SKU assignment in a multiple picking line order picking system.

Jason Matthews

Dissertation presented in for the degree of Doctor of Philosophy in the Faculty of Economic and Management Sciences at Stellenbosch University

Supervisor: Prof SE Visagie

Date: December 2015

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.
03 June, 2015

Abstract

An order picking system in a distribution center (DC) owned by Pep Stores Ltd. (PEP) is investigated. Twelve unidirectional picking lines situated in the center of the DC are used to process all piece picking. Each picking line consists of a number of locations situated in a cyclical formation around a central conveyor belt. Pickers walk in a clockwise direction around a conveyor belt picking stock for stores.

The picking lines are managed in waves due to PEPs policy to push stock to stores. For each wave of picking a subset of released stock keeping units (SKUs) is selected and assigned to an available picking line. The physical stock is then brought to the assigned picking line before multiple pickers pick all the store requirements (or orders) defined by the SKUs within that wave. Once all of the orders have been picked a new mutually exclusive set of SKUs, defining a new wave, is brought to the picking line for picking. In this way picking lines function in parallel to and independently of each other.

The order picking system is deconstructed into three decision tiers. Firstly at the start of each day SKUs are assigned to available picking lines which defines the Picking Line Assignment Problem (PLAP). Once a set of SKUs has been assigned to a picking line each SKU is assigned a specific location within the picking line which defines the SKU Location Problem (SLP). Finally once pickers are brought to the picking line the individual orders are sequenced for each picker. This defines the Order Sequencing Problem (OSP). The focus of this dissertation is on the first two subproblems namely, the SLP and PLAP as the OSP has already been solved in a previous study.

This picking line setup considered here has many similarities to carousel systems. Several heuristic approaches for arranging SKUs within carousel systems are adapted for use in this picking line environment. These heuristics are compared to two novel lower bound formulations as well as trivial lower bound to evaluate their performance. Both historical as well as generated problem instances are used to compare the relative performances of each heuristic. An average saving of 2% for large and 6.5% for medium sized problem instances is achieved if the best solution form the four heuristics is selected. Three goals are used when assigning SKUs to picking lines in the PLAP. Firstly walking distance should be reduced, secondly the number of small cartons produced should be minimal and finally the number of pallet movements required to populate any one picking line for a wave of picking should be manageable.

The concept of a maximal cut is used as an estimate for total walking distance and it is shown that by minimising the maximal cut within each picking line the total walking distance is reduced. A greedy phased insertion heuristic is introduced which minimised the maximal cut and therefore walking distance. Although the total walking distance was reduced by on average 22% compared to historical assignments the number of small cartons produced and the number of pallet movements required to populate some picking lines is undesirable.

Four measures using SKU correlations are introduced and used within a phased greedy insertion framework. These measures reduce the number of small cartons produced with a marginal increase in total walking distance compared to approaches which minimized the maximal cut only. The total walking distance is reduced by on average 20% compared to historical assignments with the number of small cartons produced within an acceptable range. However, the number of pallet movements required to populate some of the picking lines remains at an undesirable level.

A final picking line segmentation approach is introduced using a sequence of integer programming formulations. These formulations include capacity constraints which limit the total volume of stock (and therefore the number of pallet movements) assigned to any one picking line. This approach delivers individual picking lines that have a manageable number of pallet movements to populate all picking lines with stock. A final hybrid approach is also introduced which switches between this segmentation approach and a correlations approached when appropriate. This results in a 15% reduction in walking distance compared to historical assignments while maintaining a good number of small cartons produced and improving on the historical assignments in terms of the number of pallet movements required to populate any one picking line with stock.

The managers within the DC are responsible for doing both the SKU to picking line assignments as well as the SKU arrangements within each picking line. A new warehouse management system (WMS) is in the process of design and implementation. A proof of concept interface which illustrated how the approaches to both the SLP and PLAP can be implemented in the new WMS while still allowing for managerial flexibility is therefore proposed.

Opsomming

’n Bestellinguitsoekstelsel in ’n distribusiesentrum wat deur Pep Stores Bpk. (PEP) besit word, word ondersoek. PEP gebruik twaalf eenrigting uitsoeklyne wat in die distribusie sentrum is om al die items vir bestellings uit te soek. Elke uitsoeklyn bestaan uit ’n aantal vakkies wat rondom voerband geleë is. Werkers loop in ’n kloksgewyse rigting om hierdie voerband om items vir winkels te versamel.

Die uitsoeklyne opereer in golwe omdat PEP ’n beleid het om voorraad na die winkels te stuur (eerder as dat winkels voorraad bestel). ’n Subversameling van beskikbare voorraadeenhede (VE’s) word geselekteer en toegewys aan ’n beskikbare uitsoeklyn. Die voorraad word dan na die toegewysde uitsoeklyn gebring voordat ’n aantal werkers al die bestellings (wat deur die VE’s in daardie golf gedefinieer word) vir die winkels gaan versamel. Indien al die bestelling in daardie golf voltooi is, word ’n nuwe onderling uitsluitende versameling VE’s na die uitsoeklyn gebring, wat dan weer ’n nuwe golf vorm. Op hierdie manier kan die uitsoeklyne parallel aan, en onafhanklik van mekaar funksioneer.

Hierdie uitsoekstelsel kan ontbind word in drie vlakke van besluitneming. In die eerste vlak word VE’s aan beskikbare lyne toegwys, wat gedefinieer word as die uitsoeklyntoewysingsprobleem (PLAP). Nadat die VE’s aan die lyn toegewys is, moet elke VE aan ’n spesifieke vakkie binne daardie lyn toegewys word en word gedefinieer as die VE-plasingsprobleem (SLP). In die derde vlak moet die bestellings se volgorde bepaal word vir die opmaak van die bestellings. Dit word as die bestellingvolgordeprobleem (OSP) gedefinieer. Die fokus van hierdie proefskrif is op die eerste twee vlakke van besluitneming, naamlik die PLAP en SLP. Die OSP is reeds in vorige studies opgelos.

Die uitsoekstel wat hier beskou word het baie ooreenkomste met ’n rondomtaliestelsel. ’n Aantal heuristiese benaderings tot die rangskikking van van VE’s in vakkies vir rondomtaliestelsels word aangepas en ondersoek vir hierdie uitsoekstelsel. Hierdie heuristieke word vergelyk met twee nuwe ondergrensformulerings sowel as ’n triviale ondergrens. Historise data en genereerde data word gebruik om die prestasie van elke heuristiek te vergelyk. ’n Gemiddelde besparing van 2% vir groot en 9.5% vir medium opmaaklyne word verkry indien die beste oplossing van die vier heuristieke gekies word. Drie doelwitte word beskou indien VE’s aan opmaaklyne toegewys word (tydens PLAP). Eerstens moet die stapafstand van werkers geminimeer word, tweedens moet die aantal klein kartonne geminimeer word en laastens moet die hoeveelheid werk (vurkhyserbewegings) om die voorraad na ’n enkele lyn te bring binne perke gehou word.

Die beginsel van ’n maksimum snit word gebruik om die stapafstand te benader en resultate toon duidelik dat deur die maksimum snit te minimeer word die stapafstand ook verminder. ’n Gulsige gefaseerde invoegingsheuristiek (GP) word voorgestel wat die maksimum snit te minimeer. Alhoewel die totale stapafstand met 22% verminder teeenoor historiese data vermeerder die aantal klein kartonne en die aantal vurkhyserbewegins na sekere lyne word onaanvaarbaar hoog.

Vier maatstawwe om die korrelasies/verwantskappe tussen VE's te bereken word vervolgens gebruik in die GP heuristiek om VE's in lyne toe te wys. Hierdie maatstawwe verbeter die aantal klein kartonne met 'n marginale toename in stapafstand teenoor die metodes wat slegs die maksimum snit minimeer. Die totale stapafstand word nou slegs verminder met 20%, maar die aantal klein kartonne val binne 'n aanvaarbare perk. Die aantal vurkhyserbewegings na sommige lyne is egter steeds te hoog.

'n Segmenteringsbenadering word voorgestel waarin 'n aantal heeltalige programmeringsformulerings gebruik word. Hierdie formulerings sluit kapasiteitsbeperkings in wat die totale volume voorraad na 'n uitsoeklyn beperk. Hierdie formulerings lewer uitsoeklyne wat 'n aanvaarbare hoeveelheid vurkhyserbewegings benodig. 'n Finale hibriedbenadering word ook voorgestel wat 'n kombinasie van die segmentering- en korrelasiebenadering gebruik. Hierdie metode verskaf 'n 15% verbetering in stapafstand relatief tot historiese oplossings terwyl 'n goeie aantal klien kartonne gehandhaaf en daar verbeter word op die aantal vurhyserbewegings.

Die uitsoeklynbestuurders is verantwoordelik vir die oplossing van die PLAP en die SLP. 'n Nuwe pakhuisbestuurstelsel (WMS) is in die proses van implementering by PEP. 'n Voorstel van hoe hierdie oplossingmetodes in die WMS ingesluit en hoe die gebruikerskoppelvlak moet lyk sodat daar steeds 'n groot mate van vryheid aan die gebruiker oorgelaat word, word ook verskaf.

Acknowledgements

Many people played a significant role in the work leading up to and during the writing of this dissertation. The author hereby wishes to express his deepest gratitude towards:

- Prof SE Visagie for his insight, guidance, enthusiasm and above all his loyal friendship.
- Margarete and Francois Bester for their support and friendship.
- XTranda for providing valuable training and support.
- My family for their financial and emotional support over my entire academic career.
- My wife Irene for marrying and supporting me during the final stages of my PhD
- All of the staff at the Department of Logistics at Stellenbosch University for their support and friendship
- All of the employees at PEP for their friendly and enthusiastic assistance.

The Department of Logistics at Stellenbosch University are hereby thanked for the use of their computing facilities and office space. The financial support of the South African National Research Foundation (NRF), in the form of a part-time doctoral scholarship under grant number 92272, PSP in the form of a bursary towards this research is hereby acknowledged. Any opinions or findings in this dissertation are those of the author and do not necessarily reflect the views of Stellenbosch University,the NRF or PEP.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms and Abbreviations

Acronyms	Meaning
ADS	A correlations measure for a greedy insertion approach to the PLAP
ADT	A correlations measure for a greedy insertion approach to the PLAP
ASLB	The assignment lower bound approach for the SLP
CD	Classroom discipline heuristic for the SLP
COI	Cube per order index
CS	Correlated slotting heuristic
DBN	Distribution
DC	Distribution centre
DSS	Decision support system
EXD	A generated problem instance type for the SLP
GA	Greedy adjacencies heuristic for the SLP
GAP	Generalised assignment problem
GI	Greedy insertion approach to the PLAP
GP	Greedy phased insertion approach to the PLAP
GS	Greedy heuristic for the SLP
HAS_C	Hybrid assignment algorithm for the PLAP with capacity C
His	Historical results
IP	Integer programming
IP_α	Relaxed integer programming formulation for the PLAP with maximal SKU tolerance of α
JCS	A correlations measure for a greedy insertion approach to the PLAP
JCT	A correlations measure for a greedy insertion approach to the PLAP
OPA	Organ pipe heuristic for the SLP
OSP	Order sequencing problem
OXD	A generated problem instance type for the SLP
PEP	PEP Stores Ltd.
PEP_{HIS}	The set of historical results to the PLAP
PEP_{OSP}	The historical result to the PLAP where the OSP is solved
PLAP	Picking line assignment problem
SD	Steepest decent slotting heuristic
SEG_C	The pure segmentation approach to the PLAP with capacity C
SKU	Stock keeping unit
SLP	SKU location problem
SSI	Shortest spanning interval
TRLB	Trivial lower bound for the SLP
TSP	Travelling salesman problem

TSPLB	The TSP lower bound approach for the SLP
UND	A generated problem instance type for the SLP
UNS	A generated problem instance type for the SLP
VRS	Voice recognition software
WMS	Warehouse management system

CHAPTER 1

Introduction

Before a product reaches the shelf of a retailer and finally the hands of a consumer it must pass through many value adding processes. Raw materials are sourced, products are manufactured, consumers are informed of products through marketing channels and the goods are delivered. In today's economy these processes are typically performed by different organisations and business entities. Managing these different processes and the relationships between the different entities is essential to maintain competitiveness and forms the basis for the strategic function known as supply chain management.

The concept of supply chain management first appeared in industry vocabulary in the 1990s [10]. Beamon [8] describes the supply chain as a set of business entities such as suppliers, manufacturers, distributors and retailers working together in an effort to acquire new raw materials, convert them into final products and deliver these final products to retailers. Supply chain activities include product development, material sourcing, production and logistics as well as the information systems needed to conduct these activities [22]. Supply chain management typically consists of the management of these different entities with a system wide approach to costing and information.

Coyle *et al.* [10] describes supply chain management as the effective and efficient flow of products, materials, services information and financials from suppliers through various intermediate entities and organisations to the final customer. Figure 1.1 illustrates the interaction between these entities and organisations. The flow of products, services, finance and information runs through the entire chain and requires a system wide management approach.

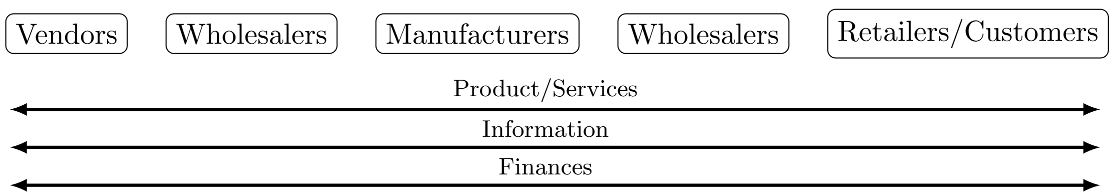


Figure 1.1: *A schematic representation of the flow of products, materials, services information and financials in a supply chain [10].*

One of the challenges of supply chain management is the physical movement of goods between entities. This is done with logistics channels which are integrated within the supply chain framework. Figure 1.2 illustrates a logistic channel where several raw material suppliers send stock to multiple manufacturers. The manufactured goods are sent to warehouses or distribution centers (DCs) where stock is consolidated and shipped to retailers. The focus of this dissertation is on the warehouse or DC connecting hub of a logistics network within a supply chain and is further discussed in more detail.

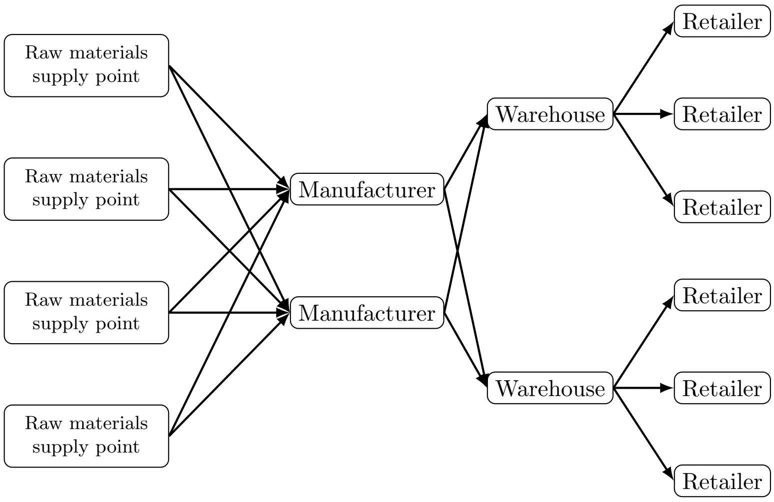


Figure 1.2: *A schematic representation of a complex logistics channel with multiple raw material suppliers, manufacturers, warehouses and retailers [10]. The arrows indicate the direction of stock movement.*

1.1 Warehousing and DCs

DCs play an important role in the logistics network by adding buffer areas to better match supply with demand. The effects of seasonality on sales can be better managed as stock levels are built up during low sales periods in preparation for high sales periods. In addition buffer areas aid to manage the stochastic lead times associated with the delivery of stock between suppliers and retailers. A further value added by DCs is the consolidation of product. Product received in bulk from multiple suppliers is consolidated into single shipments for individual retail outlets [6].

1.1.1 Types of DCs

Bartholdi & Hackman [6] identified several types of DCs by the types of customers which they serve. A retail distribution center links suppliers to retail outlets. The customers here are retail outlets which typically receive shipments on a regular basis. Orders in this environment would usually consist of hundreds of items covering the entire store catalogue. These DCs have the means to plan ahead because of the regular shipments and early availability of information about orders.

A less predictable environment is a service parts DC which is one of the most difficult types to manage. These DCs hold spare parts for capital intensive equipment such as construction vehicles or medical instruments. Two types of stock are managed in these DCs – stock for standard replacements by dealers and emergency stock usually for emergency repairs. These DCs hold a large variety of items with unpredictable demands because the majority of all parts need to be in stock in case of emergencies.

Some DCs ship items directly to end consumers and are known as catalogue fulfilment or e-commerce DCs. An example of this type of DC is the logistics network of one of the largest online retailers, namely Amazon [3]. Here a large number of customers place orders with only a few items and expect immediate shipment. A fast turn around for each order is a major concern for these DCs to ensure customer satisfaction.

DCs which handle perishable items such as food or flowers have several unique challenges. A chain of refrigeration including regulated storage and transportation vehicles is required which comes at a high cost. Managing space efficiently is therefore vital. This becomes more complicated with government regulations imposed on the handling of fresh food produce. For example, chicken should not be stored on top of other items. This reduces the chances of potential contamination from dripping juices. Inventory management is also a crucial aspect of this environment as the life spans of products are short.

Organisations may also choose to outsource their warehousing needs. A 3PL DC provides services to multiple organisations taking advantage of economies of scale and complimentary seasonality.

Frazelle [16] differentiates warehouses and DCs based on the value added to the logistics chain and to the product handled. At the start of the logistics network raw material warehouses and component warehouses hold raw materials (for example, raw metals for the manufacturing of vehicle chassis) near the point of manufacturing. As part of the manufacturing process partially completed parts or assemblies need to be stored before undergoing further manufacturing or assembly. These parts are held by work-in-progress warehouses before the finished product is stored in finished goods warehouses.

DCs may be viewed as warehouses which consolidate completed products for distribution to customers or stores [16]. In this context customers would receive regular shipments of stock. In some warehouses and DCs further value may be added to product. Possible value added services include in-house assemblies and packaging changes for marketing and repricing. Although warehouses and DCs may have different functions, customers and locations in logistics networks, they all require similar internal operations to carry out their function.

1.1.2 DC functions

Frazelle [16] identifies some major functions and activities within DCs including the receiving of goods, put away, pre-packaging, cross docking, order picking, sorting, material handling, packaging and shipping. Bartholdi & Hackman [6] groups DC activities into five main categories namely, receiving and put-away – forming the inbound activities – and order picking, checking, packing and shipping – forming the outbound activities. The interactions between these functional areas and activities are illustrated in Figure 1.3. All of the activities are connected by the material handling functionality.

The receiving function revolves around the collection and off loading of stock arriving at the DC. Stock will typically arrive at the DC in larger units compared to stock leaving the DC and will require less labour per unit of handling. Once the stock is offloaded it must undergo a quality and quantity control before it can be moved to other functional areas. During the put-away operation stock is moved to storage. Storage and transportation methods depend on the type of stock – its size, weight and handling characteristic. The location of the stock in the DC plays a critical role on DC efficiency in terms of product handling time.

Once the inbound stock has been received and stored it must be processed before leaving for customers. During the order pick operation correct quantities of different stock keeping units (SKUs) are picked for customers. Depending on the type of stock order picking can be done at the pallet, carton or individual item level. In many instances large quantities of stock are

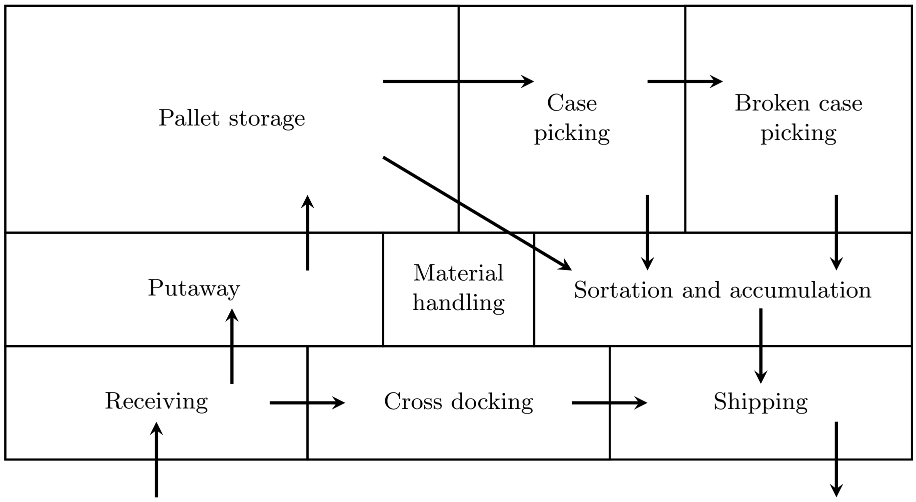


Figure 1.3: *A schematic representation of some of the functional areas in a DC and the movement of stock between them [16]. Arrows indicate the movement of stock governed by the material handling functionality.*

required to satisfy pending orders. In these cases stock can bypass storage and be directly moved from the receiving to the shipping area. This direct movement of stock from receiving to shipping is known as cross docking.

Once an order has been picked for a customer additional value added activities may be applied. Prices may be changed which typically occurs if customers are in other countries and require different currencies. In addition the basket of picked SKUs can be packed into a single package which reduces later material handling and transportation costs. These picked and packaged orders will typically be held in a holding area before being loaded onto delivery vehicles.

The focus of this dissertation is on the order picking operation and its effects on other DC functional areas for a specific DC in industry. A more detailed exposition of order picking systems is therefore provided in the next section.

1.2 Order picking

Order picking is the process of retrieving products from storage or buffer areas in response to customer requests and typically accounts for 55% of total DC costs. It involves the process of clustering and scheduling of orders, assigning stock to the orders, releasing the order to the floor for picking, physical picking and post picking clean-up [12]. Order picking may be seen as the most basic of services provided by a DC to the supply chain with all other value added services functioning around it [16].

According to Bartholdi & Hackman [6] 55% of the total time spent picking orders is associated with physical travel around the DC, 15% is spent searching for goods, 10% on extracting items and 20% for other administration type activities. Not only is physical travel the most time consuming activity, but it is also the most unproductive as no value is added to the order during travel. This activity often forms the main focus within the design and management of order picking systems.

DCs differ in terms of products received and customers served and thus order picking systems differ between DCs. The product characteristics – size, shape and weight; quantities in which products are ordered – pallets, cartons or individual items; number of SKUs in an order and the number of customers can all have an effect on the design and management of an order picking system.

Order picking systems may differ in a number of ways. Due to the advancement in technology the main area in which order picking operations differ is automation. Manual order picking requires human pickers to physically pick stock items. Automated order picking occurs when machines (and not humans) pick SKUs. This is often achieved with a system of robot arms and conveyor belts. Automatic order picking is often used when SKUs are small and uniform in shape such as in the pharmaceutical industry.

In most DCs manual order picking is used [12]. Among these systems two major types occur, namely picker-to-parts and parts-to-pickers systems. In a picker-to-parts system pickers will travel to locations holding required SKUs to pick the required items and in many cases vehicles are used. Vehicles allow pickers to reach multiple levels of storage in a high-level order picking system and increases picker movement speed. In an effort to reduce the travel times of pickers much attention in literature has been given to developing good routing procedures for different warehouse designs [13, 30, 35, 52, 54, 55].

One of the ways in which travel distance is decreased in a picker-to-parts system is to use a forward or fast pick area. A separate area of the DC is assigned as the forward pick area and functions as a mini DC within a DC. The most popular SKUs are stored in smaller quantities in this area. SKUs are therefore stored more densely giving pickers access to a wider variety of SKUs within a smaller walking distance. Pickers therefore move shorter distances to pick orders.

To maintain a dense concentration of different items and avoid stock-outs during picking the forward pick area needs to be restocked by the main DC. This creates a trade off between picking costs and restocking costs. Bartholdi & Hackman [5] addressed the problem of restocking forward pick areas by adjusting the volume assigned to a SKU within the forward pick area while minimising the total number of restocks. Accorsi *et al.* [2] considered both restocking as well as picker travel times when managing SKUs in a forward pick area. It was shown that overall improvements can be achieved by considering both objectives and in some cases the optimal allocation of stock by Bartholdi & Hackman [5] does not yield the best combined picking and restocking times.

Picker-to-parts systems may have many forms. The two basic variants occur with the presence of order batching and/or zone picking. In an order batching system a picker will pick multiple orders at the same time. Orders with a similar set of SKUs will be batched together. This reduces travel times as pickers walk the same path picking the same SKUs for multiple orders in the batch. Although travel distance is reduced the stock for a batch of orders needs to be picked in bulk and sorted between the orders. This sorting can either be done by the picker in a pick-and-sort operation or the stock for the entire batch can be picked and sorted in a later operation – known as pick-then-sort. Many studies have been conducted to improve picking efficiency by correctly batching orders for a variety of different DC configurations [14, 17, 26, 27, 35, 47, 48, 59, 64].

In zone picking the picking area (forward pick area or storage racks) is divided into zones. Each zone is then serviced by a single picker. This reduces the travel time of pickers as they only pick SKUs from a single zone and do not need to walk picking outlying SKUs in other zones. In addition pick efficiency of individual pickers increases as pickers become accustomed to the

SKUs in, and layout of their zone.

A drawback to zone picking is the re-consolidation of individual orders which span different zones. Orders are either passed sequentially between zones in a progressive zone picking system or orders are partially picked at each zone in parallel and consolidated at the end of all picking in a synchronised zone picking system. In a progressive zone picking system the issue of work balance between zones arises. Pickers stationed at the last zone of the system must wait for work to be passed down from preceding zones. The stochastic nature of picking often results in pickers either waiting for new orders from preceding zones or pickers being swamped with too much work. A similar issue also arises with synchronised zone picking, however, in this case imbalances occur between the picking and consolidation. The consolidation area is either swamped with work, or it becomes full of half completed orders waiting for SKUs from backlogged zones.

Grouping SKUs into zones to improve picking efficiency and manage work imbalances has received attention in litreature [18, 32, 34, 48, 64]. Bartholdi & Eisenstein [4] introduced a self organising system referred to as bucket brigade to address the issue of work balance in a progressive zone order picking system. Here pickers are allowed to take over orders from slower pickers upstream. In this way faster pickers pull work from slower pickers dynamically adjusting the size of the zones in which pickers operate.

Within the framework of parts-to-picker systems equipment is needed to bring stock to pickers. Two widely used systems are automatic storage and retrieval systems (AS/RS) and carousel systems. In an AR/RS system stock is brought down from storage racks and presented to pickers using specialised high-lifts. Examples of such high-lifts may be seen in Figure 1.4(a). A carousel system is a rotatable circuit of shelving which can rotate completely in both directions (bi-directional) or in a single direction (unidirectional). Figure 1.4(b) illustrates an example of a horizontal multidimensional carousel system. Pickers remain stationary as stock is presented in front of them on racks or shelves. Pickers may pick from multiple carousels, and carousels may have multiple levels of storage. There has been much attention given to the optimisation of carousel systems focusing on sequencing SKUs for a single order, sequencing a set of orders as well as arranging SKUs in a carousel [1, 7, 9, 19, 21, 23, 24, 25, 28, 29, 31, 33, 36, 37, 38, 39, 40, 49, 50, 56, 58, 60, 61, 62, 63].

(a) A photograph of an automatic storage and retrieval system. Source: [11].

(b) A photgraph of a horizontal multi-dimensional carousel. Source: [53].

Figure 1.4: *Photographs of parts-to-picker systems.*

Recent developments in the use of robots to move entire shelves through a network of pathways to pickers at picking stations has revolutionised small item parts-to-pickers systems. These robots, referred to as squat machines, are shown in Figure 1.5(a) and use advance routing and scheduling techniques. Pickers remain in a single location while squat machines queue at picking stations with required stock [20]. An example of this queue system is shown in Figure 1.5(b).

(a) A photograph of a KIVA robotic device. Source: [15].

(b) A photgraph a workstation serviced by KIVA robotic devices. Source: [15].

Figure 1.5: *Photographs of the KIVA systems robotic parts-to-picker devices and work stations.*

With the rise in technology, the way in which pickers receive pick instructions for orders has changed for both picker-to-parts and parts-to-pickers order picking systems. The first approach to manual picking was to give pickers physical printed pick slips like a type of shopping list. In this case pickers will need to manually keep track of what items have been picked for each order and search for the correct items using the location IDs on the picking slip. The first semi-automated system introduced was a pick by light system. Here pickers are directed by lights above required SKUs. With advances in voice recognition software pickers began to interact with tracking and routing software directly. Pickers wear headsets and are directed to locations holding SKUs and are provided with the pick quantities audibly.

A unique picker-to-parts order picking system using voice recognition software in a DC owned by PEP Stores Ld (PEP) is considered in this dissertation. A brief background and description of PEP's order picking system is therefore given in the next section. It is followed by a detailed problem description.

1.3 PEP's operations

PEP is the largest single brand retailer in South Africa and has been trading since 1965. PEP predominantly sells apparel and footwear, but is growing in the home décor, fast moving consumer goods and cellular device markets [51]. PEP's target market consists primarily of the low income population of Southern Africa. To reach its target market PEP uses a footprint of over 1600 stores located all around Southern Africa including Namibia, Botswana, Swaziland and Lesotho.

PEP strives to in keep prices low by managing an efficient supply chain. With three DCs located in Durban (East coast), Cape Town (West coast) and Johannesburg (Central) together with 13 transportation hubs PEP's distribution network spans more than 250 000 m^3. Figure 1.6 illustrates the framework of PEP's logistical network. Figure 1.7 illustrates the spread of the logistical footprint of PEP across Southern Africa. PEP's largest DC is in Durban which holds the largest port in South Africa. This DC receives the most stock as most of PEP's suppliers are in the far East (China and India).

A key to PEP's success is the management of its supply chain. One of the elements to PEP's

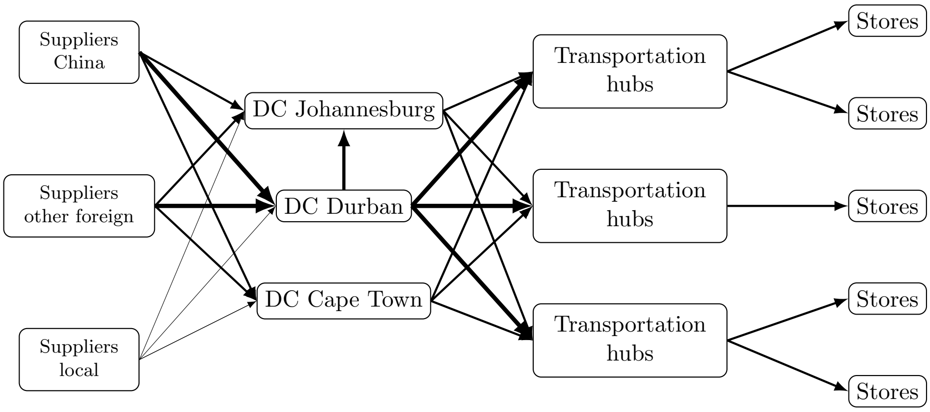


Figure 1.6: *A schematic representation of the supply chain of PEP. There are three DCs distributing to 13 transportation hubs. Each transportation hub serves a mutually exclusive set of stores. The DC in Durban sends stock to the Johannesburg DC for picking. Arrows indicate stock movement and the thickness of the lines gives an indication of the relative volume of stock moved.*

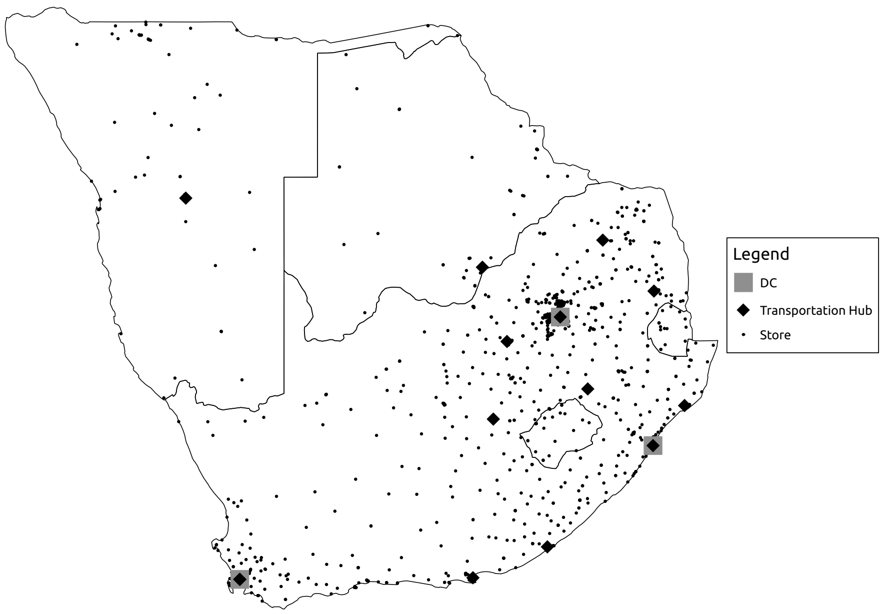


Figure 1.7: *A schematic representation the logistical footprint of PEP across Namibia, Botswana, South Africa, Lesotho and Swaziland.*

management is the distinction between replenishment and seasonal products. Replenishment products are those with a high rate of sale for the entire year and limited seasonality. Examples of these types of products include underwear and nappies. These products are monitored and replenished to stores regularly. Replenishment cycles range from weekly to monthly replenishments. The inventory at the DCs are also replenished regularly by suppliers. Seasonal products, for example winter jackets, are only managed for a single season. Future demand for the next season must be forecast and orders placed well in advance allowing for production and shipping. No additional stock for these products will be ordered once the season has started. For seasonal products the DC acts primarily as a stock consolidation facility receiving once off shipments of a SKU and distributing it to stores. In contrast, the DC plays a greater stock buffering role for replenishment products as fluctuations in store sales are absorbed in the DC's inventory.

Once an order has been placed at manufacturers the lead time until the stock arrives at a DC is typically 10 months. This includes six months production time and four months shipping time. A few weeks before a product is scheduled to be sent to stores the demand at each store is re-evaluated to account for changes during the lead time. Stock is redistributed between stores in an allocation process using more recent sales data. This allocation process is done centrally and creates a push system. Once stock has been allocated and physical stock has arrived at the DC the planning department schedules SKUs to be sent to stores. At this stage the DC can begin to pick SKUs. Each SKU has an associated out-of-DC date and must leave the DC before this date. The picked stock leaves the DC and is transported and consolidated at the transportation hubs before being delivered to stores. The delivery process typically takes 14 days.

The focus of this study is on PEP's order picking system. All of PEP's DCs use the same fundamental order picking framework, however, each has a different layout. The Durban DC will be used as a case study as it is the largest and receives the greatest amount of stock. The DC in Durban spans approximately 62 200 m^2 and is illustrated in Figure 1.8. There are several main functional areas in the DC namely receiving, decanting, rack and floor storage, order picking, dispatch and shipping. Each area will be discussed in brief with more focus placed on the order picking area.

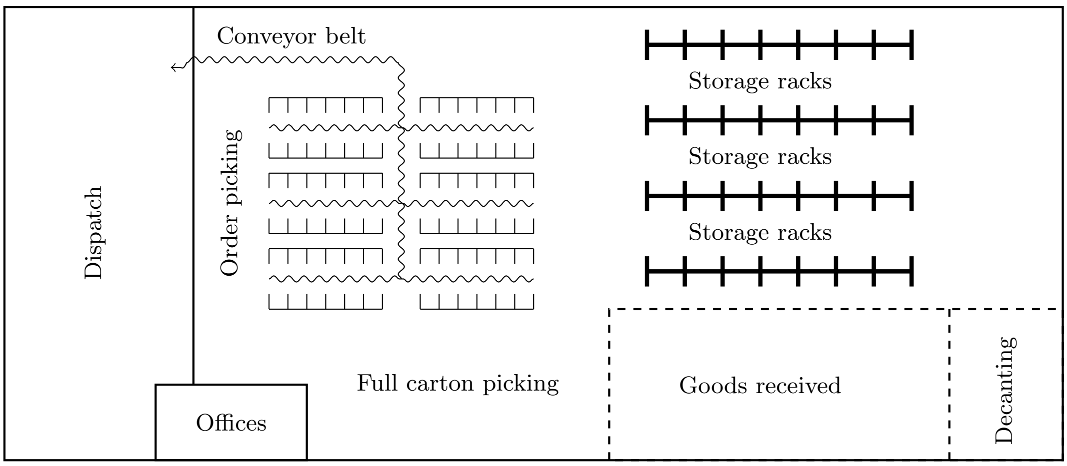


Figure 1.8: *A schematic representation the layout of PEP's DC in Durban.*

1.3.1 Receiving, storage and decanting

Stock arrives at the goods received area and is off loaded from one of the 15 loading bays. The stock, which typically arrives in cartons, is packed onto pallets and held in the goods received area until a quality control check has been completed. From here pallets are moved to either floor storage, rack storage, the order picking area or the decanting area.

The decanting area fulfils a similar role to cross docking, however, in this case stock is offloaded from containers and reloaded onto delivery vehicles destined for the Johannesburg DC. This stock will then be picked at the Johannesburg DC before being shipped to stores. Containers are not transported directly to Johannesburg to avoid container storage costs in Johannesburg and the cost of bringing empty containers back to the harbour[1]. Further transportation costs are saved as containers are often not fully loaded and delivery vehicles can hold more volume by consolidating stock when reloaded.

Figure 1.9 illustrates both the floor and the rack storage areas. The floor storage areas are typically used for stock which will be picked in a carton picking operation. Stock stored in storage racks is usually destined for piece picking. There are 23 aisles of rack storage serviced by five specialised high-lifts illustrated in Figure 1.10. Forklifts and pump trolleys, shown in Figure 1.11, are used to move pallets in and around the floor areas. Pallets destined for storage are queued at one end of the aisle and retrieved pallets are dropped off at the other end creating one direction of stock flow. The high lifts process jobs in batches. A high lift will either process a batch of put-away jobs or a batch of retrieval jobs.

(a) A photograph of the floor storage in PEP's DC in Durban.

(b) A photograph of the rack storage in PEP's DC in Durban.

Figure 1.9: *Photographs of the storage areas in PEP's DC in Durban.*

1.3.2 Order picking

A major influence on PEP's order picking system is the central inventory planning. SKUs are distributed to all stores in a single operation during the allocation process. This is achieved by collectivity assigning available stock for a SKU to all stores[2]. This allocation process is done for sets of SKUs or a distribution (DBN) consisting of SKUs of the same product type but different sizes. For example a DBN could consist of three SKUs – white T-shirts size small, medium

[1]Johannesburg is approximately 500km from Durban inland.
[2]Some stores may require no stock for a particular SKU.

(a) A photograph of an empty high lift.

(b) A photograph of a loaded high-lift lifting a pallet in the storage racks.

Figure 1.10: *Photographs of the high-lifts used to store and retrieve pallets from the storage racks in PEP's DC in Durban.*

(a) A photograph of an empty fork lift.

(b) A photograph of a loaded pump trolley.

Figure 1.11: *Photographs of pallet moving equipment used to move pallets on the floor space in PEP's DC in Durban.*

and large. Once stock has been allocated the resulting list of store requirements for the DBN are released to the DC. All store requirements for a DBN will be picked in the DC in a single operation. In this way SKUs and not orders are batched together in a single picking operation.

PEP uses 12 unidirectional picking lines to achieve SKU batching. Figure 1.12 illustrates this picking line area. Multiple pickers walk in a clockwise direction picking stock directly into cartons. Each picking line has 56 locations which can hold up to five pallet loads of the same SKU as shown in Figure 1.13. In addition managers often store additional stock on the floor space behind the location in-between picking lines and on the floor space at the ends of the picking lines if needed. Staff can quickly gain access to this stock and keep each location stocked during picking. This eliminates the complication of modelling the restocking of picking lines

SKUs can only be scheduled for picking on a picking line once the store requirements are issued by the planning department and the physical stock has been received by the DC. A batch of approximately 56 SKUs is assigned to an empty picking line and each SKU is assigned to a location. Once SKUs have been assigned to locations the physical stock is brought to the picking line before picking starts. Multiple pickers pick all the store requirements for the specific batch of SKUs before excess stock (if any) is removed and a new mutually exclusive set of SKUs is assigned to the picking line. One cycle of populating, picking and clearing of a picking line is referred to as a wave.

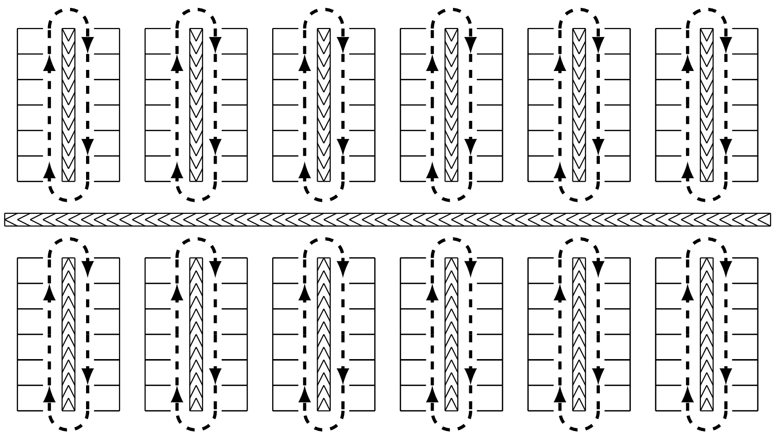


Figure 1.12: *A schematic representation of the picking line area with 12 picking lines each with 56 locations. The arrows and dashed lines indicate the walking direction of the pickers. Each picking line has a central conveyor belt conveying cartons to the main conveyor belt.*

Pickers are directed around the picking line with voice recognition software (VRS). Figure 1.14(a) illustrates the headset worn by pickers as they issue and receive instructions from the VRS. Before starting an order a picker will prepare an empty carton by placing a unique bar-code identification sticker on the carton and registering it with the VRS. Orders may be split over multiple cartons if required. For the purposes of this study, the term order will from here on refer to the set of store requirements for a single store for all the SKUs in a wave. The VRS assigns an order to a carton and directs the picker to the next location holding a required SKU for the assigned order in a clockwise direction. The VRS keeps track of the last location from which a picker picked stock. All SKUs for an order will therefore be picked in the shortest distance in a clockwise direction from where the order was issued. This implies that a picker will need to walk no longer than one cycle around the picking line to complete an order.

Figure 1.13: *A photograph of SKU locations in a picking line which is in the process of being populated with stock.*

(a) A photograph of a picker with a VRS headset placing a identification sticker on a new empty carton.

(b) A photograph of multiple pickers walking around an active picking line.

Figure 1.14: *Photographs of pickers in active picking lines.*

Orders are assigned to pickers regardless of the status and number of other pickers in the picking line. Each order must be completed by a single picker and pickers will only pick a single order at a time. In addition pickers may be added and removed from a picking line at any point before the picker starts a new order. In this way pickers can be dynamically shifted between active picking lines in the order picking area.

Pickers are able to pass each other freely while picking because there is enough space between the stock and the conveyor belt. Pickers stack empty cartons onto picking trolleys as shown in Figure 1.14(a). Pickers reuse the empty cartons from suppliers as well as new cartons and have access to empty cartons all around the picking line. The availability of empty cartons therefore does not influence the efficiency of pickers in a picking line. Figure 1.14(b) illustrates a functioning picking line with multiple pickers picking orders. Packed cartons are placed on conveyor belts which convey them to the dispatch area.

1.3.3 Dispatch and shipping

Cartons arriving at the dispatch area must often be resized as the size of cartons and the volume of stock for each order differs. In addition pickers do not know the required volume required for an order when selecting and preparing a new empty carton. Figure 1.15 illustrates the resizing and stapling of cartons at a dispatch station. A quality control check is also performed at this area on a sample of cartons to measure and manage picker accuracy.

