notably, the average best performing heuristic combination and the one which most often produced the individual best solution are the same, namely $T_dC_e\_NF$, when the results are averaged by both problem scale and flow rate, excluding a few cases where the best solutions were produced with the slightly different long-fade strategy. This suggests that for all problem classes, regardless of network characteristics and when no specific information about a problem is available, this combination of heuristics should lead to improved solutions.

The heuristic influences have greater impact for larger problems with higher flow rates. These problems are also more difficult for the optimization algorithms to solve and will naturally be the more expensive solutions. This means that

Table 8.8: Combined Results of all Topographies

			Base Fitness	Best Ave Fitness (%)	Best Ave Heuristic	Best Fitness (%)	Best Heuristic
Vertices	Edge Length		By Problem Scale				
50	100		27 089 990.17	2.01	–	3.02	$T_dC_e\_LF$
50	250		76 282 432.30	2.78	–	4.59	$T_dC_e\_LF$
100	100		78 031 980.81	3.60	–	6.39	$T_dC_e\_LF$
Flow Rate			By Flow Rate				
		Low	15 831 154.44	1.01	$T_{an}C_e\_NF$	1.71	$T_{po}T_eT_dC_d\_LF$
		Medium	44 458 463.27	3.05	–	4.92	$T_{po}T_eT_dC_d\_LF$
		High	121 884 121.37	4.35	$T_dC_e\_NF$	7.38	$T_dC_e\_NF$
Vertices	Edge Length	Flow Rate	By Problem Scale and Flow Rate				
50	100	Low	6 929 603.76	0.19	$T_dC_e\_NF$	0.32	$T_dC_e\_NF$
50	250	Low	20 077 124.90	0.42	$T_dC_e\_NF$	0.77	$T_dC_e\_NF$
100	100	Low	20 486 734.66	2.41	$T_dC_e\_NF$	4.06	$T_dC_e\_NF$
50	100	Medium	14 468 325.45	2.41	$T_dC_e\_NF$	3.25	$T_dC_e\_NF$
50	250	Medium	42 599 226.64	2.67	$T_dC_e\_NF$	4.53	$T_dC_e\_LF$
100	100	Medium	76 307 837.72	4.07	$T_dC_e\_NF$	6.98	$T_dC_e\_NF$
50	100	High	59 872 041.30	3.43	$T_dC_e\_NF$	5.48	$T_dC_e\_LF$
50	250	High	166 170 945.37	5.24	$T_dC_e\_NF$	8.48	$T_dC_e\_NF$
100	100	High	140 594 113.89	4.37	$T_dC_e\_NF$	8.22	$T_dC_e\_LF$

the heuristic influences provide the most significant improvement in network cost when it is both the most needed and the most effective. Furthermore, all problems regardless of size and scale were improved, albeit some only slightly, making a strong case for the incorporation of heuristic influences into sewer network layout optimization algorithms.

8.2.9 A Note on the Feasible Solution Space

It has been shown that the application of heuristic combinations are effective in reducing final solution costs, especially for large problems with high flow rates. However, a reduction in solution cost is not the only benefit of applying heuristics to influence decisions during the ACO process. When network characteristics are adverse, it is possible that infeasible solutions may arise, most often due to violation of the maximum cover depth constraint. As a result the algorithm may spend a significant amount of time in the infeasible solution space before finding

the first feasible solution, assuming it is able to find feasible solutions at all.

Five problems for which the basic ACO algorithm obtained infeasible solutions were randomly selected. Table 8.9 lists the cost of the best solution they obtained after the indicated percentage of the maximum number of generations has been completed. Any value with an exponent greater than 6 is in the infeasible solution space.

Table 8.9: Base Performance by Generation

Problem Class	Cost at % of Generation Limit 0%	25%	50%	75%	100%
$HI\_V50.E250.EV3.FH(1)$	$8.040E11$	$1.771E11$	$4.870E8$	$3.047E6$	$3.037E6$
$CC\_V50.E250.EV1.FH(2)$	$1.458E11$	$9.448E10$	$6.999E10$	$2.575E6$	$2.571E6$
$BI\_V100.E100.EV3.FH(3)$	$2.951E6$	$3.664E7$	$3.136E8$	$1.608E6$	$1.514E6$
$RA\_V50.E250.FH(4)$	$1.384E11$	$1.166E11$	$1.072E10$	$1.121E6$	$1.117E6$
$FL\_V100.E100.EV1.FH(5)$	$1.547E9$	$5.369E8$	$1.298E7$	$1.469E6$	$1.454E6$

Notice that for all five problems except one the first feasible solutions were obtained between 50% and 75% of the maximum number of generations. This is clearly an adverse scenario as the algorithm has spent more than half of its execution time in infeasible space. Applying heuristic combinations may help the algorithm move away from the infeasible region more quickly and consequently improve both computer runtime and the eventual solution.

Table 8.10 shows the same five problems optimzed using the best performing heuristic combination on average for that problem's class. The values listed are percentage improvement or deterioration from the base algorithm's values for the same percentage of run completion. When a percentage improvement of 99% is observed this means that a feasible solution has been found, while at this same stage the base algorithm had not yet found any feasible solutions.

High percentage improvements and deteriorations in solution fitness are observed in the earlier stages. This is an interesting result, since it implies that a heuristic combination which performs well overall might not perform well throughout all phases of the algorithm's execution. It also implies that a feasible solution region may be very near an infeasible region and may even be isolated from other feasible regions. It is therefore possible that the base algorithm may miss such a feasible region altogether without the aid of heuristic influences. Regardless of the observed negative impact in the early phases for some problems, all the problems benefited significantly from the application of the heuristic influence factors.