Figure 1.15: *A photograph of a dispatch station in PEP's DC in Durban.*

Closed cartons are conveyed to the shipping area where they are held in buffer areas. Each buffer area is designated for a specific transportation hub. Once a buffer area has a sufficient volume of stock a delivery vehicle is scheduled and the stock is loaded and delivered. Regular deliveries of stock arrive at transportation hubs and stores. A typical store would receive between one and three deliveries a week. These regular deliveries aid in the management and efficiency of batching SKUs in wave picking.

The order picking system at PEP may be seen as a picker-to-parts system as pickers walk to required SKUs. Pickers use voice recognition software and orders are not batched. Although there are multiple picking lines which resembles a synchronised zone picking system, cartons are not consolidated and multiple pickers can pick in the same zone which removes many of complexities associated with zone picking in literature. The picking line area may further be described as a type of forward pick area as stock is brought from storage racks to the picking lines. The presence of wave picking allows for all the required stock for a wave for all stores to be placed in a picking line before picking starts. Restocking during picking is therefore not a problem as in the forward picking areas described in literature.

A picking line shows many similarities to carousel systems as pickers move in a cyclical direction around the picking line. Although carousel systems are parts-to-picker systems, the relative cyclical pattern in which SKUs pass pickers has a similar mathematical structure. Two characteristics, however, differentiate these two systems. Firstly the presence of wave picking creates a deterministic planning environment for picking lines which is not the case for carousel systems in literature. Secondly the presence of multiple pickers operating in the same picking line creates a dynamic environment which differs from typical carousel systems with a single operator for each carousel.

Managing the DC's order picking in terms of waves and the unconventional picking line setup creates a novel decision making environment not documented in academic literature. Within the order picking environment there are three decision tiers which are made on a daily basis. Optimising these decision tiers forms the basis for this study. The framework and details of these decisions is further discussed in the next section.

1.4 Problem description

Within the wave picking environment at PEP there are three sequential decision tiers which are made on a daily basis. Firstly DBNs need to be assigned to available picking lines. Once DBNs have been assigned to a picking line they need to be assigned a specific location by arranging them on the picking line. Finally, before picking starts the sequence in which the VRS assigns orders to pickers must be established. These three decision tiers are summarised as

1. Assign available SKUs to available picking lines.
2. Arrange SKUs on a picking line for each wave of picking.
3. Sequence orders to be passed to pickers by the VRS for each wave of picking.

These decision tiers are made in sequence and set of possible alternatives at each decision level is defined by the previous tier. For example the set of SKUs which need to be arranged on a picking line are defined by the initial assignment of DBNs to the picking line.

Although these decisions are made in sequence from decision tier 1 to 3, optimisation and decision support models must be developed in reverse order. For example, before approaches to arranging SKUs on a picking line can be developed an approach for sequencing the resulting set of orders must be known so that the SKU arrangement can be evaluated correctly. Similarly before alternative techniques for assigning DBNs to picking lines can be developed the effects of arranging SKUs with a different set of characteristics on the same picking line must be investigated. These decision tiers and the influences they have on each other are discussed for the remainder of this section.

1.4.1 DBN assignment and SKU arrangement

At the start of each day the managers evaluate the progress of all active picking lines and determine which picking lines will become available during that day. DBNs for which store requirements have been issued by the planning department and for which stock has arrived at the DC are assigned to these picking lines. SKUs within the same DBN are assigned to the same picking line to ensure that the entire range of sizes for a product arrives at the stores at the same time.

When a DBN is released to the DC an out-of-DC date is assigned to it. DBNs are ranked according to these dates and the top ranked DBNs are selected to be scheduled on the available picking lines. Once these DBNs have been scheduled to be picked they must be distributed into waves for the different available picking lines and stock brought to each picking line. The total store requirements for each SKU in a DBN is known in advance when the DBN is assigned. Sufficient stock is therefore brought to the picking line such that restocking is not required during the picking phase. In some cases multiple adjacent locations are assigned to a SKU, but these locations are treated as a single location in the VRS.

DBNs are currently distributed into waves with the objective of balancing the workload between each wave. Management measures workload using in house estimations based on the total volume of stock assigned to the wave and the total number of picks – *i.e.* the maximum number of times picker needs to reach into a carton to pick items. The actual walking distance of pickers is not considered at this stage as it is currently not yet calculable. Management classifies SKUs as either A or B pick SKUs. Bulky or heavy items requiring two hands to pick are regarded as B picks and are considered more difficult to pick. The number of picks in a wave are then weighted when evaluating work on a picking line with B picks given a larger weighting[3].

When distributing DBNs management further avoids creating waves of picking which require an excessive number of pallets of stock to populate the picking line. These waves would require a large batch of retrieval jobs to be performed by the high-lifts. High-lifts will be tied down to this single batch of jobs for a long time – in some cases more than an entire shift (8 hours). This reduces operational flexibility in the DC. Moreover, while a picking line is being populated with SKUs it is not adding value to orders as all the stock must be retrieved before picking starts.

After performing time analysis PEP segmented the time spent by pickers in a picking line into four tasks, namely walking time, picking time, time spent interacting with the VRS and time spent preparing new cartons. Only the walking time can be improved on by assigning DBNs to picking lines as the other times are fixed. Pickers typically spend 26% of their time walking, 30% picking items from location, 32% packing stock into cartons and 12% handling empty cartons.

After consulting PEP management [57] three goals were identified which should be aspired to when assigning DBNs to picking lines and are listed below:

1. Minimise the total walking distance of pickers.

2. Limit the number of waves which require excessive numbers of pallets of stock to build the picking line.

3. Manage the number of orders requiring a low volume of stock. These orders create small cartons which increases handling costs at dispatch and total transported volume.

There are two phases to the picking line assignment decision tier. Firstly DBNs must be scheduled for picking by assigning it to a particular day. Secondly the scheduled DBNs must be distributed to available picking lines. For the purposes of this study it will be assumed that DBNs are scheduled using the current out-of-DC date ranking system. The picking line allocation problem (PLAP) is therefore defined as assigning a set of scheduled DBNs to a set

[3]Weighted picks are only used to measure picker performance and manage bonuses.

of available picking lines while minimising walking distance, managing waves requiring a large volume of stock to populate a picking line and managing the number of small cartons produced.

Once DBNs have been assigned to a picking line and before stock can be brought to the picking line each SKU from each DBN must be assigned to a specific location. Although SKUs from the same DBN must be assigned to the same picking line these SKUs need not be adjacent to each other. Any SKU may therefore be assigned to any location. This may result in SKUs from the same DBN being placed in different cartons. However, all cartons picked from the same picking line should arrive at the stores at the same time. Managers currently arrange SKUs by spreading the number of weighted picks evenly around the picking line. Managers perceive this to reduce total picking time as potential picker congestion at popular SKUs is reduced. Two goals for arranging picking lines have thus been established after consulting PEP's management and are listed below:

1. Minimise the total walking distance of pickers.
2. Manage the congestion of pickers at popular SKUs.

The SKU location problem (SLP) is defined as the arrangement of a set of SKUs already assigned to a picking line while minimising walking distance and managing picker congestion.

Figure 1.16 illustrates the interactions between the PLAP and SLP. Both problems have a shared goal of minimising walking distance. An approach for determining picker walking distances before pickers start picking is therefore required before these two problems can be solved.

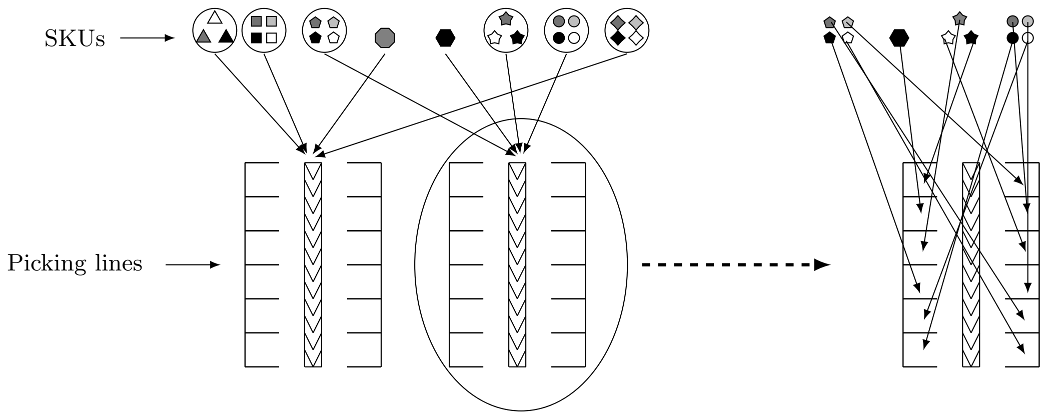


Figure 1.16: *A schematic representation of the first two decision tiers in the wave order picking operation. The picking line assignment decision tier is shown on the left and the SKU arrangement decision tier is shown on the right. Each shaded shape represents a SKU. SKUs with the same shape are part of the same DBN.*

1.4.2 Order sequencing

The total walking distance of pickers may be calculated once orders have been picked by adding the distances from the start to the end locations for each picked order. However, minimising the walking distance for a fixed SKU arrangement, before picking starts, has several complexities. Firstly the end position of the last order picked by a picker dictates the starting position of the next order. This starting position defines how far a picker needs to walk to pick all the required SKUs for that order. All preceding orders passed to a picker therefore influences the walking distance of the next order.

Further complexities are introduced with the stochastic nature of picking, the presence of multiple pickers and the dynamic way in which pickers can be added and removed from picking lines. The VRS must therefore be able to dynamically pass orders to pickers as needed while still ensuring that the total walking distance is minimised. Creating individual sequences of orders for each picker is not possible for several reasons. When pickers are added orders must be removed from other pickers' lists and assigned to the new picker. Furthermore, the order lists for a picker which is removed from a picking line must be assigned to other pickers. In addition the physical picking time is stochastic and although the walking distance may be calculated the actual time required to complete a set of orders can only be estimated. Having single order lists for pickers would create work imbalances similar to that of typical zone picking systems.

Matthews [41] investigated this final decision tier of order sequencing for multiple pickers. It was shown that this decision tier can be optimised by assigning a next/following order to a picker depending on the location where the picker finished the current order. When a picker requests a new order the VRS will select the best order based on a priority measure. This measure changes depending on the location where the picker completed the preceding order. Using this framework the final order sequencing decision tier may be described as determining a prioritised list of orders for each location while minimising the total walking distance of pickers in a dynamic picking environment.

Matthews & Visagie [42] used the concept of a span to solve the order sequencing problem (OSP) for a single picker. A span for an order is defined as the minimum length path walked by a picker which passes all required SKUs for that order from a given starting location. A span may start at any location and will end at a location holding a required SKU for that order. Figure 1.17 illustrates several spans for the same order. The concept of a cut for a location was introduced as the number of spans which pass that location. Matthews & Visagie [42] assigned spans to orders while minimising the maximal cut over all locations. This maximal cut forms a lower bound for the walking distance in terms of cycles traversed as a picker needs to completely circumvent the picking line each time it passes a location. Figure 1.18 illustrates the assigning of spans to orders. Note that the starting and ending positions of assigned spans do not necessarily link up.

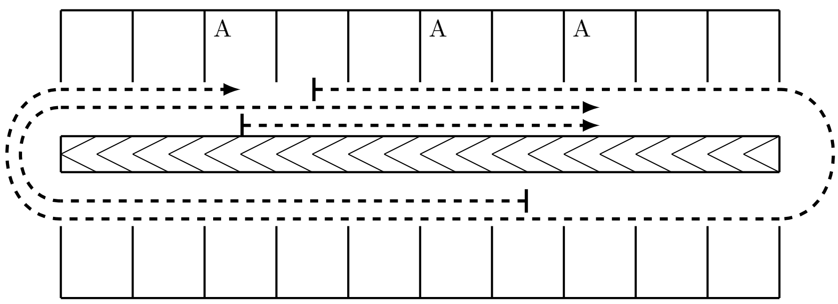


Figure 1.17: *A schematic representation of some different possible spans for the same order. Each line segment represents a possible span for the order. The letters in the locations indicate which locations hold SKUs required by the order. Each span passes all locations required by the order and finishes at a location holding a required SKU.*

It was further shown that the spans for each order could be linked up be shifting starting positions in a anticlockwise direction increasing the length of individual spans. These longer spans form a single cyclical tour which results in a feasible order sequencing solution for a single picker. Moreover the maximal cut for this new solution with longer spans will increase the original maximal cut by at most one cycle. A feasible solution to the OSP for a single picker is

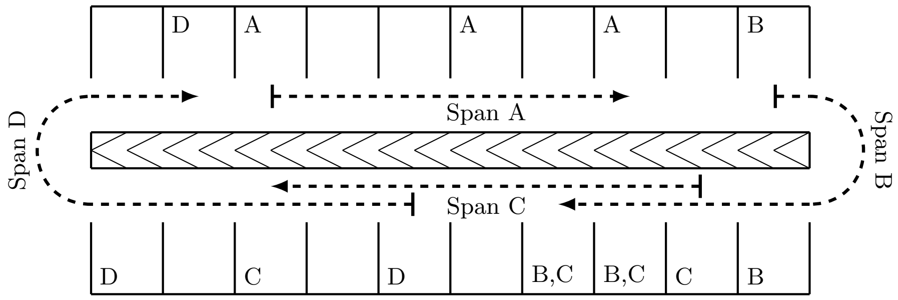


Figure 1.18: *A schematic representation of an assignment of spans to a set of orders. Each line segment is associated with a different order indicated by a letter. The letters in the locations indicate which locations hold SKUs required by each order. A span passes all the locations holding a required SKU for its order.*

therefore found which is within at most one cycle of a lower bound.

Matthews [41] applied this maximal cut solution to the multiple picker environment. A preferred starting location is assigned to each order which defines its desired span. These preferred starting locations are then used to prioritise the orders for each location. Orders are ranked for each location according to the distance to the preferred starting location of an order. Matthews [41] used a simulation model to show that the increase in walking distance when applying the maximal cut formulation solution for the OSP to a multiple picker environment is minimal. The increase in walking distance was approximately one cycle per picker in the picking line.

To compare different candidate SKU assignments and SKU arrangements in terms of walking distance the maximal cut approach to order sequencing for a single picker by Matthews & Visagie [42] will be used. The focus of this dissertation therefore falls to the first and second decision tiers, namely the PLAP and SLP.

1.5 Objectives

The aim of this dissertation consists of three parts: the development of solution methodologies for the SLP and PLAP; the gathering of representative test data for problem instances and the development of a test framework to test solution approaches to the SLP and PLAP while making provision for future studies; finally, addressing actual implementation practicalities of the proposed solution approaches. This is achieved by means of the following seven objectives:

Objective I

a Describe the internal layout and operations of the DC to better understand the problem in the DC context;

b Describe in detail the order picking operation in the DC so that the characteristics of the problem may be understood;

c Describe the different decision tiers and their interactions within the order pick operation;

Objective II

a Describe the SKU location problem (SLP) and identify the scope and assumptions;

b Identify the goals of the SLP decision tier;

c Describe the picking line allocation problem (PLAP) and identify the scope and assumptions;

d Identify the goals of the PLAP decision tier;

Objective III

a Obtain representative problem instances to test both the SLP and PLAP;

b Develop a test framework to test solution approaches to the SLP and PLAP while making provision for future research;

Objective IV

a Develop and test solution approaches to the SLP;

b Address the transitive nature of solving the SLP when evaluating solutions to the PLAP;

Objective V

a Develop and test solution approaches to the PLAP;

b Evaluate the trade-offs between the goals of the PLAP and discuss the performance of all solution approaches with regards to these trade-offs;

Objective VI

a Discuss and resolve the practical implementation issues of solution approaches to the PLAP;

b Propose a framework to integrate the PLAP solution approaches within the warehouse management system at PEP;

Objective VII

a Propose areas and directions for future research;

1.6 Dissertation layout and organisation

In Chapter 2 the problem instances which are used throughout the dissertation are introduced. The extraction and validation of historical data is discussed and the data manipulation and problem instance generation process illustrated. A test framework is presented to test solution approaches to the SLP and the PLAP while making provision for use in future studies.

The main content of the dissertation is presented in the form of four papers, two submitted and two accepted. Mathematical formulations for determining a lower bound to the SLP are introduced in Chapter 3 which is the article by Matthews & Visagie [43] that is submitted and under review. Four heuristic approaches are further tested and compared to the historical assignments and a set of random solutions. The marginal gain of solving the SLP is also discussed.

In Chapters 4 to 6 several novel approaches to solving the PLAP are introduced. All approaches are compared to the historical assignments using three goals, namely walking distance, the number of small cartons produced and wave size. Chapter 4 is the article by Matthews & Visagie [44], Chapter 5 is the article by Matthews & Visagie [46] and Chapter 6 is an article by Matthews & Visagie [45] that is submitted and under review.

Practical implementation issues of solution approaches to the PLAP are addressed in Chapter 7. The integration of managerial flexibility, data visibility and computational accuracy as well as automation is discussed. A proof of concept interface is proposed and illustrated for a decision support system that can be used in the DC.

Finally, Chapter 8 contains the dissertation conclusion including a discussion of possible directions for future work and the contributions made.

In the following chapter the different data requirements for the SLP and PLAP are discussed. The make-up of the problem instances and a test framework for testing is introduced. Due to the paper structure of this dissertation the mathematical symbols are in a few cases inconsistent across all chapters because of different requirements by different journals, but are consistent within each chapter/paper. However, the few exceptions will not hinder the overall readability of this dissertation.

Bibliography

[1] Abdel-Malek L & Tang C, 1994, *A heuristic for cyclic stochastic sequencing of tasks on a drum-like storage system*, Computers & Operationss Research, **21(4)**, pp. 385–396.

[2] Accorsi R, Manzini R & Bortolini M, 2012, *A hierarchical procedure for storage allocation and assignment within an order-picking system. A case study*, International Journal of Logistics: Research and Applications, **15(6)**, pp. 351–364.

[3] Amazon, *Amazon*, Available from `http://www.amazon.com/`.

[4] Bartholdi JJ & Eisenstein DD, 1996, *A production line that balances itself*, INFORMS, **44(1)**, pp. 21–34.

[5] Bartholdi JJ & Hackman ST, 2008, *Allocating space in a forward pick area of a distribution center for small parts*, IIE Transactions, **40(11)**, pp. 1046–1053.

[6] BARTHOLDI JJ & HACKMAN ST, 2011, *Warehouse and Distribution Science*, The Supply Chain and Logistics Institute, School of Industrial and Systems Engineering Georgia Institute of Technology Atlanta, GA 30332-0205 USA, Available from www.warehouse-science.com.

[7] BARTHOLDI JJ & PLATZMAN LK, 1986, *Retrieval strategies for a carousel conveyer*, IIE Transactions, **18(10)**, pp. 166–173.

[8] BEAMON BM, 1998, *Supply chain design and analysis: Models and methods*, International Journal of Production Economics, **55(3)**, pp. 281 – 294.

[9] BENGU G, 1995, *An optimal storage assignment for automated rotating carousels*, IIE Transactions, **27(1)**, pp. 105–107.

[10] COYLE JJ, BARDI EJ & LANGLEY JR CJ, 2003, *The Management of Business Logistics: A supply chain perspective*, 7th Edition, Thomson Learning, Canada.

[11] DAIFUKU, *Automatic storage and retrieval system*, Available from http://www.daifukumexico.com/, accessed on January 2014.

[12] DE KOSTER R, LE-DUC T & ROODBERGEN KJ, 2007, *Design and control of warehouse order picking: A literature review*, European Journal of Operational Research, **182(2)**, pp. 481–501.

[13] DE KOSTER R, ROODBERGEN KJ & VAN VOORDEN R, 1999, *Reduction of walking time in the distribution center of De Bijenkorf*, pp. 215–234 in SPERANZA MG & STÄHLY P (EDS), *New trends in distribution logistics*, pp. 215–234.

[14] DE KOSTER R, VAD DER POORT ES & WOLTERS M, 1999, *Efficient orderbatching methods in warehouses*, International Journal of Production Research, **37(7)**, pp. 1476–1504.

[15] ECOMMERCE FACTS, *Amazon acquires robot maker Kiva Systems*, Available from http://e-commercefacts.com/news/2012/03/amazon-buys-kiva-1/index.xml, accessed on January 2014.

[16] FRAZELLE EH, 2002, *World Class Warehousing and Material Handling*, McGraw-Hill, U.S.A.

[17] GADEMANN N & VAN DE VELDE S, 2005, *Order batching to minimize total travel time in a parallel-aisle warehouse*, IIE Transactions, **37(1)**, pp. 63–75.

[18] GARFINKEL M, 2005, *Minimizing Multi-zone Orders in the Correlated Storage Assignment Problem*, PhD, Georgia Institute of Technology, Georgia, (AT).

[19] GOSH JB & WELLS CE, 1992, *Optimal Retrieval Strategies For Carousel Conveyors*, Mathematical and Computer Modelling, **16(10)**, pp. 59–70.

[20] GUIZZO E, 2008, *Three Engineers, Hundreds of Robots, One Warehouse*, Spectrum, IEEE, **45(7)**, pp. 26–34.

[21] HA JW & HWANG H, 1994, *Class-based storage assignment policy in carousel system*, Computers & Industrial Engineering, **26(3)**, pp. 489–499.

[22] HANDFIELD R, *What is Supply Chain Management?*, Available from http://scm.ncsu.edu/scm-articles/article/what-is-supply-chain-management.

[23] Hassini E, 2002, *Storage Location and Space Allocation in Carousel Systems*, Doctoral Dissertation, University of Waterloo, Waterloo, Canada.

[24] Hassini E, 2009, *One-Dimensional Carousel Storage Problems: Applications, Review and Generalizations*, INFOR, **47(2)**, pp. 81–92.

[25] Hassini E & Vickson RG, 2003, *A two-carousel storage location problem*, Computers & Operations Research, **30(1)**, pp. 527–539.

[26] Henn S, 2012, *Algorithms for on-line order batching in an order picking warehouse*, Computers & Operations Research, **39(11)**, pp. 2549 – 2563, Available from `http://www.sciencedirect.com/science/article/pii/S0305054812000020`.

[27] Hsu CM, Chen KY & Chen MC, 2005, *Batching orders in warehouses by minimizing travel distance with genetic algorithms*, Computers in Industry, **56(2)**, pp. 169–178.

[28] Hwang H & Ha JW, 1994, *An Optimal Boundary for Two Class-based Storage Assignment Policy in Carousel System*, Computers & Industrial Engineering, **27(1-4)**, pp. 87–90.

[29] Hwang H, Kim Cs & Ko Kh, 1999, *Performance analysis of carousel systems with double shuttle*, Computers & Industrial Engineering, **36(1)**, pp. 473–485.

[30] Hwang H, Oh YH & Lee YK, 2004, *An evaluation of routing policies for order-picking operations in low-level picker-to-part system*, International Journal of Production Research, **42(18)**, pp. 3873–3889.

[31] Jacobs DP, Peck JC & Davis JS, 2000, *A simple heuristic for maximizing service of carousel storage*, Computers & Operations Research, **27(1)**, pp. 1351–1356.

[32] Jane CC & Laih YW, 2005, *A clustering algorithm for item assignment in a synchronized zone order picking system*, European Journal of Operational Research, **166(2)**, pp. 489–496.

[33] Jeroen P & Van Den Berg P, 1996, *Multiple Order Pick Sequencing in a Carousel System: A Solvable Case of the Rural Postman Problem*, The Journal of the Operational Research Society, **47(12)**, pp. 1504–1515.

[34] Kim BS & Smith JS, 2012, *Slotting methodology using correlated improvement for a zone-based carton picking distribution system*, Computers & Industrial Engineering, **62(1)**, pp. 286–295.

[35] Kulak O, Sahin Y & Taner M, 2012, *Joint order batching and picker routing in single and multiple-cross-aisle warehouses using cluster-based tabu search algorithms*, Flexible Services and Manufacturing, **24(1)**, pp. 52–80.

[36] Li CL & Guohua W, 2005, *Improved algorithm for maximizing service of carousel storage*, Computers & Operations Research, **32(1)**, pp. 2147–2150.

[37] Litvak N & Adan I, 2001, *The Travel Time in Carousel Systems under the Nearest Item Heuristic*, Journal of Applied Probability, **38(1)**, pp. 45–54.

[38] Litvak N & Adan I, 2002, *On a class of order pick strategies in paternosters*, Operations Research Letters, **30(1)**, pp. 377–386.

[39] LITVAK N, ADAN IJBF, WESSELS J & ZIJM WHM, 2001, *Order picking in carousel systems under the nearest item heuristic*, Probability in the Engineering and Informational Sciences, **15**, pp. 135–164.

[40] LITVAK N & MAIA V, 2010, *A survey on performance analysis of warehouse carousel systems*, Statistica Neerlandica, **64(4)**, pp. 401–447.

[41] MATTHEWS J, 2011, *Order sequencing and SKU arrangement on a unidirectional picking line*, Master of Science, University of Stellenbosch, Stellenbosch.

[42] MATTHEWS J & VISAGIE SE, 2013, *Order sequencing on a unidirectional cyclical picking line*, European Journal of Operational Research, **231(1)**, pp. 79–87.

[43] MATTHEWS J & VISAGIE SE, 2014, *SKU arrangement on a unidirectional picking line*, Submitted.

[44] MATTHEWS J & VISAGIE SE, 2015, *SKU assignment to parallel unidirectional picking lines*, The South African Journal of Industrial Engineering, **26(1)**, pp. 235–251.

[45] MATTHEWS J & VISAGIE SE, 2016, *A multi-objective approach for SKU assignment to unidirectional picking lines*, Submitted, **0(0)**, pp. 0–0.

[46] MATTHEWS J & VISAGIE SE, 2016, *SKU assignment to unidirectional picking lines using correlations*, ORiON, to appear.

[47] PAN CH & LIU SY, 1995, *A Comparative Study of Order Batching Algorithms*, Omega-International Journal of Management Science, **23(6)**, pp. 691–700.

[48] PARIKH PJ & MELLER RD, 2008, *Selecting between batch and zone order picking strategies in a distribution center*, Transportation Research Part E, **44(5)**, pp. 696–719.

[49] PARK BC, PARK JY & FOLEY RD, 2003, *Carousel system performance*, Journal of Applied Probability, **40(3)**, pp. 602–612.

[50] PARK BC & RHEE Y, 2005, *Performance of carousel systems with organ-pipe storage*, International Journal of Production Research, **43(21)**, pp. 4685–4695.

[51] PEP STORES LTD, *PEP*, Available from `www.pepstores.com`.

[52] PETERSEN CG, 1997, *An evaluation of order picking routeing policies*, International Journal of Operations & Production Management, **17(11)**.

[53] PRWEB, *Carousel*, Available from `http://www.prweb.com/`, accessed on January 2014.

[54] ROODBERGEN KJ & DE KOSTER R, 2001, *Routing methods for warehouses with multiple cross aisles*, International Journal of Production Research, **39(9)**, pp. 1865–1883.

[55] ROODBERGEN KJ & DE KOSTER R, 2001, *Routing order pickers in a warehouse with a middle aisle*, European Journal of Operational Research, **133(1)**, pp. 32–43.

[56] ROUWENHORST B, VAN DEN BERG JP & ZIJM WHM, 1996, *Performance analysis of a carousel system*, pp. 495–511 in GRAVES RJ, MCGINHIS LF, MEDEIROS DJ, WARD RE & WILHELM MR (EDS), *Progress in Material Handling Research*, pp. 495–511. The Material Handling Institute, Charlotte, NC, USA.

[57] Serdyn H, *Carton Sizes*, personal communication. Contactable at: `henniese@pepkorit.com`.

[58] Seshadri S, Rotem D & Segev A, 1994, *Optimal arrangement of cartridges in carousel type mass storage systems*, The Computer Journal, **37(10)**, pp. 873 – 887.

[59] Tang LC & Chew EP, 1997, *Order Picking Systems : Batching and Storage Assignment Strategies*, Computers & Industrial Engineering, **33(3-4)**, pp. 817–820.

[60] Vickson RG & Fujimoto A, 1996, *Optimal storage locations in a carousel storage and retrieval system*, Location Science, **4(4)**, pp. 273–245.

[61] Vickson RG & Lu X, 1998, *Optimal product and server locations in one-dimensional storage racks*, European Journal of Operational Research, **105(1)**, pp. 18–28.

[62] Wen UP & Chang DT, 1988, *Picking Rules for a Carousel Conveyor in an Automated Warehouse*, International Journal of Management Science, **16(2)**, pp. 145–151.

[63] Yeh DH, 2002, *A note on a simple heuristic for maximizing service of carousel storage*, Computers & Operations Research, **29(1)**, pp. 1605–1608.

[64] Yu M & de Koster R, 2008, *The impact of order batching and picking area zoning on order picking system performance*, European Journal of Operational Research, **198(1)**, pp. 480–490.

CHAPTER 2

Data and test framework

Test problem instances are required to evaluate different decision making approaches for both the SLP and PLAP. Furthermore a test framework is needed to model the effects on real life decision making. Different types of problem instances are required for the two types of problems (SLP, PLAP) as the time span over which decisions are made for each problem are different. An instance of the SLP needs to be solved for each individual wave on a picking line where an instance for PLAP is solved each day in the DC using multiple waves across multiple picking lines. The data requirements and test frameworks will therefore be discussed for each problem type separately.

2.1 SLP data

An SLP is solved for each individual wave of picking in the DC. The SLP has two goals, namely minimising total walking distance of pickers and managing the congestion of pickers at popular SKUs. Both of these goals can be evaluated using the store requirements, in terms of weighted number of picks, for each SKU in a wave. A problem instance is therefore defined as a set of store requirements for the SKUs already assigned to a wave and therefore a picking line.

Matthews & Visagie [2] used 22 historical OSP problem instances to evaluate the maximal cut approach to order sequencing. Each of these problem instances consisted of a set of SKUs and all store requirements. These problem instances are usable in the SLP context and Hagspihl & Visagie [1] used them to evaluate different approaches to the SLP in terms of picker congestion. The walking distance, however, was not considered in the study by Hagspihl & Visagie [1]. Included in these problem instances are the historical SKU arrangements. This allows for SLP approaches to be compared to the historical assignments. The sequential effects of optimising the OSP and SLP can also be compared to identify which decision tier has the largest impact on picking line efficiency.

The make-up of the 22 historical problem instances are illustrated in Table 2.1 and the data is available online [4]. Problem instances are split into different sizes based on the size of the maximal SKU (SKU with the most stores requiring it). The size of the maximal SKUs are larger than 500 for large problem instances, between 50 and 500 for medium problem instances and less than 50 for small problem instances.

In addition to problem instances derived from historical data further generated problem instances were also used to evaluate SLP approaches. Generated problem instances aid in evaluating SLP approaches for more generic problems and remove some of the bias of the historical

Data set	Number of SKUs	Number of orders	Size of the maximal SKU
L_{01}	49	1262	1232
L_{02}	54	1264	1226
L_{03}	51	1265	1161
L_{04}	56	1263	1011
L_{05}	51	1264	1069
L_{06}	53	1258	959
L_{07}	56	1260	855
L_{08}	54	1244	817
L_{09}	56	1264	729
L_{10}	55	1258	835
M_{01}	63	943	95
M_{02}	56	846	141
M_{03}	51	728	109
M_{04}	63	396	74
M_{05}	55	733	66
M_{06}	64	242	33
M_{07}	48	574	67
S_{01}	48	90	7
S_{02}	55	158	13
S_{03}	51	82	8
S_{04}	56	80	5
S_{05}	42	89	9

Table 2.1: *A summary of the 22 historical problem instances used to evaluate solution approaches to the SLP [3].*

data. Four sets of problem instances were generated by assigning SKUs to orders randomly using different distributions. These generated problem instances are defined as:

UND: Instances generated by assigning SKUs to orders where a SKU is assigned to an order with a probability of 0.75.

UNS: Instances generated by assigning SKUs to orders where a SKU is assigned to an order with a probability of 0.25.

EXD: Instances generated by assigning orders to SKUs where an order is assigned to a SKU with varying probability. Each order has a fixed probability of being assigned to a SKU, irrespective of the SKU, and are distributed evenly across the ranges $0.75 - 0.1$.

OXD: Instances generated by assigning SKUs to orders where a SKU is assigned to an order with varying probability. Each SKU has a fixed probability of being assigned to an order, irrespective of the order, and are distributed evenly across the ranges $0.9 - 0.15$.

A graphical representation of the generated problem instances is give in Figure 2.1. Most of the large and medium sized historical problem instances resemble the EXD pattern with an extended tail while the small historical problem instances resemble the pattern of the UNS generated problem instances.

All generated problem instances consisted of only 20 SKUs and 100 orders which is less than half the actual size of a picking line. This was done in an effort to solve the SLP with exact formulations. Heuristic solutions could then be compared to the lower bounds obtained from exact solution approaches. The properties of these generated problem instances are given in Table 2.2.

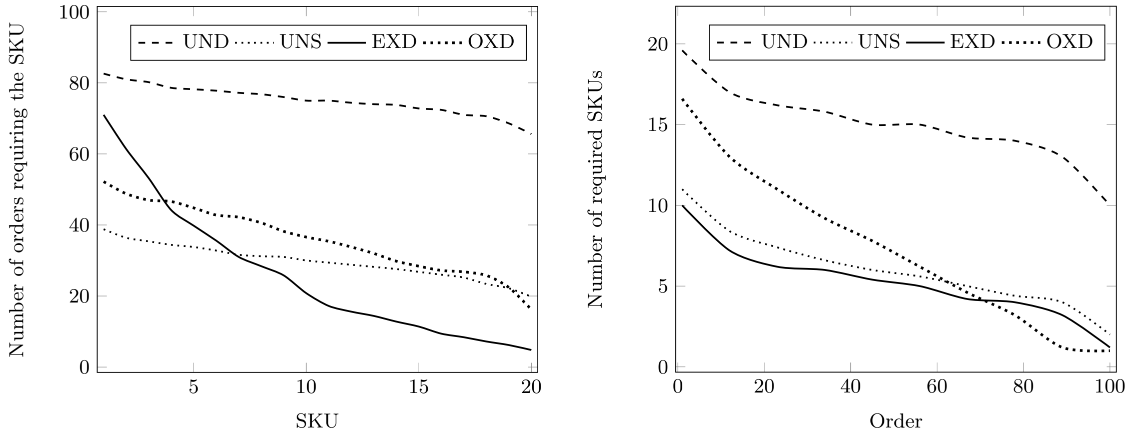


Figure 2.1: *A graphical representation of the relative size of each generated problem instance in terms of number of different SKUs required by each order and vice versa.*

Data set	Number of SKUs	Number of orders	Size of the maximal SKU
UND_1	20	100	82
UND_2	20	100	85
UND_3	20	100	83
UND_4	20	100	83
UND_5	20	100	80
UNS_1	20	100	37
UNS_2	20	100	38
UNS_3	20	100	39
UNS_4	20	100	39
UNS_5	20	100	41
EXD_1	20	100	71
EXD_2	20	100	79
EXD_3	20	100	73
EXD_4	20	100	64
EXD_5	20	100	68
OXD_1	20	100	54
OXD_2	20	100	50
OXD_3	20	100	53
OXD_4	20	100	56
OXD_5	20	100	48

Table 2.2: *A description of the 20 generated problem instances used to evaluate solution approaches to the SLP [3].*

2.2 PLAP data

Data consisting of DBN information for a set of DBNs which are assigned to a set of two or more waves each of which being assigned to a different picking line is required to test solution approaches to the PLAP. The purpose of the test data is threefold. The test data should primarily allow for different solution approaches to the PLAP to be compared to each other in terms of the three goals discussed in Section 1.4.1. This implies that sufficient information to evaluate these goals must be available for each DBN in a problem instance. A secondary aim is to directly compare different solution approaches with the historical assignment methods. This can be achieved by acquiring historical data where wave and picking line assignments are known. The historical assignments must consists of typical DBN to wave and wave to picking line assignments *i.e.* no special cases due to unrecorded internal practices. A final less imperative goal is to ensure that the data may be used for future studies. An example of such a study would be to evaluate different dynamic DBN scheduling approaches. To test different DBN scheduling approaches a set of data is required over a non-disjointed time line. This will allow DBNs to be scheduled on different days while still meeting their out-of-DC dates.

Data was extracted from PEP's warehouse management system (WMS), analysed and refined to generate test problem instances which help reach these three aims. The detailed data requirements for each objective is further discussed.

2.2.1 Data requirements

Only a set of DBNs and DBN information is required to test solution approaches to the PLAP. This DBN data forms the basic framework for a problem instance. The number of picking lines to which the DBNs are assigned can be selected arbitrarily and the size of each picking line can be set to 56 – the number of physical locations – or any other representative value.

Solutions by different PLAP approaches to the same problem instance must be evaluated according to the following three goals. The total walking distance across all picking lines should be minimised, the number of pallet movements required to populate a single picking line for a wave should be manageable and the number of orders requiring small volumes of stock (or number of small cartons produced) should be minimal. The DBN information for each DBN in a problem instance should therefore allow for each goal to be quantified. To evaluate solution approaches using these criteria the following information regarding each DBN is required:

1. The store requirements for each SKU in a DBN in terms of items or picks required. This will define the orders and therefore the walking distance within each picking line.

2. The total number of pallet movements required to move all the required stock for a DBN to a picking line. Waves requiring too many pallet movements to populate the picking line will be considered undesirable.

3. The volume of a single item for each SKU in a DBN. The exact volume picked for each store for each wave can be calculated from this data to evaluate the number of small cartons produced.

A set of historical wave and picking line assignments is required to achieve the second aim of comparing solution approaches to PEP's historical assignments. For each historical problem instance the DBNs should be scheduled for picking on the same day as the PLAP is solved daily in the DC. This models a manager's ability to assign any one of these DBNs to any one of the picking lines.

In many cases managers make decisions based on internal information which is not recorded in the online data systems. The bias of this unknown information must be reduced when selecting historical problem instances and special case waves using special picking lines removed. The following data regarding historical wave and picking line assignments is therefore required:

4. A set of DBNs assigned to waves and picking lines on the same day including the wave and picking line identification numbers to which each DBN was assigned.

5. A set of waves and picking lines which accurately depict typical internal assignment practices. *i.e.* no special case waves and picking lines.

Two areas of concern arise with regards to manager bias and special case waves and picking lines. Firstly the number of SKUs assigned to a wave should be realistic. For example, a wave with less than 15 SKUs would indicate a temporary picking line built on empty floor space or a special setup on a fixed picking line. Although these picking lines may not be built in the picking line area the orders are still processed through the order picking system as a wave of picking and is present in the online data sources. Secondly the number of adjacent locations assigned to a SKU should be known or at least practical. Should management historically assigned n adjacent locations to a SKU in a picking line any solution by a PLAP approach should also assign n adjacent locations to that SKU regardless of which picking line it is assigned to.

Finally consideration is made for future work such as DBN scheduling. Historical assignments should be extracted from a single connected time window. This allows for DBNs to be assigned to a waves and picking lines on other days while still maintaining the integrity of each of the DBN's out-of-DC dates. Furthermore the date on which DBNs became available for picking is required to simulate the daily release of DBN pick instructions to the DC by the planning department. The following data is required to achieve this aim:

6. Release dates for each DBN indicating from when a DBN can be scheduled on a picking line.

7. A connected time period from which DBNs and wave and picking line assignments are extracted.

2.2.2 Data extract

After consultation with the IT department at PEP a data extract was performed from several data sources in PEP's supply chain network [6]. The first data sources called the SKU and store master database contains both quantitative and qualitative data for all SKUs and stores. The data available for each SKU in the SKU master file are

- SKU number,
- SKU description,
- SKU volume per item in cubic meters
- pick type (A or B),
- weight and

- price.

This data is used to assign a volume to each SKU in each problem instance fulfilling the volume data requirement 3.

For each store there are a number of classifications in the store hierarchy. The disclosed data available from the store master file are

- store number,
- store name,
- store type,
- store open date and
- store close date.

Three types of stores exists, namely standard stores selling mainly apparel, home stores selling mainly home décor and cellular stores selling only cellular products. Little to no picking is done for the cellular stores in the DC because of the small size, high value and small quantities of stock in these stores. Only standard and home stores will therefore be considered.

Waves which hold stock only for new stores are processed on special case picking lines. Old stock which has already been sent to the operating stores is picked for new stores a few months before they open. These picking lines are built on empty floor space or on the bottom level of storage racks as only a handful of orders needs to be picked. New stores are identified using the store master data and these special case picking lines are excluded.

Once the allocation process has been applied to a new DBN all the store requirements for the DBN are recorded on a live database in the planning department. This data is held for three months after which it is archived. Two data tables are available the first of which consists of aggregated data for each SKU in each DBN. Included in this data table are

- DBN number,
- SKU number,
- DBN release date,
- DBN schedule date,
- number of stores,
- number of units,
- number of picks and
- a full carton indicator.

The “number of units” field represents the physical number of items which will end up on the shelves of stores. The number of picks indicates the maximum number of times a picker will put his/her hand in a carton to pick items. In many cases units are pre-packed and a single pick will contain multiple units.

The aggregated DBN data is also captured in a detailed format in the detailed DBN data table. Here the exact store requirements are included for each SKU. The available fields for this data table are

- DBN number,
- SKU number,
- store number,
- number of picks and
- a full carton indicator.

These DBN data tables are used to address most of the data requirements. From the detailed DBN data table the store requirements are established fulling data requirement 1. This data also fulfils the release date data requirement 6 and by selecting DBNs which were all scheduled within a connected time period the time period data requirement 7 is addressed.

During the allocation process stores may be assigned more than one carton load of a SKU. In these cases the full carton portion of the store requirements will be picked separately in the full carton area. Both full carton and piece picking instructions are therefore released to the DC for the same DBN number. The WMS processes the full carton DBNs on a virtual picking line holding stock for one DBN. Included in the DBN data is a field indicating whether DBN pick instructions are for full carton or piece picking.

When a wave of picking is planned in the DC that wave is given a unique wave and picking line identification number and is a combination of wave and picking line information. All the data for the DBNs assigned to a wave and a picking line is recorded on a local picking line data table and held for three months before being archived. For each wave of picking the following data are available

- wave and picking line ID,
- wave schedule date,
- wave completion date,
- DBN number,
- SKU number,
- number of items,
- location ID and
- a full carton indicator.

This picking line data table is used to fulfil the historical data requirement 4. Using this data table further analysis can be performed to evaluate the bias of management's picking line assignments and address the unbiased data requirement 5.

Of all the data requirements only the pallet movement requirement 2 is not addressed. Due to the current WMS no historical pallet movement data is stored. Moreover the actual number of pallets required to move all required stock for a DBN to a picking line is not known. As an alternative the total volume of stock required for a wave will be used as a measure of the size of the wave instead of total pallet movements. Using this measure, waves will be considered undesirable if too much stock is required to populate the picking line.

All of the above data tables were extracted from PEP's IT system on 25 April 2013. The merging, analysis and exclusion of data will further be discussed in the next section.

2.2.3 Data merging

The extracted data from different data tables is merged by using unique identification keys. Problem instances are generated from this merged data. Figure 2.2 illustrates the merging of data between different data tables. The SKU data table is mapped to the detailed DBN data table using the SKU number field. All SKU numbers in the detailed DBN data table had a corresponding match in the SKU master data table. Similarly store data is merged with the DBN detailed data using the store number field. For all stores in the DBN data a store was found in the store master data table. The DC picking line data is matched with the DBN data by using a composite mapping key (DBN number, SKU number, Full carton indicator).

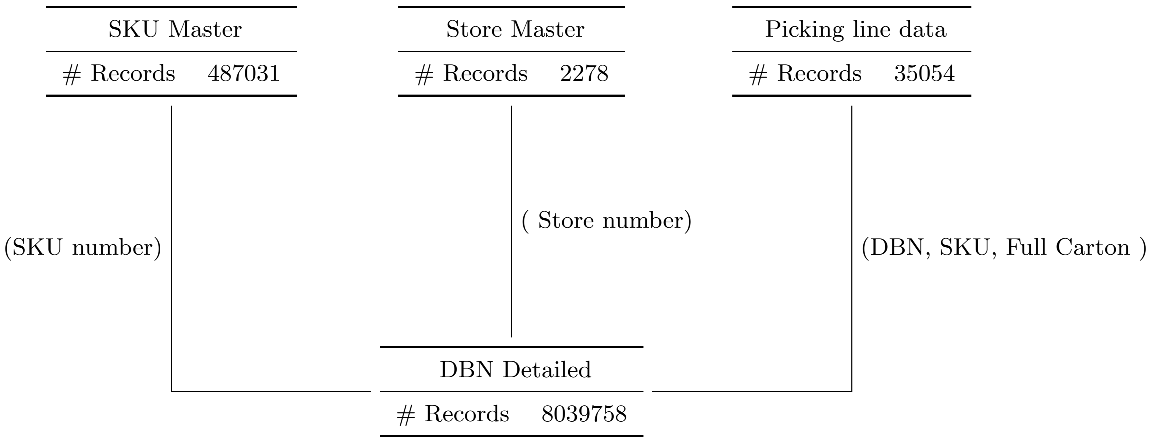


Figure 2.2: *A schematic representation of the merging procedure of the different data tables. The lines indicate a mapping using the relevant key. The total number of raw records in each data set is also given to illustrate the size of the data merge.*

In many cases records in the DBN data did not have any matches in the DC picking line data. This is expected as the DBN data contains all released DBNs including those not yet scheduled for picking. In some cases there were DBNs within the DC picking line data table with no match in the DBN data. This is due to the archiving time line of the DBN data. DBNs which were released more than three months before the extraction date, although scheduled after this date, were not present in the data.

Analysis was performed once the data had been merged into a single data table. Using this data a number of exclusions are made on both DBNs and picking lines. This reduced the data to a subset of picking lines and addressed the historical management assignment biases discussed in data requirement 5.

2.2.4 Exclusions

The exclusions imposed on the data consists of two types namely DBN exclusions and wave/picking line exclusions. DBNs are excluded from the data if the DBN contains SKUs for which there is no volume. This exclusion is rare and was applied to less than 0.5% of the total DBNs. Furthermore all DBNs marked as full carton DBNs are excluded as they are not picked in the picking lines. Finally any DBN with a total volume in excess of 300m^3 was excluded as these DBNs are handled as a special case in the DC and are infrequent. This exclusion was applied to one DBN.

Wave/picking line exclusions were based on several criteria. Firstly any wave with more than five excluded SKUs were excluded as too much historical information was lost. It was observed that this exclusion would apply to most of the waves scheduled before 31 January 2013. Only waves scheduled after 31 January 2013 were therefore included as this created a connected time period. After this exclusion there were only eight waves with at most four SKUs with missing data. Each of these waves were included in the problem instances as the effects of these missing SKUs would be minimal.

A further exclusion was made for waves which only have stock for new stores. This was achieved by comparing the schedule dates of waves with the open dates of the stores. Waves which only have orders for stores with open dates after the schedule date are excluded.

Further analysis was made into the number of SKUs assigned to each picking line as well as the number of locations assigned to each SKU. The actual number of locations assigned to a SKU in a picking line is not recorded and the only information provided is the location number. In many cases all of the 56 locations in a picking line are not used for each picking line. This leaves space for additional stock, empty cartons or empty unused locations. Table 2.3 illustrates the size of all the waves in the merged data set after all the DBN, full carton and new stores exclusions are applied. In some cases picking lines are built on empty floor space similarly to those for new stores. The waves for these picking lines would have very few SKUs and are excluded. Waves which were assigned significantly less than 56 SKUs suggests a special case. There is not sufficient data to identify the cause or purpose of these small picking lines and they are excluded. After consulting management waves with less than 30 SKUs allocated to it were excluded as they were deemed to be special cases which could not be understood with the limited data provided. In some rare cases more than 56 locations are used. This is achieved by adding additional location labels to the floor space around the picking line and storing stock on the floor. This is a manageable procedure and these picking lines are included.

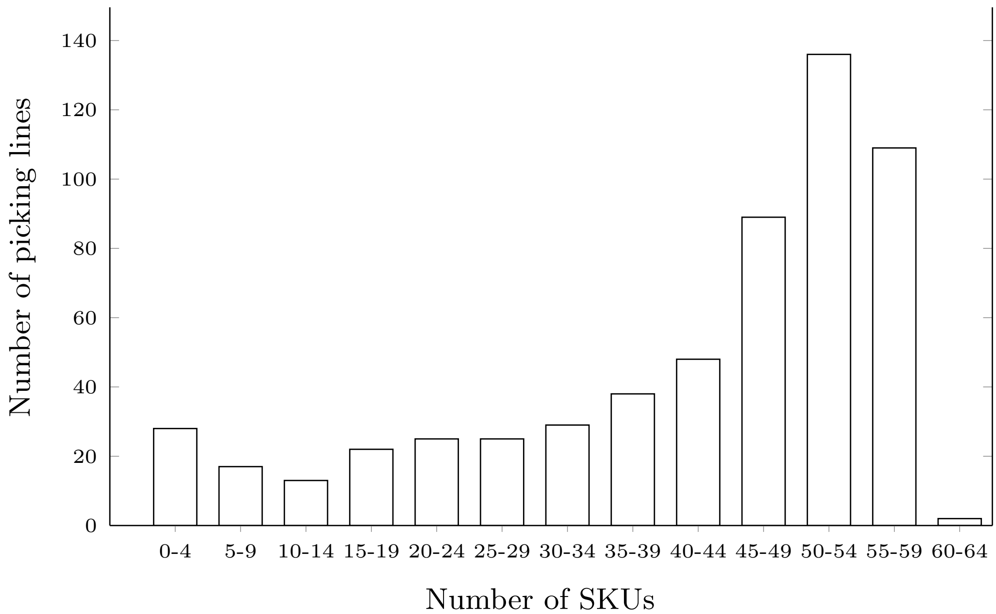


Figure 2.3: *A plot of the number of SKUs assigned to each wave in the data extract after the DBN, full carton and new stores exclusions are applied.*

Although the historical location of each SKU in a picking line is known, the exact number of locations assigned to the SKU for additional stock is not known. The number of adjacent empty locations can, however, be deduced, but it is still not known on which side of the assigned location additional stock is stored, if any. The effect of this lack of information is illustrated in

Figure 2.4. In an attempt to estimate the number of assigned locations the number of empty adjacent locations for a SKU either to the left or the right of its assigned location is compared to the total volume of stock required for that wave of picking. This comparison is illustrated in Tables 2.3 and 2.4. In many cases large volumes of stock are stored in just a single location. In these cases additional pallets are either stored on the floor behind the storage location between picking lines or in a staging area next to the picking line for fast and easy access by pump trolleys. There are many SKUs for which there is a relatively small volume of stock with many empty locations adjacent to it. Additional data is therefore required to determine the number of locations assigned to SKUs as volume cannot be used as an accurate estimate.

In an attempt to compare solution approaches to the PLAP it will be assumed that each SKU can be assigned to a single location in a picking line regardless of the volume required by the SKU and the available space adjacent to it in the historical assignments. To reduce the effects of this assumption on model bias each historical wave will be assumed to have been processed on a unique picking line with a capacity equal to the number of SKUs assigned to the wave. In this way should empty locations be present for any reason when processing a wave on a picking line they would remain present in the problem instances.

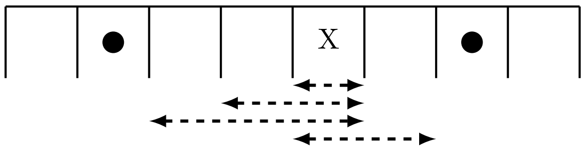


Figure 2.4: *A schematic representation of the lack of information regarding the number of locations assigned to a SKU in a picking line. All SKU locations assigned to a SKU are indicated with a dot or an X. Arrows indicate possible storage capacity for the SKU marked with an X.*

Number of locations including	Total volume rounded up to the nearest 10 m^3										
empty locations on the right	10	20	30	40	50	60	70	80	90	100	110
1	14951	538	63	18	2	2	1	-	-	-	-
2	617	289	149	39	23	12	7	-	1	-	-
3	92	44	60	37	15	7	5	3	1	-	1
4	25	2	2	3	2	2	1	-	-	-	-
5	21	-	1	-	-	-	-	-	1	-	-
6	23	-	-	-	-	-	-	-	-	-	-
7	17	-	-	-	-	-	-	-	-	-	-
8+	49	-	-	-	-	-	-	-	-	1	-

Table 2.3: *A comparison between the number of locations available for a SKU including empty locations on the right of the assigned location and the total volume for the SKU. Volume is rounded up to the nearest unit of 10 cubic meters. Each cell entry contains the number of SKUs in the historical data for the respective number of locations and volume category.*

A potential drawback on the assumption that each SKU is only assigned a single location is the generation of extremely large waves. This may be caused by assigning 56 SKUs with large volumes of stock to the same picking line. In practice less SKUs would have been assigned as some SKUs would occupy multiple locations. To evaluate the impact of this assumption the total volume assigned to historical picking lines is analysed. Table 2.5 illustrates the total volume assigned to historical picking lines and is compared to the number of SKUs assigned to the picking line. There are many picking lines which process waves with a large volume of stock, in excess of 450 m^3. This suggests that waves with large volumes of stock are practically manageable and the potential affects of the assumption are manageable.

Number of locations including	Total volume rounded up to the nearest 10 m^3										
empty locations on the left	10	20	30	40	50	60	70	80	90	100	110
1	14290	708	213	66	33	10	11	3	3	1	1
2	955	83	29	13	3	6	-	-	-	-	-
3	217	23	7	4	1	1	1	-	-	-	-
4	28	3	1	-	-	-	-	-	-	-	-
5	17	1	-	-	-	-	-	-	-	-	-
6	20	-	-	-	-	-	-	-	-	-	-
7	8	-	-	-	-	-	-	-	-	-	-
8+	29	-	-	-	-	-	-	-	-	-	-

Table 2.4: *A comparison between the number of locations available for a SKU including empty locations on the left of the assigned location and the total volume for the SKU. Volume is rounded up to the nearest unit of 10 cubic meters. Each cell entry contains the number of SKUs in the historical data for the respective number of locations and volume category.*

Total Volume in	Number of SKUs							
cubic meters	30-34	35-39	40-44	45-49	50-54	55-59	60-64	Total
$[0, 50)$	19	21	15	5	5	14	1	80
$[50, 100)$	-	-	1	4	27	35	-	67
$[100, 150)$	-	2	-	7	25	23	-	57
$[150, 200)$	1	-	4	8	23	14	-	50
$[200, 250)$	-	-	4	9	14	5	-	32
$[250, 300)$	1	1	5	9	9	-	-	25
$[300, 350)$	-	2	1	9	4	-	-	16
$[350, 400)$	-	-	5	6	1	-	-	12
$[400, 450)$	-	-	3	2	-	-	-	5
$[450, 500)$	-	2	2	1	1	-	-	6
$[500, 550)$	1	-	1	1	-	-	-	3
$[550, 600)$	-	2	1	-	-	-	-	3
$[600, 650)$	-	-	-	-	-	-	-	0
$[650, 700)$	-	-	1	-	-	-	-	1
Total	22	30	43	61	109	91	1	357

Table 2.5: *A summary of the total volume of stock in a wave compared to number of SKUs in the wave. The volume is given in ranges of 50 cubic meters and the number of SKUs is given in ranges of five SKUs. The total number of waves for each volume and number of SKUs segment is also given.*

Following these exclusions a number of historical problem instances are identified for use and are summarised in Table 2.6. For each problem instance a single wave will be assigned to a single unique picking line. From here on a wave/picking line combination will be represented by a unique picking line ID. Picking line IDs will therefore be different within each problem instance. To better compare results between different PLAP solution approaches several scenarios each having problem instances with the same number of available picking lines are generated from this historical data. The connected extraction window is only required for future studies. A set of disjointed problem instances is therefore used to generate these scenarios. Within each scenario the number of available picking lines for each problem instance is the same. These scenarios are generated by selecting a random subset of picking lines from each problem instance for each scenario (where possible). For example a historical problem instance with five picking lines is used to generate problem instances for four scenarios one with two, three, four and five picking lines respectively. A summary of these scenarios is given in Table 2.7. Although there are historical problem instances with more than eight picking lines (as shown in Table 2.6) there are two few to draw significant statistical comparisons between solution approaches. Only scenarios having problem instances with between two and eight picking lines is therefore used.

Number of picking lines per problem instance	Number of problem instances	Number of DBNs
2	9	385
3	5	329
4	9	742
5	7	745
6	11	1510
7	8	1248
8	7	1048
9	3	526
10	2	324
11	1	237
13	1	260

Table 2.6: *A summary of the makeup of the historical problem instances for the PLAP.*

Number of picking lines per problem instance	Number of problem instances	Number of DBNs
2	61	2592
3	53	3437
4	49	4146
5	38	4109
6	32	4161
7	22	3177
8	14	2148

Table 2.7: *The composition of the scenarios generated from historical problem instances for the PLAP.*

Approaches to the PLAP can be tested using these scenarios. Although it is assumed that for the problem instances one location is assigned to a SKU, PLAP solution approaches must be able to handle problem instances where multiple locations are assigned to a SKU for practical implementation. A test framework in which to test different approaches taking these assumptions into account is further discussed in the next section.

2.3 PLAP test framework

A test framework is required to simulate different decision making strategies for the PLAP. Any test framework should address two goals. Firstly new decision making strategies should only have access to information which is currently accessible or can be made accessible should the strategy be implemented. Secondly the effects of new decision strategies should be measurable and reflect realistic implications in the DC. Following the release date and time window data requirements 6 and 7 provision should be made for DBN scheduling in the test framework. A test framework is therefore designed which allows for the testing of approaches to the PLAP as well as the dynamic DBN scheduling problem.

All coding was done using the AIMMS optimisation suit [5]. The optimisation suite serves as an interface between data sources and mathematical solvers and offers interface design capabilities. The general framework for the test environment is described using a flow diagram shown in Figure 2.5. There are three main modules within the framework, namely the data input, decision test and data output modules.

A single iteration within the decision simulation module reflects one day in the DC. At the start of each iteration DBN and picking line data is obtained from an external database. This simulates the ongoing release of DBN pick instructions by the planning department and availability of new picking lines. Once data has been obtained the decision simulation module will assign DBNs to picking lines with the user defined approach. The performance of each picking line is evaluated after DBNs have been assigned to all the available picking lines. At the end of each iteration output data is recorded in the external database. Each of these three modules are further discussed with reference to the static PLAP and dynamic DBN scheduling functionalities.

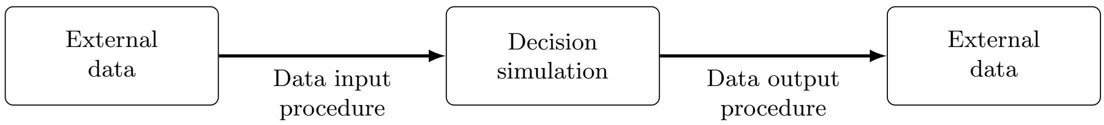


Figure 2.5: *A schematic representation of the main modules within the test framework for the PLAP.*

2.3.1 Data input

Two elements form part of the data input module. Firstly an SQL database stores the required data for the tests. Included in this database is the set of DBNs with the required information as well as the picking line information (historical or custom set). A separate database is used for each of the seven scenarios illustrated in Table 2.7. For each database three data tables are present, namely a DBN detailed table holding all the required data for each individual order; a DBN aggregated data holding aggregated data for each DBN as well as the release and schedule dates; and a picking line data holding information regarding historical picking lines.

The DBN detailed data table contains the

- DBN number,
- SKU number,
- store number,
- number of items,

- volume and
- historical picking line ID.

The DBN aggregated table contains the

- DBN number,
- SKU number,
- volume,
- number of required locations,
- release date,
- schedule date,
- out-of-DC date and
- historical picking line ID.

This data is mapped to the DBN detailed data table using a composite key consisting of the DBN and SKU numbers. The number of required locations field is included in the data to make provision for future problem instances where multiple locations are assigned to a SKU. This is a redundant field for the problem instances introduced in Section 2.2 as it is assumed that each SKU is assigned one location only.

The picking line data table consists of the

- picking line scenario ID,
- picking line ID,
- schedule date and
- number of SKUs.

Although the data is stored such that the entire set of DBNs may be seen as a single problem instance, each historical problem instance can be identified by the schedule date. Therefore each of the individual PLAP problem instances can still be solved and evaluated on its own using the schedule date as an identification key.

At the start of each test data is obtained from the external database. Two different data input procedures are developed for the different decision environments. The procedure for the static PLAP environment obtains all the picking line data for the picking lines which are scheduled for the specified test date. All the DBN data which maps to one of the now active picking lines is then obtained. In this way only DBNs which were historically scheduled on the specified test date are considered for a decision simulation iteration. In this way each problem instance for the PLAP is evaluated independently.

For the dynamic allocation environment DBNs can be scheduled on different dates. The data input procedure here requires a temporary data table containing a list of all the DBNs already scheduled for picking during one of the previous iterations or days. This table is created at the start of the test and is populated as DBNs are scheduled. The procedure will obtain all DBNs which have not yet been scheduled (*i.e.* not present in the temporary DBN data table) and have a release date before the next test date.

Once the list of pending DBNs has been updated the DBNs must be assigned to a set of pending picking lines. Both historical and custom generated scenarios can be used. Custom scenarios can be generated by changing the number and size of available picking lines for each day. These scenarios are stored in the picking line data table in the input database.

2.3.2 User input

Each test requires a number of user inputs dictating which data to be used, the type of decision environment as well as the algorithm to be tested. The following parameters are set for each test:

- Test ID
- Type of test (static PLAP or dynamic DBN scheduling)
- Problem type scenario (historical picking lines for the PLAP and custom scenarios for dynamic DBN scheduling)
- Algorithm identification.

During a test the model parameters must be set and assignment algorithms run by calling several procedures. One iteration of the decision simulation module is described using a flow diagram shown in Figure 2.6. Generic parameters and variables include picking line sizes, DBN assignment variables, SKU location variables and OSP variables among others. A number of unique parameters and variables may be required depending on the solution approach being tested. For each candidate solution approach only two procedures must be coded. Firstly a procedure for calculating new parameters and secondly a procedure for assigning DBNs to picking lines. A case statement is used within the decision simulation module to select the appropriate procedure for the tested solution approach. In this way new solution approaches to the PLAP can be easily implemented and tested once the test framework is developed by using algorithm identification keys.

2.3.3 Data output

During each test data is written to external output data tables. At the start of each test the meta-data regarding that test is recorded in a past tests table. This table is used to record information concerning each tested decision approach and contains the

- test ID (user defined),
- test description (user defined),
- type of test (PLAP or DBN scheduling),
- picking line data,
- algorithm identification and
- a completion flag.

The completion field is used for automation. The test environment can be set to run through all test which have not been completed. The appropriate user input is read from the past tests table. In this way many tests can be read into the test framework and evaluated in a single program run.

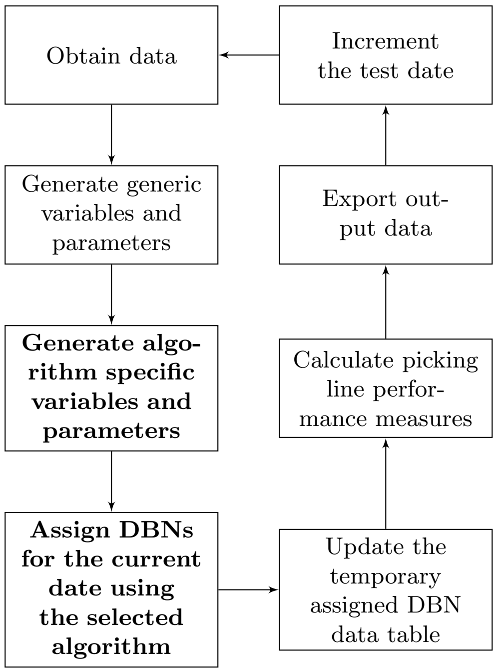


Figure 2.6: *A flow diagram of the decision simulation module within the PLAP test framework. Bold states indicate procedures which are algorithm specific.*

At the end of each problem instance regarding the algorithm performances and DBN assignments are recorded. Firstly picking line data is stored for each assignment in a picking line data table with the

- test ID,
- picking line ID,
- schedule date,
- volume,
- number of items,
- number of location visits,
- maximal SKU value,
- number of locations,
- number of SKUs,
- number of cycles walked and
- computational time.

A field for both the number of available locations as well as the number of SKUs is included to make provision for future data where SKUs are assigned multiple locations. From this output data table the total walking distance for each DBN assignment is obtained as well as the size of each picking line in terms of volume. Furthermore the required computational time to assign DBNs to each picking line is recorded for algorithm comparison.

The data recorded in the DBN data for each scheduled SKU are

- test ID,

- picking line ID,
- DBN number,
- SKU number,
- location ID,
- DBN release date,
- DBN schedule date and
- DBN out-of-DC date.

From this output data table analysis can be performed on the time spent in the DC for all DBNs for the dynamic DBN scheduling environment.

Finally, information for each order is stored in the store data table with the

- test ID,
- picking line ID,
- store number,
- volume and
- number of items.

The actual volume for each order is obtained from this output data table and is used to determine the number of small cartons produced.

By ensuring that each output record has a test ID tests can be deleted and rerun during their development phases. A test can easily be deleted from each data table for a test rerun. A subset of tests can be extracted for analysis after the model development and testing phases. Approaches to the PLAP can be tested using the historical problem instances as well as the proposed test framework. For the rest of this dissertation solution approaches to the SLP and PLAP are introduced and discussed as well as implementation issues when changing these decision making environments in the DC.

Bibliography

[1] Hagspihl R & Visagie S, 2014, *The number of pickers and stock-keeping unit arrangement on a uni-directional picking line*, The South African Journal of Industrial Engineering, **25(3)**, pp. 169–183.

[2] Matthews J & Visagie SE, 2013, *Order sequencing on a unidirectional cyclical picking line*, European Journal of Operational Research, **231(1)**, pp. 79–87.

[3] Matthews J & Visagie SE, *Picking Line Data*, Available from `http://scholar.sun.ac.za/handle/10019.1/86110`.

[4] Matthews J & Visagie SE, 2014, *SKU arrangement on a unidirectional picking line*, Submitted.

[5] Paragon Decision Technology, *AIMMS*, Available from `www.aimms.com`.

[6] Serdyn H, *Carton Sizes*, personal communication. Contactable at: `henniese@pepkorit.com`.

CHAPTER 3

SKU arrangement on a unidirectional picking line

3.1 Introduction

Distribution centres (DCs) play a key role in many supply chains. DCs typically match supply with demand by consolidating product, resulting in buffers of stock and a reduction in transportation costs [1]. The order pick operation plays a significant role within most DCs and typically accounts for 60% of all DC costs [18]. De Koster [3] describes order picking as the process of retrieving products from storage or buffer areas in a response to customer requests. The order picking operations for the DCs owned by PEP, a major retailer in South Africa, is considered in this paper. PEP has three distribution centres in Southern Africa, as well as 14 distribution hubs. Together the DCs occupy more than 230 000 m^2 and distribute over 600 million items per year across Southern Africa [14].

PEP preponderantly sells apparel, but has also been growing in the home décor and cellular device market. They serve a target market consisting mainly of the low income population in South Africa. In an effort to keep costs and prices low, PEP is known for its very efficient supply chain. PEP requires a large footprint of approximately 1500 retail outlets (stores) to reach its market. PEP adopted a central inventory planning and management approach to keep costs low with such a large number of stores. All inventory levels for the stores are managed centrally by the planning department at the head office. Stock is thus pushed to stores by a central planner rather than pulled by a store manager placing orders.

The order pick operation in PEP's DCs is greatly influenced by this central planning approach. All store requirements for the subset of stock keeping units (SKUs) scheduled to be picked during a specific operations window (typically weekly) will be released by a central planner to a DC for all stores. This allows the DC to process all store requirements for a single SKU in a single operation. PEP therefore batches SKUs for collective picking, rather than batching orders as is often the case in literature [4, 7, 12].

PEP uses 12 independent picking lines which operate in parallel to batch SKUs. Figure 3.1 schematically illustrates the layout of a single picking line with m (typically 56) locations. Multiple pickers (typically eight) are assigned to a picking line and walk in a clockwise direction picking stock. Collectively pickers walk approximately 720 km per day during picking which equates to 160 man hours of walking per day.

Before any picking can take place on a picking line a set of SKUs must be assigned to the

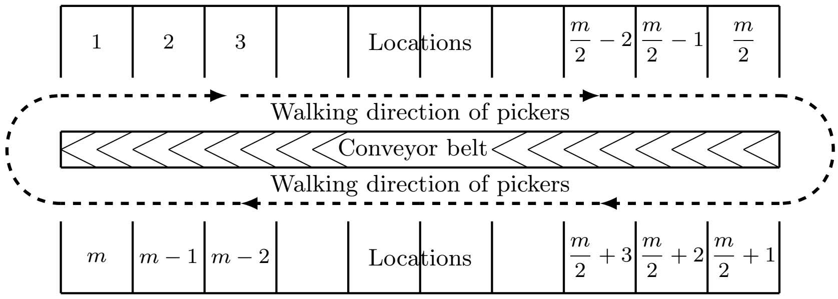


Figure 3.1: *A schematic representation of a picking line with m locations.*

picking line and arranged by allocating each SKU to a location. The physical stock is then brought to the picking line before pickers commence with picking. All store requirements for the SKUs on the picking line will be picked before the left over stock (if any) is removed and a new mutually exclusive set of SKUs is assigned to the picking line. One cycle of populating, picking and removing leftover stock is referred to as a wave of picking. The term *order* will refer to the set of picks required by a store for the set of SKUs within a wave of picking on a picking line. Because the store requirements for each SKU in a wave are known when a wave commences the orders are deterministic for each wave of picking.

This process of managing waves generates three sequential and dependant tiers of decisions. In the first decision tier SKUs are assigned to the available picking lines in a planning horizon (typically daily). SKUs can only be processed after central office has released all the store requirements for that SKU. The required quantities for all the SKUs that must be picked in a picking line are thus known when SKUs are assigned to a picking line. SKUs are ranked according to priority and a subset of the top ranked SKUs are selected to fill the available picking lines. Once this subset is selected these SKUs are assigned to individual picking lines. Managers assign these SKUs to each picking line based on experience and in-house rules.

The second decision tier occurs once a batch of SKUs has been assigned to a specific picking line and the SKUs are arranged in that picking line. Currently the arrangement of the SKUs is determined by picking line managers each of which using his/her own approach, but the main philosophy is to evenly distribute SKUs with a high number of branch orders around the picking line. Each of the m locations in the picking line can store up to five pallet loads of stock which is sufficient to satisfy all store requirements for each SKU in a wave. Restocking the picking line is thus not a problem as is normally the case in literature [3]. Figure 3.2 illustrates the first and second decision tiers.

The third and final decision tier occurs just before picking starts. Here the sequence in which orders are assigned to pickers is determined as each order does not require all the SKUs and may be assigned to any picker. PEP uses voice recognition software (VRS) to manage the order-picking and assign orders. Each picker is equipped with a headset which sends instructions to and receives feedback from the picker. The VRS assigns an order to a picker while the picker prepares a new empty carton. Empty cartons are available at any location in the picking line and have no affect on the optimisation of any decision tier. The picker places a unique identification sticker onto the empty carton and registers the identification number with the VRS. The VRS then pairs the identification number with the assigned order before directing the picker, in a clockwise direction, to the next required SKU for the assigned order.

The VRS keeps track of the positions of the pickers. Pickers are therefore directed to the closest SKU required by the currently assigned order. The closest SKU is the next required SKU for

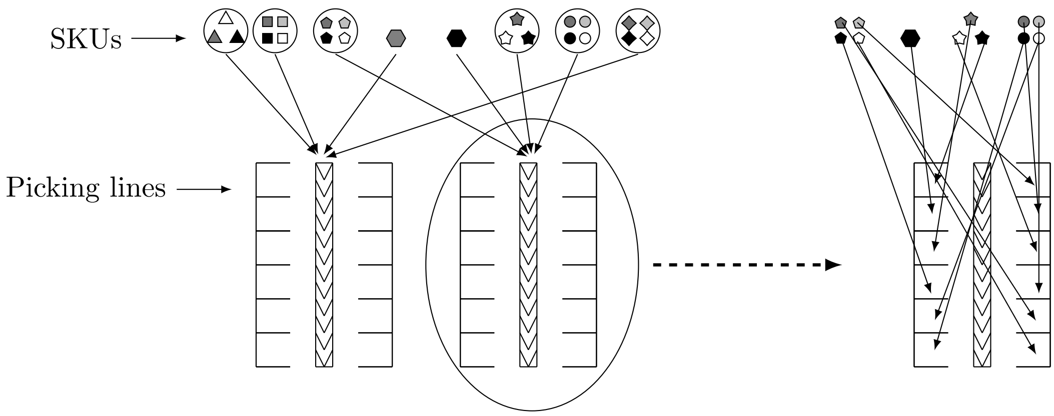


Figure 3.2: *A schematic representation of the first two decision tiers in the wave order picking operation. Decision tier 1 is shown on the left and decision tier 2 is shown on the right. Each shaded shape represents a SKU. Shapes are grouped in decision tier 1 for display purposes only.*

the current order in a clockwise direction from where the previous SKU was picked. Once a picker has completed all the picks for the order (which can take at most one pick cycle), the VRS assigns a new order to the picker. Each picker will therefore pick all the required SKUs for a single order before being assigned a new order to ensure pick accuracy. Pickers can work in parallel, but each order is only picked by a single picker as pickers sequentially pick his/her list of orders.

The focus of this paper is on the middle decision tier, namely the arrangement of SKUs within a picking line. Several heuristic approaches are tested and lower bounds introduced. The remainder of this paper is structured as follows: A brief problem description and discussion on work concerning the sequencing of orders in this picking line set-up is provided in §3.2. A brief discussion on related work in literature is provided in §3.3 and the details and adaptation of several heuristic methods from literature for use in this picking line environment are discussed in §3.4. A tight lower bound or an optimal solution is necessary to measure the performance of different heuristics. In §3.5 two mathematical formulations as well as a trivial approach for determining lower bounds, on the distance travelled by pickers are introduced. Computational results of all the heuristics for both historical and generated problem instances are presented in §3.6. After a discussion of the results the paper is finally concluded in §3.7.

3.2 Problem description

The SKU location problem (SLP) considers the arrangement of a set of SKUs in a single picking line while minimising the total distance walked by pickers. An optimal sequence of orders must be calculated for the arrangement to evaluate the total walking distance for a candidate SKU arrangement. Therefore both an approach to arranging SKUs within a picking line as well as their sequencing of orders for pickers is required to solve the SLP.

The problem of sequencing a set of orders for a picker in this unidirectional picking line set-up was considered by Matthews & Visagie [10]. They considered the following assumptions regarding the order sequencing process to align the mathematical model with the actual process in the DC:

1. Pickers are required to complete an entire order before starting another one. Order batching is not possible as pickers place items directly into cartons which are closed and shipped

as is.

2. The physical pick time of a SKU is constant over all orders.

3. A picker walks at a constant speed.

4. Pickers may be assigned an order, signifying the start of that order, at any location regardless of whether the order requires the SKU at that location or not. The order will be completed at the last location which holds a required SKU.

5. The change-over time when changing between two orders is constant over all pickers and orders.

6. Once a picker completes an order the picker may not physically pick stock for the next order from that location. The VRS system will therefore register the starting location of the next order as the following location in the picking line. In this way, should the next order require the last SKU of the previous order, the VRS will route the picker around the entire picking line. This assumption is made due to the requirements by PEP to improve picking accuracy.

Matthews & Visagie [10] showed that an exact approach to this order sequencing problem (OSP) was not solvable due to the size of real life instances. However, by translating the distance walked into the number of cycles walked by pickers they proposed an approach for determining a good number of cycles needed to pick all orders on a picking line with a fixed SKU arrangement. A sequence of orders obtained with this method was shown to be within one pick cycle (the distance walked by a picker passing all locations in a picking line) of the minimum number of pick cycles required to pick all the orders for the given SKU arrangement. This method to solve the OSP will thus be used to evaluate any candidate SKU arrangement to the SLP and is summarised below.

Given a SKU arrangement, spans (paths which cover all required SKUs for an order) are assigned to orders. The cut of each location, defined as the number of assigned spans which pass that location, is determined and the maximum number of spans passing any one location or maximal cut is minimised. Two different assignments of spans to orders are illustrated in Figure 3.3. For a span to be feasible for an order two conditions must hold: the span must pass all the locations which hold a SKU required by that order, and the span must start and end at a location holding a required SKU. It is important to note that the starting and ending locations for different spans do not necessarily link up, while determining the maximal cut.

Matthews & Visagie [10] showed that once spans have been assigned to orders while minimising the maximal cut these spans can be linked into a single tour connecting all spans within 1 cycle of the maximal cut. The maximal cut therefore forms a lower bound in terms of the number of cycles traversed to pick all the orders for the given SKU arrangement. The following variables are needed to model the assignment of spans to orders given a fixed SKU arrangement. Let

$$\hat{s}_l^o = \begin{cases} 1 & \text{if the span for order } o \text{ starts at location } l \\ 0 & \text{otherwise,} \end{cases}$$

and

C be the maximal cut.

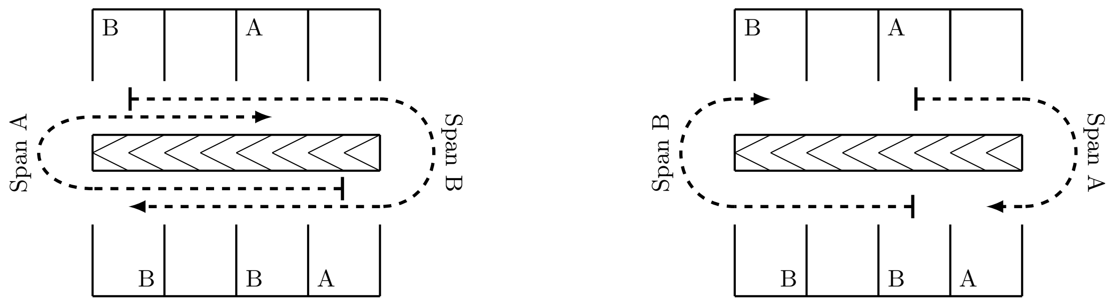


Figure 3.3: *A schematic representation of two different assignments of spans for the same set of orders. Each line segment is associated with a different order indicated by a letter. The letters in the locations indicate which locations hold SKUs required by each order. A span should always pass all the locations holding a required SKU for its order.*

The following parameters are set in the model. Let

$\mathcal{O}$ be the set of all orders,

$\mathcal{S}$ be the set of SKUs,

$\mathcal{L}$ be the set of locations,

$\hat{\mathcal{L}}_o$ be the set of locations which holds a SKU required by order o, and

$$\hat{d}^o_{lk} = \begin{cases} 1 & \text{if the span for order } o \text{ starting at location } l \text{ passes location } k \\ 0 & \text{otherwise.} \end{cases}$$

In terms of these symbols the objective is to

$$\text{minimise} \quad C \tag{3.1}$$

subject to

$$\sum_{l \in \hat{\mathcal{L}}_o} \hat{s}^o_l = 1 \qquad o \in \mathcal{O}, \tag{3.2}$$

$$\sum_{o \in \mathcal{O}} \sum_{l \in \hat{\mathcal{L}}_o} \hat{d}^o_{lk} \hat{s}^o_l \leq C \qquad k \in \mathcal{L}, \tag{3.3}$$

$$\hat{s}^o_l \in \{0, 1\} \qquad o \in \mathcal{O} \text{ and } l \in \mathcal{L}.$$

The objective function (3.1) minimises the maximal cut. Constraint set (3.2) ensures that each order is allocated a starting location which holds a required SKU. Constraint set (3.3) calculates the size of the maximal cut over all locations. The parameter $\hat{d}^o_{lk}$ is pre-calculated and the details of this calculation may be found in Matthews & Visagie [10].

Due to the size of the OSP alone it is clear that an exact solution approach to the more complex SLP is not solvable by means of an integer programming formulation as it depends on the OSP to evaluate candidate SKU arrangements. Heuristic approaches should thus be investigated. Related literature is therefore discussed in the next section and heuristics are identified for adaptation for the picking line system considered here.

3.3 Related literature

A carousel has a number of shelves which are linked together in a closed loop. This loop of shelves rotates automatically presenting stock to a picker (human or robotic) located in a fixed position [9]. The picking line considered here forms a type of carousel system in which the picking line forms a fixed carousel and multiple pickers “rotate” in a clockwise direction around it. Hassini [6] provides a review of various carousel problems and applications. According to the classification by Hassini [6] the picking line approach under consideration is a single layer, single bin, unidirectional carousel because the SKUs are not stored above each other, pickers process single orders sequentially and pickers walk in a clockwise direction around SKUs. The major difference between the picking line approach considered here and carousels in literature is the deterministic and finite nature of the set of orders being picked during a wave. Moreover bidirectional carousels are more common in practice, as they are most efficient, and thus receives the majority of the attention in the literature on carousels.

When sequencing orders on a bidirectional carousel Bartholdi & Platzman [2] showed that a solution within one rotation of a lower bound could be achieved by picking all orders on their shortest spanning intervals (SSIs). A SSI is the shortest path, in both directions, which needs to be passed in a bidirectional carousel to pick all the required SKUs for that order. Both a shortest matching technique as well as a nearest neighbour approach for sequencing the SSIs were tested. Van den Berg ([17]) later developed a polynomial algorithm for linking SSIs within 1.5 rotations of a lower bound. However, Matthews & Visagie [10] showed that for a unidirectional carousel picking orders on SSIs rarely yields solutions close to the optimum.

Once order sequencing can be performed in carousels, the location of SKUs on the carousels becomes the next focus. Vickson & Fujimoto [19] proved that using an organ pipe arrangement (OPA) minimizes the long-run average travel time for a sequence of single independent and identically distributed orders for a bidirectional carousel. Vickson & Lu [20] further showed that assigning SKUs to locations in a greedy sequential manner (GS) is optimal for the unidirectional case. Both of these SKU arrangements are applied to carousels with an infinite set of stochastic orders and have not been proven to be optimal for a finite deterministic set of orders considered here. Litvak & Maia [9] briefly mentions a SKU allocation approach by Stern [16] based on a maximal adjacency principle which places SKUs next to each other if the probability of them appearing in the same order is high.

Hagspihl & Visagie [5] used a simulation approach to test the effects of SKU arrangements on congestion in a picking line. Hagspihl & Visagie [5] proposed a Classroom Discipline (CD) heuristic which – using an ordered set based on pick frequency – sequentially inserts SKUs into the gaps between already assigned SKUs to reduce picker congestion. It was shown that congestion, in comparison to the GS and OPA heuristics as well as the historical configuration was reduced when using this heuristic, however, the effect of this heuristic on the number of cycles traversed was not tested.

The OPA, GS and CD heuristics as well as the SKU allocation approach by Stern [16] are suitable for adaptation (where necessary) and use in the picking line system considered here. These heuristics are therefore discussed in more detail in the next section.

3.4 Heuristics

Following the studies by Vickson & Fujimoto [19] and Vickson & Lu [20] the OPA and GS approaches are adapted for the picking line setup and presented in Algorithm 1 and Algorithm 2 respectively. Figure 3.4 illustrates the SKU arrangement for both the OPA and GS approaches with the aid of a small example.

Figure 3.4: *A schematic representation of the layouts for both the OPA and GS heuristics. The height of each bar represents the probability that an order will require a SKU for carousel systems or the actual number of orders requiring that SKU in the case of the picking line under consideration.*

Algorithm 1: Organ Pipe Heuristic

Data: A set of SKUs $\mathcal{S}$ ordered by pick frequency with s_i the i^{th} element of $\mathcal{S}$
Result: An assignment of SKUs to locations

Let $\mathcal{T}$ be an ordered sets of SKUs
Set $\mathcal{T} = \emptyset$
for i = *1 to* $|\mathcal{S}|$ **do**
  if $i \mod 2 = 1$ **then**
    Set $\mathcal{T} = \mathcal{T} \cup s_i$; `/* where` $\mathcal{T} \cup s_i$ `implies that` s_i `is added to the front of` $\mathcal{T}$ `*/`
  end
  else
    Set $\mathcal{T} = s_i \cup \mathcal{T}$; `/* where` $s_i \cup \mathcal{T}$ `implies that` s_i `is added to the back of` $\mathcal{T}$ `*/`
  end
end
Assign SKUs to the locations in the same order as they appear in $\mathcal{T}$

Algorithm 2: Greedy Sequential Heuristic

Data: A set of SKUs $\mathcal{S}$ ordered by pick frequency
Result: An assignment of SKUs to locations

Assign SKUs to the locations in the same order as they appear in $\mathcal{S}$

Stern [16] used an adjacency measure to assign SKUs to locations in a carousel. In the carousel environment the probability of two SKUs being required by the same order is used as an adjacency measure. A Greedy Adjacency (GA) heuristic is therefore introduced in this deterministic picking line environment by considering the actual number of orders requiring two SKUs as the number of adjacencies between those two SKUs. The GA heuristic initially allocates the SKU with the highest pick frequency to the first location and then sequentially allocates the SKU with the highest number of adjacencies with the previously allocated SKU to the next available location. The GA heuristic is described in Algorithm 3 and the SKU arrangement for a small example is illustrated in Figure 3.5.

Adjacencies

	A	B	C	D	E
A	20	10	11	4	5
B	10	15	5	6	1
C	11	5	12	8	2
D	4	6	8	10	1
E	5	1	2	1	6

Figure 3.5: *An illustration of how the GA assigns SKUs to locations based on adjacencies. The adjacency matrix on the left indicates the number of adjacencies between each SKU. The diagonal indicates the total number of orders requiring that SKU. The figure on the right illustrates the final allocation of the five SKUs. The height of each bar represents the number of orders requiring that SKU.*

Algorithm 3: Greedy Adjacencies Heuristic

Data: A set of SKUs $\mathcal{S}$
  A adjacency matrix (A) where element (a_{ij}) represents the number of orders requiring both SKUs $s_i \in \mathcal{S}$ and $s_j \in \mathcal{S}$
Let $\mathcal{L}$ be a set of locations with element l_i representing the i^{th} element of $\mathcal{L}$
Result: An assignment of SKUs to locations

Assign the SKU with the largest number of orders requiring it to location l_1
for $p = 2$ *to* $|\mathcal{L}|$ **do**
  Assign to location l_p the unassigned SKU with the highest number of adjacencies to the SKU assigned to location l_{p-1}
end

Hagspihl & Visagie [5] tested the CD heuristic on the same picking line system considered in this paper. Although it was shown that this approach reduces the effects of picker congestion the effect of this heuristic on the number of cycles traversed was not tested. This heuristic is therefore included in this paper and is described in Algorithm 4. A small example of its implementation is illustrated in Figure 3.6. The algorithm sequentially assigns SKUs in a greedy sequence to locations by placing them in-between already assigned SKUs.