Table 8.10: Best Heuristic Performance by Generation

Problem Class	% Cost difference at % of Generation Limit 0%	25%	50%	75%	100%	Best Ave Heuristic
(1)	49.93	−34.61	−2295.98	3.53	3.52	$T_eC_d\_LF$
(2)	69.03	31.66	99.84	8.46	8.32	$T_{ao}T_eC_e\_LF$
(3)	−95568.41	−4815.45	79.02	−1.33	5.06	$T_{ao}T_eT_dC_e\_LF$
(4)	56.54	39.39	99.97	9.92	9.60	$T_{an}T_eT_dC_e\_LF$
(5)	57.53	66.18	−892.21	4.76	4.87	$T_dT_eC_e\_LF$

8.3 Summary and Conclusions

In this chapter the focus was on enhancing the optimality of sewer networks by combining heuristic influence factors during the optimization process. All possible heuristic combinations were applied in 5 optimization runs of each of the 3060 problems of the problem library and the results were averaged. The results were analysed by terrain topography, problem scale, flow rates and a combination thereof. In all cases the heuristic factors lead to improved solutions. For some topographies, such as hill and convex topographies, specific combinations emerged as best performers. However, for others such as the flat and simplex noise topographies, no clear winner emerged. It was observed that, in most cases, the beneficial impact of the heuristic factors increased with problem scale and flow rate.This is an encouraging result since the heuristic influence factors and their combinations have the most benefit for the most expensive networks.

When observing the averaged results for all topographies combined, the same pattern of increased benefit with increased problem scale and flow rate emerged. Furthermore, a clear best performer for the most problems emerged, namely applying the distance rank influence factor to the tree growing phase while using elevation ranks to perform the completion phase selections. This is an important result. However, since this combination did not perform the best for all problem classes, even better results may be obtained if a problem's class is known and more specialised heuristic combinations are used.

It was also shown that the application of heuristic influences not only improves the eventual results of the optimization, it also has a beneficial effect on the algorithm's ability to locate the feasible search space.

A number of interesting and previously unknown characteristics of sewer network layouts, that further the quest for optimality were identified.

Chapter 9

Conclusion and Recommendations

This dissertation set out to explore sewer network design as an optimization problem. The intent was to improve on the current state of the art and to gain a better understanding of what makes a network a better solution for a given class of problem. New algorithms were proposed for both the network layout development and the hydraulic optimization which performed better than existing algorithms for all problems tested. Furthermore, using newly uncovered understanding of how network characteristics change during the course of optimization, heuristic influence factors were proposed which further improved the results for a broad range of problem classes.

9.1 Conclusions

Chapter 2 described sewer network analysis and design as an optimization problem that requires the simultaneous solution of three sub-problems, namely the spatial layout, the flow-directional layout and the hydraulics of the network. It was postulated that spatial layout takes place a-priori due to practical considerations such as existing infrastructure and the functional requirements of the network. Consequently, layout optimization algorithms are concerned only with the flow-directional design of the network. The constraints to which the layout are subject were introduced and motivated. Most notably, only branched networks were considered. This is a common assumption in sewer network optimisation which does not significantly limit the value of proposed algorithms. The hydraulic design of sewer networks is also subject to a number of constraints which were described in detail. Contributor hydrograph theory was selected as the flow rate calculation model despite its shortcomings. Since the accurate determination of design flow rates is the most import requirement for hydraulic optimization, which

van Heerden (2014) showed that contributor hydrograph theory is capable of, its use was deemed justified.

In Chapter 2 the so-called Fitness Warping phenomenon of network optimization was defined for the first time. It refers to the adverse scenario which may arise when a good layout is selected by the optimization algorithm, but the accompanying selection of hydraulic parameters is poor, or the other way around, resulting in an overall poor solution fitness. To avoid this phenomenon it was decided to combine two individual optimization algorithms, one for each sub-problem. Since hydraulic optimization must be carried out for each candidate layout, the computational complexity of this approach can be excessive. To counteract this, an extremely efficient hydraulic optimization algorithm, called the HOMS algorithm, was developed and shown to yield good results, as described in Chapter 3.

Chapter 4 addresses the flow-directional layout design problem. It was stated that the problem is a variant of the degree-constrained Minimum Spanning Tree (d-MST) problem, a well known NP-Hard problem in graph theory. Drawing from the literature on d-MST problems it was concluded that Ant Colony Optimization (ACO) was the best candidate metaheuristic. ACO was briefly described, along with modifications to the algorithm that were employed. It was uncovered that strategies which operate on either the nodes, i.e. manholes, or the edges, i.e. pipes, of a network perform differently. Drawing inspiration from the work of Moeini and Afshar (2012), who combined a tree-growing algorithm with a Max-Min Ant System to simultaneously create layouts and select diameters, two node-based and two-edge based layout creation algorithms were developed. Two of these algorithms construct a permutation of identities, rather than operating directly on the network itself, since Merkle *et al.* (2002) had achieved success using this approach in resource-constrained project scheduling. Some limitations inherent to the layout creation algorithms, due to their internal selection models, were identified and described.

Chapter 5 describes the combined hydraulic and layout optimization algorithms, starting from a given problem's base layout through to eventual convergence. Finding benchmark problems to evaluate the performance of the four proposed algorithms was problematic. Three new example problems, with varying topography and scale, were developed to reveal performance differences of the proposed layout selection strategies. The MMAS algorithm of (Moeini and Afshar, 2012) was used as benchmark. The newly proposed algorithms performed significantly better than Moeini and Afshar (2012) for the larger, more difficult problems and the MMAS algorithm showed severe fitness warping in the early trial solutions. A variation in performance of the proposed algorithms over the three examples indicated that differences in the heuristics they employ are important and should be investigated further.