Algorithm 4: Classroom Discipline Heuristic

Data: A set of SKUs $\mathcal{S}$ ordered by pick frequency
Result: An assignment of SKUs to locations

Let $\mathcal{T}$ and $\mathcal{U}$ be ordered sets of SKUs, with u_j the j^{th} element of $\mathcal{U}$
Set $\mathcal{T} = \emptyset$
for i = *1 to* $\lceil \log_2 (|\mathcal{S}| + 1) \rceil$ **do**
  Set $\mathcal{U}$ equal to the top $2i - 1$ SKUs in $\mathcal{S} \setminus \mathcal{T}$;   `/* The set of unasigned SKUs */`
  for $j = 1$ *to* $|\mathcal{U}|$ **do**
    if $i = 1 \mod 2$ **then**
      Insert u_j into position $2j - 1$ of $\mathcal{T}$
    end
    else
      Insert u_j into position $|\mathcal{T}| - 2j + 3$ of $\mathcal{T}$
    end
  end
  Assign SKUs to the locations in the same order as they appear in $\mathcal{T}$
end

To evaluate the performance of the heuristics presented here a tight lower bound or optimal solution is necessary. Several approaches to determine a tight lower bound are therefore presented in the next section.

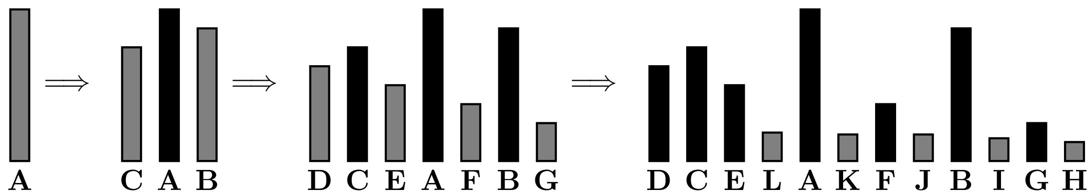


Figure 3.6: *A schematic representation of the Classroom Discipline (CD) heuristic for a picking line with 12 SKUs (A-L). The arrows indicate the transition between groups of SKU insertions. The grey bars indicate those SKUs that are inserted during the current round of insertion and the height of the bars indicate the number of orders requiring that SKU. Depending on the number of SKUs in the line the last iteration of insertion does not necessarily fill all possible gaps between previously inserted SKUs.*

3.5 Lower bounds

Utilising the maximal cut approach for the OSP when solving the SLP would require each SKU to be assigned to a location after which spans, defined by starting locations, need to be assigned to spans. Each different SKU arrangement redefines the set of possible starting positions for (and thus lengths of) candidate spans for each order as all orders do not necessarily require all SKUs. It is only once these spans have been allocated that the maximal cut can be determined.

A trivial approach to determining a lower bound for any (minimisation) mathematical programming problem is to reduce the problem size by relaxing some constraints. A lower bound for the SLP can therefore be obtained by only considering a subset of the SKUs and the subsequent set of orders. Moreover if only one SKU is selected from the full set of SKUs no SKU arrangement is necessary. Following Assumption 6 the presence of only one SKU implies that the number of cycles required to pick all the store requirements for the picking line is precisely the number of stores requiring that SKU. A trivial lower bound (TRLB) to the SLP may therefore be seen as the number of stores requiring the SKU with the highest pick frequency (its number of order visits).

If a larger subset of SKUs was considered for use in a lower bound an exhaustive search of all non-isomorphic SKU arrangements must be performed, or a mathematical approach devised which can narrow the search for the best arrangement. Two approaches for determining the optimal arrangement of a set of SKUs are further described in this section each one using a different approach to evaluating the number of cycles traversed.

3.5.1 A travelling salesman approach

When considering the physical layout of the picking line the assignment of SKUs to locations resembles a travelling salesman problem (TSP). A formulation, which will be referred to as the TSP lower bound (TSPLB), is presented which uses a TSP approach to determine which SKUs are adjacent to each other on the picking line or neighbours in the TSP cycle. Using this TSP solution representation for the SKU arrangement further variables and constraints are added to evaluate the walking distance of pickers. This is achieved by adding variables which assign spans to all the orders and constraints which ensure that each span follows the route imposed by the TSP subtour breaking constraints. Once the spans for each order have been assigned the maximal cut can then be calculated. Figure 3.7 illustrates the methodology with two SKU configurations. The solid lines indicate the adjacency assignments by the TSP variables and constraints while the dashed lines indicate the span associated with an order. The span for any

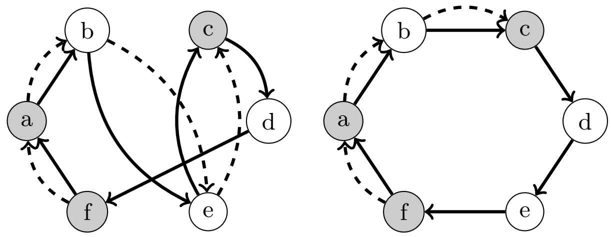


Figure 3.7: *A graphical representation of two examples of the TSPLB formulation configurations for six SKUs and a single order. The solid lines indicate that the SKUs are placed adjacent to each other. Grey nodes indicate that the order requires that SKU. Dashed lines indicate the span of the order (starting at SKU f) for the specific SKU arrangement.*

order can only follow the direction of the solid (TSP cycle) lines.

To model the SLP using this TSP approach let

$$\hat{x}_{st} = \begin{cases} 1 & \text{if SKU } s \text{ is adjacent to SKU } t \text{ in a clockwise direction} \\ 0 & \text{otherwise,} \end{cases}$$

$$p^o_{st} = \begin{cases} 1 & \text{if the span of order } o \text{ includes the arc from SKU } s \text{ to SKU } t \\ 0 & \text{otherwise,} \end{cases}$$

$\hat{l}_s$ be the location number of SKU s

and

C be the maximal cut.

The following parameters are set in the model. Let

$\mathcal{O}$ be the set of all orders,

$\mathcal{S}$ be the set of SKUs,

$\hat{\mathcal{S}}_o$ be the set of SKUs required by order o, and

$\hat{\mathcal{O}}_s$ be the set of orders requiring SKU s.

In terms of these symbols the objective is to

$$
\begin{aligned}
&\text{minimise} \quad C && && (3.4)\\
&\text{subject to} && && \\
&\sum_{s\in\mathcal{S}} \hat{x}_{st} = 1 && t \in \mathcal{S}, && (3.5)\\
&\sum_{t\in\mathcal{S}} \hat{x}_{st} = 1 && s \in \mathcal{S}, && (3.6)\\
&(|\mathcal{S}|-1)\cdot \hat{x}_{st} + \hat{l}_s - \hat{l}_t \leq |\mathcal{S}|-2 && s,t \in \mathcal{S}, s \neq 1 \text{ and } t \neq 1, && (3.7)\\
&\hat{l}_s \geq 1 && s \in \mathcal{S} \text{ and } s \neq 1, && (3.8)\\
&\hat{l}_s \leq |\mathcal{S}|-1 && s \in \mathcal{S} \text{ and } s \neq 1, && (3.9)\\
&\sum_{s\in\mathcal{S}} \sum_{t\in\hat{\mathcal{S}}_o} p^o_{st} = |\hat{\mathcal{S}}_o| - 1 && o \in \mathcal{O}, && (3.10)\\
&\sum_{s\in\hat{\mathcal{S}}_o} \sum_{t\in\mathcal{S}} p^o_{st} = |\hat{\mathcal{S}}_o| - 1 && o \in \mathcal{O}, && (3.11)\\
&\sum_{s\notin\hat{\mathcal{S}}_o} \sum_{t\in\mathcal{S}} p^o_{st} = \sum_{s\in\mathcal{S}} \sum_{t\notin\hat{\mathcal{S}}_o} p^o_{st} && o \in \mathcal{O}, && (3.12)\\
&\sum_{o\in\mathcal{O}} p^o_{st} \leq |\mathcal{O}|\cdot \hat{x}_{st} && s,t \in \mathcal{S}, && (3.13)\\
&C \geq \sum_{o\notin\hat{\mathcal{O}}_s} \sum_{t\in\mathcal{S}} p^o_{st} + |\hat{\mathcal{O}}_s| && s \in \mathcal{S}, && (3.14)\\
&\hat{x}_{st} \in \{0,1\} && s,t \in \mathcal{S}, && \\
&p^o_{st} \in \{0,1\} && s,t \in \mathcal{S} \text{ and } o \in \mathcal{O}, && \\
&\hat{l}_s \in \mathrm{Z} && s \in \mathcal{S}. &&
\end{aligned}
$$

The objective function (3.4) minimises the maximal cut as defined by Matthews & Visagie [10]. Constraint sets (3.5) and (3.6) ensure that each SKU is assigned an adjacent SKU both to the left and to the right of itself. The subtours are broken by constraint sets (3.7)–(3.9) which are based on the MTZ subtour breaking constraints [15]. These subtour breaking constraints ensure that the assignment of adjacent SKUs forms a single cycle. Constraint sets (3.5)–(3.9) for the TSP constraints and determine which SKUs are adjacent to each other on the picking line. The spans for each order are now considered.

Constraint sets (3.10) and (3.11) ensure that a span for each order starts and ends at a required SKU. This is achieved by limiting the total number of in-degrees and out-degrees for all required SKUs for the span of an order to one less than the total number of required SKUs by the order. Figure 3.7 illustrates the effect of these constraints as the dotted arcs do not necessarily form a cycle. Constraint set (3.12) ensures that if a span passes a SKU which it is not required by the corresponding order the span will proceed to the next SKU. All orders must be assigned spans which follow the cycle defined by constraint sets (3.5)–(3.8). This is achieved by constraint set (3.13) which only allows spans to pass from one SKU to another if the SKUs are adjacent in the TSP. Finally the maximal cut is determined by constraint set (3.14). The cut for each SKU (or location which holds the SKU) is determined by adding the number of orders which require the SKU to the number of orders which do not require the SKU, but have spans which pass the SKU.

3.5.2 An assignment approach

An assignment model (ASLB) approach was investigated which uses assignment variables and constraints to assign SKUs to locations. Similarly to the TSPLB approach additional variables and constraints are added to evaluate the walking distance. For the ASLB approach, however, there are no SKU adjacency variables or subtour breaking constraints and therefore a different structure of span assignment variables and constraints are needed to assign spans to orders. The following formulation is presented to link the assignment of SKUs with the assignment of spans to orders. Three sets of decision variables are defined. Let

$$x_{sl} = \begin{cases} 1 & \text{if SKU } s \text{ is assigned to location } l \\ 0 & \text{otherwise,} \end{cases}$$
$$\hat{s}_l^o = \begin{cases} 1 & \text{if the span for order } o \text{ starts at location } l \\ 0 & \text{otherwise,} \end{cases}$$
$$d_l^o = \begin{cases} 1 & \text{if the span for order } o \text{ passes location } l \\ 0 & \text{otherwise} \end{cases}$$

and

C be the maximal cut.

The following parameters are set in the model. Let

$\mathcal{O}$ be the set of all orders,
$\mathcal{S}$ be the set of SKUs,
$\mathcal{L}$ be the set of locations,
$\hat{\mathcal{S}}_o$ be the set of SKUs required by order o, and
$\hat{\mathcal{O}}_s$ be the set of orders requiring SKU s.

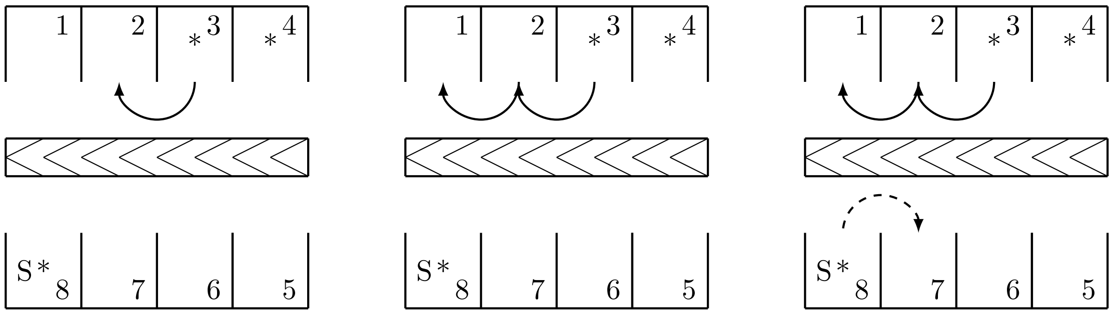
In terms of these symbols the objective is to

$$\begin{aligned}
&\text{minimise} \quad C && && (3.15)\\
&\text{subject to} && &&\\
&\sum_{s\in\mathcal{S}} x_{sl} = 1 && l \in \mathcal{L}, && (3.16)\\
&\sum_{l\in\mathcal{L}} x_{sl} = 1 && s \in \mathcal{S}, && (3.17)\\
&\sum_{l\in\mathcal{L}} \hat{s}^o_l = 1 && o \in \mathcal{O}, && (3.18)\\
&\sum_{s\in\hat{\mathcal{S}}_o} x_{sl} \leq d^o_l && o \in \mathcal{O} \text{ and } l \in \mathcal{L}, && (3.19)\\
&d^o_l \geq d^o_{l+1 \bmod |\mathcal{L}|} - \hat{s}^o_{l+1 \bmod |\mathcal{L}|} && o \in \mathcal{O} \text{ and } l \in \mathcal{L}, && (3.20)\\
&\sum_{o\in\mathcal{O}} d^o_l \leq C && l \in \mathcal{L}, && (3.21)\\
&x_{sl} \in \{0,1\} && s \in \mathcal{S} \text{ and } l \in \mathcal{L}, &&\\
&\hat{s}^o_l \in \{0,1\} && o \in \mathcal{O} \text{ and } l \in \mathcal{L}, &&\\
&d^o_l \in \{0,1\} && o \in \mathcal{O} \text{ and } l \in \mathcal{L}. &&
\end{aligned}$$

Similar to the previous formulation, the objective function (3.15) minimises the maximal cut. Constraint sets (3.16) and (3.17) ensure that each location is assigned to a single SKU and each SKU is assigned to a single location as there will be the same number of SKUs and locations in a picking line. Constraint set (3.18) assigns a starting location to each span. Constraint set (3.19) ensures that spans for orders will pass all locations which have been assigned a SKU required by that order. Constraint set (3.20) determines the set of locations which must be passed to pick all the required SKUs of an order given the starting position assigned to its span. If a span is required to pass location l and the span does not start at location l then the preceding location $(l-1)$ must also be passed. If a span starts at a location l the preceding location $(l-1)$ would only be passed if it held a SKU required by the corresponding order and if an order starts at the first location the last location would only be passed if it held a required SKU. Finally, constraint set (3.21) calculates the maximal cut.

Constraint set (3.20) is further illustrated with the aid of a small example consisting of eight locations and a single order (o) shown in Figure 3.8. All locations holding a SKU must be passed by the span for order o including location 3 which implies $d^o_3 = 1$. For a picker to arrive at location 3 walking in a clockwise direction location 2 must also be passed. Constraint set (3.20) therefore forces $d^o_2 = 1$ as shown by the solid arrow in Figure 3.8 (a). Location 2 does not hold a required SKU but must now be passed. This implies that location 1 must also be passed for a picker to reach location 2 shown by the solid arrow in Figure 3.8 (b). Location 8 must be passed as it holds a required SKU. However, because it is the starting position of the order the picker does not need to pass location 7 to arrive at location 8. Constraint set (3.20) will therefore set $d^o_7 = 0$ indicated by the dashed arrow in Figure 3.8 (c).

Both the TSPLB and ASLB represent formulations which cover the entire solution space of candidate SKU arrangements. For each SKU arrangement a span is assigned to each order while minimising the maximal cut. Therefore each candidate SKU arrangement is evaluated using the maximal cut lower bound which is within one cycle the minimum number of cycles required for the specific SKU arrangement. Both of these formulations therefore form a lower



(a) (b) (c)

Figure 3.8: *A schematic representation of cuts described by equation set* (3.20) *for a single order. An asterisk indicates a location that holds a SKU required by order o and S indicates the assigned starting location. The span for this order thus runs from location 8 to location 4 in a clockwise direction.*

bound to the SLP which is within one pick cycle of an optimal SKU arrangement.

3.6 Results

In this section the problem instances used to test both the heuristic approaches as well as the lower bound approaches are introduced. A comparison is made between the different lower bound approaches and the results presented. Finally the performances of the heuristic approaches are compared and evaluated based on historical results and random arrangements.

3.6.1 Problem instances

A set of 22 historical problem instances based on historical data obtained from PEP was used [11]. All three of PEP's DCs function with the same methodology, but for the purpose of this paper the data for the largest one of the three, located in Durban (South Africa), is considered. A further 20 generated problem instances are also used to compare the approaches. These problem instances are included in an effort to avoid the bias (if any) of the historical data when comparing approaches. All tests were performed on an Intel i7 2GHz processor with eight GB ram running the Windows 7 operating system. All mathematical formulations were solved with CPLEX 12.3 and coded in AIMMS 3.12 [8, 13].

Each generated problem instance only comprised of 20 SKUs and 100 orders which is significantly smaller than the historical problem instances due to the complexity of the mathematical formulations used to generate lower bounds. This was done to draw comparisons between the TSPLB and ASLB approaches. Furthermore, the generated problem instances where divided into for sets:

UND: Instances generated by assigning SKUs to orders where a SKU is assigned to an order with a probability of 0.75.

UNS: Instances generated by assigning SKUs to orders where a SKU is assigned to an order with a probability of 0.25.

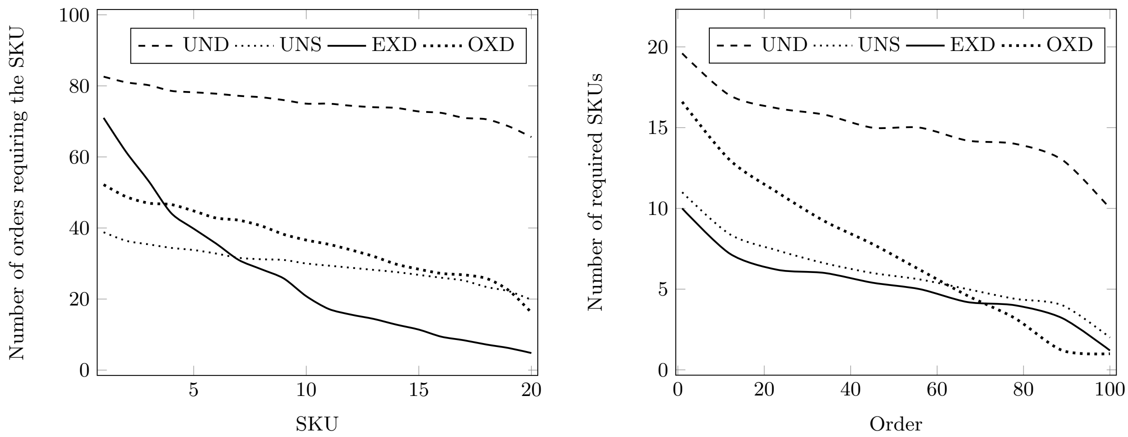


Figure 3.9: *A graphical representation of the relative size of each generated problem instance in terms of number of different SKUs required by each order and vice versa.*

EXD: Instances generated by assigning orders to SKUs where an order is assigned to a SKU with varying probability. Each order has a fixed probability of being assigned to a SKU, irrespective of the SKU, and are distributed evenly across the ranges $0.75 - 0.1$.

OXD: Instances generated by assigning SKUs to orders where a SKU is assigned to an order with varying probability. Each SKU has a fixed probability of being assigned to an order, irrespective of the order, and are distributed evenly across the ranges $0.9 - 0.15$.

A graphical representation of the configuration of the different generated problem instances is given in Figure 3.9. Most of the larger historical problem instances resemble the SKU frequency pattern of the EXD set of problem instances, but most historical problem instances have extended tails. Many of the smaller historical problem instances, however, resemble the SKU frequency distribution of the UNS set of problem instances.

3.6.2 Lower bounds

Initial results showed that the lower bound formulations could not be solved for the entire set of SKUs for each problem instance due to their computational complexity. A subset of SKUs was therefore used and the size of the subset increased (in an attempt to improve the lower bound) until computational times reached a threshold of two hours. SKUs were selected for inclusion in the subset on the basis of highest pick frequency. The solutions obtained when solving subsets of SKUs for the generated problem instance with the TSPLB and ASLB formulations are shown in Table 3.1. For the UND, UNS and OXD sets of problem instances the trivial lower bound was strengthened by using the proposed formulations. Moreover the lower bounds are strengthened as the sizes of the subset of SKUs increases. Note that for four instances of the EXD set of problem instances an optimal solution was found because the best known solution equals the lower bound.

The computational times required to solve for subsets of SKUs for the generated problem instances are shown in Table 3.2. The ASLB outperforms the TSPLB as in most cases the TSPLB could not solve instances with more than eight SKUs within the two hour time limit. The UNS set of problem instances require the longest computational times for both the TSPLB and ASLB approaches. The convergence to an optimal solution in the mathematical formulations is thus computationally more expensive with a sparse allocation of SKUs to orders.

Problem instance	Size $(\|\mathcal{S}\|, \|\mathcal{O}\|)$	TRLB	Best kown solution	TSPLB 5	TSPLB 8	TSPLB 10	TSPLB 12	ASLB 5	ASLB 8	ASLB 10	ASLB 12
				Number of SKUs in sample							
UND_1	(20,100)	82	90	82	83	84	-	82	83	84	85
UND_2	(20,100)	85	91	85	85	-	-	85	85	86	87
UND_3	(20,100)	83	91	84	85	-	-	84	85	86	87
UND_4	(20,100)	83	90	83	85	86	-	83	84	85	86
UND_5	(20,100)	80	90	83	84	85	-	82	84	85	86
UNS_1	(20,100)	37	63	39	45	-	-	39	45	49	-
UNS_2	(20,100)	38	64	40	46	-	-	40	47	50	-
UNS_3	(20,100)	39	66	41	48	-	-	41	48	51	-
UNS_4	(20,100)	39	65	41	47	-	-	41	47	51	-
UNS_5	(20,100)	41	65	43	48	-	-	43	48	51	-
EXD_1	(20,100)	71	71^*	71	71	71	71	71	71	71	71
EXD_2	(20,100)	79	79^*	79	79	79	79	79	79	79	79
EXD_3	(20,100)	73	73^*	73	73	73	73	73	73	73	73
EXD_4	(20,100)	64	69	64	65	-	-	64	65	66	66
EXD_5	(20,100)	68	68^*	68	68	68	68	68	68	68	68
OXD_1	(20,100)	54	62	54	54	-	-	54	54	55	56
OXD_2	(20,100)	50	63	51	55	-	-	51	55	56	57
OXD_3	(20,100)	53	65	53	56	-	-	53	56	57	-
OXD_4	(20,100)	56	62	56	56	-	-	56	56	57	57
OXD_5	(20,100)	48	62	51	53	-	-	51	53	55	-

Table 3.1: *A comparison of the value of the maximal cut when solving generated problem instances for different SKU sample set sizes for both the TSPLB and ASLB approaches. Entries with a dash indicate that no solution was obtained within two hours. Known optimal solutions are indicated with an asterisk.*

Problem instance	Size $(\|\mathcal{S}\|, \|\mathcal{O}\|)$	TSPLB 5	TSPLB 8	TSPLB 10	TSPLB 12	ASLB 5	ASLB 8	ASLB 10	ASLB 12
		Number of SKUs in sample							
UND_1	(20,100)	6	478	1503	-	1	40	240	2912
UND_2	(20,100)	3	32	-	-	1	11	25	73
UND_3	(20,100)	4	516	-	-	1	30	119	5976
UND_4	(20,100)	5	506	3864	-	1	12	207	2730
UND_5	(20,100)	11	674	2122	-	1	27	193	3865
UNS_1	(20,100)	1	515	-	-	1	39	434	-
UNS_2	(20,100)	1	681	-	-	2	37	396	-
UNS_3	(20,100)	3	743	-	-	1	75	310	-
UNS_4	(20,100)	2	848	-	-	1	41	420	-
UNS_5	(20,100)	1	522	-	-	1	44	403	-
EXD_1	(20,100)	3	29	114	139	1	1	4	3
EXD_2	(20,100)	2	46	68	168	1	2	5	9
EXD_3	(20,100)	4	33	130	213	1	1	2	10
EXD_4	(20,100)	3	1404	-	-	1	28	46	112
EXD_5	(20,100)	3	42	78	122	1	3	2	2
OXD_1	(20,100)	4	32	-	-	1	13	192	2945
OXD_2	(20,100)	2	479	-	-	1	65	302	4585
OXD_3	(20,100)	2	892	-	-	1	60	313	-
OXD_4	(20,100)	2	38	-	-	1	10	302	3199
OXD_5	(20,100)	2	544	-	-	1	42	525	-

Table 3.2: *A comparison of the computation times (rounded to the nearest second) when solving generated problem instances for different SKU sample set sizes for both the TSPLB and ASLB approaches. Entries with a dash indicate that no solution was obtained within two hours.*

Problem instance	Size $(\lvert\mathcal{S}\rvert, \lvert\mathcal{O}\rvert)$	TRLB	Best kown solution	TSPLB				ASLB			
				Number of SKUs in sample							
				5	8	10	12	5	8	10	12
L_1	(49, 1262)	1232	1232*	1232	1232	1232	1232	1232	1232	1232	1232
L_2	(54, 1264)	1226	1226*	1226	1226	1226	1226	1226	1226	1226	1226
L_3	(51, 1265)	1161	1161*	1161	1161	1161	1161	1161	1161	1161	1161
L_4	(56, 1263)	1011	1011*	1011	-	1011	1011	1011	1011	1011	1011
L_5	(51, 1264)	1069	1069*	1069	1069	1069	1069	1069	1069	1069	1069
L_6	(53, 1258)	959	1002	959	959	959	-	959	959	959	959
L_7	(56, 1260)	855	955	855	855	855	855	855	855	855	855
L_8	(54, 1244)	817	974	817	817	817	-	817	817	817	817
L_9	(56, 1264)	729	947	729	729	-	-	729	729	729	-
L_{10}	(55, 1258)	835	872	835	835	835	-	835	835	835	835
M_1	(63, 943)	95	248	99	-	-	-	99	105	110	-
M_2	(56, 846)	141	221	141	-	-	-	141	142	148	-
M_3	(51, 728)	109	149	109	109	109	-	109	109	109	-
M_4	(63, 396)	74	74*	74	74	74	74	74	74	74	74
M_5	(55, 733)	66	114	66	66	66	66	66	66	66	66
M_6	(64, 242)	33	41	33	33	33	33	33	33	33	33
M_7	(48, 574)	67	74	67	67	67	67	67	67	67	67
S_1	(48, 90)	7	7*	7	7	7	7	7	7	7	7
S_2	(55, 158)	13	14	13	13	13	13	13	13	13	13
S_3	(51, 82)	8	8*	8	8	8	8	8	8	8	8
S_4	(56, 80)	5	6	5	5	5	5	5	5	5	5
S_5	(42, 89)	9	9*	9	9	9	9	9	9	9	9

Table 3.3: *A comparison of the value of the maximal cut when solving historical problem instances for different SKU sample set sizes for both the TSPLB and ASLB approaches. Entries with a dash indicate that no solution was obtained within two hours. Known optimal solutions are indicated with an asterisk.*

The TSPLB and ASLB approaches were further used to solve for subsets of SKUs for the historical problem instances and the solutions obtained are shown in Table 3.3. For most of the instances of the historical problem instances a subset of up to 12 SKUs does not yield a stronger lower bound than the trivial lower bound. Note that for the largest and smallest instances of the historical problem instances a feasible solution is known which reaches the trivial lower bound. Although stronger lower bounds are found for the generated problem instances when using either the TSPLB or ASLB formulations for a subset of SKUs this is rarely the case for the historical problem instances. Either a stronger lower bound does not exists (as for problem instances L_1–L_5, S_1, S_3, S_5) or a larger subset of SKUs is required rendering the tighter lower bound to computationally expensive.

The number of orders in the model does not significantly influence the relative computational times when comparing the computational times between the historical (Table 3.4) and the smaller generated problem instances (Table 3.2). Moreover the set of SKUs shared between orders plays a significant role in the strength of the trivial lower bound as well as the required computational times to solve the TSPLB and ASLB formulations.

For all lower bound solutions only a subset of the total set of SKUs was selected for the mathematical models. These SKUs were selected based on their pick frequency and may not be the best set of SKUs with which to determine a lower bound. Therefore, for all the problem instances (both generated and historical) for which an optimal solution was not known the effect of selecting a different subset of SKUs not based on pick frequency was further investigated. Subsets of SKUs were generated by selecting SKUs randomly while ensuring that for each subset the SKU with the highest pick frequency was included to ensure that at least the trivial

Problem instance	Size ($\|\mathcal{S}\|, \|\mathcal{O}\|$)	TSPLB 5	TSPLB 8	TSPLB 10	TSPLB 12	ASLB 5	ASLB 8	ASLB 10	ASLB 12
		Number of SKUs in sample							
L_1	(49, 1262)	559	90	122	170	1	25	4	52
L_2	(54, 1264)	322	106	147	292	1	22	37	40
L_3	(51, 1265)	253	108	160	261	4	37	76	88
L_4	(56, 1263)	454	-	293	1747	32	40	139	380
L_5	(51, 1264)	205	93	180	286	28	69	111	567
L_6	(53, 1258)	325	99	167	-	30	74	412	199
L_7	(56, 1260)	255	4084	163	281	32	57	207	718
L_8	(54, 1244)	417	4733	399	-	1	238	-	-
L_9	(56, 1264)	153	5006	-	-	47	227	-	-
L_{10}	(55, 1258)	344	4982	474	-	18	97	2250	-
M_1	(63, 943)	25	-	-	-	28	718	2732	-
M_2	(56, 846)	10	-	-	-	8	424	2747	-
M_3	(51, 728)	2	6	878	-	2	21	59	-
M_4	(63, 396)	1	20	198	603	1	2	7	-
M_5	(55, 733)	2	16	146	484	1	14	26	-
M_6	(64, 242)	1	4	20	475	1	2	4	-
M_7	(48, 574)	2	13	209	768	3	15	59	-
S_1	(48, 90)	1	1	1	1	1	1	1	2
S_2	(55, 158)	1	1	2	3	1	1	3	4
S_3	(51, 82)	1	1	1	1	1	1	1	2
S_4	(56, 80)	1	1	1	15	1	1	1	4
S_5	(42, 89)	1	1	1	1	1	1	2	2

Table 3.4: *A comparison of the computational times (rounded to the nearest second) when solving historical problem instances for different SKU sample set sizes for both the TSPLB and ASLB approaches. Entries with a dash indicate that no solution was obtained within two hours.*

Problem instance	Size ($\lvert\mathcal{S}\rvert, \lvert\mathcal{O}\rvert$)	Ordered subset	Random subset Min	Max	μ	σ	Problem instance	Size ($\lvert\mathcal{S}\rvert, \lvert\mathcal{O}\rvert$)	Ordered subset	Random subset Min	Max	μ	σ
UND_1	(20,100)	83	82	82	82.00	0.00	L_6	(53, 1258)	959	959	959	959	0
UND_2	(20,100)	85	85	85	85.00	0.00	L_7	(56, 1260)	855	855	855	855	0
UND_3	(20,100)	85	83	84	83.50	0.58	L_8	(54, 1244)	817	817	817	817	0
UND_4	(20,100)	84	83	84	83.50	0.58	L_9	(56, 1264)	729	729	729	729	0
UND_5	(20,100)	84	81	83	82.50	1.00	L_{10}	(55, 1258)	835	835	835	835	0
UNS_1	(20,100)	45	40	43	41.75	1.26	M_1	(63, 943)	105	95	95	95	0
UNS_2	(20,100)	47	39	45	42.50	2.65	M_2	(56, 846)	142	141	141	141	0
UNS_3	(20,100)	48	44	45	44.50	0.58	M_3	(51, 728)	109	109	109	109	0
UNS_4	(20,100)	47	41	46	43.25	2.06	M_5	(55, 733)	66	66	66	66	0
UNS_5	(20,100)	48	43	45	44.00	0.82	M_6	(64, 242)	33	33	33	33	0
EXD_4	(20,100)	65	64	64	64.00	0.00	M_7	(48, 574)	67	67	67	67	0
OXD_1	(20,100)	55	54	54	54.00	0.00	S_2	(55, 158)	13	13	13	13	0
OXD_2	(20,100)	56	51	55	52.30	1.17	S_4	(56, 80)	5	5	5	5	0
OXD_3	(20,100)	57	53	56	53.60	0.82							
OXD_4	(20,100)	57	56	56	56.00	0.00							
OXD_5	(20,100)	55	49	54	50.70	1.41							

Table 3.5: *A summary of the range of solution qualities for 20 subsets of nine randomly selected SKUs and the SKU with the highest pick frequency for all problem instances with no proven optimal solution. Each instance was solved with the when solved using the ASLB approach. The symbol μ indicates the average and σ the standard deviation.*

lower bound would be obtained by each set. Table 3.5 illustrates the range of solutions for the different subsets of SKUs when solved with the ASLB approach. It is clear that selecting a subset based on pick frequency does no worse than considering a different subset of SKUs.

3.6.3 Heuristics

Following the approaches for determining a lower bound to the SLP both the GS and OPA as well as the GA and CD heuristics were used to solve all the problem instances. These heuristics are compared to a set of 50 random SKU arrangements for each problem instance as well as the best lower bound. The solutions obtained when solving the generated problem instances with the heuristics are shown in Table 3.6 and the results for the historical problem instances in Table 3.7. Included in Table 3.7 are the historical number of cycles traversed (PEP_{HIS}) as well as the number of cycles which would have been traversed had the maximal cut formulation been used to sequence orders for the historical SKU arrangements (PEP_{OSP}). These historical results are included to illustrate the compounded effect of addressing each different decision tier. Both the GS and OPA heuristics do not necessarily yield an optimal solution and in two cases the best known solution was found by randomly arranging SKUs. This is in contrast to the results for bi-directional and unidirectional carousel systems with an infinite set of stochastic orders where the OPA and GS approaches yield optimal solutions respectively. In addition there are multiple instances where the GS and OPA heuristics yield solutions of different quality. Although in a few cases the GA yields the best solution there is no significant difference between the GA, CD, OPA and GS heuristics when tested using the Friedman test and in most cases the best solution was obtained by one of these heuristics. The heuristics were compared using this non-parametric statistical test based on ranked values for each size classification (small, medium, and large) at a 95% confidence level. However, in 17 of the 22 problem instances at least one of the heuristics found the best known solution.

All four of the proposed heuristics have fast computation times (less than one second). This enables all of the heuristics to be included in any implementation and the best solution selected for use. If the best solution between the four heuristics is selected it results in an average saving in walking distance of approximately 2% for the large and 6.5% for the medium sized problem instances when compared to PEP's arrangements after the OSP was solved correctly. Note that no saving was possible for the small picking lines as PEP's method already found a best known solution. Also note that the best saving achieved by a heuristic for a problem instance was 15%. This would result in an average saving of roughly 60 km per month. It is therefore suggested that PEP run all four of the heuristics and select the best solution.

Problem instance	Size $(\|\mathcal{S}\|, \|\mathcal{O}\|)$	ASLB	OPA	GS	GA	CD	Random (min, max)	(μ, σ)
UND_1	(20,100)	85	91	91	91	91	(90*,92)	(91,0.4)
UND_2	(20,100)	87	91	91	91	91	(90*,92)	(91,0.2)
UND_3	(20,100)	87	91*	91*	92	92	(91*,92)	(91,0.3)
UND_4	(20,100)	86	91	91	91	91	(90*,92)	(91,0.5)
UND_5	(20,100)	86	90*	91	91	91	(90*,92)	(91,0.5)
UNS_1	(20,100)	49	67	66	65	65	(63*,67)	(66,1)
UNS_2	(20,100)	50	67	68	66	66	(64*,69)	(67,1)
UNS_3	(20,100)	51	69	68	68	68	(66*,71)	(68,1)
UNS_4	(20,100)	51	67	68	67	67	(65*,69)	(67,0.8)
UNS_5	(20,100)	51	68	65*	70	70	(67 ,70)	(68,0.8)
EXD_1	(20,100)	71	71*	71*	71*	71*	(71*,73)	(71,0.5)
EXD_2	(20,100)	79	79*	79*	79*	79*	(79*,79)	(79,0)
EXD_3	(20,100)	73	73*	73*	73*	73*	(73*,73)	(73,0)
EXD_4	(20,100)	66	70	69*	69*	69*	(69*,73)	(71,0.8)
EXD_5	(20,100)	68	68*	70	70	70	(68*,72)	(70,1)
OXD_1	(20,100)	56	64	64	63	63	(62*,73)	(70,1)
OXD_2	(20,100)	57	65	66	66	66	(63*,74)	(70,1)
OXD_3	(20,100)	57	65*	66	65*	65*	(65*,75)	(70,1)
OXD_4	(20,100)	57	65	64	65	65	(62*,76)	(70,1)
OXD_5	(20,100)	55	64	64	64	64	(62*,77)	(70,1)

Table 3.6: *A comparison of the value of the maximal cut when solving the generated problem instance with the OPA, GS, GA and CD heuristics to a set of randomly generated solutions as well as the best known lower bound. The symbol μ indicates the average and σ the standard deviation.*

3.7 Conclusion

An order picking problem that is a variation of a unidirectional carousel system was identified. The problem of arranging SKUs within this unidirectional carousel/picking line to minimise the rotations of the carousel/walking distance of pickers (SLP) was considered. Two heuristics which are known to be optimal for many carousel problems were tested and it was found that these heuristics often did not yield the best solutions. Two additional heuristics were also tested and it was shown that all four heuristic approaches yields similar results.

An optimal solution or tight lower bound was needed to measure the performance of the heuristics because all the heuristics displayed similar performances. Following a maximal cut approach to the order sequencing problem (OSP) two mathematical formulations for determining a lower bound to the SLP were introduced. These lower bounds were tested against a trivial lower bound and were shown to only solve small problem instances within a reasonable time. The formulations were used to determine lower bounds by only considering a subset of SKUs for larger problem instances. In most cases the sample of SKUs needed to determine a lower bound was too large to strengthen the trivial lower bound within a reasonable time frame.

A tight lower bound for large problem instances was not found due to the complexity of the

Problem instance	Size $(\lvert\mathcal{S}\rvert, \lvert\mathcal{O}\rvert)$	ASLB	PEP_{HIS}	PEP_{OSP}	OPA	GS	GA	CD	Random (min, max)	(μ, σ)
L_{01}	(49,1262)	1232	1262	1232*	1232*	1232*	1232*	1235	(1232*,1242)	(1237,2.7)
L_{02}	(54,1264)	1226	1255	1226*	1226*	1226*	1226*	1226*	(1226*,1226)	(1226,0)
L_{03}	(51,1265)	1161	1254	1161*	1161*	1161*	1161*	1161*	(1161*,1183)	(1171,8.8)
L_{04}	(56,1263)	1011	1224	1072	1060	1011*	1063	1063	(1011*,1019)	(1014,2.4)
L_{05}	(51,1264)	1069	1234	1069*	1069*	1069*	1069*	1069*	(1069*,1069)	(1069,0)
L_{06}	(53,1258)	959	1222	1005	1021	1002*	1007	1007	(1021,1069)	(1042,10.2)
L_{07}	(56,1260)	855	1227	955*	971	968	971	971	(962,982)	(973,4.3)
L_{08}	(54,1244)	817	1242	992	989	980	990	990	(974*,994)	(988,4.3)
L_{09}	(56,1264)	729	1202	947*	967	957	960	960	(952,969)	(962,3.5)
L_{10}	(55,1258)	835	1177	1025	941	872*	897	897	(876,910)	(893,6.7)
M_{01}	(63,943)	99	640	259	280	285	248*	248*	(268,286)	(279,4.5)
M_{02}	(56,846)	145	615	232	221*	225	230	230	(227,250)	(237,4.5)
M_{03}	(51,728)	109	457	152	150	149*	162	162	(154,168)	(161,3.5)
M_{04}	(63,396)	74	224	90	124	81	100	100	(74*,86)	(79,3.1)
M_{05}	(55,733)	66	461	125	114*	121	126	126	(118,134)	(127,2.9)
M_{06}	(64,242)	33	142	45	69	56	41*	41*	(65,81)	(73,4)
M_{07}	(48,574)	67	324	80	74*	80	88	88	(81,95)	(88,3.6)
S_{01}	(48,90)	7	40	7*	7*	7*	7*	7*	(7*,8)	(8,0.5)
S_{02}	(55,158)	13	82	14*	16	18	15	15	(14*,18)	(16,0.9)
S_{03}	(51,82)	8	36	8*	8*	8*	8*	8*	(8*,9)	(8,0.4)
S_{04}	(56,80)	5	38	6*	7	6*	7	7	(6*,7)	(7,0.1)
S_{05}	(42,89)	9	40	9*	10	10	10	10	(9*,10)	(10,0.5)

Table 3.7: *A comparison of the value of the maximal cut when solving the historical problem instances with the OPA, GS, GA and CD heuristics to a set of randomly generated solutions, PEP's historical solution as well as the best known lower bound. in the last column μ indicates the average and σ the standard deviation.*

lower bound formulations. The heuristic approaches were therefore further compared to a set of random arrangements as well as the historical number of cycles traversed and the results obtained after solving the OSP for the historical SKU arrangements. All four heuristics had fast computation times and could all therefore be run and the best solution selected for use. By selecting the best solution an average saving of 2% for the large and 6.5% for the medium sized problem instances was achieved.

From a managerial perspective a SKU arrangement module within the warehouse management system (WMS) of PEP is in the process of being implemented. It is suggested that all four of these heuristics be run in the back end of this module in the WMS. The best solution would then be presented to the managers who would be able to make changes to this proposed arrangement if they so wish. No change management is thus required to implement these heuristics.

The whole order picking process involves three tiers of decision making where SKUs are initially assigned to available picking lines, the SKUs are then arranged into locations and finally the orders are sequenced for pickers. Following on this research a natural progression would be to investigate the assignment of pending SKUs to available picking lines (SKU assignment problem). Candidate SKU to picking line assignments can be evaluated in terms of walking distance by using any one of the proposed heuristics for the SLP. The walking distance of pickers, a number of other secondary issues in the DC as well as factors downstream in the supply chain should all be taken into account when solving the SKU assignment problem. These additional considerations may include storage locations in the DC, product promotions as well as the grouping of SKUs from the same department to be picked into the same carton.

Bibliography

[1] BARTHOLDI JJ & HACKMAN ST, 2011, *Warehouse and Distribution Science*, The Supply Chain and Logistics Institute, School of Industrial and Systems Engineering Georgia Institute of Technology Atlanta, GA 30332-0205 USA, Available from `www.warehouse-science.com`.

[2] BARTHOLDI JJ & PLATZMAN LK, 1986, *Retrieval strategies for a carousel conveyer*, IIE Transactions, **18(10)**, pp. 166–173.

[3] DE KOSTER R, LE-DUC T & ROODBERGEN KJ, 2007, *Design and control of warehouse order picking: A literature review*, European Journal of Operational Research, **182(2)**, pp. 481–501.

[4] GADEMANN N & VAN DE VELDE S, 2005, *Order batching to minimize total travel time in a parallel-aisle warehouse*, IIE Transactions, **37(1)**, pp. 63–75.

[5] HAGSPIHL R & VISAGIE S, 2014, *The number of pickers and stock-keeping unit arrangement on a uni-directional picking line*, The South African Journal of Industrial Engineering, **25(3)**, pp. 169–183.

[6] HASSINI E, 2009, *One-Dimensional Carousel Storage Problems: Applications, Review and Generalizations*, INFOR, **47(2)**, pp. 81–92.

[7] HSU CM, CHEN KY & CHEN MC, 2005, *Batching orders in warehouses by minimizing travel distance with genetic algorithms*, Computers in Industry, **56(2)**, pp. 169–178.

[8] IBM, *CPLEX*, Available from `www.ibm.com`.

[9] LITVAK N & MAIA V, 2010, *A survey on performance analysis of warehouse carousel systems*, Statistica Neerlandica, **64(4)**, pp. 401–447.

[10] MATTHEWS J & VISAGIE SE, 2013, *Order sequencing on a unidirectional cyclical picking line*, European Journal of Operational Research, **231(1)**, pp. 79–87.