A thorough study of why certain algorithms perform better for certain conditions require the analysis of a large number of problem instances. Drawing inspiration from other fields of optimization, where standardised libraries exist that contain problems with controlled characteristics, techniques were developed and implemented to generate a library of 3060 sewer network problems. The number represents 20 different problems in each of 153 classes with different characteristics, as described in Chapter 6.

In Chapter 7 an investigation into the nature of sewer network layouts was made. Thirteen layout characteristic parameters were proposed. These include parameters that measure hydraulic characteristics, such as slopes, and characteristics that measure graph theoretical parameters, such as betweenness centrality. It was hypothesised that if the layout creation could be guided to make selections which favour beneficial layout parameters, the eventual solutions would improve. The proposed optimization algorithms of Chapter 5 were used to solve each of the 3060 library problems multiple times in order to generate sufficient data to establish correlations between the proposed layout parameters and solution fitness, or cost. While most of the parameters demonstrated some degree of correlation, three were selected for further investigation, namely distance rank, elevation rank and betweenness centrality. These parameters were used to define heuristic influence factors, η_{ij} in Equation 4.1, which would augment the decision making of the ants.

The heuristic influence factors, individually and in combinations thereof, were incorporated into the ACO algorithms. Extensive testing using the problem library showed that different heuristic combinations perform better for different problem classes, which is consistent with the finding of Chapter 5. Furthermore, the effectiveness of heuristic combinations in reducing solution cost increased as the problem scale and total flow rate in the networks increased. A heuristic combination that performed best on average for all topographies was identified, namely using distance ranking, which encourages the formation of shorter branches, in the tree growing phase and elevation ranking during the completion phase. This combination is a safe and effective strategy for all networks. Furthermore it was shown that the heuristics facilitated the identification of the feasible solution space.

While the heuristic influence factors were not as uniform in performance expected, it was shown that their application certainly improved the eventual results of the optimization algorithms and that their inclusion in future sewer network optimization algorithms is warranted. Furthermore it was shown that by studying the development of optimized solutions, improved understanding of the nature of optimality for specific problem classes may be gained and exploited to enhance solutions even further.

9.2 Recommendations

Novel sewer network layout and hydraulic optimization algorithms were proposed and their behaviour was evaluated. It is the nature of research to reveal areas where further investigation is required. In this section potential areas which can be explored by future researchers are listed. To structure the section, recommendations are grouped into four parts, namely hydraulic optimization, layout optimization, problem library and understanding optimality. A brief discussion of each area where further investigation may be beneficial is presented and some approaches to overcome the problems are suggested.

9.2.1 Hydraulic Optimization

In this section, research that can be carried out to improve the HOMS algorithm of Chapter 3 is described.

9.2.1.1 Remove all assumptions and restrictions of the hydraulic model

In this investigation, and those before it, many simplifying restrictions were made to reduce the complexity of the hydraulic model. Only branched networks are considered, thereby removing all cycles from the network. Furthermore divergence structures, pumping stations and rising mains are excluded from the design. Each of these assumptions present a different challenge when included.

Branched network layout constraint: While this constraint is active during layout creation, the end result influences the hydraulic model as well. In the HOMS algorithm the branched network layout characteristic is exploited to perform a topological sort of the layout graph. Topological ordering is possible if and only if there are no directed cycles in the network. Consequently, if cycles are introduced the routing of flow and computation of flow rates will need to be handled differently. A possible solution is to introduce a hydraulic model capable of handling cycles, such as the fully dynamic wave model, if the computational complexity of the model can be accommodated in the optimization process.

Divergence structures: Removing the single out degree constraint from the layout creation algorithm will result in divergence structures at some manholes where multiple outlet pipes are present. To determine the flow rate for each outlet pipe the hydraulic analysis model must be expanded.

Pumping stations and rising mains: If these are included, the hydraulic analysis of the pumping stations and rising mains must be incorporated into the analysis procedure. However, their introduction essentially split the network into smaller sub-networks, one part flowing towards a pumping station and the other flowing

away from the pumping station. A possible solution is to apply the standard HOMS algorithm separately to each sub-network, with a pumping station acting as an outlet manhole the flow rate simply carried over to the succeeding sub-network.

9.2.1.2 Use the HOMS algorithm to seed a metaheuristic hydraulic optimization algorithm

The HOMS algorithm was developed with computational efficiency in mind while maintaining solution quality. Performance of the algorithm exceeded expectations since it found solutions that are accurate to within 2% of solutions obtained by sophisticated metaheuristic models. However, combining the HOMS algorithm with a metaheuristic algorithm, to create a hybrid hydraulic optimization algorithm, should yield an algorithm which performs exceedingly well. The most obvious application is to use the HOMS algorithm to limit the eligible set of diameters that the metaheuristic algorithm uses for each pipe to diameters around the result of the HOMS algorithm. This will significantly reduce the number of decisions available to the metaheuristic algorithm, thereby reducing the size of the search space,potentially improving the performance of the algorithm both in terms of convergence time and solution quality.

Alternatively, it may be possible to develop a hybrid algorithm which uses HOMS to select diameters and then uses its metaheuristic mechanism to select slopes. Since most hydraulic optimization algorithms are concerned with selecting diameter sizes and calculating the other parameters from hydraulic equations, this novel approach may prove to be an effective strategy.

9.2.1.3 Investigate the severity of full flow assumption for velocity

Due to the fitness warping phenomenon the hydraulic optimization algorithm's core constraint was computational efficiency while maintaining solution quality. To achieve this, inspiration was drawn from the dynamic programming algorithms of the past where, due to hardware constraints of the time, algorithms were necessarily computationally efficient by modern standard. An algorithm was presented, named HOMS, which relied on clever manipulation of the hydraulic design constraints and knowledge of the engineering space to increase diameters in order to reduce required slopes. The benefit of increasing a diameter is evaluated by a predefined parameter, γ_b. The parameter is essentially a required reduction in slope factor before accepting a diameter increment as beneficial.

An interesting discrepancy was uncovered during this investigation between the mathematical theory, and standard design practice in the engineering workspace.

It is standard practice in engineering design work to assume full flow conditions when determining flow velocities in pipes. In this case, increasing the diameter reduces the required slope to meet a specified velocity. However, if the assumption of full flow conditions is not made and one dimensional optimization algorithms are employed to solve the highly implicit equation the opposite is observed. The required slope to achieve a specific velocity within a pipe increased with a diameter increase. The reason for this discrepancy was identified. However, a full analysis of the consequence of this finding did not form part of this dissertation's scope, consequently no definitive recommendation can be made at this stage. However, this finding certainly warrants further investigation.

9.2.1.4 Compare the hydraulic optimization model to an experienced engineer

With optimization problems it is often difficult to determine exactly how well the algorithm performed. Most commonly, algorithms are compared to one another to gauge performance. However, it may be beneficial to apply the HOMS algorithm to case studies where designs have been carried out by experienced sewer network design engineers, comparing the cost of the solutions. Regardless of the performance outcome, the solutions should be compared to determine the differences and how these affected the eventual solution cost. Furthermore, practical considerations which design engineers take into account, which are currently not included the HOMS algorithm, may be included to good effect.

9.2.1.5 Investigate alternative objective functions

In this study the objective of all the algorithms was to minimize the capital investment cost of a network. However, other sewer network characteristics may also be minimized or maximized. Commonly in engineering, risk factors are proposed for designs which attempt to minimize the adverse consequences that a partial failure may cause. It would be interesting to incorporate sewer network risk factors into the hydraulic optimization. In so doing the resulting network should not only be cost efficient, but also more resilient to partial failure.

9.2.2 Layout Optimization

In this section areas of investigation that may improve understanding of layout optimization is proposed.

9.2.2.1 Alternative layout creation strategies

In Chapter 4, four layout creation strategies were presented. It was shown that their performance differed depending on the problem being solved. An investigation was conducted to explain why and when certain strategies perform better, but addtional layout creation strategies were not considered.

The strategies presented in this dissertation all rely on a specific tree growing algorithm. However, alternative tree growing algorithms exist, which exhibit different characteristics. Undertaking a thorough investigation of alternative tree growing algorithms may both improve the performance of the layout creation algorithm and further the understanding of optimal network layouts. The network layout parameters defined in Chapter 7 can be used to investigate the layouts produced by the different tree growing algorithms.

9.2.2.2 Investigate the degree-constrained minimum spanning tree problem

As mentioned in Chapter 7 the layout creation problem is a variant of the degree-constrained minimum spanning tree (d-MST) problem. While an initial investigation into d-MST algorithms was made, it may be useful to expand the scope of such an investigation. It is possible that useful graph theoretical models and techniques may be uncovered that will improve the current layout creation strategies and algorithms.

9.2.2.3 Further investigate Fitness Warping

The adverse effects of the fitness warping phenomenon, defined in Section 2.4, was demonstrated in Chapter 5. The research described here did not reveal the extent to which fitness warping negatively impacts an algorithm's overall performance, only that it does. Nor was an attempt made to mitigate its impact, instead it was completely avoided. By creating two similar algorithms, one with fitness warping present and another without, it would be possible to study exactly how the phenomenon affects the optimization process, which may be used to mitigate its impact. Furthermore, quantifying its adverse effects would not only serve to improve understanding of simultaneous sewer network optimization, but all optimization problems in which two sub-problems have to be optimized.

9.2.2.4 Compare the layout optimization model to an experienced engineer

See section 9.2.1.4 above. Exactly the same arguments and approach can be applied for the layout optimization.

9.2.2.5 Investigate alternative objective functions

See section 9.2.1.5 above. Exactly the same arguments and approach can be applied for the layout optimization.

9.2.2.6 Implement and compare different metaheuristics

In the research described here, Ant Colony Optimization (ACO) algorithms were used to perform the layout creation based on the results of Bui and Zrncic (2006) who found that it performed very well for degree-constrained minimum spanning tree problems. However, a thorough comparison of alternative metaheuristic algorithms is warranted. Additionally, variations of the algorithms, for example the rebirthing genetic algorithm or max-min ant system algorithm, should be investigated.

9.2.3 Standard Problem Library

In Chapter 6 a software model was developed to produce sewer network instances with controlled characteristics. However, no attempt was made to propose a definitive standard benchmark problem library, examples of which are found in other fields of optimization. This section identifies areas in which the software model can be improved in order to deliver a definitive tool capable of producing networks which allow accurate and direct comparison of optimization algorithms.

9.2.3.1 Expand the controllable characteristics

The software model presented in Chapter 6 allowed control of certain characteristics of a sewer network instance. However, there are some notable shortcomings in the current implementation.