[11] MATTHEWS J & VISAGIE SE, *Picking Line Data*, Available from `http://scholar.sun.ac.za/handle/10019.1/86110`.

[12] PAN CH & LIU SY, 1995, *A Comparative Study of Order Batching Algorithms*, Omega-International Journal of Management Science, **23(6)**, pp. 691–700.

[13] PARAGON DECISION TECHNOLOGY, *AIMMS*, Available from `www.aimms.com`.

[14] PEP STORES LTD, *PEP*, Available from `www.pepstores.com`.

[15] PUNNEN AP, 2007, *The traveling salesman problem: Applications, forumlations and variations*, pp. 1–28 in GUTIN G & PUNNEN AP (EDS), *The traveling salesman problem and its variationss*, pp. 1–28. Springer, New York (NY).

[16] STERN HI, 1986, *Parts locationand optimal pickig rules for a carousel conveyour automatic storage and retrieval system*, Proceedings of the Proceedings of the Seventh International Conference on Automation in Warehousing, Springer, California, pp. 185–193.

[17] Van Den Berg JP, 1996, *Multiple Order Pick Sequencing in a Carousel System: A Solvable Case of the Rural Postman Problem*, The Journal of the Operational Research Society, **47(12)**, pp. pp. 1504–1515.

[18] Van Den Berg JP & Zijm WHM, 1999, *Models for warehouse management: Classication and examples*, International Journal of Production Economics, **59(1-3)**, pp. 519–528.

[19] Vickson RG & Fujimoto A, 1996, *Optimal storage locations in a carousel storage and retrieval system*, Location Science, **4(4)**, pp. 273–245.

[20] Vickson RG & Lu X, 1998, *Optimal product and server locations in one-dimensional storage racks*, European Journal of Operational Research, **105(1)**, pp. 18–28.

CHAPTER 4

SKU assignment to parallel unidirectional picking lines

4.1 Introduction

Order picking is the process of retrieving products from storage (or buffer areas) in response to a specific customer request [22]. It is estimated that the cost of order picking can be in excess of 55% of all warehousing costs and has the potential to become the bottleneck of a supply chain due to its labour intensive nature [4]. Optimising the order pick operation therefore has the potential to improve overall supply chain efficiency as well as reducing costs.

The order picking system used in a distribution centre (DC) operated by PEP Stores Ltd. (PEP) is investigated in this paper. PEP is the largest single brand retailer in Southern Africa [20]. PEP sells predominately apparel that is distributed by means of three DCs located on the West Coast (Cape Town), East Coast (Durban) and mainland (Johannesburg) servicing over 1500 retail outlets across Southern Africa. All three DCs have structural differences, but perform order picking by means of the same fundamental picking system. The DC on the East Coast has the highest throughput as most of the suppliers are situated in Asia and therefore forms the focus of this paper. However, the models presented in this paper could easily be applied to the other two DCs as well.

Due to the nature of the market as well as the physical characteristics of apparel (size and shape) a large proportion of PEP's picking is piece picking in that individual units of stock are handled by pickers. The DC is designed to handle both carton and piece picking, but the piece picking operation forms the largest proportion in the DC and is the focus of the study. In the remainder of this paper the term order picking will refer to the piece picking operation.

The DC uses a unique variation of a forward pick area, as described by Bartholdi & Hackman [2], consisting of unidirectional picking lines. This system has evolved to manage the large number of different stock keeping units (SKUs) sold annually by PEP (in excess of 40000). Each picking line has 56 locations with the capacity to hold five pallets of the same SKU. These picking lines are managed in waves of picking – which are sets of SKUs placed in a picking line and picked together in a single operation, independently from all other SKUs not on that picking line. In addition there are enough pickers available such that picking lines can be operated in parallel. Once all the store requirements for all the SKUs in a wave have been picked the excess stock (if any) is removed and all the locations in the picking line become available for a new wave of picking.

Orders, which are typically defined as all stock requirements for a store across the entire DC, are here defined for each specific wave of picking. An order is thus defined by the store requirements for SKUs in a specific wave. In this way each store may have multiple orders associated with it at any given time, each one associated with a different wave. This segmentation of orders is possible as there are frequent shipments of stock to each store (at least weekly).

Figure 4.1 shows the general layout of the picking line area. There are six picking lines on either side of a main conveyor. The main conveyor is elevated and the smaller picking line conveyors rise to create a bridge allowing for pickers to completely circumvent the picking line by walking under the bridge. Figure 4.2 illustrates a functioning picking line with pickers walking around the conveyor belt picking orders. While picking, pickers interact via a headset to a voice recognition system (VRS) which directs them in a clockwise direction around the picking line. The VRS tracks the location of the last pick made by a picker and therefore always directs the picker to the nearest required SKU for the active order. Once an order is completed a new order will be assigned to the picker independent of the position or number of other pickers in the picking line. The VRS can therefore assign or remove pickers at any time between the completion of an old, and assignment of a new order.

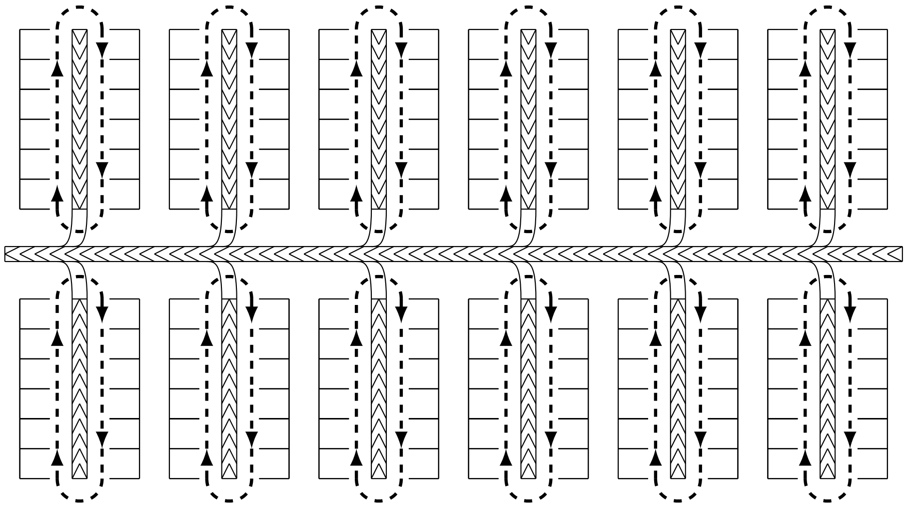


Figure 4.1: *A schematic representation of the layout of the 12 picking lines in the DC, six on either side of the main conveyor. The dashed lines indicate the movement of the pickers around the conveyor belts. The jagged lines indicate the direction of movement of the conveyor belts. The conveyor belts in each smaller picking line link with the main conveyor belt with bridges.*

Figure 4.3 shows a typical location populated with stock. In some cases SKUs may require more space to accommodate all the stock. Management then either uses the floor space behind the location or additional adjacent locations to store this excess stock. Due to the level of safety stock at each location stock outs rarely occur during a wave of picking as all the stock required for all the orders associated with that wave are known and may be stored on the line before picking starts.

Pickers do not use totes but instead pack items directly into cartons, as shown in Figure 4.2. This reduces material handling, as waves are picked independently allowing for the cartons to be shipped as is. The DC also reuses the cartons which held the bulk stock received from suppliers which becomes available as picking takes place, thereby reducing packaging costs. Pickers place unique bar code stickers on each carton and may be required to use multiple cartons for a single order, depending on the number and size of the SKUs in that order. Once an order is completed

Figure 4.2: *A photograph of a functioning picking line.*

Figure 4.3: *A photograph of a SKU location in a picking line holding five pallet loads of a SKU next to an empty SKU location.*

or a carton is full, it is placed on the conveyor belt which conveys the carton to the dispatch area via the main conveyor. Each carton holds stock from a single picking line only and thus may have excess capacity as neither the cartons used by the pickers nor the volume of stock required by a store from a line are uniform. Cartons are therefore resized, if necessary, before being stapled at the dispatch area as shown in Figure 4.4.

Figure 4.4: *A photograph of the dispatch area where cartons are resized to reduce the volume for transportation.*

Before a SKU can be assigned to a wave its picking instructions must be released by the central planning department at central office. The planning department calculates the store requirements and assigns an out-of-DC date to it. To achieve this, SKUs of the same product type, but of different sizes, are grouped together as a distribution (DBN) and are planned collectively. At the start of each day the DC schedules available DBNs for processing for that day based on the out-of-DC date, number of available picking lines and location of the physical stock in the DC. Once the DBNs have been scheduled all SKUs in the same DBN are placed in the same wave, and therefore on the same picking line. PEP forces DBNs to be placed on the same picking line to ensure that all the SKUs for a DBN will arrive at a store on the same day. This enables the store to pack out all the sizes of a product type on the shelves in one batch.

The picking line area considered here may be described as a synchronised zone picking system, as described by de Koster *et al.* [4], as each picking line runs independently of, and in parallel to the other. The pickers which were assigned to a picking line may join any other picking line when a wave of picking is completed, because the picking lines run independently. In this way the challenge of empty capacity in a single picking line (zone) due to work imbalance between picking lines (zones) is not a major problem as is typically the case in zone picking systems [2]. However, some balancing is required as too many pickers on a single line cause congestion and there are a limited number of forklifts available to build picking lines [7]. This balancing revolves around the rate at which picking lines are picked and the rate at which new picking lines are built and therefore falls out of the scope of this study.

The focus of this paper is on the assignment of already scheduled DBNs into specific waves for a given day while minimising the total walking distance of pickers in the picking lines. The remainder of this paper is structured as follows: A brief discussion on related work in literature is given in §4.2 followed by a more detailed exposition of the problem and solution approaches in §4.3. The results are discussed in §4.4 and the paper is concluded in §4.5.

4.2 Literature review

Due to the cyclical paths which pickers walk in a picking line it has many similarities to carousel systems. A carousel is an automated storage and retrial system and typically consists of a number of shelves which rotate to present required stock to a stationary picker [12]. Hassini [8] composed an extensive literature review on carousel systems, both applications and generalizations thereof. Multiple carousel configurations are discussed with reference to both the order sequencing as well as storage location problems. He mentioned that for almost all cases SKUs have been assumed to have independent demand and that storage policies for demand correlated SKUs have received little attention. It was further suggested that demand correlations between SKUs may play a role in both location of SKUs in carousels as well as the carousel to which the SKU is assigned in the multi-carousel case.

Litvak & Maia [12] expands on carousel systems with multiple carousels and discusses several configurations. In most cases the problem statement consisted of multiple carousels and a single picker. Emerson & Schmatz [5] considered a carousel configuration consisting of 22 carousels where each pair of carousels has its own picker (11 pickers in total). They used simulations to test three storage schemes, namely a random storage scheme, a sequential alternating storage scheme and a scheme which stores SKUs in the carousel with the largest number of openings. Moreover, the effects of a floating operator which moved between pairs of carousels were included in the simulation. Litvak & Maia [12] noted that Emerson & Schmatz [5] did not address the problem of optimally assigning items to carousel bins nor did they discuss any analytical results.

In all cases carousels are modelled with the assumption that demand is stochastic with the probability of an order requiring a SKU used as a measure of demand. The deterministic nature of the orders in picking lines considered in this paper therefore limits the direct use of the carousel approaches found in literature.

Having multiple picking lines PEP's order picking system also resembles a zone based system. SKU assignment in zone based order picking has received some attention, but zone based order picking designs often differ significantly. Jane & Laih [10] considered the assignment of SKUs to locations in a warehouse using a synchronised pick-by-light zone picking system. Here an order is simultaneously picked in each zone and only once the order is completed in all zones can a new order begin. Therefore the time required to complete an order is equal to the maximum time required to pick all the SKUs from one of the zones. Due to this zone structure the objective of the SKU assignment was to balance the workload between zones by reducing the maximum pick time in each zone. To achieve this, Jane & Laih [10] used a similarity measure between SKUs (that is calculated as the number of orders which require both SKUs) in conjunction with a natural clustering approach to minimise the similarity of SKUs within zones. By minimising the similarity within zones the amount for work required for orders is spread more evenly over all zones.

Garfinkel [6] focused on minimising the number of zones visited (both sequential and synchronised zones) when picking all orders by assigning SKUs to storage locations. Normally walking distance is minimised, but it was suggested that minimising the number of zones visited is beneficial under circumstances when orders require few SKUs, batching is not desirable and sorting is expensive.

Chiang *et al.* [3] used data mining techniques to create similarity measures between SKUs for use in a storage assignment model. The model was applied to a DC with an S-shape picking strategy within parallel aisles. Both complementary and substitutable relationships[1] between SKUs were calculated. Using these relationships, an association index was developed between every available storage location and the new SKUs. A generalised assignment model was then used to assign the SKUs which was shown to reduce the number of aisles over which orders were required to pick and therefore the total picking time.

Pan & Wu [18] considered the storage assignment problem for a single aisle pick-and-pass picking line. Markov chains were used to estimate the walking distances of pickers based on the probability that SKUs are requested by the same order. Storage assignments to both single, multiple equally sized and multiple different sized zones were considered. Three optimal storage assignment policies were proposed taking into account total distance travelled as well as the balance of work between zones. Both the dependence of pickers on each other as well as the physical layout of the linear zones limit the use of these approaches on picking lines.

Although picking lines may be seen as different zones an order is defined for a single wave of picking only. Pickers can also freely switch between picking lines which run in parallel. Approaches which seek to manage the spread of orders across zones, either by balancing or minimising, are not suited for the picking line system investigated in this paper.

There has also been some attention to the assignment of SKUs to non-zoned order picking systems. Accorsi *et al.* [1] simultaneously considered both the storage allocation (inventory levels for each SKU) as well as the storage assignment (location of the SKU in the DC) problems. Using a case study with forward and reserve areas they extensively compared combinations of different forward storage allocation strategies, storage assignment rules, SKU clustering algorithms, percentile threshold cut-offs for SKU similarity and picker routing policies. Relevant combinations were tested and compared based on travel distances of picking and restocking as well as the total cost of restocking. The results demonstrated that the issues related to allocation and assignment are correlated and both should be taken into account when designing and managing an order pick system.

Although Accorsi *et al.* [1] suggests considering both SKU allocation and SKU assignment simultaneously, SKU allocation falls outside of the scope of the problem considered in this paper. The volume of stock for a SKU for a wave of picking is predetermined by the planning department. In addition the number of required locations for a SKU is predetermined by the DC as all the stock required for a wave of picking will be stored at the picking line before picking starts.

Manzini [13] compared allocation algorithms for correlated products in a less than unit load picker to parts order picking system. A correlation measure between SKUs is calculated as the number of orders requiring both SKUs. Three algorithms which use this correlation measure are proposed. A case study is presented which compares these algorithms to a Cube-per-Order Index (COI), a frequency based and a random approach. The case study comprised of parallel and orthogonal aisles where a picking vehicle of finite capacity used a composite routing strategy. All approaches significantly reduced overall picking costs when compared to the random approach while the approaches using correlation measures showed a marginally better performance when compared to the COI approach. Due to the cyclical nature of the picking lines and the ability of pickers to pick directly into cartons these approaches cannot be directly applied to the order picking system considered in this paper.

The multiple picking line system considered at PEP shows many similarities to a study by Kim

[1]Complementary SKUs have a high probability to be required together and substitutionary SKUs have a low probability to be required together

& Smith [11] who focused on a dynamic warehouse slotting problem where SKUs are re-slotted on a daily basis in a zone-based carton picking DC. After the completion of a day's scheduled orders (or pick wave) the orders required for the next pick wave are known which creates a deterministic planning environment for re-slotting. During each night shift SKU re-slotting takes place based on the SKU correlations created by the orders scheduled for the next pick wave. The objective of the re-slotting is to reduce the overall picking time of each pick wave.

The order picking configuration considered by Kim & Smith [11] consisted of multiple zones comprising of a single shelf with uniform slots. Each zone is serviced by a single picker who walks between the shelf and a parallel conveyor belt. Pickers collect pending orders from one end of the zone (the start) and pick all required SKUs for the order before placing the carton onto the conveyor belt and returning to the start of the zone to begin a different order. In this way the distance which is travelled by a picker to complete an order, for a zone, is twice the distance from the start of the zone to the furthest SKU. The conveyor belt conveys cartons either to the dispatch area for completed orders, or to another zone for further picking which allows cartons to hold stock from multiple zones. A schematic representation of this layout is given in Figure 4.5.

Kim & Smith [11] used a Mixed Integer Programming (MIP) formulation to assign SKUs to slots while minimising the overall distance travelled. Due to the nature of the order picking configuration the distance travelled could be determined with a linear system of constraints. The MIP was not solvable due the size of the problem. They therefore investigated four further heuristic approaches: A steepest descent neighbourhood slotting (SD) heuristic which, given an initial solution based on a heuristic using the COI, sequentially compares all pairwise interchanges, selects the best interchange in terms of total pick distance and implements it until no improvements are possible; A correlated slotting (CS) heuristic which uses a similar pairwise comparison approach, except that correlations (number of orders requiring both SKUs) are considered; Two simulated annealing approaches based either on the SD or CS heuristics were used to aid in the movement away from local optima. They showed that the simulated annealing approaches yielded the best results and the worst performing heuristic was the SD heuristic. Although the order picking system considered by them is similar to PEP's system the cyclical paths walked by pickers creates a more complex environment to calculate walking distance.

4.3 Model

The focus of this study is on the assignment of scheduled DBNs to available picking lines (as waves) for a single day while minimising the walking distance of pickers. Because the scheduling of DBNs takes into account the number and capacity of available picking lines all the DBNs that are scheduled for a day will be assigned to a picking line. This problem, which will be referred to as the Picking Line Assignment Problem (PLAP), may therefore be described as a Generalized Assignment Problem (GAP) with a set of available picking lines, or knapsacks, to which a set of DBNs or items, must be assigned. Each DBN requires a number of locations, or weight, in a picking line and each DBN must be assigned to a picking line with a limited number of available locations.

Two further problems must be solved to calculate the walking distance of pickers once DBNs have been assigned to a picking line. Firstly SKUs must be assigned to specific locations in their respective picking lines using an assignment model. Thereafter the orders defined by the DBNs within each picking line must be sequenced for pickers using a clustered travelling salesman model. All three problems must therefore be solved to solve the PLAP.

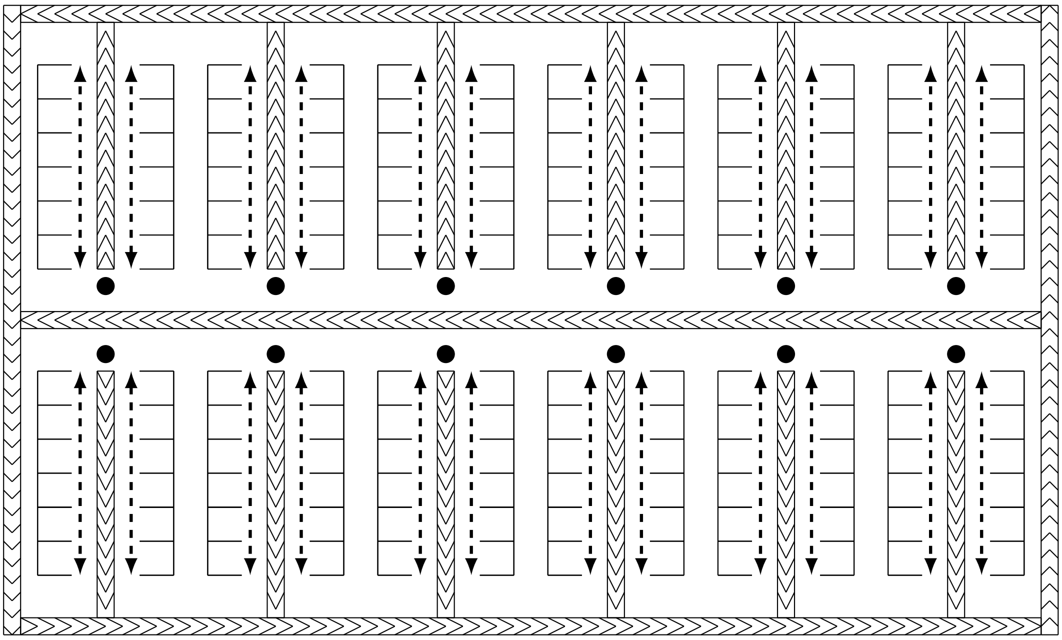


Figure 4.5: *A schematic representation of the layout of the zones in the DC investigated by Kim & Smith [11]. The dashed lines indicate the movement of the pickers and the dot indicates the point at which an order is started in a zone. The jagged lines indicate the direction of movement of the conveyor belts.*

Matthews & Visagie [15] focused on the sequencing of orders in a unidirectional picking line with fixed SKU locations while minimising total walking distance. Their approach is based on a maximal cut formulation which calculates a solution within one walking cycle of a lower-bound. A cycle is the distance walked by a picker to circumvent the picking line once. In other words a cycle is the distance that a picker walks to pass all the locations once. The solution determines the preferred order to assign to a picker given his/her current location and is suitable for the current VRS system.

Matthews & Visagie [17] showed that the scope for optimisation by arranging the SKUs assigned to a picking line is limited. Several heuristics were tested against a lower bound and a set of random arrangements. It was shown that there is no significant difference in performance between any approaches including random arrangements. These algorithms included the Greedy and Organ pipe heuristics which are known from literature to be optimal for carousel systems with a stochastic set of orders [8]. Furthermore it was shown that integer programming approaches to SKU arrangement in a single picking line while minimising walking distance was too computationally complex to solve. Therefore the conclusion was to use greedy approaches to allocate SKUs to locations within a picking line.

The mathematical structure of the PLAP consists of simultaneously solving a GAP problem as DBNs are allocated to picking lines, an assignment problem as SKUs are assigned to specific locations in their picking line, and a clustered travelling salesman problem to sequence the orders defined within each picking line. All three problems are untractable on their own, resulting in the PLAP being unctractable as well. A less complex objective function or estimator for total walking distance is therefore presented here to solve the GAP and only after DBNs have been assigned to picking lines is the actual walking distance calculated by arranging the SKUs and sequencing the orders.

Matthews & Visagie [17] proposed a lower bound for the number of cycles traversed in a picking line by considering the SKU which has the most number of stores requiring it, referred to as a maximal SKU. Pickers would need to walk at least once around the picking line per store requiring a maximal SKU, which would constitute a lower bound. The number of stores requiring a maximal SKU (or size of a maximal SKU) for each picking line could therefore be considered as an estimate of the total walking distance in terms of the number of cycles traversed. An integer programming formulation (IP) with an objective function which seeks to minimise the sum of the sizes of the maximal SKUs is proposed. Once the DBNs have been assigned to picking lines using this formulation the exact number of cycles traversed is calculated. This is achieved by arranging SKUs in the picking line using a greedy approach proposed by Matthews & Visagie [17] after which the orders are sequenced using the maximal cut formulation by Matthews & Visagie [15]. To model the assignment of DBNs to picking lines the following parameters are set

$\mathcal{L}$	be the set of all picking lines with elements l,
$\lvert l \rvert$	be the number of SKU locations available for picking line l,
$\mathcal{D}$	be the set of all DBNs with elements d,
$\lvert d \rvert$	be the number of locations required by DBN d,
$\lceil d \rceil$	be the size of a maximal SKU accociated with DBN d.

To initially assign DBNs using this objective define

$$x_{dl} = \begin{cases} 1 & \text{if DBN } d \text{ is assigned to picking line } l \\ 0 & \text{otherwise} \end{cases}$$

and

$$y_l \text{ as the size of a maximal SKU for picking line } l.$$

In terms of these symbols the objective is to

$$\text{minimise} \sum_{l \in \mathcal{L}} y_l, \tag{4.1}$$

subject to

$$\sum_{l \in \mathcal{L}} x_{dl} = 1 \qquad d \in \mathcal{D}, \tag{4.2}$$

$$\sum_{d \in \mathcal{D}} (x_{dl} \cdot |d|) = |l| \qquad l \in \mathcal{L}, \tag{4.3}$$

$$y_l \geq x_{dl} \cdot \lceil d \rceil \qquad d \in \mathcal{D} \text{ and } l \in \mathcal{L}, \tag{4.4}$$

$$x_{dl} \in \{0, 1\} \qquad d \in \mathcal{D} \text{ and } l \in \mathcal{L}, \tag{4.5}$$

$$y_l \geq 0 \qquad l \in \mathcal{L}. \tag{4.6}$$

The objective function (4.1) minimises the sum of the sizes of all maximal SKUs for each picking line. Constraint set (4.2) assigns each DBN to a single picking line while constraint set (4.3) ensures that the number of locations needed by the DBNs assigned to a picking line should be equal to the number of locations available in that picking line. The size of the maximal SKU for each picking line is determined by constraint set (4.4).

Experiments showed that Formulation (4.1)–(4.4) was not solvable in a reasonable time (within 10 minutes) for instances with five or more picking lines. A further relaxation of the formulation is therefore introduced. The computational complexity is reduced by adjusting the value for the size of all maximal SKUs by rounding to the nearest multiple of a parameter α. These adjusted sizes, which are minimised, lead to a reduction in the number of possible objective function values during the optimisation procedure and a potential faster convergence to an optimal solution. To model this relaxed formulation (IP$_\alpha$) constraint set (4.4) is replaced with

$$y_l \geq x_{dl} \cdot \lceil d \rceil^\alpha \qquad \forall\, d \in \mathcal{D} \text{ and } l \in \mathcal{L}, \tag{4.7}$$

where $\lceil d \rceil^\alpha$ is the adjusted size of the maximal SKU for DBN d.

Although the use of adjusted sizes as an objective function significantly reduced the computational times the formulations were still not solvable within a reasonable time for instances with seven or more lines. A further heuristic was developed based on an insertion heuristic for GAPs developed by Martello & Toth [14]. The approach by Martello & Toth [14] ranks all unassigned items in decreasing order based on the difference between each item's best and second best possible assignment into a knapsack and assigns the top item to its best knapsack. This process is repeated until all items have been assigned or there is insufficient space for an item to be assigned to a knapsack which results in an infeasible solution. Using this framework as a starting point a greedy insertion algorithm (GI) for the PLAP is introduced in Algorithm 5.

Algorithm 5: Greedy insertion of DBNs while minimising the maximal SKU.

```
Data: A set of picking lines 𝓛 in descending order according to |l|
      A set of DBNs 𝓓
Result: An assignment of DBNs to picking lines
for Each line l ∈ 𝓛 do
    while an unassigned DBN exists which fits into the remaining locations of picking line l do
        Select an unassigned DBN with the largest maximal SKU which fits into picking line l
        Assign this DBN to picking line l
    end
end
```

After experimenting with real life data it was found that for rare instances Algorithm 5 does not yield feasible results as some DBNs remain unassigned. In an attempt to always find feasible solutions Algorithm 5 was adjusted to have two phases, resulting in a phased greedy insertion heuristic (GP). The first phase would assign all DBNs which require more than one location or is required by more than β stores to a picking line using Procedure 1. After this initial assignment the remaining DBNs (each requiring a single location) are then assigned in the same fashion. A feasible solution can always be found by removing enough DBNs requiring only one location from the initial insertion. Additionally by choosing a good value for β the effects on the size of a maximal SKU may be reduced. In an effort to keep β as small as possible and thereby reducing the effects on a maximal SKU, β is incrementally increased (line 6, Algorithm 6) until a feasible solution is found. This phased approach is described by Algorithm 6.

Procedure 1: A partial greedy insertion of DBNs while minimising the maximum SKU.

Data: A set of picking lines $\mathcal{L}$ in descending order according to $|l|$
A set of DBNs $\mathcal{D}$
A set of pre-assigned DBNs $\mathcal{D}_l$ associated with each picking line
Result: A final set of assigned DBNs $\mathcal{D}_l$ associated with each picking line

for *Each picking line* $l \in \mathcal{L}$ **do**
  while *an unassigned DBN exists which fits into the remainging locations of picking line* l **do**
    Select the DBN with the largest maximal SKU, $\lceil d \rceil$, which fits into set $\mathcal{D}_l$
    Assign this DBN to set $\mathcal{D}_l$
  end
end

Algorithm 6: A sequential phased insertion of DBNs while minimising the maximum SKU.

Data: A set of picking lines $\mathcal{L}$
A set of DBNs $\mathcal{D}$
Result: An assignment of DBNs to picking lines

$\beta = 0$
while *an unassigned DBN exists* **do**
  Clear all assignments of DBNs
  Insert all DBNs where $|d| > 1$ or $\lceil d \rceil > \beta$ using Procedure 1
  Insert all remaining DBNs using Procedure 1
  $\beta = \max_{d/|d|>\beta} |d|$
end

4.4 Data and results

To test the approaches introduced in the previous section historical data from PEP was captured over a three month period and is available on-line [16]. The data consisted of 353 picking lines over 61 work days with the number of parallel picking lines operating per day ranging from 2 to 11. The dataset contained 7354 unique DBNs and 7510 unique SKUs[2]. Due to the wide range of number of lines scheduled for a day the master data was reorganised into seven scenarios. This allowed for a more comprehensive comparison between algorithms. Each scenario consisted of a more uniform set of instances (days) where each instance in a scenario had the same number of picking lines. These scenarios were constructed by taking each historical instance with n picking lines and removing picking lines to create additional instances with $n-1,\ n-2\ldots 2$ picking lines. Each of these new instances still comprises of a set of historical picking lines which were scheduled for the same day and thus allow solution approaches to be compared to the historical case. The composition of these adjusted scenarios is given in Table 4.1.

All tests were performed on an Intel i7 2GHz processor with eight GB ram running the Windows 7 operating system. All mathematical formulations were solved with CPLEX 12.3 and coded in AIMMS 3.12 [9, 19]. Each instance was run for a maximum of two hours after which the best solution obtained would be reported on.

A summary of the total walking distance by pickers for each scenario is shown in Table 4.2. It is

[2]Some SKUs are present in multiple DBNs due to replenishment cycles.

Number of lines per instance	Number of days	Number of DBNs
2	61	2592
3	53	3437
4	49	4146
5	38	4109
6	32	4161
7	22	3177
8	14	2148

Table 4.1: *The composition of the scenarios from historical data.*

clear that all the solution approaches significantly improve on the current assignment approach by PEP. As expected the reduction in the accuracy to which the size of maximal SKUs are calculated reduces the solution quality. However, this is not a significant reduction relative to the improvement on the historical case. It is noted that the GI and GP approaches appear to outperform the IP approaches. This small improvement is attributed to the sequence in which picking lines are populated which is more suited for the greedy approach to the SKU arrangement and final order sequencing.

Scenario	His	GI	GP	IP	IP_{25}	IP_{50}	IP_{100}
2	7168	6019	6019	6053	6124	6150	6192
3	9289	<u>7514</u>	7515	7556	7601	7660	7788
4	11532	<u>9014</u>	9015	9072	9139	9233	9372
5	11259	<u>8614</u>	8618	-	-	8765	8877
6	10706	<u>8121</u>	8129	-	-	8284	8396
7	8412	<u>6350</u>	6366	-	-	6497	6558
8	6011	<u>4703</u>	4715	-	-	4782	4845

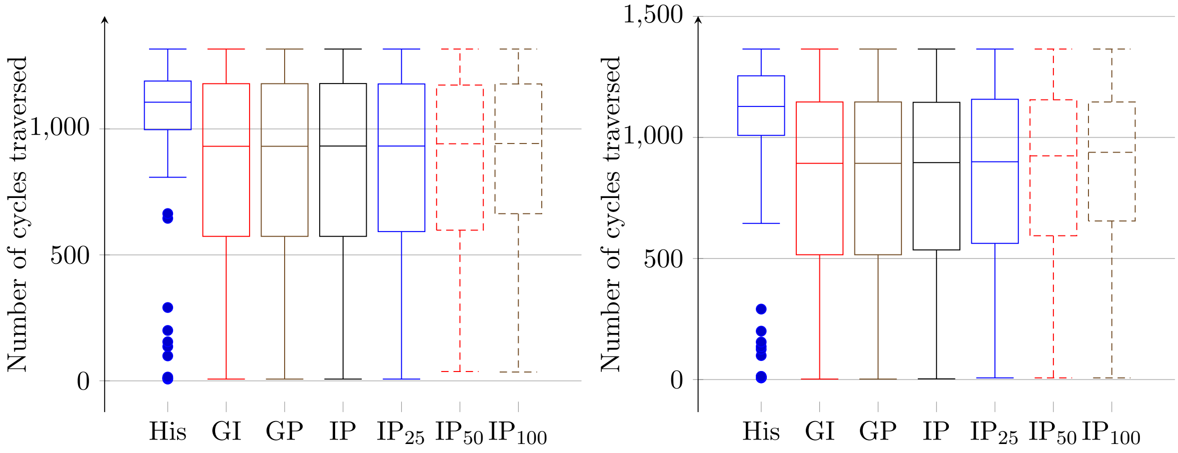
Table 4.2: *A comparison of greedy insertion (GI), the phased greedy insertion (GP), the integer programming formulations (IP_α) as well as the historical assignment (His) in terms of total number of kilometres walked in a scenario. A dash indicates that the particular solution approach was not run for a scenario due to excessive computation times. An underlined entry indicates that for a few solitary instances the solution obtained was not feasible. These results are still included for comparison purposes.*

To illustrate the loss of accuracy when using the size of the maximal SKU as an estimate of total walking distance the sum of the maximal SKUs for each picking line was converted to distance and summarised in Table 4.3. This conversion was achieved by taking the maximal SKU as the final number of physical cycles walked and converting this to kilometres in the same fashion as Table 4.2. The actual walking distance is on average 5% greater than the distance estimator calculated by means of the size of the maximal SKUs for the historical case. The difference between the actual walking distance and the distance estimator calculated by means of the maximal SKU are significantly larger for the proposed solution approaches and range from 25% to 60%, depending on the size of the scenario. This suggests that the sum of the maximal SKUs is only an accurate measure of the total walking distance for poor SKU to picking line assignments. This occurs when the size of the maximal SKUs within each picking line, which is a lower bound, tends towards the total number of orders in the picking line, which is an upper bound. Although this measure does not accurately estimate the final walking distance it is effective at reducing the total walking distance when used as an estimate in the objective function.

The GP shows similar results to the IP approach with respect to the sum of the sizes of the maximal SKUs for scenarios with four or fewer picking lines. Moreover, in many cases after determining the actual walking distances by arranging the SKUs in the picking line and sequencing

Scenario	His	GI	GP	IP	IP_{25}	IP_{50}	IP_{100}
2	6776	4845	4852	4845	4865	4888	4938
3	8852	5642	5656	5642	5669	5727	5826
4	11106	6443	6470	6443	6488	6568	6746
5	10724	5801	5828	-	-	5927	6105
6	10242	5340	5395	-	-	5484	5640
7	8054	4069	4130	-	-	4199	4319
8	5711	2937	2984	-	-	3014	3096

Table 4.3: *A comparison of greedy insertion (GI), the phased greedy insertion (GP), the integer programming formulations (IP_α) as well as the historical assignment (His) in terms of the sum of the sizes of the maximal SKUs converted to kilometres. A dash indicates that the particular solution approach was not run for a scenario due to excessive computation times. An underlined entry indicates that for a few solitary instances the solution obtained was not feasible. These results are still included for comparison purposes.*



Scenarios with three lines per instance

Scenarios with four lines per instance

Figure 4.6: *A graphical box-plot representation of the distribution of the number of cycles traversed for each line after scheduling scenarios with three and four lines per instance using the greedy insertion (GI), the phased greedy insertion (GP), the integer programming formulations (IP_α) as well as the historical assignment (His).*

the orders the GP outperforms the IP approach. This suggests that improving solution methods for solving the exact MIP approach in (4.1)–(4.6) would have an insignificant effect on the final walking distance for scenarios with more picking lines.

A box plot representation of the number of cycles traversed for each picking line in scenarios with three and four lines per instance respectively is given in Figure 4.6. This representation illustrates the median (50^{th} percentile), Q1 (25^{th} percentile) and Q3 (75^{th} percentile) as the horizontal lines in the closed box. The individually plotted coordinates are associated with the outliers in terms of cycles traversed. These outliers are all picking lines either 1.5 times the inter quartile range (IQR) smaller than Q1 or larger than Q3. The whisker lines indicate the minimum and maximum number of cycles for non-outlier picking lines. From the results in Figure 4.6 it is clear that the overall distribution of cycles traversed when minimising the sum of the sizes of the maximal SKUs is more spread towards a smaller numbers of cycles. It is clear that the current assignment methodology at PEP creates a distribution which is skewed towards larger lines. This pattern was also observed for all other scenarios as well.

A summary of the computational times for all the approaches are summarised in Table 4.4.

These computational times do not include the time required to arrange SKUs and solve the order sequencing problems which ranged from 0 to 40 seconds. It is clear that the computation times for the exact approaches decrease significantly as the accuracy of the adjusted sizes of the maximal SKUs decreases. It is also clear that even if the size of the maximal SKUs are rounded to the nearest multiple of 100 the computational times are excessive for scenarios consisting of seven or more picking lines. The Greedy insertion heuristics consistently require less than a second to solve which suggests that the insertion approach is preferred overall.

Algorithm	Lines per instance	μ	σ	Max	Q_1	Median	Q_3	Min
GI	2	0.01	0.01	0.02	0.01	0.00	0.00	0.00
	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	4	0.02	0.02	0.06	0.02	0.02	0.00	0.00
	5	0.02	0.02	0.06	0.04	0.02	0.01	0.00
	6	0.03	0.02	0.10	0.05	0.03	0.02	0.00
	7	0.03	0.02	0.08	0.05	0.03	0.02	0.00
	8	0.05	0.02	0.07	0.05	0.05	0.04	0.02
GP	2	0.01	0.00	0.02	0.01	0.01	0.01	0.00
	3	0.01	0.01	0.03	0.02	0.01	0.01	0.00
	4	0.02	0.01	0.04	0.03	0.02	0.02	0.01
	5	0.07	0.23	1.43	0.04	0.03	0.02	0.02
	6	0.09	0.30	1.75	0.05	0.03	0.03	0.02
	7	0.21	0.52	1.93	0.05	0.05	0.03	0.02
	8	0.39	0.87	2.53	0.06	0.04	0.03	0.02
IP	2	0.02	0.01	0.05	0.03	0.02	0.02	0.00
	3	0.42	0.42	1.51	0.78	0.20	0.16	0.05
	4	55.06	131.79	635.16	27.02	8.58	4.84	0.23
IP_{25}	2	0.03	0.01	0.08	0.03	0.02	0.02	0.02
	3	0.15	0.05	0.28	0.19	0.14	0.11	0.05
	4	0.44	0.36	1.50	0.37	0.34	0.27	0.13
IP_{50}	2	0.02	0.01	0.03	0.02	0.02	0.02	0.00
	3	0.10	0.04	0.22	0.13	0.09	0.06	0.03
	4	0.29	0.22	1.31	0.30	0.25	0.22	0.11
	5	3.27	2.51	13.62	3.56	2.40	1.95	0.45
	6	60.24	121.13	489.86	37.81	16.65	6.24	1.11
	7	1095.90	2380.75	7205.92	688.84	56.45	21.34	1.69
	8	3206.77	3590.89	7200.10	7200.05	426.88	154.78	21.14
IP_{100}	2	0.01	0.01	0.03	0.02	0.02	0.02	0.00
	3	0.07	0.02	0.16	0.08	0.06	0.06	0.03
	4	0.20	0.14	1.01	0.22	0.17	0.14	0.09
	5	1.97	1.64	6.71	3.01	1.83	0.50	0.34
	6	12.68	12.52	56.93	14.84	6.95	5.37	1.28
	7	712.85	1892.60	7200.04	92.35	24.96	15.57	1.65
	8	1733.33	2813.67	7200.09	2847.60	89.05	33.31	9.56

Table 4.4: *A comparison of computational times in seconds between the greedy insertion (GI), the phased greedy insertion (GP) and the integer programming formulations (IP_α) for each scenario. Both the average (μ) and the standard deviation (σ) are presented. Standard measures of spread are also presented with the maximum time (Max) the 25^{th} percentile (Q_1), the 50^{th} percentile (Median), the 75^{th} percentile (Q_3) and the minimum time (Min) given.*

It is clear from the results presented in Table 4.2 that all the proposed solution approaches significantly improve on the total walking distance required by pickers to pick all the SKUs. Further analysis was also performed to evaluate the effects on other areas in the DC. One of these areas is carton utilisation. Should a carton be under utilised this will require additional material handling to resize it. To evaluate poor carton utilisation PEP supplied a threshold carton size of 0.006m^3 (the size of a typical shoe box) [21]. Orders with less volume were considered undesirable. Should an order require less than this threshold of stock in a picking line it would be identified as a small order. Table 4.5 summarises the effect of small orders which are generated by each solution approach. It is clear that the number of small orders is increased when using any one of the proposed solution approaches. The total volume of stock which are packaged in small orders is very small suggesting no significant impact on transportation costs, however, the increase in the number of cartons increases carton handling in the DC. Although this increase is undesirable the operations management at PEP has agreed that it is manageable.

Scenario	His	GI	GP	IP	IP_{25}	IP_{50}	IP_{100}
2	0.05%	0.14%	0.14%	0.14%	0.13%	0.13%	0.12%
3	0.04%	0.13%	0.13%	0.12%	0.13%	0.13%	0.12%
4	0.03%	0.12%	0.12%	0.12%	0.11%	0.11%	0.11%
5	0.03%	0.13%	0.13%	-	-	0.13%	0.13%
6	0.04%	0.12%	0.12%	-	-	0.12%	0.12%
7	0.03%	0.12%	0.13%	-	-	0.12%	0.11%
8	0.03%	0.12%	0.13%	-	-	0.12%	0.12%

Table 4.5: *A comparison of the percentage of total volume generated by orders with less than* 0.006*m*^{3} *of stock between the greedy insertion (GI), the phased greedy insertion (GP), the integer programming formulations (*IP_α*) and the historical assignment (His).*

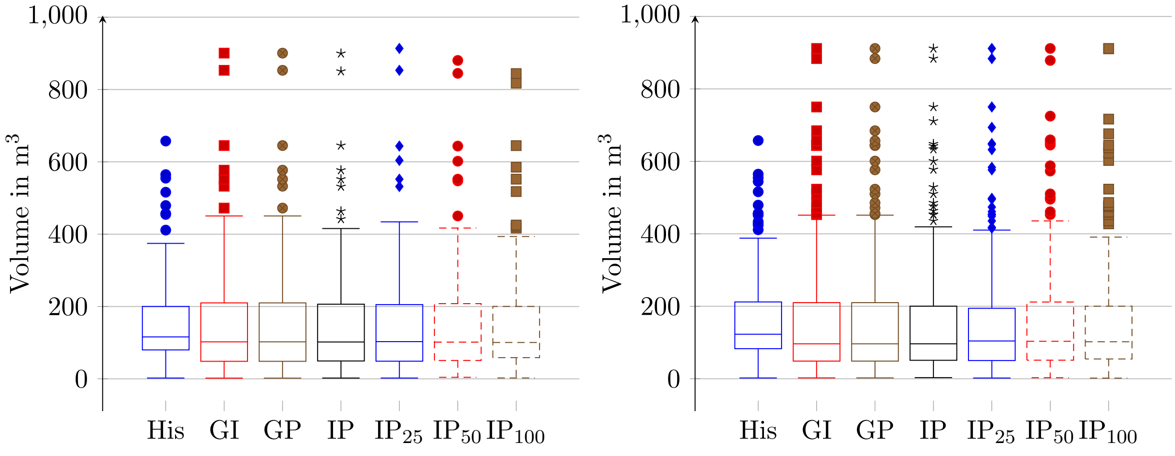
A further area of concern is the distribution of total volume over picking lines. Picking lines with large volumes require long building and picking times which may become an operational risk. Figure 4.7 illustrates distribution of volume (in m^3) to picking lines for the scenarios with three and four lines per instance, respectively. It is clear that when assigning DBNs to minimise the sum of the maximal SKUs the size and number of the large picking lines, in terms of volume, increases. This pattern was also observed for the other scenarios. These large picking lines are undesirable, both in the historical data as well as the proposed assignments, but they are all manageable at these low occurrences.