A characteristic should be introduced which allows the user to more narrowly define the shape in plan of the generated network. Currently, networks shapes are generated entirely at random, which means that two networks with the same characteristics can potentially differ significantly from one another. Strongly coupled to the problem of shape is the introduction of the outlet node. Allowing the user to specify the location of the outlet node relative to the desired shape will also be beneficial. The position of the network relative to its topography should be controllable. Currently, the position of network elements relative to the topography is uncontrolled. Consequently it may happen that networks with similar characteristics otherwise, have significantly varying slope requirements since they are positioned differently relative to their topography. The current set of 7 topography functions should be expanded to include a wider range of topographies.

Once the controllable characteristics of the package have been sufficiently expanded, it would be of significant benefit to sewer network optimization as a whole if it were used to propose a definitive standard problem library which future researchers can use to compare the results of the their optimization algorithms.

9.2.3.2 Classify a given network

The current software model is capable of producing networks with varied characteristics, which were combined to define a set of network classes. It would be beneficial to leverage the software's capabilities to determine the characteristics of a given network with the aim of classifying the network. That would pave the way towards using the best suited combination of heuristic influence factors in the optimization of the given network.

9.2.4 Understanding Optimality

The evolution of sewer network characteristics as solutions improved during the lifespan of the optimization process was studied. While many interesting results have been obtained and used to improve optimality, there are some still some stones left unturned which may further our understanding of sewer network optimality.

9.2.4.1 Expand the investigation to include hydraulic characteristics

Network characteristics were chosen which are controllable during layout creation with what little information is available at the time. It may be beneficial to perform a similar study as the one conducted in Chapter 4, but which focusses instead on hydraulic characteristics of the network. The results obtained from such an investigation can be used to propose heuristic influence factors that operate on metaheuristic hydraulic optimization algorithms, or to improve the decision making of the HOMS algorithm proposed in Chapter 3.

9.2.4.2 Investigate sewer network layouts from a graph theory perspective

Thirteen layout parameters were presented in section 7.1 and their effects on optimality investigated. I is possible there are other characteristics which would offer additional insight into the nature of a layout. Only two parameters from graph theory, namely degree and betweenness centrality, were investigated. It revealed a direct relationship between betweenness centrality of the graph and the diameters of a sewer network. Undertaking an investigation form a formal, mathematical viewpoint, i.e. analysing the network from graph theory perspective, may reveal

new characteristics of sewer networks that not only improve their optimization, but also lead to new design guidelines and methods.

9.2.4.3 Incorporate a machine learning algorithm

Machine learning algorithms learn over time, similar to metaheuristic optimization algorithms, but a key difference is that machine learning algorithms use existing data to improve its decisions. This technology could be leveraged to study the results of sewer network optimizations using different heuristic combinations and different problem classes. Learning from the results, the algorithm should be able to recommend the best heuristic combinations to use for a given problem. If a sufficiently advanced machine learning algorithm was created, it might even propose new heuristic influence factors to enforce certain networks characteristics that it deems beneficial. This is certainly an interesting and exciting avenue of investigation.

9.2.4.4 Changing heuristic influence factors during algorithm lifespan

In Chapter 7 it was shown that different heuristic influence factors performed better, or worse, for certain phases of the optimization algorithm's lifespan. Certain combinations found the feasible solution space almost instantaneously but performed poorly overall, while others struggled to find the feasible solution space and still performed the best overall. It would be interesting to investigate the effect on algorithm performance if different heuristic combinations were applied at different times during the algorithm's lifespan.

Appendices

Appendix A

Standard Unit Hydrographs

Used with permission from GLS Consulting (Pty) Ltd.