All solution approaches show significant improvements compared to the historical case in terms of walking distance. The two heuristic insertion approaches consistently obtained better solutions in less than one second of computational time. Moreover the GP approach achieved feasible solutions for all instances. It is therefore recommended that the GP approach should be used by PEP when assigning DBNs.

4.5 Conclusion

An order picking operation in a DC owned by PEP was investigated. The system functions with unidirectional cyclical picking lines to process waves of SKUs. These SKUs are grouped together into DBNs (if they differ only by size) and are planned and scheduled as a group. The objective is to minimising the total walking distance of pickers by assigning the DBNs to available picking lines for picking.

Several approaches for SKU assignment in literature were studied, but the cyclical nature of the routes of the pickers around the picking lines created a unique system which renders existing approaches not usable. An IP formulation was therefore suggested and tested using historical data from PEP. It was shown that this approach reduced the walking distance of pickers on average by 22%, but it was not solvable in a reasonable time for scenarios where there are five or more picking lines in a day. A further relaxation of this IP approach was introduced which



Scenarios with three lines per instance Scenarios with four lines per instance

Figure 4.7: *A graphical box-plot representation of the distribution of the total volume for each line after scheduling scenarios with three and four lines per instance using the greedy insertion (GI), phased greedy insertion (GP), the integer programming formulations (IP_α) as well as the historical assignment (His). The median, Q_1 and Q_3 are represented as the horizontal lines in the closed box. The individually plotted coordinates are associated with the outliers. The whisker lines indicate the minimum and maximum number of cycles for non-outlier picking lines.*

adjusted the size of the maximal SKUs by rounding. This relaxed formulation showed similar results in terms of cycles with a computation time 50 times faster. However, scenarios with seven or more picking lines could still not be solved in a reasonable time.

To solve larger instances a heuristic approach was introduced based on an algorithm for GAPs by Martello & Toth [14] which greedily inserted DBNs. This approach yielded good results, but in some cases not all the DBNs were assigned to a picking line. A further phased greedy insertion approach was developed which held back smaller DBNs for later insertion to achieve feasibility. The approach showed good results and achieved a feasible solution for all test instances.

Following the improvement in terms of picker walking distance the effects of the DBN assignments on other factors was investigated. Carton utilisation was measured by determining the number of orders which consisted of a small volume of stock. These small orders would require additional material handling to resize their cartons. The number of these small orders increased when minimising walking distance. This increase was considered manageable by PEP although undesirable.

The second factor was the distribution of volume over different picking lines. Picking lines with too large volumes are considered as an operational risk. The proposed algorithms increased the number and volume of these large picking lines when compared to the historical case. Once again these large picking lines, both from the historical case and proposed assignments, are manageable, but undesirable.

It is proposed that PEP use an assignment approach which minimises the sum of the maximal SKUs. The GP approach is recommended because its solution quality is good and its computational times short. Future studies should consider the trade-offs seen here between walking distance, small orders and large volumes of stock on picking lines. Future work may also include developing assignment strategies which reduce the number of small orders generated and control the occurrences of large lines with respect to volume while still reducing the most important factor, namely the walking distance of the pickers.

Bibliography

[1] Accorsi R, Manzini R & Bortolini M, 2012, *A hierarchical procedure for storage allocation and assignment within an order-picking system. A case study*, International Journal of Logistics: Research and Applications, **15(6)**, pp. 351–364.

[2] Bartholdi JJ & Hackman ST, 2011, *Warehouse and Distribution Science*, The Supply Chain and Logistics Institute, School of Industrial and Systems Engineering Georgia Institute of Technology Atlanta, GA 30332-0205 USA, Available from `www.warehouse-science.com`.

[3] Chiang DMH, Lin CP & Chen MC, 2011, *The adaptive approach for storage assignment by mining data of warehouse management system for distribution centres*, Enterprise Information Systems, **5(2)**, pp. 219–234.

[4] de Koster R, Le-Duc T & Roodbergen KJ, 2007, *Design and control of warehouse order picking: A literature review*, European Journal of Operational Research, **182(2)**, pp. 481–501.

[5] Emerson CR & Schmatz DS, 1981, *Results of modeling an automated warehouse system*, Industrial Engineering, **13(8)**, pp. 28–32.

[6] Garfinkel M, 2005, *Minimizing Multi-zone Orders in the Correlated Storage Assignment Problem*, PhD, Georgia Institute of Technology, Georgia, (AT).

[7] Hagspihl R & Visagie S, 2014, *The number of pickers and stock-keeping unit arrangement on a uni-directional picking line*, The South African Journal of Industrial Engineering, **25(3)**, pp. 169–183.

[8] Hassini E, 2009, *One-Dimensional Carousel Storage Problems: Applications, Review and Generalizations*, INFOR, **47(2)**, pp. 81–92.

[9] IBM, *CPLEX*, Available from `www.ibm.com`.

[10] Jane CC & Laih YW, 2005, *A clustering algorithm for item assignment in a synchronized zone order picking system*, European Journal of Operational Research, **166(2)**, pp. 489–496.

[11] Kim BS & Smith JS, 2012, *Slotting methodology using correlated improvement for a zone-based carton picking distribution system*, Computers & Industrial Engineering, **62(1)**, pp. 286–295.

[12] Litvak N & Maia V, 2010, *A survey on performance analysis of warehouse carousel systems*, Statistica Neerlandica, **64(4)**, pp. 401–447.

[13] Manzini R, 2006, *Correlated Storage Assignment in an Order Picking System*, International Journal of Industrial Engineering, **13(4)**, pp. 384–394.

[14] Martello S & Toth P, 1990, *Knapsack Problems Algorithms and Computer Implementations*, Johhn Wiley & Sons, West Sussex England.

[15] Matthews J & Visagie SE, 2013, *Order sequencing on a unidirectional cyclical picking line*, European Journal of Operational Research, **231(1)**, pp. 79–87.

[16] MATTHEWS J & VISAGIE SE, *Picking Line Data*, Available from http://scholar.sun.ac.za/handle/10019.1/86110.

[17] MATTHEWS J & VISAGIE SE, 2014, *SKU arrangement on a unidirectional picking line*, Submitted.

[18] PAN JCH & WU MH, 2009, *A study of storage assignment problem for an order picking line in a pick-and-pass warehousing system*, Computers & Industrial Engineering, **57(1)**, pp. 261–268.

[19] PARAGON DECISION TECHNOLOGY, *AIMMS*, Available from www.aimms.com.

[20] PEP STORES LTD, *PEP*, Available from www.pepstores.com.

[21] SERDYN H, *Carton Sizes*, personal communication. Contactable at: henniese@pepkorit.com.

[22] TOMPKINS JA, 2010, *Facilities planning*, 4th Edition, John Wiley & Sons, Hoboken, NJ.

CHAPTER 5

SKU assignment using correlations to unidirectional picking lines

5.1 Introduction

Warehouses form a central part of supply chains. The role of warehouses is typically to match supply with demand and to consolidate product from multiple suppliers [2]. To play this role, stock must be stored, moved and picked in the warehouse using one or more of a variety of different layouts, mechanisms, picking systems and storage equipment depending on product and market characteristics. One of the essential parts of warehouse management is the placing of stock into locations that improve stock movement and picking efficiency.

The warehouse slotting problem is described by Kim & Smith [8] as determining an assignment of stock keeping units (SKUs) to picking slots[1] to support carton picking. Ideally SKUs which are usually placed in the same carton should be placed as near to each other as possible. Warehouses typically do not re-slot in the short term because in most cases long term SKU correlations are used as a desirability measure to slot SKUs close to each other. Furthermore the time and cost involved to re-slot is often too high.

Following on a study by Matthews & Visagie [14] a distribution centre (DC) owned by Pep Stores Ltd[2] (PEP) is considered [16]. A major influence on the order pick system in use at their DCs is the centralised stock management of PEP. Stock levels for each store are managed collectively and stock outflows are planned centrally at SKU level creating a push system. During an outflow for a SKU all stores requiring that SKU are stocked with the SKU in a single pick operation. Distributions (DBNs) which consist of a set of SKUs that are of the same product but differ in size are used to achieve this. The make-up of a DBN is determined by the central planning department which sets how much of each SKU in the DBN should go to each store. These DBN pick instructions are issued (or released) to the DC. All the SKUs in the DBN are picked in the same batch once the DBN pick instructions are released.

A type of forward pick area – as described by Bartholdi & Hackman [2] – consisting of 12 picking lines is used to pick these DBNs. A picking line has 56 slots (or locations) each holding up to five pallet loads of the same SKU and is used for all piece picking in the DC. These picking lines are serviced by multiple pickers and are able to operate in parallel to each other. Figure 5.1 illustrates the layout of the picking lines in this forward pick area.

[1] A picking slot is a storage location which is directly accessible by pickers.
[2] Pep is the largest single brand retailer in South Africa.

Picking lines operate in waves, defined as a set of SKUs and their corresponding store requirements which are to be picked in a batch. Each wave of SKUs is placed on its own picking line and all the store requirements for those SKUs are picked in a single operation. Once all the picking is completed any remaining stock is removed and a new wave of SKUs is brought to the picking line. All the SKUs associated with the same DBN are placed on the same picking line ensuring that all the SKUs in the same DBN arrive at the store at the same time. This process of populating, picking and clearing stock on a picking line may take anything from four hours to two days depending on the number and size of orders associated with, and the characteristics of, the SKUs assigned to that wave on the picking line.

Due to the varying rates at which picking lines are completed and the parallelisation of the picking line area the number of picking lines which become available for new waves during each day varies. DBNs are scheduled onto available picking lines using a first-in, first-out (FIFO) system in an attempt to ensure that all DBNs are processed within the desired threshold of seven days from receiving both the pick instructions from the planning department and the physical stock from the suppliers.

During the picking phase pickers walk in a clockwise direction around a picking line sequentially picking orders. Order consolidation is not performed to ensure that picking lines are managed independently from each other. Instead pickers pick directly into cartons placing the completed cartons onto a conveyor belt. Completed cartons are then stored in buffer areas located in the outbound section of the DC which are emptied on a regular basis based on delivery schedules.

New cartons as well as re-cycled cartons from suppliers are used and are available around the picking line. Cartons only hold stock from a single picking line and are closed and shipped as they come from the picking line. When pickers select a carton to hold the stock for an order they do not know what volume of stock is required for that order and are unable to select an appropriate sized carton. Moreover, in some cases stores only require a small volume of stock within a picking line and are unable to fill even the smallest available carton in the DC. Many cartons which are selected for orders with small volumes of stock have excess capacity. These cartons are manually resized into smaller dimensions to reduce volume and are undesirable as they increase the per-volume handling cost throughout the supply chain.

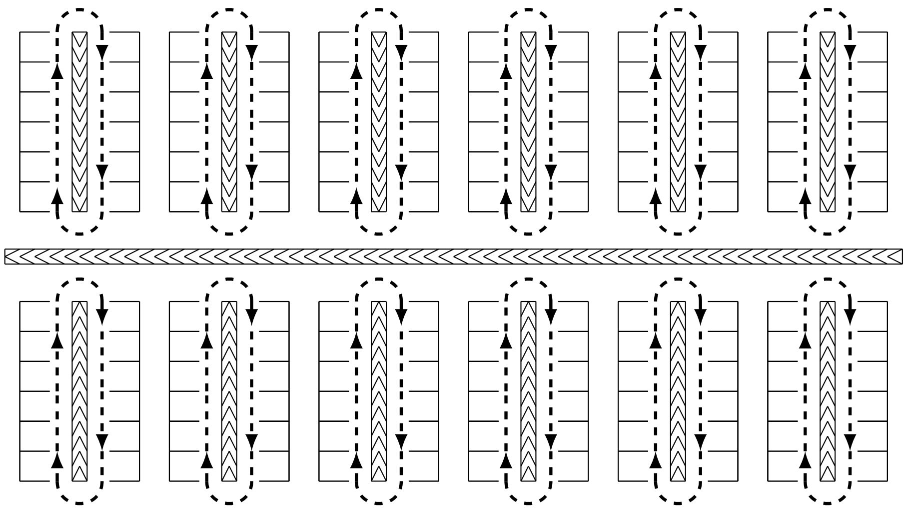


Figure 5.1: *A schematic representation of the layout of the 12 picking lines in the DC, six on either side of the main conveyor. The dashed lines indicate the movement of the pickers around the conveyor belts [14]*

Matthews & Visagie [14] suggested approaches to assign DBNs to picking lines while minimising the total walking distance of pickers by using the concept of a maximal SKU[3]. Significant improvements were made on the historical method by using both integer programming (IP) and heuristic approaches. They pointed out that focusing on walking distance alone increased the number of small cartons produced, as many stores required a small volume of stock from certain picking lines. In addition, operational risk was increased as more picking lines required excessively large volumes of stock which increased the overall time which a picking line is occupied by a single wave of picking and might result in a need to replenish stock during a wave of picking. The focus of this paper is to address these two additional issues by using SKU correlations while still achieving satisfactory walking distances.

The remainder of this paper is structured as follows. A discussion of related work in existing literature is given in §5.2. The four solution approaches using correlations are introduced in §5.3 with the results presented in §5.4. The paper is concluded in §5.5.

5.2 Literature review

Accorsi *et al.* [1] addressed both the storage allocation and storage assignment problems simultaneously. The storage allocation problem focuses on the amount of stock stored in each location, typically addressing the issues around replenishment costs, while the storage assignment problem focuses on the physical location of stock in an effort to minimise order picking costs. Three main problems were identified when optimising order picking systems with a forward pick area, namely:

1. Which SKUs should be in the forward pick area?
2. How much of each SKU should be in the forward pick area?
3. Where should each SKU be stored?

Bartholdi & Hackman [2] addressed the first problem and introduced three approaches for the second problem namely the equal space, equal time and optimal allocation strategies. Accorsi *et al.* [1] addressed the final storage assignment problem and identified three main approaches, namely the class-based, ranked-index-based and correlation storage assignment policies. The clustering storage assignment policy was further expanded as three sequential steps, namely correlation analysis, clustering and priority list determination with cluster assignment.

Accorsi *et al.* [1] also proposed a top down hierarchical procedure for overall order picking optimisation which was applied to a case study. Numerous combinations of different approaches to each problem were considered including different allocation rules, correlation measures and clustering algorithms. It was shown that considering both SKU allocation and SKU assignment simultaneously yields better overall order picking costs compared to sequentially solving each problem.

Although Accorsi *et al.* [1] proposes a generalised framework for order picking optimisation the detail of the SKU assignment approaches have often been governed by DC layouts and management practices. Chiang *et al.* [4] used data mining techniques to assign newly arriving SKUs to available slots in a rectangular DC with a S-shaped picking strategy. An association index was developed between SKUs and available locations based on association rule mining, SKU

[3] A maximal SKU is a SKU within a set of SKUs with the highest number of stores requiring it.

turnover rates and the distance from a location to the exit. The association rule mining used SKU correlations between already assigned and unassigned SKUs. A generalised assignment problem was formulated to assign SKUs to available locations which maximised the sum of these association indices.

Bindi *et al.* [3] also investigated storage allocation rules for a rectangular DC with parallel shelves. Two processes were identified, namely family grouping and storage allocation. A proposed similarity measure, based on SKU correlations and a stock turn coefficient (total stock movement over average stock level) was developed and compared to a Jaccard statistic. Several storage assignment rules were also tested as part of the storage allocation process. Extensive testing by means of what-if analysis for a case study showed that correlation measures significantly improve overall throughput of the DC.

Manzini [9] minimised total picking time in a rectangular warehouse by using SKU correlations to arrange SKUs. In this case, however, the warehouse had two orthogonal sets of shelves. The DC employed a picker-to-parts system and picking vehicles with a finite capacity using a composite picking strategy. Three solution approaches based on correlations were proposed, namely a clustering approach, a parallel algorithm and a sequential algorithm. It was noted that the sequential approach, which used the last assigned SKU to determine the next SKU to assign to a cluster, had a risk of generating correlated couplets of SKUs instead of maximising overall correlations.

A SKU assignment problem in a synchronised zone order picking system was investigated by Jane & Laih [7] using correlation and clustering techniques. All zones processed the same order at the same time and only once an order has been picked in all zones can a new order begin for any zone. The completion time of an order was thus seen as the longest completion over all zones and an objective was therefore defined by using correlations which balanced the workloads of each order over all zones

The structure and layout of the DC play a role in determining appropriate SKU assignment techniques as seen by the previous studies. Kim & Smith [8] investigated a carton order picking system which has many similarities to the order picking system discussed in this paper. The DC considered by Kim & Smith [8] also had re-slotting which was performed on a daily basis. Different sets of SKUs were picked on different days of the week which created vastly different SKU correlations for each wave of picking. The time and cost required to re-slot was reasonable and re-slotting was therefore performed during the night followed by a wave of picking during the next day.

The picking area considered by Kim & Smith [8] consisted of a number of single aisle zones, each with a single picker. Cartons requiring SKUs from multiple zones are conveyed between zones which removes the need for later consolidation. Pickers receive new orders[4] for their zone from a designated starting point at one end of the aisle and proceed to pick all required SKUs before placing the carton on a conveyor belt and walking back to the start to begin a new order. In this way the distance walked by a picker to complete an order in a zone is equal to twice the distance from the start to the furthest required SKU.

Kim & Smith [8] considered the SKU slotting problem while minimising the total completion time to pick a wave. Orders typically require stock from other zones and thus the total completion time of a wave was determined by the zone with the longest completion time to pick all orders. An IP formulation was developed which minimised walking distance, but it was too complex to solve. Three further heuristic approaches were therefore introduced, namely a

[4]New orders for a zone includes orders picked in other zones.

steepest descent neighbourhood slotting heuristic, a correlated slotting heuristic and a simulated annealing slotting heuristic. These approaches used correlations to determine good pairs of SKUs to switch by reasoning that SKUs with high correlations should be near each other. Once a switch is made the total walking distance is then re-calculated. It was shown that the simulated annealing approach performed best.

Although the order picking system addressed in this paper has a similar structure and re-slotting methodology to the study by Kim & Smith [8] the structure of each zone/picking line is fundamentally different. Instead of the linear zone the cyclical structure of a picking line shows many similarities to a carousel system. A carousel system is an automated storage and retrieval system with a set of shelves which rotate to present stock to pickers. Hassini [5] presents an extensive literature study on carousel systems. Hassini [5] noted that correlations between SKUs have received little attention in the carousel context. It is further suggested that SKU correlations should be used when assigning SKUs to carousels as well as locations within a carousel.

There has also been some attention paid in literature to the exact order picking setup discussed in this paper with some of the different decision tiers of this order picking system being addressed. These decisions include the sequence in which orders are passed to pickers, the arrangement of SKUs in a picking line and the assignment of DBNs to picking lines. Matthews & Visagie [11] considered the problem of sequencing orders for pickers in a unidirectional picking line, with fixed SKU slotting, while minimising the total distance travelled. The concept of a maximal cut was introduced and an IP formulation was developed which generated a solution within one pick cycle of a lower bound.

Matthews & Visagie [13] considered the SKU arrangement on a single picking line. SKUs which have already been assigned to the picking line are arranged while minimising the total walking distance using the maximal cut approach as suggested by Matthews & Visagie [11]. An IP formulation was presented which was shown not to be solvable in a realistic time frame for problem instances with more than 15 SKUs. Matthews & Visagie [13] also tested several heuristic methods including an organ pipe and a greedy approach, both of which are optimal for some carousel systems which have many similarities to unidirectional picking lines. These heuristics were tested against historical arrangements as well as a set of random solutions. It was found that the scope for optimisation when arranging SKUs on a single picking line was minimal. A lower bound for the number of cycles traversed was also identified by considering the SKU with the maximum number of stores requiring it, called a maximal SKU. At least one cycle would need to be traversed for each store requiring the maximal SKU (*i.e.* the size of the maximal SKU) which generated this lower bound.

Matthews & Visagie [14] generalised their study considering the assignment of DBNs over multiple picking lines. It was reasoned that the maximal SKU measure of a lower bound should be correlated with the actual number of cycles traversed. Matthews & Visagie [14] therefore minimised the sum of the sizes of the maximal SKUs for each picking line to reduce total walking distance. An IP formulation with this new objective was developed which was not solvable for problem instances with more than four picking lines. A further relaxation of this formulation was therefore developed which rounded the size of the maximal SKUs in an effort to reduce the computational effort of proving an optimal solution. This relaxation showed faster computational times but was still not solvable in a realistic time frame (within 10 minutes) for problem instances with more than six picking lines.

A greedy insertion approach based on the algorithm by Martello & Toth [10] for multiple knapsacks was therefore developed. It assigns DBNs to picking lines in a greedy fashion based on the size of the maximal SKUs and the available space in the picking line. Although this approach yielded good results in a short computational time, in some cases a feasible solution could not be found because all the DBNs were not assigned to a picking line. A phased insertion approach was thus developed which held certain small DBNs back for a second round of insertion to ensure feasibility. The results were similar in terms of walking distance and in all cases a feasible solution could be found.

For all the studies mentioned above, the performance of solution approaches are only compared based on a single measure, namely picking speed and efficiency. It is clear from the studies discussed in this section that for many DC configurations using SKU correlations to assign SKUs to slots improves the speed of the order pick operation. In most cases only correlations between adjacent SKUs and not a broader neighbourhood are included in the objective function. No correlation approaches have yet been adapted for or applied to the order picking system presented in this paper.

5.3 Solution approaches

When applying the top down hierarchical procedure by Accorsi *et al.* [1] to the order picking system at PEP only the storage assignment phase needs to be applied as all piece picking must be processed on a picking line. Replenishment while picking is in progress is not present at PEP because all required stock for a pick wave is stored in the picking line before a wave of picking begins.

The storage assignment phase may further be simplified by only considering a clustering based approach. An index based approach for the storage assignment problem, which typically addresses restock travel distances, is not appropriate for the order picking system at PEP as all picking lines may be viewed as equidistant from the reserve area and restocks are rare. Only the correlation and clustering steps are required when assigning SKUs to picking lines as the number of SKUs assigned to each cluster should match the number of available locations for an available picking line. This removes the need for the priority list determination and cluster assignment.

Using the simplified top down hierarchical procedure by Accorsi *et al.* [1] as well as the study by Matthews & Visagie [14] the assignment of SKUs may be seen as two phased. Firstly each DBN (d) in the set of DBNs ($\mathcal{D}$) needs to be assigned to a picking line. Once a set of DBNs has been assigned to a picking line the SKUs associated with those DBNs should be arranged by assigning them to individual locations in a SKU arrangement phase. The walking distance of the pickers can be calculated only once each SKU has been assigned to a location. The first phase is illustrated on the left and the second phase on the right hand side of Figure 5.2.

Matthews & Visagie [13] investigated approaches for the SKU arrangement phase for a single picking line and made use of the maximal cut approach described in Matthews & Visagie [11] to evaluate the resulting walking distances of the different arrangements. They showed that savings were minimal during this phase and that the problem was too complex to be solved exactly suggesting that the two assignment phases need to be handled independently. The SKU arrangement phase will therefore be solved separately using the greedy approach by Matthews & Visagie [13] as it is fast, easy to implement and is known to achieve good results. The focus therefore moves specifically to the assignment of DBNs to a set of picking lines ($\mathcal{L}$).

In the picking line assignment phase each DBN requires a number of locations ($|d|$) in a picking lines. Each picking line (l) in the set of available picking lines ($\mathcal{L}$) has a number of available loca-

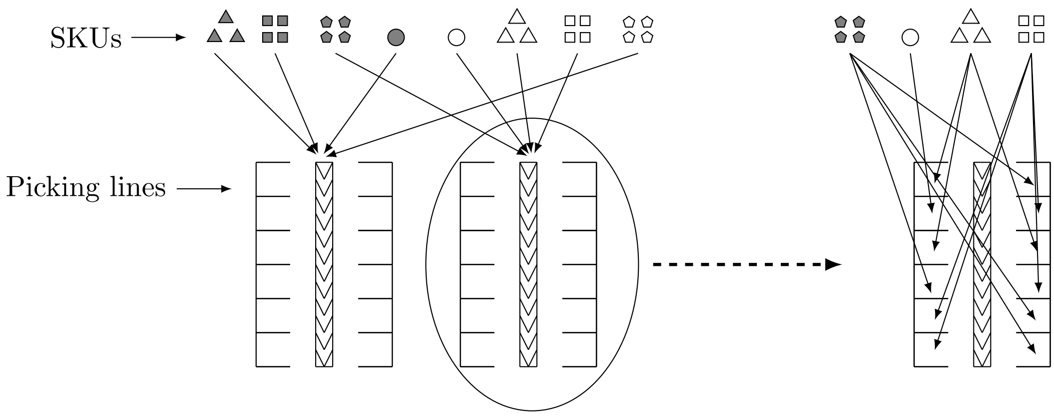


Figure 5.2: *A schematic representation of the slotting phases in the DC. Each shape represents a SKU and clusters of the same shape with the same shading represent DBNs.*

tions ($|l|$). The current approach used to assign DBNs is to spread work, measured by volume of stock, evenly between available picking lines which does not take into account walking distances of pickers. An underlying principle of each approach by Kim & Smith [8] was to interchange SKUs between slots followed by an objective function re-evaluation. Several characteristics of the problem considered here points against the use of SKU interchanges when considering uni-directional picking lines. DBNs which vary in size (number of required locations) would need to be interchanged in their entirety between picking lines. This creates more complexity as either only DBNs of the same size can be interchanged or sets of DBNs with the same number of SKUs collectively need to be interchanged. A phased greedy insertion approach was therefore introduced by Matthews & Visagie [14] to insert DBNs into available picking lines. Here DBNs are ranked according to some desirability measure and inserted sequentially into available picking lines. If a feasible solution is not found the DBNs are segmented into two subsets according to their size (number of SKUs and number of stores). These different subsets are then inserted into the available picking lines in two phases. These subsets iteratively change in size until a feasible solution is found.

Matthews & Visagie [14] used a maximal SKU measure with the phased greedy insertion algorithm (GP) which minimised the sum of the sizes of the maximal SKUs. For each DBN the SKU which has the highest number of stores requiring it (referred to as a maximal SKU) is considered and DBNs are ranked according to the size of this maximal SKU denoted as $\lceil d \rceil$. It is, however, proposed that correlations between DBNs should be considered to reduce the number of small cartons produced while still maintaining good walking distances. A correlation measure is therefore introduced as $\mathcal{B}_a \cap \mathcal{B}_b$, where $\mathcal{B}_a$ represents the set of stores requiring at least one SKU from DBN set $\mathcal{D}_a \subseteq \mathcal{D}$. By assigning DBNs with strong correlations in terms of this measure to the same picking line it would be expected that both the walking distance would be shorter and the size of each order in a picking line would be larger because more SKUs that have to be picked for the same stores will be grouped together in the same wave of picking.

Four possible desirability scores which use correlation measures were used to rank DBNs during the phased greedy insertion procedure. The first desirability score denoted as ADT considers the number of stores required by the candidate DBN and which requires at least one DBN already assigned to the picking line. This is achieved by merging all assigned DBNs in a picking line and considering them as a single DBN. The intersection of the set of stores requiring the candidate DBN and this new merged DBN (or correlation between the two DBNs) is then calculated. This desirability score is defined as

$$S(\mathcal{D}_l, d) = |\mathcal{B}_l \cap \mathcal{B}_d|. \tag{5.1}$$

The second desirability score (ADS), defined as

$$S(\mathcal{D}_l, d) = \sum_{a \in \mathcal{D}_l} |\mathcal{B}_a \cap \mathcal{B}_d|, \tag{5.2}$$

considers the correlations of a candidate DBN with all preassigned DBNs individually. This is achieved by calculating the sum of all the correlations between assigned DBNs and the candidate DBN. By assigning DBNs using these desirability scores picking lines should have fewer stores which only require one or two SKUs resulting in fewer small cartons being produced. In addition by increasing the number of shared stores the physical pick density (picks per store) of each store should increase which should create efficient pick cycles as pickers will be picking from more locations per cycle.

Bindi *et al.* [3] proposed a similarity measure which used both an adjusted Jaccard statistic and a stock turn coefficient, defined as the ratio between the total stock movement and average stock quantity. The nature of the order picking system considered in this paper does not, however, lend itself to the use of stock turn in a similarity measure due to the wave principle and the frequency at which picking lines are built. A third desirability score, defined as

$$S(\mathcal{D}_l, d) = \frac{\mathcal{B}_l \cap \mathcal{B}_d}{\mathcal{B}_l \cup \mathcal{B}_d}, \tag{5.3}$$

is based on the Jaccard statistic (JCT) and is included in the tests. Finally a desirability score (JCS)

$$S(\mathcal{D}_l, d) = \sum_{a \in \mathcal{D}_l} \frac{\mathcal{B}_a \cap \mathcal{B}_d}{\mathcal{B}_a \cup \mathcal{B}_d} \tag{5.4}$$

is introduced. The JCS measure is similar to the ADS measure, but scaled relative to the number of DBNs in the two subsets. Here the sum of the Jaccard statistics between all DBNs and the candidate DBN is calculated. The GP algorithm used to insert DBNs based on a desirability measure is illustrated in Algorithm 7.

Procedure 6: A partial greedy insertion of DBNs using a desirability measure.

Data: A set of picking lines $\mathcal{L}$ in descending order according to $|l|$
A set of DBNs $\mathcal{D}$
A set of pre-assigned DBNs $\mathcal{D}_l$ associated with each picking line
Result: A final set of assigned DBNs $\mathcal{D}_l$ associated with each picking line

```
for Each picking line l ∈ L do
    while an unassigned DBN exists which fits into the remainging locations of picking line l do
        Select the DBN with the largest desirability score, S(D_l, d), which fits into set D_l
        Assign this DBN to set D_l
    end
end
```

Algorithm 7: A sequential phased insertion of DBNs using a desirability measure.

Data: A set of picking lines $\mathcal{L}$
A set of DBNs $\mathcal{D}$
Result: An assignment of DBNs to picking lines
$\beta = 0$
while *an unassigned DBN exists* **do**
  Clear all assignments of DBNs
  Insert all DBNs where $|d| > 1$ or $\lceil d \rceil > \beta$ using Procedure 6
  Insert all remaining DBNs using Procedure 6
  $\beta = \max_{d/|d|>\beta} |d|$
end

5.4 Results

The four proposed desirability scores were tested using a phased greedy insertion approach on seven scenarios from real life historical data. Each problem instance comprised of a number of picking lines which were scheduled for the same historical day and the historical DBNs assigned to them. Each test scenario consisted of a set of these problem instances each with the same number of picking lines per day which allows for easier comparison. All the scenarios are available on-line [12]. A summary of the properties of these scenarios is given in Table 5.1. All testing was performed on an Intel i7 2 GHz processor with eight GB ram running the Windows 7 operating system. All mathematical formulations were solved with CPLEX 12.3 and coded in AIMMS 3.12 [6, 15].

Number of lines per day	Number of problem instances	Number of DBNs
2	61	2592
3	53	3437
4	49	4146
5	38	4109
6	32	4161
7	22	3177
8	14	2148

Table 5.1: *The composition of the scenarios from historical data.*

The results of all the approaches were compared to the maximal SKU phased greedy insertion approach (GP) by Matthews & Visagie [14]. The performance of the approaches are compared using three measures, namely walking distance, the number of small cartons produced and volume distribution. Table 5.2 illustrates the total distance walked for each scenario for each approach. It is clear that the GP approach performs the best in terms of walking distance, while the ADT approach shows the best results for approaches using correlations. All approaches using correlations have marginally longer walking distances (within 5%) compared to the GP. All the presented approaches still improve on the historical results by approximately 20%. In both cases the summed correlation measures (ADS, JCS) perform worse than their parent scores (ADT, JCT).

A summary of the number of small cartons produced by each approach is given in Table 5.3. Small orders (that cause cartons to have excess volume capacity) have less than 0.006 m^3 volume of stock from a picking line assigned to them [17]. In terms of number of small orders the worst performing approach is the GP approach. All the proposed correlation measures show lower numbers of small cartons produced compared to the GP approach as the number of small orders

Scenario	His	GP	ADT	ADS	JCT	JCS
2	7168	6019	6193	6212	6191	6224
3	9289	7515	7710	7749	7712	7791
4	11532	9015	9282	9311	9289	9363
5	11259	8618	8922	8932	8923	8988
6	10706	8129	8363	8384	8371	8442
7	8412	6366	6540	6559	6551	6608
8	6011	4715	4847	4856	4852	4870

Table 5.2: *The total number of kilometres walked in each scenario for all the solution approaches (ADT, ADS, JCT, JCS) as well as the historical assignment (His) and GP approach by [14].*

is roughly halved. These approaches still perform worse compared to the historical assignments, but this is offset by the improvement in walking distance.

Scenario	His	GP	ADT	ADS	JCT	JCS
2	0.05%	0.14%	0.09%	0.09%	0.09%	0.08%
3	0.04%	0.13%	0.08%	0.08%	0.08%	0.08%
4	0.03%	0.12%	0.07%	0.07%	0.07%	0.07%
5	0.03%	0.13%	0.06%	0.07%	0.07%	0.08%
6	0.04%	0.12%	0.06%	0.07%	0.06%	0.07%
7	0.03%	0.13%	0.06%	0.07%	0.06%	0.07%
8	0.03%	0.13%	0.06%	0.06%	0.06%	0.06%

Table 5.3: *The proportion of total volume of stock attributed to small orders (i.e. orders with less than* 0.006 m^3 *of stock). For the historical assignment (His) the GP approach by [14] and the correlation approaches (ADT, ADS, JCT, JCS).*

A scatter plot between these two measures is given in Figure 5.3 to better visualise the trade off between walking distance and the number of small cartons produced. Each marker indicates the total walking distance, in kilometres, as well as the proportion of total picked volume attributed to small orders. The historical assignments forms a cluster of solutions with long walking distance and good number of small cartons produced while the GP approach shows many more solutions with poor number of small cartons produced and shorter walking distances. The solutions obtained using the ADT approach are clustered with shorter walking distances and good number of small cartons produced relative to the historical solutions.

A summary of the computational times required for each approach is given in Table 5.4. The use of correlations in a desirability score increases the computation times compared to the GP approach, which only considers maximal SKUs. This is attributed to the need to dynamically change the desirability score after each insertion of a DBN. The two measures which consider the sum of correlations (ADS, JCS) have shorter computational times than their parents (ADT, JCT). This is due to the ability to calculate $\mathcal{B}_a \cap \mathcal{B}_d$ and $\mathcal{B}_a \cup \mathcal{B}_d$ for each pair of DBNs only once and use this pre-calculated value for each iteration of the ADS and JCS approaches. It is also noted that the maximum computation time for problem instances with eight picking lines is high. This is due to the number of additional insertion phases required to find a feasible solution.

A comparison of the size of the picking lines in terms of volume was also performed. Figure 5.4 illustrates the distribution of volume over all the lines for scenarios with four and five lines per problem instance respectively. Similar results were obtained for the other problem instances. It is clear that the approaches using correlations have reduced the size of largest picking lines with

Algorithm	Lines per problem instance	μ	σ	Max	Q_1	Median	Q_3	Min
ADT	2	1.86	0.59	3.07	2.35	1.92	1.41	0.65
	3	3.88	1.34	7.11	4.86	3.82	2.72	1.46
	4	6.49	2.23	13.11	7.61	5.96	5.31	1.98
	5	14.35	39.72	252.40	9.74	7.14	6.25	3.93
	6	19.31	50.63	296.30	13.02	9.97	8.29	5.99
	7	40.13	94.95	357.99	12.01	11.14	9.44	6.22
	8	87.58	206.04	734.06	14.36	12.81	11.45	7.08
ADS	2	0.99	0.42	1.96	1.26	0.98	0.72	0.06
	3	1.96	0.90	4.82	2.45	1.84	1.36	0.16
	4	3.06	1.63	8.32	3.78	2.88	1.90	0.26
	5	2.71	1.28	6.29	3.78	2.64	1.57	0.50
	6	3.25	1.85	7.81	4.63	2.81	1.74	0.76
	7	2.52	0.97	4.16	3.41	2.50	1.63	0.98
	8	3.13	4.35	16.02	2.60	1.52	0.89	0.61
GP	2	0.01	0.00	0.02	0.01	0.01	0.01	0.00
	3	0.03	0.09	0.69	0.02	0.01	0.01	0.00
	4	0.07	0.30	2.13	0.03	0.02	0.02	0.01
	5	0.07	0.23	1.43	0.04	0.03	0.02	0.02
	6	0.09	0.30	1.75	0.05	0.03	0.03	0.02
	7	0.32	0.71	2.53	0.05	0.05	0.03	0.02
	8	0.39	0.87	2.53	0.06	0.04	0.03	0.02
JCT	2	1.83	0.56	2.74	2.35	1.86	1.45	0.69
	3	4.58	1.52	8.09	5.58	4.68	3.59	1.86
	4	6.83	2.19	13.76	8.03	6.27	5.48	2.31
	5	9.87	2.90	17.53	11.82	8.94	7.63	4.77
	6	13.22	3.88	22.41	15.42	12.73	10.60	6.87
	7	14.00	4.74	25.08	15.15	13.21	12.17	7.53
	8	60.14	42.64	204.77	56.21	49.38	45.03	28.89
JCS	2	0.88	0.38	1.73	1.14	0.87	0.64	0.06
	3	1.92	0.87	4.59	2.40	1.80	1.34	0.15
	4	7.47	6.99	28.96	12.00	3.87	2.53	0.24
	5	2.84	1.26	5.01	4.07	2.83	1.68	0.51
	6	3.26	1.68	6.52	4.58	2.81	1.90	0.83
	7	2.37	0.97	4.15	3.12	2.40	1.52	0.90
	8	98.75	101.89	296.06	171.08	72.19	9.39	0.61

Table 5.4: *A comparison of computational times in seconds between the different solution approaches for each scenario. Both the average times (μ) and the standard deviation (σ) thereof are presented. Standard measures of spread are also presented with the maximum time (Max) the 25^{th} percentile (Q_1), the 50^{th} percentile (Median), the 75^{th} percentile (Q_3) and the minimum time (Min) given.*

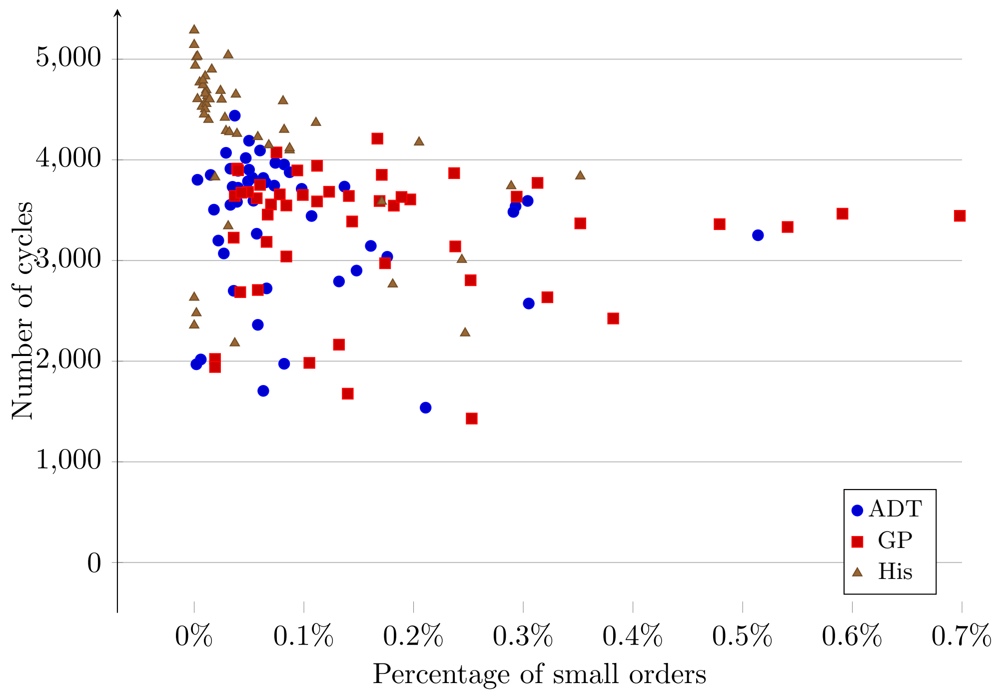


Figure 5.3: *A scatter plot between the number of cycles traversed and the percentage of the total picked volume attributed to small orders. Each marker represents a single problem instance with four picking lines.*

respect to volume. For the scenario with five picking lines per problem instance the spread of volume over lines is aligned to that of the historical case. Similar patterns were observed for the other scenarios. Correlation measures provide the best trade offs when used to assign DBNs if all three measures are taken into account. Following all the results it is proposed that the ADT approach should be used to assign DBNs.

5.5 Conclusion

A real life order picking system where re-slotting is performed on a daily basis as implemented by PEP was investigated. This investigation follows on a study by Matthews & Visagie [14]. The order picking system consisted of unidirectional picking lines in a forward pick area where all the piece picking is processed. SKUs, which are grouped together into DBNs by PEP, are batched into waves and processed in a single operation on a picking line. The number of picking lines which became available for the assignment of DBNs each day vary as the time required to stock, pick and clear picking lines varies. The assignment of DBNs to available picking lines forms the focus of this study. Assignments are evaluated in terms of the distance walked to pick all orders, the number of small cartons produced as well as the spread of volume over picking lines.

Matthews & Visagie [14] used a phased greedy insertion technique to minimise the sum of the maximal SKUs in an effort to minimise the walking distance of pickers. It was shown that this objective had negative effects on other operational areas such as the number of small cartons produced and volume distribution. Many approaches in literature use correlations to assign SKUs to locations although the main objective was still to reduce total picking time. It is therefore proposed to use correlations between DBNs as a measure to assign DBNs to picking

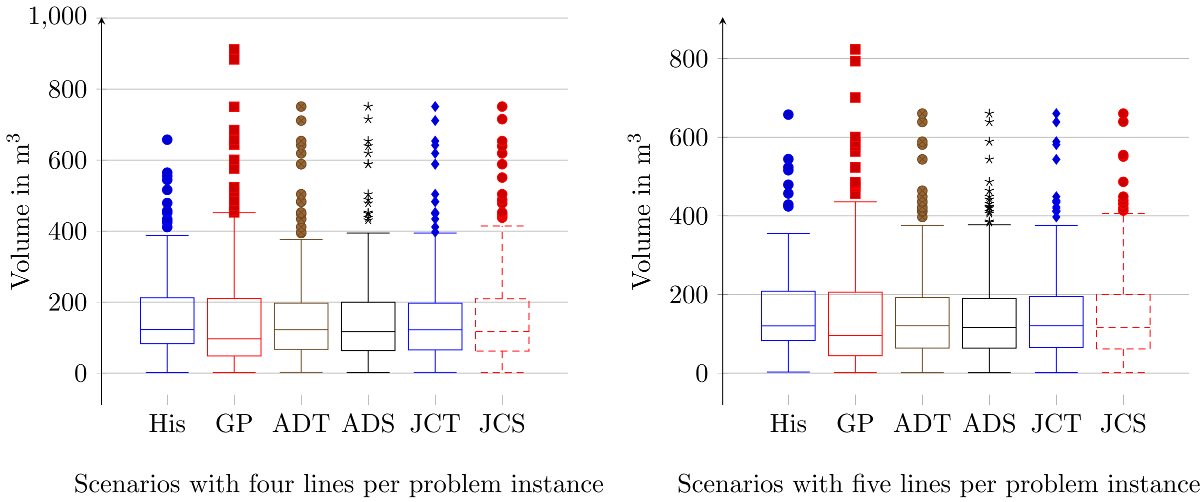


Figure 5.4: *A graphical box-plot representation of the distribution of the total volume for each line after scheduling scenarios with four and five lines per problem instance. The median (50^{th} percentile), Q1 (25^{th} percentile) and Q3 (75^{th} percentile) are represented as the horizontal lines in the closed box. The individually plotted coordinates are associated with the outliers which are either 1.5 times the inter quartile range (IQR) smaller than Q1 or larger than Q3. The whisker lines indicate the minimum and maximum number of cycles for non-outliers.*

lines to reduce the number of small cartons produced while still maintaining acceptable walking distances.

Four desirability scores were tested and compared to the historical case as well as the maximal SKU approach (GP) by Matthews & Visagie [14]. The first two scores (ADT and ADS) considered the total number of stores sharing SKUs in DBNs. The second two approaches (JCT and JCS) use the Jaccard statistic as a measure of correlation.

It was shown that the total walking distance marginally increased in comparison to the GP approach while still significantly improving on the historical case. In addition the number of small orders generated was roughly half that of the GP approach. It was also shown that using correlations resulted in a slightly better distribution of volume over picking lines, although, the large picking lines are still undesirable. It is recommended that the ADT desirability scores be used to assign DBNs to picking lines.

Using correlation measures have reduced the number of small cartons produced and improved volume distribution slightly with only a marginal increase in walking distance. Future work may include approaches to reduce these large picking lines using capacity constraints or goal programming techniques.

Bibliography

[1] Accorsi R, Manzini R & Bortolini M, 2012, *A hierarchical procedure for storage allocation and assignment within an order-picking system. A case study*, International Journal of Logistics: Research and Applications, **15(6)**, pp. 351–364.

[2] Bartholdi JJ & Hackman ST, 2011, *Warehouse and Distribution Science*, The Supply Chain and Logistics Institute, School of Industrial and Systems Engineer-

ing Georgia Institute of Technology Atlanta, GA 30332-0205 USA, Available from `www.warehouse-science.com`.

[3] BINDI F, MANZINI R, PARESCHI A & REGATTIERI A, 2009, *Similarity-based storage allocation rules in an order picking system: an application to the food service industry*, International Journal of Logistics Research and Applications, **12(4)**, pp. 233–247.

[4] CHIANG DMH, LIN CP & CHEN MC, 2011, *The adaptive approach for storage assignment by mining data of warehouse management system for distribution centres*, Enterprise Information Systems, **5(2)**, pp. 219–234.

[5] HASSINI E, 2009, *One-Dimensional Carousel Storage Problems: Applications, Review and Generalizations*, INFOR, **47(2)**, pp. 81–92.

[6] IBM, *CPLEX*, Available from `www.ibm.com`.

[7] JANE CC & LAIH YW, 2005, *A clustering algorithm for item assignment in a synchronized zone order picking system*, European Journal of Operational Research, **166(2)**, pp. 489–496.

[8] KIM BS & SMITH JS, 2012, *Slotting methodology using correlated improvement for a zone-based carton picking distribution system*, Computers & Industrial Engineering, **62(1)**, pp. 286–295.

[9] MANZINI R, 2006, *Correlated Storage Assignment in an Order Picking System*, International Journal of Industrial Engineering, **13(4)**, pp. 384–394.

[10] MARTELLO S & TOTH P, 1990, *Knapsack Problems Algorithms and Computer Implementations*, Johhn Wiley & Sons, West Sussex England.

[11] MATTHEWS J & VISAGIE SE, 2013, *Order sequencing on a unidirectional cyclical picking line*, European Journal of Operational Research, **231(1)**, pp. 79–87.

[12] MATTHEWS J & VISAGIE SE, *Picking Line Data*, Available from `http://scholar.sun.ac.za/handle/10019.1/86110`.

[13] MATTHEWS J & VISAGIE SE, 2014, *SKU arrangement on a unidirectional picking line*, Submitted.

[14] MATTHEWS J & VISAGIE SE, 2015, *SKU assignment to parallel unidirectional picking lines*, The South African Journal of Industrial Engineering, **26(1)**, pp. 235–251.

[15] PARAGON DECISION TECHNOLOGY, *AIMMS*, Available from `www.aimms.com`.

[16] PEP STORES LTD, *PEP*, Available from `www.pepstores.com`.

[17] SERDYN H, *Carton Sizes*, personal communication. Contactable at: `henniese@pepkorit.com`.

CHAPTER 6

A multi-objective approach for SKU assignment to unidirectional picking lines

6.1 Introduction and background

Distribution centre (DC) management typically revolves around two major process groups, namely inbound and outbound processes. Inbound processes include the receiving and put-away of stock, while outbound processes include order-picking, quality control, packing and shipping operations [2]. The order picking operation is typically the most time and cost intensive operation and may account for approximately 60% of all DC costs [3]. Order picking may be described as the process of consolidating product to satisfy customer orders and is often labour intensive. Following on a study by Matthews & Visagie [15] an order picking system in a DC owned by PEP Stores Ltd (PEP), the largest single retail brand in Southern Africa, is investigated [19].

PEP has approximately 1600 stores in Southern Africa selling mostly apparel. All stock planning, procurement and distribution is coordinated at central office to achieve low costs and availability in stores. Store management has little to no control over the stock sent to their stores as all decisions are made by central management. At the start of each SKU's planning cycle store requirements for the entire company are determined. Due to the central control of decisions stock flows are planned at a SKU level and not on a store level as is more common in literature [18]. SKUs have different planning cycles ranging from weekly replenishments, for products such as underwear, to a single outflow per season for products such as winter jackets. The central planning department groups these SKUs of the same product, but of different sizes, together in a distribution (DBN) to determine the store requirements of SKUs. All the SKUs in the same DBN are then planned together. Pick instructions for DBNs are generated for the entire company by taking store specific demand and size profiles into account. These instructions are then released to the DC which schedules each DBN to be picked. DBNs consisting of SKUs (and not orders) are then batched together in waves and all the store requirements for all SKUs in the same wave are picked together.

The DC uses a unique type of forward pick area consisting of unidirectional picking lines to process SKUs in waves. All piece picking is processed in this forward pick area and constitutes approximately 70% of all stock volume and 90% of all units picked. Figure 6.1 illustrates the layout of the DC with five main areas, namely the goods receiving, decanting, storage rack, picking line and full carton picking areas. The picking line area consists of 12 picking lines, six on either side of a main conveyor. A graphical representation of a picking line is shown in

Figure 6.2. Each picking line has 56 SKU locations which can hold up to five pallet loads of a single SKU. Pickers walk in a clockwise direction around the conveyor belt and may pass each other. Pickers complete orders sequentially and are assigned new orders independent of the presence of other pickers in the picking line. This independence allows pickers to be removed and added to picking lines as needed by the DC.

Pickers place items directly into cartons while picking. Full cartons are placed on a conveyor belt which conveys the cartons to a dispatch area. Here the cartons are cut to size if needed, closed with a staple gun and sent to a holding area for delivery to hubs or stores which removes the need for order consolidation. In this way each carton only holds stock picked from a single picking line for a single store.

An implication of this wave picking on parallel picking lines and no order consolidation is that SKUs assigned to different picking waves may arrive at a store on different days. It is desirable for PEP to have all of the SKUs in the same DBN arriving at a store at the same time and it is therefore required that SKUs in the same DBN be placed in the same pick wave and therefore on the same picking line.

A wave of picking has four phases, namely planning, building, picking and clearing. During the planning phase the DC assigns pending DBNs to available picking lines. The SKUs within each picking line are then arranged around the picking line and the sequence in which orders are passed to pickers is determined. The building phase requires a team of forklifts, pump trolleys and specialised high-lift cranes to place the required stock in the picking line. The picking phase commence once all the required stock is placed in the picking line. Multiple pickers then pick all the required orders. During the final clearing phase leftover stock (if any) is taken to the storage racks and the picking line is tidied for a new wave of picking consisting of a new mutually exclusive set of DBNs. The time required to build, pick and clear a picking line can range from a few hours to a few days depending on the characteristics of the SKUs and their physical location in the DC.

The order picking system considered in this paper may be seen as a parallel zone order picking system. Zone order picking typically requires work balance over zones which are underutilised or form bottlenecks [2]. When considering work balance within waves of picking two resources need to be taken into account, namely pickers and stock moving equipment.

Pickers are only required by the picking line area and may move between picking lines as needed. This independence of pickers and freedom to move between picking lines (zones) minimises the effects of work balance as long as there are still picking lines in their picking phase. Therefore the typical picker work balance issues found in literature are negligible in this parallel picking lines setup.

The movement of stock around the DC, however, occurs between all the functional areas shown in Figure 6.1. Each aisle in the storage rack may only be serviced by a single specialised high-lift crane which receives pallets from, and places retrieved pallets for pump trolleys at the base of the aisle. In addition the flow of stock in an aisle is one-way as stock for put-away is received at the storage rack on the opposite side of the picking lines and retrieved stock placed on the picking line side of the storage racks. Similarly to picking lines batches of stock movements are typically processed together and are grouped by function – such as populating a picking line with SKUs, clearing a picking line or put-away. Flexibility is essential when managing the stock movements and it is therefore undesirable for a single picking line to require a large number of stock pallet movements during the building phase.

The focus of this paper is on the assignment of DBNs to available picking lines in a single

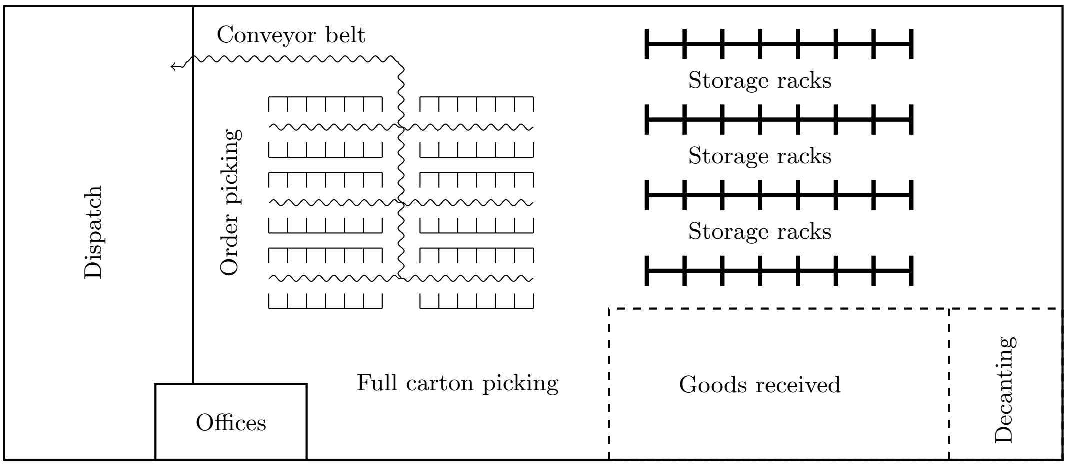


Figure 6.1: *A schematic representation of the layout of the functional areas in the DC owned by PEP.*

shift during the planning phase. Following the operational requirements for picking lines three goals are identified which should be considered when assigning DBNs to picking lines, namely minimising the walking distance of pickers, reducing the number of small cartons produced and controlling the number of stock movements required to build any one picking line.

The remainder of this paper is structured as follows, a discussion of related work in literature is given in §6.2 and a discussion of the proposed models is supplied in §6.3. The results are presented in §6.4 and the paper is concluded in §6.5.

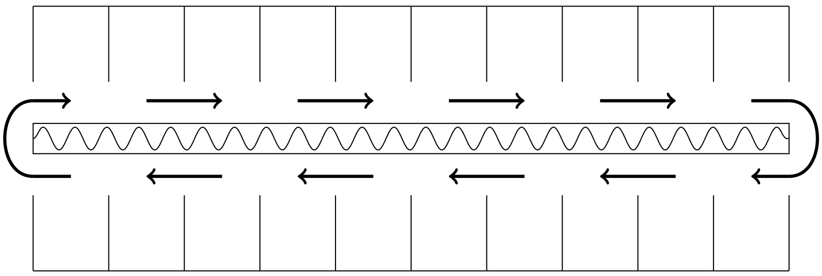


Figure 6.2: *A schematic representation of a picking line. The arrows indicate the direction in which pickers walk around the central conveyor belt picking orders.*

6.2 Literature review

The structure and layout of the DC often defines the type of SKU assignment approaches which can be used. Several configurations which show resemblance of the picking line system described in this paper are discussed in this section. These include carousel systems, synchronized zone order picking as well as previous studies on unidirectional picking lines. Some less similar, but still relevant configurations are also discussed.

A carton order picking operation with a similar re-slotting framework to the order picking system presented in this paper was investigated by Kim & Smith [9]. Different SKUs were picked on different days with re-slotting occurring during the night. Each zone comprised of a single aisle with uniform slots and is operated by a single picker. Pickers collect pending orders from a starting position at one end of the aisle and walk down the aisle picking required SKUs directly into cartons. Completed cartons are placed on an adjacent conveyor and are conveyed either to the dispatch area or another zone if SKUs from that zone are required to fulfil the order. Once the carton is placed on the conveyor the picker walked back to the start to collect a new order. The total walking distance for an order in a zone is thus twice the distance from the starting position to the furthest SKU. Kim & Smith [9] introduced a mixed integer programming (MIP) formulation to minimise the walking distance and did not consider work balance. The MIP formulation was not solvable and three further heuristic approaches were introduced, namely a sequential steepest descent neighbourhood slotting heuristic, a correlated slotting heuristic and a simulated annealing slotting heuristic. The simulated annealing slotting heuristic showed the best results The approaches by Kim & Smith [9] cannot be used in this study due to the cyclical structure of the picking lines and thus the focus is shifted to carousel systems.

A carousel is an automated storage and retrieval system typically consisting of a number of shelves which are linked together and rotate, usually in both directions, presenting stock to pickers. Carousel systems show many similarities to picking lines due to their cyclical nature. Hassini [6] provides an extensive literature review on carousel systems and expands on many carousel related systems and solution methodologies for order sequencing and SKU arrangement on carousels. In all cases, however, a stochastic set of orders is modelled for each planning horizon in contrast to the sets of deterministic orders found in wave picking.

Litvak & Maia [10] further provides an overview of research on the performance evaluation and design of carousel systems. Litvak & Maia [10] discusses problems with multiple carousels but with a single picker only. Emerson & Schmatz [4] mentions a carousel configuration consisting of 22 carousels where each pair of carousels has its own picker (thus 11 in total), but the problem of assigning SKUs to carousel bins which is relevant here was not addressed. Thus the focus shifts to unidirectional picking lines.

Matthews & Visagie [12] considered the sequence in which orders are passed to pickers for a fixed arrangement of SKUs while minimising walking distance. An integer programming (IP) formulation was developed using the concept of a maximal cut which generates a solution within one pick cycle of a lower bound. The arrangement of SKUs around the picking lines to minimise total walking distance was considered [14]. Two IP formulations were proposed for determining a lower bound on walking distance, both of which were not solvable for typical real life problem instances. A maximal SKU (a SKU with the most stores requiring it) was identified as a lower bound. Several heuristic approaches were tested, including the organ pipe and greedy heuristics. These heuristics are known to be optimal for some carousel systems [20, 21]. The results were compared to a set of historical arrangements and a set of random arrangements. A conclusion was that there is a minimal marginal benefit to better arranging SKUs in a picking line if the deterministic set of orders are sequenced correctly.

Matthews & Visagie [15] focussed on assigning DBNs to available picking lines for a single day while minimising the total walking distance. Due to the computational complexity of sequencing orders and the minimal impact of arranging SKUs on a picking line the maximal SKU was used as an estimate of total walking distance. The objective was therefore to minimise the sum of all the sizes of the maximal SKUs over all picking lines while assigning all scheduled DBNs to available picking lines. Only after all the DBNs have been assigned to picking lines using this approximation would the actual number of cycles traversed be calculated by arranging the SKUs with a greedy approach and solving the order sequencing problem. An IP formulation was introduced to assign DBNs to picking lines which was not solvable within a reasonable time for real life problem instances. A greedy phased insertion approach (GP) was therefore developed and compared to historical results. Significant improvements were made in terms of walking distance compared to historical cases. However, the number of small cartons produced increased and picking waves were created which needed an undesirable number of pallet movements to populate the picking line with SKUs.

Matthews & Visagie [16] considered correlations between DBNs to assign them to available picking lines using a phased greedy insertion approach. It was shown that using correlations significantly reduced the number of small cartons produced with a small increase in walking distance compared to minimising the sum of the sizes of maximal SKUs. However, large picking lines were still generated which required an undesirable number of pallet movements to populate them.

The parallel nature of the picking lines shows many similarities to synchronised zone picking. Some studies have been published concerning work balance between synchronised zones. Jane & Laih [8] considered SKU assignment in a manual, synchronized zone, pick by light order picking system. Stock requirements for each zone would be picked independently within each zone followed by order consolidation. The zone which required the most time to pick its stock requirements for the set of scheduled orders would therefore determine the overall time required to pick the entire set of orders across all the zones. The time required to pick a set of orders is therefore determined by the zone which requires the longest time or has the most work for pickers. The objective was thus to balance work between zones to manage the most dense zone. Jane & Laih [8] used a similarity measure between SKUs – measured as the number of orders requiring both SKUs – and developed a formulation which minimised the similarities between SKUs within the same zone. Work is balanced between zones by minimising similarities within zones resulting in orders that require a more evenly distributed number of SKUs from each zone. A mathematical formulation was presented which was solved by means of a heuristic approach.

Garfinkel [5] considered both synchronized and sequential zoned order picking systems and minimised the number of zones visited to pick orders. Normally walking distance would be minimised but Garfinkel [5] minimised the number of zones visited to pick orders. Garfinkel [5] suggests that minimising the number of zones visited would be beneficial in cases where only a few SKUs are required per order, batching is not desirable and sorting is expensive. This approach is not conducive to the order picking system in this paper due to the presence of wave picking and the large number of SKUs required by stores.

Although Jane & Laih (2005) as well as Garfinkel [5] considered balancing work between zones this is not suitable for the order picking system considered in this paper. Merely balancing work between picking lines would be undesirable as in most cases this would yield worst case scenarios in terms of walking distance as each picking line is assigned a SKU with a large maximal SKU. Moreover, it has been shown by Matthews & Visagie [16] that increasing similarities within picking lines improves the goal of reducing the number of small cartons generated.

SKU slotting in literature focuses on minimising overall picking time. Manzini [11] used SKU correlations to assign SKUs to storage locations in a DC with two sets of storage racks orthogonal to each other. A composite routing strategy was used by a set of picking vehicles with finite capacity. Several heuristic approaches to solve the assignment problem were proposed. Due to the layout of the DC work balance was not an issue and only total picking time was considered as a performance measure and is thus not usable in the context considered here.

Accorsi *et al.* [1] considered both SKU allocation (size of space allocated to a SKU) and SKU assignment (location of a SKU) in a forward pick area. The objective was to reduced the overall travel distances for both picking and restocking. Several assignment and allocation approaches were compared using simulation and what-if analysis. They concluded that considering both problems simultaneously decreases the overall restocking distance and cost. Although both the picking and restocking operations are considered the restocking did not follow a wave principle and only overall movement was considered. These findings are thus not suitable for the order picking system considered in this paper.

6.3 Models and algorithms

The problem considered in this paper may be seen as a generalised assignment problem with three desirability criteria. A set of DBNs of various sizes must be assigned to a set of picking lines while minimising walking distance, reducing the number of small cartons produced and avoiding a large number of stock movements to build any one picking line for a wave of picking. Matthews & Visagie [15] showed that the maximal SKU can be used as an estimator for the total walking distance. However, the actual number of cycles traversed by pickers must be calculated after each assignment is finalised. After consultation with PEP and following a previous study by Matthews & Visagie [15] the number of small cartons will be estimated by calculating the number of orders which require less than 0.006 m^3 (approximately the size of a typical shoe box) of stock. The warehouse management system (WMS) at PEP currently does not record and store the individual pallet movements and thus the physical volume of stock required to fulfil all the orders in a wave is used as a proxy. This data is available in the WMS and will therefore be used as a measure of the number of stock movements required to build a picking line[1].

Using volume as a measure of the number of stock movements for a wave capacity (C) is introduced. It is desirable that there should always be less than C cubic meters of stock assigned to a single picking line for a wave of picking. A formulation by Matthews & Visagie [15] is adapted to assign DBNs to picking lines while ensuring that no picking line has more than C m^3 of stock. This formulation assigns DBNs while minimising the sum of the sizes of the maximal SKUs. An additional capacity constraint is now added to limit the volume of stock assigned to a picking line. The following parameters have to be set in the model. Let

$\mathcal{L}$	be the set of all picking lines with elements l,
$\lvert l\rvert$	be the number of SKU locations available for picking line l,
$\mathcal{D}$	be the set of all DBNs with elements d,
$\lvert d\rvert$	be the number of locations required by DBN d,
$\lceil d\rceil$	be the size of the maximal SKU associated with DBN d,
$\tilde{d}$	be the total volume of stock associated with DBN d and
C	be the maximum allowable volume on a picking line.

Moreover, two sets of variables are needed to formulate the assignment of DBNs to picking lines. Let

$$x_{dl} = \begin{cases} 1 & \text{if DBN } d \text{ is assigned to picking line } l \\ 0 & \text{otherwise} \end{cases}$$

and

y_l as the size of the maximal SKU for picking line l.

[1]All models may be easily adjusted should the number of stock movements required for a wave become available.

In terms of these symbols the objective is to

$$\text{minimise} \sum_{l\in\mathcal{L}} y_l \tag{6.1}$$

subject to

$$\sum_{l\in\mathcal{L}} x_{dl} = 1 \qquad d \in \mathcal{D}, \tag{6.2}$$

$$\sum_{d\in\mathcal{D}} (x_{dl} \cdot |d|) = |l| \qquad l \in \mathcal{L}, \tag{6.3}$$

$$y_l \geq x_{dl} \cdot \lceil d \rceil \qquad d \in \mathcal{D} \text{ and } l \in \mathcal{L}, \tag{6.4}$$

$$\sum_{d\in\mathcal{D}} (x_{dl} \cdot \tilde{d}) \leq C \qquad l \in \mathcal{L} \tag{6.5}$$

$$x_{dl} \in \{0,1\} \qquad d \in \mathcal{D} \text{ and } l \in \mathcal{L}, \tag{6.6}$$

$$y_l \geq 0 \qquad l \in \mathcal{L}. \tag{6.7}$$

The objective function (6.1) minimises the sum of the sizes of all maximal SKUs for each picking line. Constraint set (6.2) assigns each DBN to a single picking line while constraint set (6.3) ensures that the capacity of each picking line is not exceeded. The size of the maximal SKU for each line is determined by constraint set (6.4). The maximum volume for each picking lines is set with volume capacity constraint set (6.5).

Computational results by Matthews & Visagie [15] showed that Formulation (6.1)–(6.5) is not solvable in a reasonable time (within 10 minutes) for problem instances with more than three picking lines. Matthews & Visagie [15] reduced computational times by proposing a phased greedy heuristic (GP). Experimental results showed that using a phased greedy approach with two binding capacity constraints, the number of locations and volume, rarely yielded feasible solutions. A segmentation methodology (SEG_C) is therefore introduced. Picking lines are segmented into clusters and DBNs are first assigned to a cluster of picking lines. DBNs are later assigned to a specific picking line within the cluster using independent instances of Formulation (6.1)–(6.5). This process is illustrated in Figure 6.3. The following parameters are defined to model the assignment of DBNs into picking line clusters. Let

$\mathcal{I}$	be the set of all of clusters with elements i,
$\mathcal{L}_i$	be the set of picking lines in cluster i,
$\|\mathcal{L}_i\|$	be the total number of SKU locations available for cluster i,
$\lvert\mathcal{L}_i\rvert$	be the number of picking lines in cluster i.

The following variables are defined to model the assignment of DBNs to clusters. Let

$$\hat{x}_{di} = \begin{cases} 1 & \text{if DBN } d \text{ is assigned to cluster } i \\ 0 & \text{otherwise} \end{cases}$$

and

$\hat{y}_i$ as the size of the maximal SKU for cluster i.

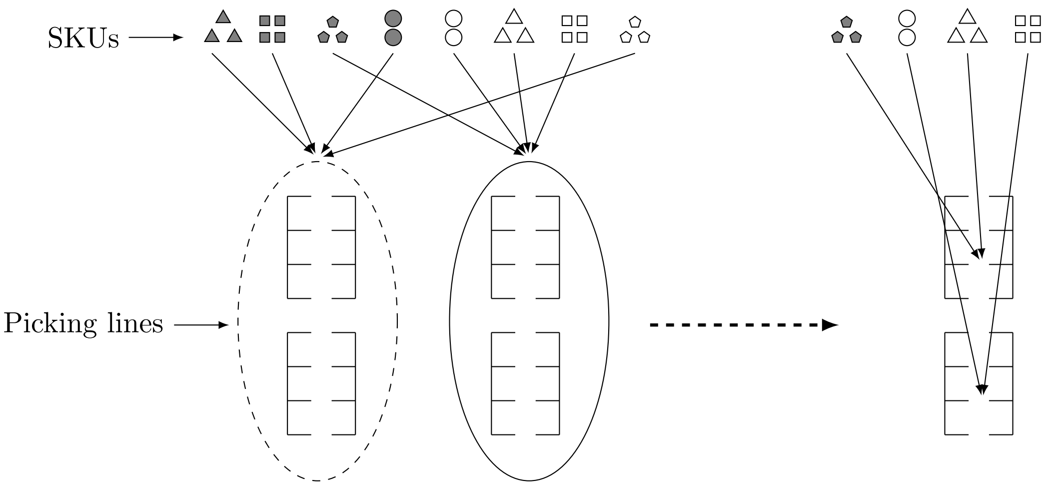


Figure 6.3: *A schematic representation of the segmentation of picking lines into cluster and the assignment of DBNs. Each shape represents a SKU and clusters of the same shape with the same shading represent DBNs.*

In terms of these symbols the objective is to

$$\text{minimise} \sum_{i \in \mathcal{I}} \hat{y}_i, \tag{6.8}$$

subject to

$$\sum_{i \in \mathcal{L}_i} \hat{x}_{di} = 1 \qquad d \in \mathcal{D}, \tag{6.9}$$

$$\sum_{d \in \mathcal{D}} (x_{di} \cdot |d|) = \|\mathcal{L}_i\| \qquad i \in \mathcal{I}, \tag{6.10}$$

$$\hat{y}_i \geq \hat{x}_{di} \cdot \lceil d \rceil \qquad d \in \mathcal{D} \text{ and } i \in \mathcal{I}, \tag{6.11}$$

$$\sum_{d \in \mathcal{D}} (\hat{x}_{di} \cdot \tilde{d}) \leq C \cdot |\mathcal{L}_i| \qquad i \in \mathcal{I}, \tag{6.12}$$

$$\hat{x}_{di} \in \{0, 1\} \qquad d \in \mathcal{D} \text{ and } i \in \mathcal{I}, \tag{6.13}$$

$$\hat{y}_i \geq 0 \qquad i \in \mathcal{I}. \tag{6.14}$$

The objective function (6.8) minimises the sum of the sizes of all maximal SKUs for each cluster. Constraint set (6.9) assigns each DBN to a single cluster while location capacity constraint set (6.10) ensures that the capacity of each cluster is not exceeded in terms of locations. The size of the maximal SKU for each cluster is determined by constraint set (6.11) and the volume per cluster is constrained with average volume capacity constraint (6.12).

Based on the computational times for Formulation (6.8)–(6.12) it is proposed that three clusters are used. Therefore a set of picking lines should be segmented into at most three clusters. Should a cluster contain more than three picking lines it would then undergo a further segmentation using Formulation (6.8)–(6.12) again until all clusters consist of no more than three picking lines. Note that using this segmentation approach does not guarantee a solution which will satisfy location capacity constraint set (6.3) for all of the clusters. Should this occur additional side constraints may be introduced which spread the small DBNs ($|d| = 1$) between the picking line clusters. However, for all problem instances a solution was found which satisfied constraint (6.3). In addition, the assignment of DBNs to clusters using Formulation (6.8)–(6.12) may force volume capacity constraint set (6.5) to be infeasible for a cluster. Should this occur the value of C is temporarily increased incrementally by a predefined value until a feasible solution is found. This approach was used as constraint set (6.12) is desirable but not binding from a practical implementation perspective and may thus be viewed as a soft constraint.

The primary goal of segmenting picking lines into clusters is to reduce the maximum volume assigned to picking lines. In some problem instances greedy insertion approaches naturally generate solutions which satisfy volume constraint set (6.5) which becomes non-binding due to the characteristics of the DBNs in those problem instances. It is therefore proposed to integrate a greedy insertion approach together with the segmentation methodology. Matthews & Visagie [16] introduced the Adjacency approach (ADT) which used correlations when inserting DBNs. A correlation measure is calculated for each pending DBN as the number of stores requiring the pending DBN and at least one already inserted DBN. The pending DBN with the highest measure or score is then inserted into the picking line. The use of correlations yielded similar results to the GP approach but with a significantly lower number of small cartons produced. Therefore the ADT insertion approach is integrated with the segmentation formulation. Figure 6.4 illustrates this hybrid assignment algorithm (HAS_C) where the correlations approach (ADT) introduced by Matthews & Visagie [16] is used. Initially a single iteration of the ADT heuristic is run and if the solution satisfies the volume capacity constraints that solution is used. Should the volume capacity constraints be violated the segmentation formulation is used and the set of picking lines are segmented into clusters of two or three picking lines. The DBNs in each of these clusters are then assigned independently by using the ADT approach. If the solution violates the volume capacity constraints an IP formulation is used to assign DBNs. In some cases after solving the IP formulation the capacity constraints may still be infeasible in which case the capacity threshold is increased in a similar fashion to the segmentation phase.

6.4 Results

To test the segmentation approaches the real life historical problem instances introduced by Matthews & Visagie [15] were used and are available online [13]. The data consists of seven scenarios described in Table 6.1 each of which has a different number of picking lines per problem instance which aids in the comparison of solution approaches. All testing was performed on an Intel i7 2GHz processor with eight GB ram running the Windows 7 operating system. All mathematical formulations were solved with CPLEX 12.3 and coded in AIMMS 3.12 [7, 17].

Number of lines per day	Number of problem instances	Number of DBNs
2	61	2592
3	53	3437
4	49	4146
5	38	4109
6	32	4161
7	22	3177
8	14	2148

Table 6.1: *The composition of the historical data scenarios [13].*

A comparison in terms of cycles traversed between the HAS_{300} approach the historical case, the ADT approach and the GP approach is summarised in Table 6.2. For comparison purposes a pure segmentation approach (SEG_∞) using both IP formulations without volume capacity constraint sets (6.5) and (6.12) is also tested and the results presented.

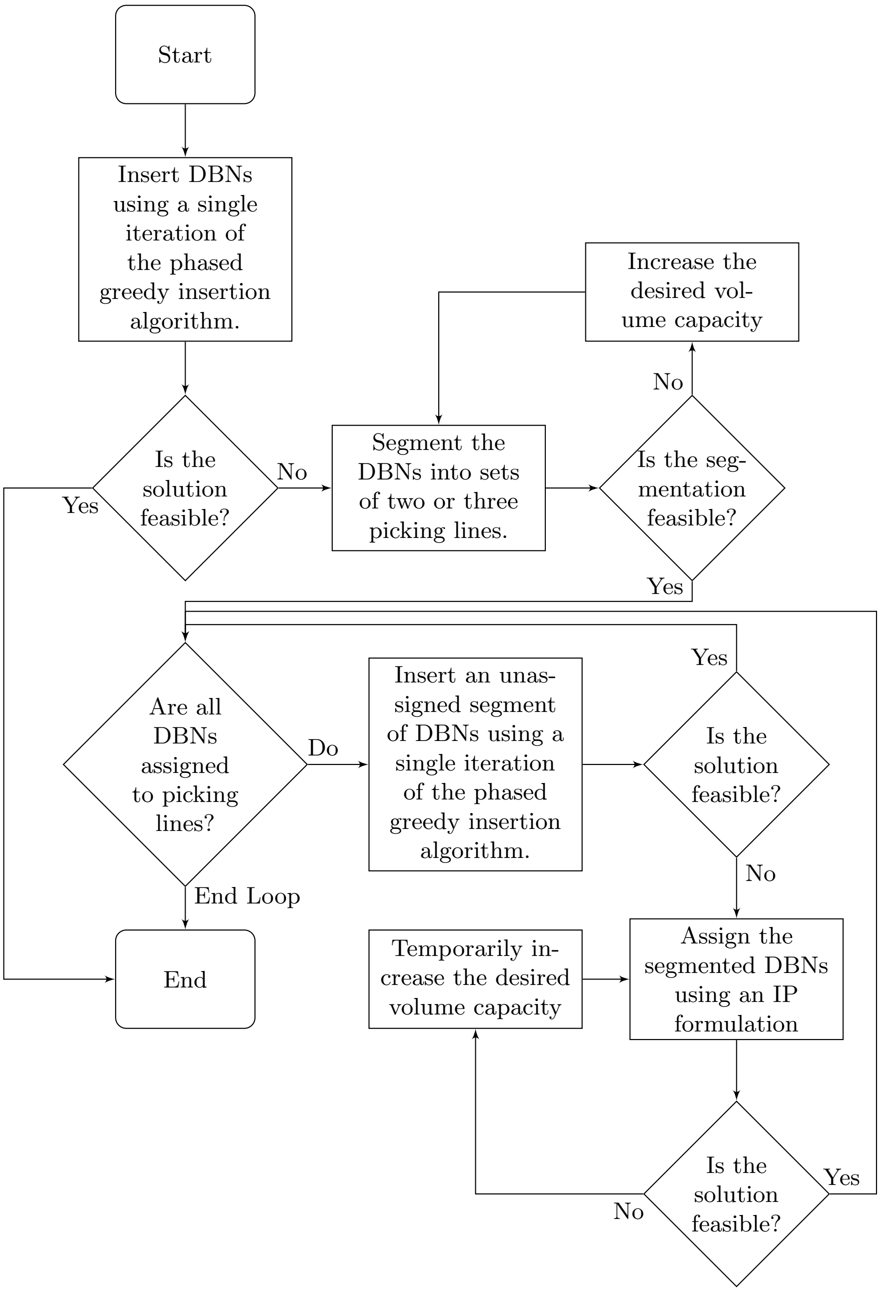


Figure 6.4: *A flow chart representation of the hybrid assignment approach.*

Scenario	His	GP	SEG_∞	SEG_{300}	ADT	HAS_{300}
2	7168	6019	6046	6273	6193	6387
3	9289	7515	7551	7917	7710	8080
4	11532	9015	9090	9693	9282	9847
5	11259	8618	8653	9094	8922	9248
6	10706	8129	8154	8620	8363	8758
7	8412	6366	6386	6656	6540	6749
8	6011	4715	4729	4885	4847	4975

Table 6.2: *A summary of the total walking distance (in kilometres) between the historical assignments (His), a segmentation approach with no volume capacity (SEG_∞), a segmentation and hybrid approach with a capacity of 300 m^3 (SEG_{300}, HAS_{300}), and the greedy phased (GP) and adjacency (ADT) approaches.*

The SEG_∞ approach is marginally outperformed by the GP approach in terms of total walking distance which is the best known approach to minimise the number of cycles traversed. Although the HAS_{300} approach has additional constraints it still outperforms the historical case. The ADT approach outperforms the HAS_{300}. The difference in solution quality only occurs for problem instances where the ADT generates a solution which violates the volume capacity constraints. In these cases the HAS_{300} approach solves a more constrained problem and therefore the objective function value can at best be the same.

For further clarity on the effect of the volume capacity constraints on the number of cycles traversed, all problem instances where the ADT generates an initial solution which violates the volume capacity constraints were considered separately. The summary of these results are given in Table 6.3. The greatest impact of capacity constraints occurs for scenarios with four or less picking lines where the walking distance is increased by 15% compared to the SEG_∞ approach. However, the total walking distance using the HAS_{300} approach is still 14% less than the historical assignments.

(Scenario, Number of problem instances)	His	GP	SEG_∞	SEG_{300}	ADT	HAS_{300}
(2, 61)	1955	1542	1551	1772	1574	1768
(3, 53)	2968	2184	2199	2551	2216	2586
(4, 49)	5351	3970	4013	4590	4041	4606
(5, 38)	5470	3920	3939	4376	4041	4367
(6, 32)	6298	4497	4510	4967	4578	4973
(7, 22)	5057	3565	3576	3845	3626	3834
(8, 14)	2958	2124	2129	2286	2151	2278

Table 6.3: *A summary of the total walking distance (in kilometres) between the historical assignments (His), a segmentation approach with no volume capacity (SEG_∞), a segmentation and hybrid approach with a capacity of 300 m^3 (SEG_{300}, HAS_{300}), and the greedy phased (GP) and adjacency (ADT) approaches for problem instances where the ADT does not find a solution which satisfies all volume capacity constraints.*

The computation times for all of the algorithms are summarised in Table 6.4. The best performing algorithm is the GP. The segmentation approaches show consistent computation times and are all solvable within one minute.

A summary of the effects on the number of small cartons produced is given in Table 6.5. The HAS_{300} approach has a smaller number of small cartons produced than the GP approach although the ADT approach still has the best performance for most scenarios. Note that for the scenario with two picking lines per day the inclusion of volume capacity constraints improves the number of small cartons produced. This is attributed to a smaller number of lines to which excess volume must be distributed and therefore a smaller number of picking lines with small volumes of stock assigned to it.

A graphical representation of the spread of volume over all the picking lines for scenarios with four and five picking lines respectively is given in Figure 6.5. The median (50^{th} percentile) Q1 (25^{th} percentile) and Q3 (75^{th} percentile) are shown as the horizontal lines in the box plot. Individually plotted points are outliers that lie outside 1.5 times the inter quartile range (Q3-Q1) from either Q1 or Q3. The exposed horizontal lines indicate the minimum and maximum volumes for non-outlier picking lines. The inclusion of volume capacity constraints thus reduced the number and size of outlying waves in terms of volume. When using the HAS_{300} approach there are waves which have in excess of 300 m^3 of stock. This may be attributed to problem instances where a segmented set of DBNs cannot be assigned to individual picking lines without violating the volume capacity constraints with a desired threshold of 300 m^3. The threshold

Algorithm	Picking lines per problem instance	μ	σ	Max	Q_1	Median	Q_3	Min
GP	2	0.01	0.00	0.02	0.01	0.01	0.01	0.00
	3	0.01	0.01	0.03	0.02	0.01	0.01	0.00
	4	0.02	0.01	0.04	0.03	0.02	0.02	0.01
	5	0.07	0.23	1.43	0.04	0.03	0.02	0.02
	6	0.09	0.30	1.75	0.05	0.03	0.03	0.02
	7	0.21	0.52	1.93	0.05	0.05	0.03	0.02
	8	0.39	0.87	2.53	0.06	0.04	0.03	0.02
ADT	2	1.86	0.60	3.07	2.35	1.91	1.41	0.65
	3	3.91	1.34	7.11	4.86	3.89	2.96	1.46
	4	3.48	1.31	6.32	4.27	3.28	2.56	1.29
	5	8.43	2.46	15.04	10.13	7.72	6.70	4.09
	6	19.31	50.63	296.30	13.02	9.97	8.29	5.99
	7	40.13	94.95	357.99	12.01	11.14	9.44	6.22
	8	87.58	206.04	734.06	14.36	12.81	11.45	7.08
SEG_∞	2	0.03	0.04	0.28	0.03	0.02	0.02	0.00
	3	0.42	0.41	1.72	0.75	0.19	0.14	0.09
	4	0.10	0.03	0.22	0.11	0.09	0.08	0.05
	5	1.01	0.65	2.78	1.42	1.07	0.34	0.16
	6	1.72	1.28	6.68	2.11	1.53	0.97	0.31
	7	3.68	3.16	13.67	4.24	3.38	1.37	0.41
	8	6.07	6.91	27.19	6.46	4.03	1.54	0.48
SEG_{300}	2	0.03	0.01	0.05	0.03	0.03	0.02	0.00
	3	0.47	0.44	1.98	0.78	0.23	0.16	0.09
	4	0.11	0.03	0.22	0.12	0.11	0.09	0.08
	5	0.97	0.62	2.70	1.42	0.97	0.30	0.11
	6	2.05	2.53	14.17	2.20	1.60	0.68	0.25
	7	6.06	8.92	32.81	4.24	3.27	1.64	0.30
	8	60.54	208.60	785.22	6.99	4.98	2.48	0.52
HAS_{300}	2	0.77	0.37	1.65	1.05	0.72	0.52	0.22
	3	2.71	1.85	8.50	3.09	2.28	1.42	0.56
	4	4.07	1.96	10.19	4.93	3.72	2.65	1.15
	5	6.71	3.52	15.15	9.29	5.58	3.96	2.63
	6	10.36	6.30	37.66	11.84	8.62	7.19	2.98
	7	13.19	5.72	29.46	15.78	11.54	9.58	5.73
	8	19.96	19.13	76.13	19.07	12.94	10.44	6.15

Table 6.4: *A comparison of computational times in seconds between a segmentation approach with no volume capacity (SEG_∞), a segmentation and hybrid approach with a capacity of 300 m^3 (SEG_{300}, HAS_{300}), and the greedy phased (GP) and adjacency (ADT) approaches. Standard measures of spread are presented including the 25^{th} percentile (Q_1) and the 75^{th} percentile (Q_3).*

Scenario	His	GP	SEG_∞	SEG_{300}	ADT	HAS_{300}
2	0.05%	0.14%	0.14%	0.11%	0.09%	0.07%
3	0.04%	0.13%	0.12%	0.11%	0.08%	0.08%
4	0.03%	0.12%	0.12%	0.1%	0.07%	0.06%
5	0.03%	0.13%	0.13%	0.11%	0.06%	0.07%
6	0.04%	0.12%	0.12%	0.1%	0.06%	0.06%
7	0.03%	0.13%	0.12%	0.1%	0.06%	0.07%
8	0.03%	0.13%	0.12%	0.11%	0.06%	0.07%

Table 6.5: *A comparison of the number of small cartons produced between a segmentation approach with no volume capacity (SEG_∞), a segmentation and hybrid approach with a capacity of 300 m^3 (SEG_∞, SEG_{300}), the historical assignments (His) and the GP and ADT approaches. The results are presented in terms of the percentage of total volume of stock attributed to orders with less than 0.006 m^3 of stock.*

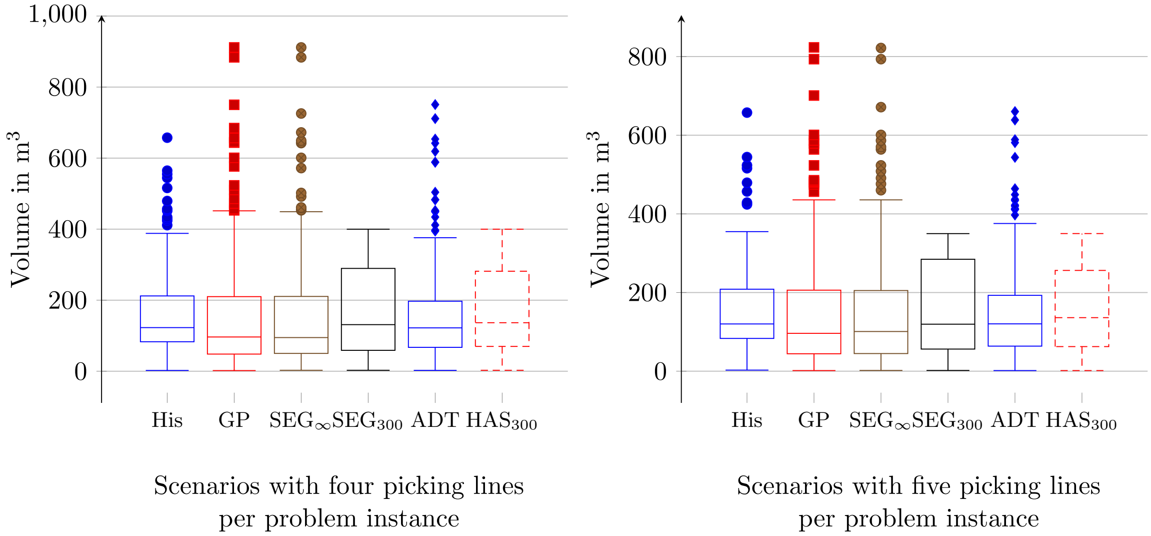


Figure 6.5: *A graphical box-plot representation of the distribution of the total volume for each picking line after scheduling for scenarios with four and five picking lines per problem instance. The median (50^{th} percentile), Q1 (25^{th} percentile) and Q3 (75^{th} percentile) are represented as the horizontal lines in the closed box. The individually plotted coordinates are associated with the outliers which are either 1.5 times the inter quartile range (IQR) smaller than Q1 or larger than Q3. The whisker lines indicate the minimum and maximum number of cycles for non-outliers.*

is therefore marginally increased until a feasible solution is found. The distribution of volume shows similar patterns for the other scenarios which are not presented here.