	Land use(s) following typical UH pattern														
	Very high Inc. Res.	High Inc. Res.	Medium Inc. Res.	Low Inc. Res.	Cluster	Flats	Bus_Comm	Educational	Gov_Inst	Industrial	Other	Farm_AH	None (eg. P.O.S)	Unknown	Large (per kl AADD)
Hour	Dimensionless flow ordinates (relative to hydrograph peak)														
1	0.15	0.17	0.15	0.09	0.15	0.15	0.08	0.08	0.08	0.09	0.08	0.17	0.00	0.09	0.09
2	0.08	0.10	0.09	0.05	0.09	0.09	0.07	0.07	0.07	0.07	0.07	0.10	0.00	0.07	0.07
3	0.06	0.05	0.07	0.06	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.05	0.00	0.06	0.06
4	0.05	0.05	0.05	0.19	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.00	0.05	0.05
5	0.05	0.05	0.15	0.49	0.15	0.15	0.06	0.06	0.06	0.06	0.06	0.05	0.00	0.08	0.08
6	0.11	0.44	0.77	0.80	0.77	0.77	0.08	0.08	0.08	0.10	0.08	0.44	0.00	0.25	0.25
7	0.67	1.00	1.00	0.83	1.00	1.00	0.15	0.15	0.15	0.47	0.15	1.00	0.00	0.69	0.69
8	1.00	0.91	0.98	0.93	0.98	0.98	0.34	0.34	0.34	0.68	0.34	0.91	0.00	0.95	0.95
9	0.87	0.94	0.73	1.00	0.73	0.73	0.83	0.83	0.83	0.84	0.83	0.94	0.00	1.00	1.00
10	0.85	0.84	0.66	0.96	0.66	0.66	0.94	0.94	0.94	0.93	0.94	0.84	0.00	0.89	0.89
11	0.82	0.74	0.72	0.89	0.72	0.72	1.00	1.00	1.00	0.94	1.00	0.74	0.00	0.87	0.87
12	0.71	0.59	0.65	0.75	0.65	0.65	0.98	0.98	0.98	0.89	0.98	0.59	0.00	0.83	0.83
13	0.56	0.48	0.61	0.71	0.61	0.61	0.94	0.94	0.94	0.75	0.94	0.48	0.00	0.60	0.60
14	0.50	0.41	0.57	0.74	0.57	0.57	0.89	0.89	0.89	0.81	0.89	0.41	0.00	0.59	0.59
15	0.46	0.40	0.63	0.76	0.63	0.63	0.88	0.88	0.88	0.95	0.88	0.40	0.00	0.53	0.53
16	0.44	0.38	0.66	0.73	0.66	0.66	0.92	0.92	0.92	1.00	0.92	0.38	0.00	0.53	0.53
17	0.41	0.39	0.70	0.70	0.70	0.70	0.84	0.84	0.84	0.89	0.84	0.39	0.00	0.47	0.47
18	0.38	0.48	0.68	0.66	0.68	0.68	0.35	0.35	0.35	0.66	0.35	0.48	0.00	0.37	0.37
19	0.45	0.53	0.72	0.59	0.72	0.72	0.22	0.22	0.22	0.35	0.22	0.53	0.00	0.28	0.28
20	0.49	0.52	0.70	0.51	0.70	0.70	0.15	0.15	0.15	0.22	0.15	0.52	0.00	0.24	0.24
21	0.45	0.51	0.68	0.38	0.68	0.68	0.12	0.12	0.12	0.17	0.12	0.51	0.00	0.20	0.20
22	0.50	0.49	0.57	0.26	0.57	0.57	0.11	0.11	0.11	0.14	0.11	0.49	0.00	0.16	0.16
23	0.40	0.42	0.44	0.18	0.44	0.44	0.10	0.10	0.10	0.12	0.10	0.42	0.00	0.15	0.15
24	0.29	0.28	0.22	0.13	0.22	0.22	0.09	0.09	0.09	0.10	0.09	0.28	0.00	0.13	0.13
	Unit hydrograph parameters (1 ℓ/min)														
Hydrograph Peak	1.69	1.04	0.64	0.39	0.37	0.30	2.46	4.97	1.93	2.19	1.75	0.59	0.00	0.55	2.00
% of AADD	50%	55%	60%	75%	65%	75%	55%	55%	55%	55%	55%	60%	0%	55%	60%
Leakage & base flow	0.26	0.21	0.19	0.15	0.14	0.11	1.05	2.12	0.83	1.04	0.75	0.15	0.00	0.23	0.21

	Land use(s) following typical UH pattern														
	Very high Inc. Res.	High Inc. Res.	Medium Inc. Res.	Low Inc. Res.	Cluster	Flats	Bus_Comm	Educational	Gov_Inst	Industrial	Other	Farm_AH	None (eg. P.O.S)	Unknown	Large (per kl AADD)
Hour	Flows (ℓ /min)														
1	0.51	0.39	0.29	0.19	0.20	0.16	1.25	2.52	0.98	1.24	0.89	0.25	0	0.28	0.39
2	0.40	0.31	0.25	0.17	0.17	0.14	1.22	2.47	0.97	1.19	0.87	0.21	0	0.27	0.35
3	0.36	0.26	0.23	0.17	0.17	0.13	1.20	2.42	0.95	1.17	0.86	0.18	0	0.26	0.33
4	0.34	0.26	0.22	0.22	0.16	0.13	1.17	2.37	0.93	1.15	0.84	0.18	0	0.26	0.31
5	0.34	0.26	0.29	0.34	0.20	0.16	1.20	2.42	0.95	1.17	0.86	0.18	0	0.27	0.37
6	0.45	0.67	0.68	0.46	0.42	0.34	1.25	2.52	0.98	1.26	0.89	0.41	0	0.37	0.71
7	1.39	1.25	0.83	0.47	0.51	0.41	1.42	2.87	1.12	2.07	1.01	0.74	0	0.61	1.59
8	1.95	1.16	0.82	0.51	0.50	0.40	1.89	3.81	1.49	2.53	1.35	0.69	0	0.75	2.11
9	1.73	1.19	0.66	0.54	0.41	0.33	3.09	6.25	2.43	2.88	2.20	0.70	0	0.78	2.21
10	1.70	1.08	0.61	0.52	0.38	0.31	3.36	6.79	2.64	3.08	2.40	0.65	0	0.72	1.99
11	1.65	0.98	0.65	0.50	0.41	0.33	3.51	7.09	2.76	3.10	2.50	0.59	0	0.71	1.95
12	1.46	0.82	0.61	0.44	0.38	0.31	3.46	6.99	2.72	2.99	2.47	0.50	0	0.69	1.87
13	1.21	0.71	0.58	0.43	0.37	0.29	3.36	6.79	2.64	2.68	2.40	0.43	0	0.56	1.41
14	1.11	0.64	0.55	0.44	0.35	0.28	3.24	6.54	2.55	2.81	2.31	0.39	0	0.55	1.39
15	1.04	0.63	0.59	0.45	0.37	0.30	3.21	6.49	2.53	3.12	2.29	0.39	0	0.52	1.27
16	1.00	0.61	0.61	0.43	0.38	0.31	3.31	6.69	2.61	3.23	2.36	0.37	0	0.52	1.27
17	0.95	0.62	0.64	0.42	0.40	0.32	3.12	6.29	2.45	2.99	2.22	0.38	0	0.49	1.15
18	0.90	0.71	0.63	0.41	0.39	0.31	1.91	3.86	1.51	2.49	1.36	0.43	0	0.43	0.95
19	1.02	0.76	0.65	0.38	0.41	0.33	1.59	3.21	1.25	1.81	1.14	0.46	0	0.38	0.77
20	1.09	0.75	0.64	0.35	0.40	0.32	1.42	2.87	1.12	1.52	1.01	0.46	0	0.36	0.69
21	1.02	0.74	0.63	0.30	0.39	0.31	1.35	2.72	1.06	1.41	0.96	0.45	0	0.34	0.61
22	1.11	0.72	0.55	0.25	0.35	0.28	1.32	2.67	1.04	1.35	0.94	0.44	0	0.32	0.53
23	0.94	0.65	0.47	0.22	0.30	0.24	1.30	2.62	1.02	1.30	0.93	0.40	0	0.31	0.51
24	0.75	0.50	0.33	0.20	0.22	0.18	1.27	2.57	1.00	1.26	0.91	0.32	0	0.30	0.47
	Flow hydrograph volumes (ℓ/d)														
Regular flow	1090	697	507	313	293	238	1513	3057	1187	1490	1076	395	0	333	1210
Leakage & base flow	374	302	274	216	202	158	1512	3053	1195	1498	1080	216	0	331	302
TOTAL FLOW	1464	999	780	529	495	396	3025	6109	2382	2988	2156	611	0	664	1512
AADD	2929	1817	1301	706	761	528	5500	11108	4331	5432	3920	1019	0	1207	2520

Appendix B

Digital Appendix

This appendix serves as a reference to the dissertation's digital appendix, which can be found ftp://gcvr_ftp:2ks8P8Rk@ftp.sun.ac.za/.

The data for Chapters 6, 7 and 8 is contained in the digital appendix, which comprises approximately 400GB of predominantly plain text files or .csv files. The appendix contains many 'readme' files to assist with folder navigation and understanding file structures. In addition to the raw data, some processed data is present which show the final rolled up results used to draw the conclusions of Chapters 7 and 8, respectively.

A runnable .jar file, SNO.jar, is included in the digital appendix. This is a lightweight software package which will allow the reader to use the techniques and algorithms presented throughout this dissertation to perform problem generation, optimization and network characteristic calculations. A brief tutorial to assist the reader with using the software is also included.

List of References

Afshar, M. (2012). Rebirthing genetic algorithm for storm sewer network design. *Scientia Iranica*, vol. 19, no. 1, pp. 11–19.

Afshar, M., Rohani, M. and Sargolzaei, M. (2011). Application of cellular automata to sewer network optimization problems. *Scientia Iranica*, vol. 18, no. 3, pp. 304–312.

Bau, Y., Ho, C. and Ewe, H. (2008). Ant colony optimization approaches to the degree-constrained minimum spanning tree problem. *Journal of Information Science and Engineering*, pp. 1081–1094.

Blum, C. and Roli, A. (2001). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, vol. 35, pp. 268–308.

Blumensaat, F., Wolfram, M. and Krebs, P. (2012). Sewer model development under minimum data requirements. *Environ Earth Sci*, vol. 65, pp. 1427–1437. Special Issue.

Buck, J. (2015). *Mazes for Programmers*. The Pragmatic Programmers, LLC.

Bui, T. and Zrncic, C. (2006). An ant-based algorithm for finding degree constrained minimum spanning tree. Tech. Rep., The Pennsylvania State University at Harrisburg, Middletown, PA 17057. Computer Science Program.

Daniel, W. (1990). *Applied nonparametric statistics*, pp. 358–365. 2nd edn. PWS-Kent.

de Berg, M., Cheong, O., van Kreveld, M. and Overmars, M. (1997). *Computational Geometry: Algorithms and Applications*. 3rd edn. Springer-Verlag Berlin Heidelberg.

Diogo, A. and Graveto, V. (2006). Optimal layout of sewer systems: A deterministic versus stochastic model. *Journal of Hydraulic Engineering*, vol. 132, no. 9, pp. 927–943.

Diogo, A., Walters, G., Sousa, E. and Graveto, V. (2000). Three-dimensional optimization of urban drainage systems. *Computer-Aided Civil and Infrastucture Engineering*, vol. 15, no. 6, pp. 409–426.

Dorigo, M. (1992). *Optimization, learning and natural algorithms*. Ph.D. thesis, Politecnico di Milano, Italy.

Dorigo, M. and Gambardella, L. (1997). Ant colony system: a cooprative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66.

Dorigo, M., Maniezzo, V. and Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on systems, man, and cybernetics-part B: cybernetics*, vol. 26, no. 1, pp. 29–41.

Dorigo, M. and Stützle, T. (2004). Ant colony optimization. MIT Press.

Freeman, R. (1978). Job satisfaction as an economic variable. *American Economic Review*, vol. 68, no. 2, pp. 135–141.

Giustolisi, O. and Savic, D. (2006). A symbolic data-driven techniques based on evolutionary polynomial regression. *Journal of Hydroinformatics*, vol. 8, no. 3, pp. 207–222.

Haghighi, A. (2013). Loop-by-loop cutting algorithm to generate layouts for urban drainage systems. *Journal of Water Resource Planning and Management*, vol. 139, no. 6, pp. 693–703.

Haghighi, A. and Bakhshipour, A. (2012). Optimization of sewer networks using an adaptive genetic algorithm. *Water Resource Management*, vol. 26, no. 12, pp. 3441–3456.

Haghighi, A. and Bakhshipour, A. (2015). Deterministic integrated optimiozation model for sewage collection networks using tabu search. *Journal of Water Resource Planning and Management*, vol. 141, no. 1.

Knuth, D. (1974). Postscript about np-hard problems. *ACM SIGACT News*, vol. 6, no. 2, pp. 15–16.

Kolisch, R., Sprecher, A. and Drexl, A. (1992). Characterization and generation of a general class of resource-contrained project scheduling problems. Tech. Rep. 301, Instituten für Betriebswirtschaftslehreder Universität Kiel.

Leeuwen, V. (1998). *Handbook of Theoretical Computer Science*, vol. A. Amsterdam: Elsevier.

Lejano, R.P. (2006). Optimizing the layout and design of branched pipeline water distribution systems. *Irrigation and Drainage Systems*, vol. 20, pp. 125–137.

Leonora, B., Dorigo, M., Gambardella, L. and Gutjahr, W. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing: an international journal*, vol. 8, no. 2, pp. 239–287.

Li, G. and Matthew, R. (1990). New approach for optimization of urban dranage systems. *Journal of Environmental Engineering*, vol. 116, no. 5, pp. 927–944.

Li, G.Y. (1986). The optimal desogn of sewer networks by dddp. *China Water Supply and Sewearge*, vol. 2, no. 2, pp. 18–23. In Chinese.

Mays, L. and Wenzel, H. (1976). Optimal design of multilevel branching sewer systems. *Water Resources Research*, vol. 12, no. 5, pp. 913–917.

Meredith, D. (1972). Dynamic programming with case study on planning and design of urban water facilities. Tech. Rep., Colorado State University, fort Collins. Treaties on urban water systems.

Merkle, D., Middendorf, M. and Schmeck, H. (2002). Ant colony optimization for resource-constrained project scheduling. *IEEE Transaction on evolutionary computation*, vol. 6, no. 4.

Miles, S. and Heany, J. (1988). Better than 'optimal' method for designing dranage systems. *Journal of Water Resource Planning and Management*, vol. 114, no. 5, pp. 477–499.

Moderl, M., Butler, D. and Rauch, W. (2008). A stochastic approach for automatic generation of urban drainage systems. *11th International Conference on Urban Drainage*, vol. 11.

Moeini, R. and Afshar, M.H. (2012). Layout and size optimization of sanitary sewer network using intelligent ants. *Advances in Engineering Software*, vol. 51, pp. 49–62.

Orlowski, S., Wessäly, R., Pióro, M. and Tomaszewski, A. (2009). Sndlib 1.0 - survivable network design library. *Networks: An International Journal*, vol. 55, no. 3, pp. 276–286.

Pan, T. and Kao, J. (2009). Ga-qp model to optimize sewer system design. *Journal of Environmental Engineering*, vol. 135, no. 1, pp. 17–24.

Pasupathy, R. and Henderson, S.G. (2011). Simopt: A library of simulation optimization problems. In: *Proceedings of the 2011 Winter Simulation Conference (WSC)*.

Perlin, K. (2001). Noise hardware. In: *Real-Tme Shading*, chap. 2.

Ramkissoon, Y. (2014). Water is life. sanitation is dignity: Accountability to people who are poor. Tech. Rep., South African Human Rights Commission.

Rao, S. (2009). *Engineering Optimization: theory and practice*, pp. 260–262. 4th edn. Wiley.

Reingold, E. and Tilford, J. (1981). Tider drawings of trees. *IEEE Transactions of Software Engineering*, vol. 7, no. 2, pp. 223–228.

Robinson, D. and Labadie, J. (1981). Optimal design of urban storm water drainage system. In: *International Symposium on Urban Hydrology, Hydraulic and Sediment Control*, pp. 145–156. University of Kentucky, Lexington, KY, USA.

Rohani, M. and Afshar, M. (2015). Ga-ghca model for the optimal design of pumped sewer networks. *Canadian Journal of Civil Engineering*, vol. 42, no. 1, pp. 1–12.

Royte, E. (2017 August). A place to go: Open defecation threatens health. *National Geographic*.

Shaw, V. (1963). The development of contributor hydrographs for sanitary sewers and their use in sewer design. *Civil Engineering in South Africa*, vol. 5.

Sitzenfrei, R., Fach, S., Kizenl, H. and Rauch, W. (2010*a*). A multi-layer cellular automata approach for algorithmic generation of virtual case studies: Vibe. *Water Science and Technology*, vol. 61, no. 1, pp. 37–45.

Sitzenfrei, R., Moderl, M. and Rauch, W. (2010*b*). Graph-based approach for generating virtual water distribution systems in the software vibe. *Water Science and Technology: Water Supply*, vol. 10, no. 6, pp. 923–932.

Stephenson, D. and Hine, A. (1982). Computer analysis of johannesburg sewers. *Proc. Instn. Munic. Engrs. S.A. IMESAF*, , no. 4, pp. 13–23.

Stützle, T. and Hoos, H. (2000). Max min ant system. *Future Generation Computer Systems*, vol. 16, pp. 889–914.

Trifunovic, N., Maharjan, B. and Varavamoorthy, K. (2013). Spatial network generation tool for water distribution network design and performance analysis. *Water Science and Technology: Water Supply*, vol. 13, no. 1, pp. 1–19.

van Heerden, G. (2014). *Evaluation of alternatives for hydraulic analysis of sanitary sewer systems*. Master's thesis, University of Stellenbosch, South Africa.

Walters, G. (1985). The design of the optimal layout for a sewer network. *Engineering Optimization*, vol. 9, no. 1, pp. 37–50.