Problem instances were classified or grouped according to their average volume per picking line to illustrate the spread of volume between different problem instances (days) within each scenario. Figure 6.6 illustrates the number of problem instances within each of these volume groups. The results suggest that there is scope to reduce the number of days which have a high volume of scheduled DBNs. By changing the strategy in which DBNs are scheduled across different days in a planning horizon the effects of the capacity constraints could be alleviated.

6.5 Conclusion

A DC with an order picking system which uses a unique forward picking area comprising of unidirectional picking lines was investigated. The DC has 12 picking lines on which all piece picking is performed. During the planning phase SKUs are grouped together into DBNs if they are of the same product but have different sizes. All the store requirements for a DBN are then established for a specific planning period. DBNs are assigned to picking lines where all the store requirements are picked in a single operation or wave. Three goals were considered when assigning DBNs to waves. In literature the third issue of a large number of pallet movements required to build picking lines was not addressed. Managing the number of pallet movements required to build picking lines while minimising walking distance and reducing number of small cartons produced is therefore addressed in this paper.

Due to lack of captured data the total volume of stock sent to a picking line, instead of the actual number of pallet movements, was used as a measure of the number of pallet movements. An MIP formulation was introduced which assigned DBNs to picking lines while minimising an estimator for the number of cycles traversed (maximal SKU) and constraining the total volume assigned

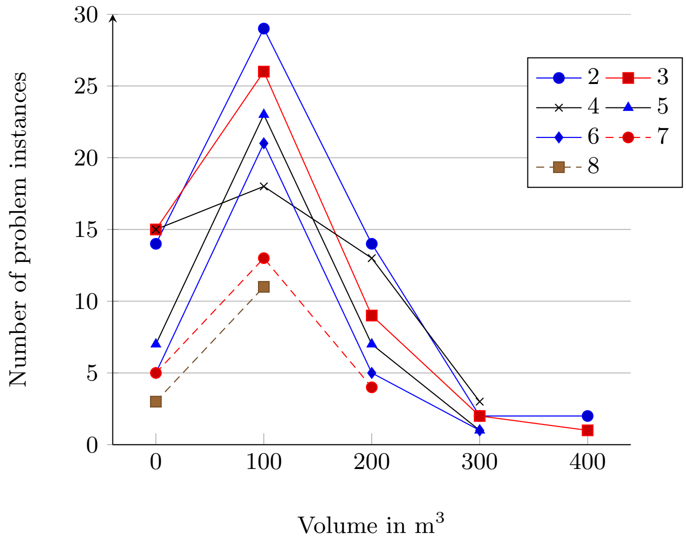


Figure 6.6: *The number of problem instances in each scenario for each average volume category.*

to the picking line. This MIP was not solvable in a reasonable time and thus a segmentation approach was introduced which sequentially segmented picking lines and DBNs into clusters. A hybrid assignment approach was developed which incorporated both an insertion approach using correlations and the MIP formulations for segmentation to reduce the number of small cartons produced.

The HAS$_{300}$ achieved walking distance on average 15% better than historical assignments over the historical problem instance set and achieved a level of small cartons produced similar to the ADT approach. The use of volume capacity constraints improved the distribution of volume over picking lines such that the volume assigned to picking lines were all within a given feasible threshold. Following the results presented here it is proposed that PEP uses the HAS$_{300}$ approach to assign DBNs to picking lines.

Although volume was used as a measure for the number of pallet movements required to build a picking line both IP formulations – segmentation and normal – can easily be adapted to incorporate pallet movements or the total pallet movement cost/time. It is therefore suggested that this data be captured and included in the WMS.

The distribution of volume between problem instances in each scenario is not uniform. This result suggests that further improvements may be made when scheduling DBNs into different days or shifts. A natural progression of these models is to consider the scheduling of DBNs across days taking into account building costs, picking costs and the out-of-DC dates of DBNs.

Bibliography

[1] ACCORSI R, MANZINI R & BORTOLINI M, 2012, *A hierarchical procedure for storage allocation and assignment within an order-picking system. A case study*, International Journal of Logistics: Research and Applications, **15(6)**, pp. 351–364.

[2] BARTHOLDI JJ & HACKMAN ST, 2011, *Warehouse and Distribution Science*, The Supply Chain and Logistics Institute, School of Industrial and Systems Engineer-

ing Georgia Institute of Technology Atlanta, GA 30332-0205 USA, Available from www.warehouse-science.com.

[3] de Koster R, Le-Duc T & Roodbergen KJ, 2007, *Design and control of warehouse order picking: A literature review*, European Journal of Operational Research, **182(2)**, pp. 481–501.

[4] Emerson CR & Schmatz DS, 1981, *Results of modeling an automated warehouse system*, Industrial Engineering, **13(8)**, pp. 28–32.

[5] Garfinkel M, 2005, *Minimizing Multi-zone Orders in the Correlated Storage Assignment Problem*, PhD, Georgia Institute of Technology, Georgia, (AT).

[6] Hassini E, 2009, *One-Dimensional Carousel Storage Problems: Applications, Review and Generalizations*, INFOR, **47(2)**, pp. 81–92.

[7] IBM, *CPLEX*, Available from www.ibm.com.

[8] Jane CC & Laih YW, 2005, *A clustering algorithm for item assignment in a synchronized zone order picking system*, European Journal of Operational Research, **166(2)**, pp. 489–496.

[9] Kim BS & Smith JS, 2012, *Slotting methodology using correlated improvement for a zone-based carton picking distribution system*, Computers & Industrial Engineering, **62(1)**, pp. 286–295.

[10] Litvak N & Maia V, 2010, *A survey on performance analysis of warehouse carousel systems*, Statistica Neerlandica, **64(4)**, pp. 401–447.

[11] Manzini R, 2006, *Correlated Storage Assignment in an Order Picking System*, International Journal of Industrial Engineering, **13(4)**, pp. 384–394.

[12] Matthews J & Visagie SE, 2013, *Order sequencing on a unidirectional cyclical picking line*, European Journal of Operational Research, **231(1)**, pp. 79–87.

[13] Matthews J & Visagie SE, *Picking Line Data*, Available from http://scholar.sun.ac.za/handle/10019.1/86110.

[14] Matthews J & Visagie SE, 2014, *SKU arrangement on a unidirectional picking line*, Submitted.

[15] Matthews J & Visagie SE, 2015, *SKU assignment to parallel unidirectional picking lines*, The South African Journal of Industrial Engineering, **26(1)**, pp. 235–251.

[16] Matthews J & Visagie SE, 2016, *SKU assignment to unidirectional picking lines using correlations*, ORiON, to appear.

[17] Paragon Decision Technology, *AIMMS*, Available from www.aimms.com.

[18] Parikh PJ & Meller RD, 2008, *Selecting between batch and zone order picking strategies in a distribution center*, Transportation Research Part E, **44(5)**, pp. 696–719.

[19] Pep Stores Ltd, *PEP*, Available from www.pepstores.com.

[20] Vickson RG & Fujimoto A, 1996, *Optimal storage locations in a carousel storage and retrieval system*, Location Science, **4(4)**, pp. 273–245.

[21] Vickson RG & Lu X, 1998, *Optimal product and server locations in one-dimensional storage racks*, European Journal of Operational Research, **105(1)**, pp. 18–28.

CHAPTER 7

Implementation

In Chapter 3 several approaches to the SLP were investigated. Solution approaches were compared using total walking distance only. Two mathematical formulations were introduced to find a lower bound to the SLP in terms of walking distance. These formulations built on the maximal cut framework introduced by Matthews & Visagie [2] for the OSP. It was shown that the problem is too computationally intensive to be solved exactly and several heuristic approaches were therefore investigated.

Included in the set of heuristic approaches were two approaches known to be optimal for some carousel systems, namely the greedy approach (GS) and organ pipe approach (OPA). Furthermore an adjacencies approach (GA) was tested as well as the classroom discipline heuristic (CD) introduced by Hagspihl & Visagie [1]. All of the heuristic approaches performed approximately equally well and collectively outperformed the historical arrangements by on average 2% and up to 15% in terms of walking distance.

When comparing the initial improvements made by only solving the OSP (on average 20%) to the solutions obtained by solving the SLP as well (an additional 2%) the marginal impact of solving the SLP is small. Moreover, a set of random solutions was also generated for each test instance. It was shown that the range of objective function values for this set of random solutions was also small. Any approach to the SLP which has a manageable congestion level can therefore be used in a decision support system (DSS). It is suggested that al of the heuristics be run within the DSS and the best solution presented to the manager as the computational time for each heuristic is negligible. Managers could then change the arrangement if they so wish.

Following the study of the SLP the PLAP was investigated. An exact solution approach is not solvable due to the compounding complexity of sequentially solving the PLAP, SLP and OSP. The maximal cut formulation can only be used to evaluate the walking distance of a final solution to the PLAP. The formulation is too computationally intensive to evaluate all candidate solutions for a PLAP solution methodology. The walking distances of candidate solutions to the PLAP were therefore compared using the maximal SKU for each picking line as an estimator of the walking distance. The exact walking distance was only calculated for the final solutions obtained by any PLAP solution approach.

An initial approach to the PLAP which minimised the size of the maximal SKU within each picking line is introduced in Chapter 4. A mathematical formulation with this objective is introduced which is not solvable for real life data instances. A phased greedy insertion approach (GP) was therefore introduced which generated feasible solutions for all test instances. It was shown that minimising the maximal SKU reduced the total walking distances of pickers by on average 22%. Trade-offs, however, exist between the three goals discussed in Section 1.4, namely walking distance, picking line size and carton utilisation. Minimising the maximal SKU generated a large number of large undesirable picking lines and increased the number of small cartons.

Correlation measures between DBNs were used to assign DBNs to picking lines to reduce the number of small cartons. Using the phased greedy insertion methodology four different correlation measures were introduced in Chapter 5. It was shown that the correlation approaches are marginally (on average 2%) out performed by the GP approach in terms of walking distances. However, the number of small orders indicating small cartons was halved and the size of the large picking lines was marginally reduced.

Although the number of small cartons is at a satisfactory level using correlation measures the number and size of large picking lines was still undesirable. A mathematical formulation using a segmentation framework was therefore introduced in Chapter 6 in an effort to introduce volume capacity constraints. Using this segmentation approach all data instances were solved and feasible solutions obtained no more than 5% of the picking lines exceeded the volume threshold of 300 m^3 for any one scenario. The walking distance of pickers was on average 15% less than the historical assignments. Carton utilisation, however, was similar to the GP approached. A hybrid approach (HAS_{300}) incorporating correlation measures as well as a segmentation approach was therefore introduced. This approach showed the most favourable trade-off between all of the PLAP objectives and is selected for use in the DC.

All solution approaches to the PLAP were compared using test instances discussed in Chapter 2. Although these test instances are adequate to compare different solution approaches there are a number of implementation practicalities not represented in the test data which should be considered before implementing the HAS_{300} approach in a DSS. These implementation practicalities are further discussed in Section 7.1. A proof of concept interface is introduced and is discussed in Section 7.2. The interface addresses these implementation considerations and provides a framework for future WMS developments.

7.1 Implementation considerations

It was assumed that all SKUs were assigned a single location in each picking line when generating test instances for the PLAP. Although this assumption was shown to have only a small impact on the test results, future implementations of the HAS_{300} approach should handle cases where multiple locations are assigned to SKUs for additional stock. DBNs typically consist of multiple SKUs and thus typically require multiple locations in a picking line. The HAS_{300} (and all other PLAP approaches) therefore assign DBNs of various sizes ($|d|$) to picking lines and can accommodate SKUs requiring multiple locations by increasing the DBN size.

Following the analysis comparing available locations with SKU volume in Section 2.2, it was shown that an automatic procedure for assigning multiple locations to SKUs based on volume is not possible with the current data. The actual number of locations assigned to a SKU must therefore be assigned manually by a manager. Provision must therefore be made in a DSS for manual adjustments of this parameter (by line managers) before the HAS_{300} assigns DBNs to picking lines.

The HAS_{300} uses volume capacity constraints to reduce the number of picking lines requiring large numbers of pallet movements. Future developments in the current WMS may allow for the tracking of pallets in the DC. Provision should therefore be made to use this more accurate data in the future. The mathematical structure of the volume capacity constraints in the

HAS_{300} approach can be changed to include total pallet movements by adjusting the input parameters. The actual distance from different storage locations (floor and rack storage) to the physical picking lines can also be used to estimate the required build time for each picking line. Estimated building time can therefore be used as a measure of the picking line size. Applying a capacity constraint on the total build time may also improve general stock movement efficiency by reducing total pallet movement distances.

A further adjustment may be made to the HAS_{300} formulation by adding additional capacity constraints. For example, the total number of picks in a picking line may be more evenly distributed by applying a capacity on each picking line. This may improve picker moral as work is more evenly balanced between picking lines. Additional analysis and testing should, however, be completed on the performance of such formulations before implementation.

Before arriving at a final set of test instances for the PLAP a number of exclusions were made on the extracted data. Picking lines which were set up on empty floor space were excluded to reduce the effects of management bias. These picking lines consisted of very few SKUs which often required special attention – such as SKUs for new stores. Provision must therefore be made in a DSS for managers to manually set up picking lines on empty floor space.

Managers currently schedule and assign DBNs to picking lines using a manual paper based system although a more automated WMS is in the process of design and implementation. DBN pick instructions are printed and managers organise these instructions into piles for each wave. There is little to no data visibility for the mangers using this system and the effects of different assignment decisions cannot be quickly calculated and evaluated. Moving from a manual to a computerised DSS requires change management. The interfacing between managers and the DSS should aid in the integration of management knowledge and expertise, data visibility and algorithmic speed and accuracy. An easy to use interface offering flexibility and accuracy is thus required.

A proof of concept DSS interface is proposed in this section to address the implementation issues discussed. This interface is coded in JAVA, an object orientated programming language often used for user interface and web design [3]. The design, use and future improvements to this interface are discussed in more detail in the following section.

7.2 User interface

A proof of concept interface for an order picking DSS was designed to illustrate and provide a framework for the implementation of the HAS_{300} approach. The main page of the interface is illustrated in Figure 7.1 and consists of a picking line data panel on the left, a DBN data panel in the middle and a controls panel on the right. This page is the first point of interaction between the user and the DSS.

In the "controls" panel there are four buttons shown in Figure 7.2. The "Get SKUs" button retrieves new DBN pick instructions from the planning department. The "Add picking line" button adds a picking lines to the pending list of picking lines shown in the picking line data panel. The "Auto Assign" button implements the HAS_{300} approach. Finally the "Update" button calculates the number of cycles required to pick the pending picking lines and adjusts the metadata for each pending picking line.

Once the DBN pick instructions have been obtained from planning DBN and SKU level data is visibly provided to the user to aid in decision making. The urgency of DBNs is visualised

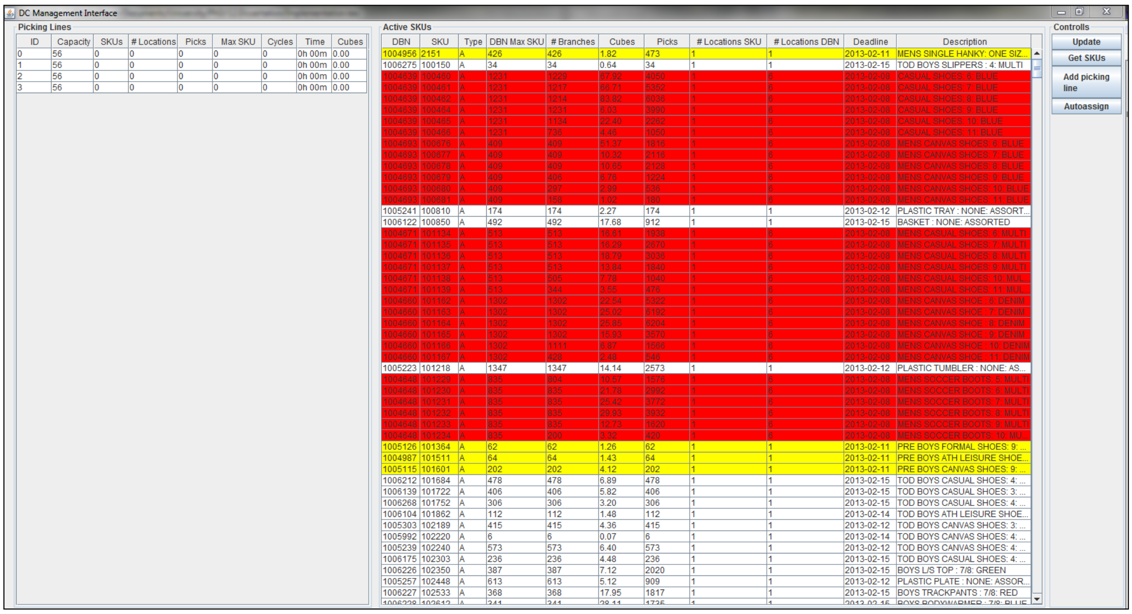


Figure 7.1: *A screen shot of the main page for the proof of concept interface for the order picking DSS.*

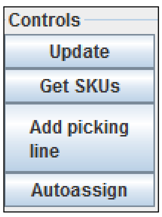
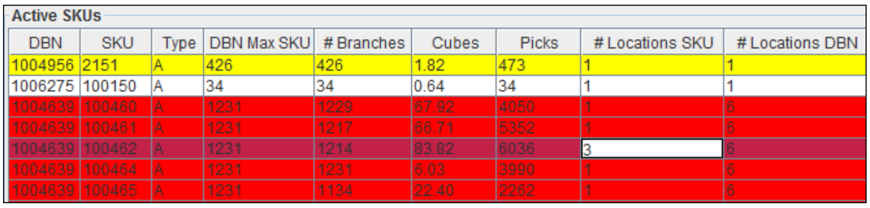


Figure 7.2: *A screen shot of the controls panel in the main page of the proof of concept interface for the order picking DSS.*

using different colours depending on how soon a DBN needs to be processed. DBNs that are due within the next two days are highlighted red and DBNs highlighted yellow are due within the next four days. Additional data fields are also provided and include the

- DBN number,
- SKU number,
- pick type,
- DBN max SKU – The size of the maximal SKU associated with this DBN,
- # stores – The number of stores requiring this SKU,
- # locations SKU – The number of locations required by this SKU,.
- # locations DBN – The number of locations collectively required by the SKUs in the DBN,
- deadline and
- description.

The SKUs may be sorted according to anyone of these data fields. This gives the user additional data visibility and improves the user experience. The number of locations per SKU field can be adjusted by the user and is illustrated in Figure 7.3. This functionality addresses the need for managers to manually adjust the number of assigned locations to SKUs with large volumes of stock.


Active SKUs

DBN	SKU	Type	DBN Max SKU	# Branches	Cubes	Picks	# Locations SKU	# Locations DBN
1004956	2151	A	426	426	1.82	473	1	1
1006275	100150	A	34	34	0.64	34	1	1
1004639	100460	A	1231	1229	67.92	4050	1	6
1004639	100461	A	1231	1217	66.71	5352	1	6
1004639	100462	A	1231	1214	83.82	6036	3	6
1004639	100464	A	1231	1231	6.03	3990	1	6
1004639	100465	A	1231	1134	22.40	2262	1	6

Figure 7.3: *A screen shot of the proof of concept interface for the order picking DSS illustrating how the user can change the number of locations assigned to a SKU.*

Picking lines are added to the pending list using the "add picking" line button when the user plans waves of picking. Empty picking lines are added with a default capacity of 56 locations. This capacity can be changed by the user as needed. DBNs can be added to these pending picking lines using the auto-assign button or by manually assigning DBNs. The auto-assign button will automatically scheduled the top n DBNs according to out-of-DC dates which fit into the pending picking lines. These DBNs are then assigned to individual picking lines using the HAS_{300} approach to the PLAP.

DBNs are manually assigned to picking lines by selecting their entries and right clicking on the selection. A popup menu will appear providing an option to move the DBNs to a pending picking line. This process is shown in Figure 7.4. When a SKU is moved to a picking line all the SKUs within the same DBN will also be moved to that same picking line. This ensures that DBNs are picked in their entirety in a single wave. This functionality also allows for managers to manually build special case picking lines on the floor of the DC.

Once DBNs have been assigned to picking lines the walking distance and an estimate for the total picking time for a team of eight pickers[1] can be calculated using the "update" button in

[1]The DSS may easily be modified to allow for a user to assign a custom number of pickers to each picking line.

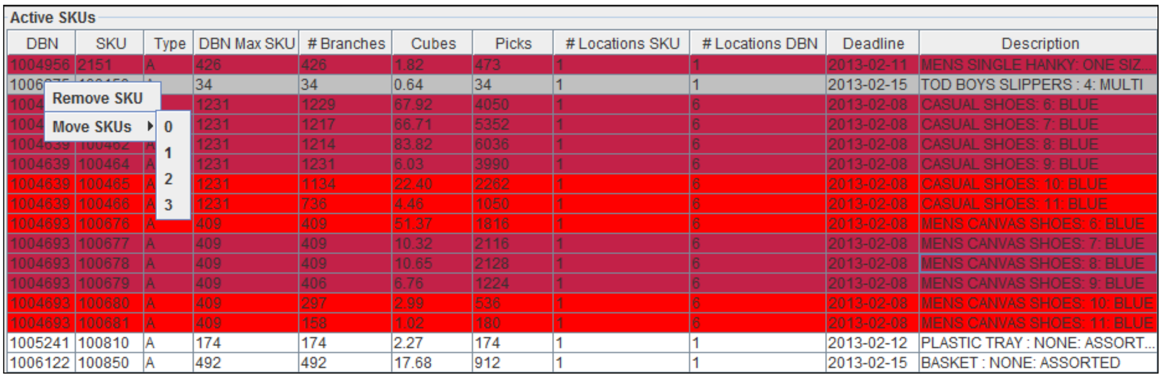


Figure 7.4: *A screen shot the of the popup menu in the proof of concept interface for the order picking DSS illustrating how to manually assign DBNs to picking lines*

the control panel. Picking lines can also be updated individually by right clicking on the picking line row. This drop down menu is shown in Figure 7.5. A pending picking line can be removed, exported for processing, updated, filled automatically or viewed.

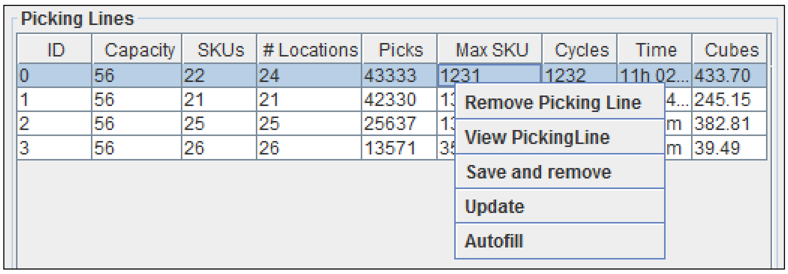


Figure 7.5: *A screen shot of the popup menu in the main page in the proof of concept interface for the order picking DSS illustrating the order picking DSS picking line options.*

The "view picking line" button will open up a picking line specific window shown in Figure 7.6. This window consists of a picking line data panel in the top left corner, a SKU data panel on the right and an actions panel in the bottom left corner. The picking line can be updated or exported for building using the respective button in the actions panel.

The actual locations for the different SKUs can be set in the picking line window by moving SKUs up and down the list. DBNs can also be removed from the picking or sent to another picking line. This is achieved by selecting a set of SKU entries and right clicking on them and selecting the appropriate option from the popup menu shown in Figure 7.7.

Using this proposed DSS interface the HAS_{300} can be integrated into the current WMS. Managers can easily adjust parameters and maintain manual control of DC decisions with improved data visualisation and performance evaluations. Further functionality can be integrated into the DSS as more data are recorded and becomes available.

Bibliography

[1] HAGSPIHL R & VISAGIE S, 2014, *The number of pickers and stock-keeping unit arrangement on a uni-directional picking line*, The South African Journal of Industrial Engineering, **25(3)**, pp. 169–183.

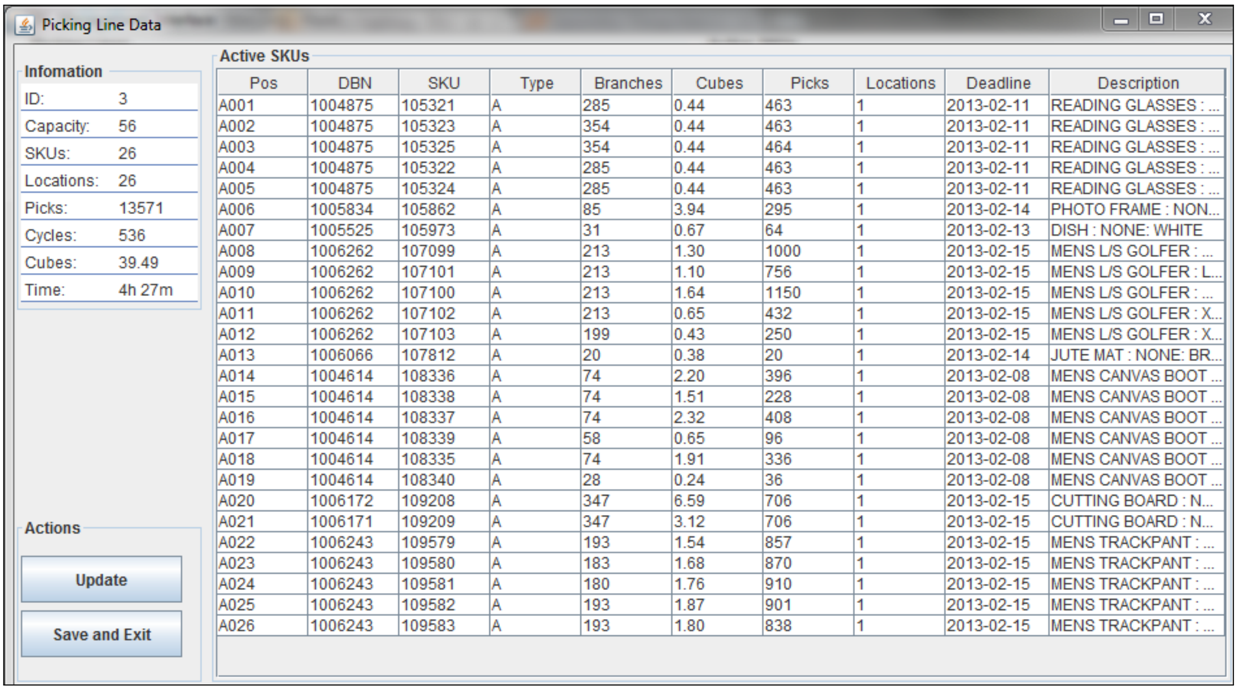


Figure 7.6: *A screen shot of the picking line window in the proof of concept interface for the order picking DSS.*

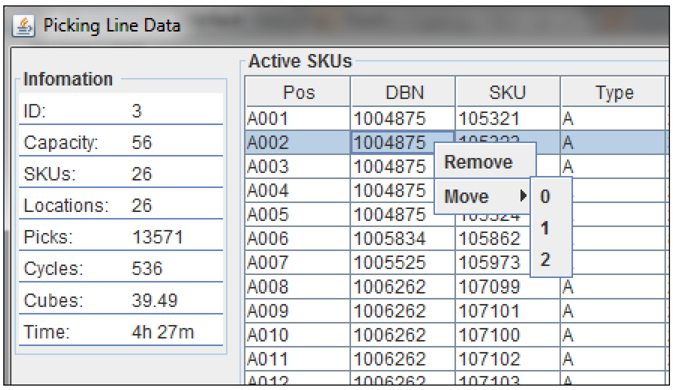


Figure 7.7: *A screen shot of the picking line window in the proof of concept interface for the order picking DSS illustrating the SKU options.*

[2] MATTHEWS J & VISAGIE SE, 2013, *Order sequencing on a unidirectional cyclical picking line*, European Journal of Operational Research, **231(1)**, pp. 79–87.

[3] ORACLE, *JAVA*, Available from `http://www.oracle.com/`.

CHAPTER 8

Conclusion

A brief summary of this dissertation is provided in this chapter highlighting the major contributions of the work. In addition final conclusions and remarks are given including recommendations for PEP. Finally areas of future work which follow on this study are discussed.

8.1 Dissertation summary

A general overview of supply chains and warehousing was provided in Chapter 1. An order picking system used in a DC owned by PEP Stores Ltd (PEP) was identified. The DC layout and operations were further discussed in detail, with additional focus given to the order picking system and management processes associated with it. The order picking system investigated here makes use of a unique configuration of unidirectional picking lines operating under wave picking which creates a tiered decision level environment. Three decision tiers were identified within this system which defined three sub problems, namely the Order Sequencing Problem (OSP), SKU Location Problem (SLP)and Picking Line Allocation Problem (PLAP). The OSP focused on sequencing orders for multiple pickers while minimising the total walking distance of pickers. This subproblem was solved in previous studies and the focus of the dissertation turned to the SLP and PLAP. The SLP addressed the problem of arranging SKUs on a picking line or assigning individual SKUs to different locations while minimising the total walking distance of pickers. The PLAP has a broader scope and addressed the problem of assigning DBNs (or groups of similar SKUs) to different picking lines to achieve good trade-offs between three objectives namely, total walking distance, the number of small cartons produced and operational risk due to the number of waves requiring large volumes of stock to populate picking lines.

In Chapter 2 test problem instances were derived from historical data sources. Both the OSP and SLP problem definitions are localised to a single wave on a single picking line and historical problem instances which were used for OSP studies were therefore also used in conjunction with generated problem instances to test SLP solution approaches. A further set of real life data was extracted from many data sources at PEP. This data consisted of a representative set of completed waves of picking covering a connected time period in the DC. Test scenarios were generated from the data historical extract to test the PLAP. A test framework was developed to test the SLP and PLAP. In this framework provision was made for features in future studies such as DBN scheduling problems.

Solution approaches to the SLP were introduced and tested in Chapter 3. Two mathematical formulations using the results from the OSP studies were introduced in an attempt to produce a lower bound. These approaches were, however, not solvable for real life problem instances. Four heuristic approaches were tested against the historical assignments as well as a set of random solutions. It was shown that if the OSP is solved there is a marginal gain of on average 2% when solving the SLP. Furthermore, it was shown that any heuristic could be used to evaluate the walking distance of a solution to the PLAP. It was proposed that all of the heuristics be run in the DSS and the best solution presented to the managers. The managers could then change the arrangement if they so wished which gave them ultimate control of this decision tier.

An initial mathematical formulation was proposed to solve the PLAP in Chapter 4. The size of the maximal SKU in each picking line is used as an estimator for the total walking distance. DBNs were assigned to picking lines while minimising the sum of the sizes of the maximal SKUs in each picking line. Once DBNs had been assigned to picking lines the SLP was solved for each picking line using a greedy approach before the total walking distance was determined by solving the resulting OSPs. This mathematical formulation was not solvable for large problem instances and a phased greedy insertion approach (GP) was introduced. The total walking distance of pickers reduced by 22% compared to the historical assignments after solving the SLP and OSP. However, the number of small cartons increased and the number of large picking lines (in terms of total volume) increased to an undesirable level creating operational risk.

In Chapter 5 phased greedy approaches using correlation measures were introduced to reduce the number of small cartons produced. Four correlation measures were tested using a greedy insertion technique and it was shown that the number of small cartons produced can be reduced with a small increase in walking distance compared to the GP approach. The number of picking lines requiring a large volume of stock still remained at an undesirable level.

A segmentation approach using mathematical formulations was introduced in Chapter 6 to include capacity constraints. Picking lines were segmented into clusters of no more than three picking lines per cluster and DBNs would initially be assigned to a cluster before being assigned to a specific picking line using a smaller subproblem formulation. Using these capacity constraints all of the volume of stock sent to any one picking line was of a satisfactory size with the number of picking lines requiring a large volume of stock less than the historical assignments. The walking distance was also on average 20% less than the historical assignments. A further hybrid approach (HAS_C) was introduced to incorporate correlations with the capacity constraints. It was shown that this approach yielded the best trade-off between the three goals of walking distance, the number of small cartons produced and the number of picking lines requiring a large volume of stock.

Practical implementation issues of the PLAP in the DC were discussed in Chapter 7. The effects of the assumptions made in Chapter 2 to standardise test problem instances were addressed. A proof of concept for a user interface was proposed which integrates manger experience, PLAP solution approaches and the WMS. The proposed approaches to the PLAP are used to calculate an initial solution which managers could then change if needed. This decision making framework was proposed for use in the DC. Parts of it are already implemented and functional in PEP's WMS.

8.2 Recommendations

In Chapter 3 it was shown that there is a small marginal benefit when solving the SLP if the OSP is solved correctly. Anyone of the tested heuristics can therefore be used to evaluate the total walking distance within a picking line.

Managers currently spread work around picking lines in an attempt to reduce congestion which resembles the CD heuristic. They also consider other factors such as the position of staging areas and additional storage space behind locations when arranging SKUs. After consulting with PEP's management it is recommended that managers should retain a high level of control of this decision tier. This would encourage manager ownership of the picking line and reduce the need for change management. It is therefore proposed that all of the heuristic approaches be run in the DSS and the best solution proposed to the managers who could change the arrangement if they so wish.

In Chapters 4 to 6 several approaches to the PLAP were proposed and compared using three goals. It was shown that a trade-off exists between walking distance, the number of small cartons produced and the volume of stock assigned to any one picking line. A greedy phased approach (GP) which seeks to minimise the sum of the sizes of the maximal SKUs for each picking line was introduced in Chapter 4. This approach minimised the walking distance which generated picking lines requiring large volumes of stock and was not desirable for implementation. Several approaches using correlation measures were introduced in Chapter 5. These approaches had a small increase in walking distance and reduced the number of small cartons produced, however, the number of picking lines requiring a large volume of stock was still undesirable for implementation.

A segmentation approach (SEG_C) was introduced in Chapter 6 included volume capacity constraints. These constraints helped assign desirable volumes of stock to each picking line. A hybrid approach (HAS_C) was further introduced to improve on the number of small cartons produced by the SEG_C approach. This approach used aspects of the ADT correlation approach as well as the SEG_C approach and showed the best trade-off between the three goals. Both the SEG_C and the HAS_C approaches offer desirable results with respect to all three goals and are implementable. It is, however, recommended that the HAS_C should be used to solve the PLAP in the WMS as it has a more desirable trade-off between the number of small cartons produced and walking distance for problem instances where capacity constraints are not binding.

A proof of concept interface was designed in Chapter 7 to integrate the HAS_C approach with manager decision making and the WMS. It is proposed that this framework be implemented in the decision making environment in the DC as it allows sufficient manager flexibility and provides useful data visualisations. The effects of different decisions can quickly be evaluated and visualised by means of this interface. Moreover, an automatic analytical method would be used to generate an initial solution which could be changed by management. This allows for the benefits of automation and human insight.

8.3 Future work

The scope of this dissertation was to describe and address the SLP and PLAP. A natural continuation of these problems is to consider the scheduling of DBNs during a planing horizon, typically weekly. Using the out-of-DC dates as a constraint DBNs can be scheduled to match complimentary DBNs on the same picking line. Focus may also be applied to balancing workloads across shifts during the week. Similar to the picking line volume capacity goals for the PLAP total work (picking by pickers and stock movements by high lifts) should be balanced across shifts to maintain a constant level of staff and reduce overtime.

The effects of assigning DBNs to different days of the week may be evaluated by formulating a PLAP problem which includes waves from multiple days in a connected time period. This creates a static instance of the DBN scheduling problem where all DBNs to be scheduled in a time period are known at the start of the time period. A similar clustering framework presented in Chapter 6 may be used to assign DBNs. Waves from different days should be assigned into the same cluster. Additional constraints can then be added to ensure that DBNs are not assigned to a cluster which is scheduled after its out-of-DC date.

Although solving a static scheduling problem provides insights into problem characteristics and potential effects of new schedules, DBNs are released at the start of each day and not the start of each week/time period creating a dynamic environment. A proposed approach to this dynamic problem would be to adjust the current ranking of DBNs, which is done according to out-of-DC date, and rather assign each DBN into a priority group. For example, group A would consist of all DBNs which are past their deadline, group B all DBNs due in the next two days, group C all DBNs due in the next five days and group D all other DBNs. At the start of each day available picking lines would first be filled with DBNs from group A and only if additional capacity exists will DBNs from group B be selected. This selection would continue until all capacity is full or all DBNs have been scheduled. In this way DBNs within the same group may be picked in a different sequence to their out-of-DC dates. This approach as well as any other approaches should be tested with a long term simulation which incorporates the dynamic releasing of DBNs to the DC.

For both the static and dynamic scheduling problems the historical data presented in Chapter 2 can be used to evaluate the effects of different DBN schedules. Moreover, the test framework presented in Chapter 2 can be used and easily adapted for both scheduling problems.

All three of the different DCs used by PEP run on the same fundamental order picking system although each DC has a different structural layout. Natural questions surrounding structural layouts which arise after developing decision making frameworks include:

1. How many locations should picking lines for each layout have?
2. What is the optimal picking line size mix?
3. What should the ratio of picking line space to overall DC space be?

These questions will require extensive research and scenario testing as structural changes are strategic and cannot be reversed in the short term. Question 1 may be approached by testing different custom picking line configurations using the historical DBN data presented in Chapter 2. A long term simulation model could be used to evaluate the long term effects of a picking line mix in Question 2. A DC simulation would be required to evaluate the space allocations in the DC proposed by Question 3 and would require additional pallet movement analysis. Both the picking line as well as the storage rack areas would need to be simulated.

One of the special case picking lines mentioned in Chapter 2 are those for new stores. These picking lines are not suited for the current picking line setup as they consist of a few stores and many SKUs. Further study may be conducted on how to integrate these picking lines into the standard waves. Two issues must be considered, firstly stock for these new stores must be held in the DC until the store is able to receive stock near its opening date. This may require an increase in the size of the dispatch area. Furthermore, forecasting the sales for the store and assigning stock is complex. This decision is typically made as late as possible to reduce the effects of uncertainty and store development changes. Analysis of new stores can be made to see the accuracy of sales forecasts and the stability of the initial product mix assigned to the store. Poor stability would suggest that the forecast may be made earlier allowing the DC to prepare for these new stores during normal operations.

8.4 Achievement of objectives

In §1.5 the following seven objectives were identified:

Objective I

a Describe the internal layout and operations of the DC to better understand the problem in the DC context;

b Describe in detail the order picking operation in the DC so that the characteristics of the problem may be understood;

c Describe the different decision tiers and their interactions within the order pick operation;

Objective II

a Describe the SKU location problem (SLP) and identify the scope and assumptions;

b Identify the goals of the SLP decision tier;

c Describe the picking line allocation problem (PLAP) and identify the scope and assumptions;

d Identify the goals of the PLAP decision tier;

Objective III

a Obtain representative problem instances to test both the SLP and PLAP;

b Develop a test framework to test solution approaches to the SLP and PLAP while making provision for future research;

Objective IV

a Develop and test solution approaches to the SLP;

b Address the transitive nature of solving the SLP when evaluating solutions to the PLAP;

Objective V

a Develop and test solution approaches to the PLAP;

b Evaluate the trade-offs between the goals of the PLAP and discuss the performance of all solution approaches with regards to these trade-offs;

Objective VI

a Discuss and resolve the practical implementation issues of solution approaches to the PLAP;

b Propose a framework to integrate the PLAP solution approaches within the warehouse management system at PEP ;

Objective VII

a Propose areas and directions for future research;

Objective I was achieved in Chapter 1 where a detailed discussion of PEP's DCs was given. Moreover, a detailed discussion of the order picking system at PEP's DC in Durban was provided. Three decision tiers, resulting in three optimisation problems, namely the OSP, SLP and PLAP were identified and discussed. In §1.4 the SLP and PLAP were discussed in detail addressing the scope, assumptions and goals of each. This was done in fulfilment of Objective II.

In Chapter 2 test problem instances were introduced for both the SLP and PLAP. These problem instances were derived from historical data and made provision for future studies. A test framework was also introduced to test different solution approaches to the SLP, PLAP as well as envisaged future approaches to DBN scheduling. Objective III was therefore achieved.

Objective IV was achieved in Chapter 3. Different solution approaches to the SLP were compared with respect to walking distance. It was shown that there is less marginal benefit when solving this decision tier than solving the OSP correctly. It was therefore proposed that all of the four fast heuristic must be used to solve the SLP for different solutions to the PLAP.

In Chapter 4 a phased greedy approach was introduced to assign DBNs to picking lines. Here the sum of the sizes of the maximal SKUs within each picking line was minimised. Phased greedy insertion approaches using DBN correlation measures were further introduced in Chapter 5 to reduce the number of small cartons produced. In Chapter 6 a segmented mathematical formulation approach was introduced to include volume capacity constraints. These capacity constraints limited the volume of stock assigned to each picking line. A further hybrid approach was introduced to incorporate the advantages of using correlations and the capacity constraints. All of the introduced approaches were compared using the goals listed in §1.4, namely total walking distance, the number of small cartons produced and the volume of stock required to populate a picking line for a wave. Objective V was therefore fulfilled.

Objective VI was achieved in Chapter 7. Practical implementation issues were discussed and addressed. Moreover a proof of concept user interface was proposed to integrate the PLAP solution procedures with management decision making and the WMS. Finally directions for future work was discussed in §8.3 in fulfilment of Objective VII.

8.5 Contribution

A unique order picking system consisting of three decision tiers was investigated in this dissertation. The focus of this dissertation was to develop novel solution approaches for the two unsolved decision tiers, namely the SLP and PLAP. The structure of a picking line shows many similarities to carousel systems in literature, but differs with the presence of wave picking and multiple pickers. Using the maximal cut formulation for the OSP two new mathematical formulations for generating a lower bound to the SLP were introduced. The first formulation used assignment type variables and constraints to assign SKUs to locations. These were combined with additional variables and constraints to determine spans for orders and calculate the size of the maximal SKU. A second model used a TSP type model to sequence the SKUs into a cycle as on a picking line. A different set of additional variables and constraints were then used to assign spans to orders and calculate the size of the maximal SKU. In both cases it was shown that a solution was obtained within one cycle of a lower bound.

Four heuristic approaches were further tested and compared to a set of random solutions. Both the GS and OPA heuristics known to be optimal for some carousel systems were tested as well as the CD heuristic proposed by Hagspihl & Visagie [1] to minimise congestion on a unidirectional picking line. Furthermore a new approach using adjacencies (GA) was introduced and tested. These heuristics were tested and compared in terms of total walking distance. This was the first time that these heuristics had been compared using walking distance on a unidirectional picking line. It was shown that there is minimal benefit from solving the SLP if the OSP is solved correctly. Moreover all the solutions obtained fell into a small range and variance in terms of total walking distance. It was concluded that the effects of this decision tier on walking distance is minimal compared to the order sequencing decision tier. Candidate solutions to the PLAP may therefore be evaluated by using all of the computationally inexpensive heuristic approaches to solve the SLP. This result reduces the complexity of the PLAP as the SLP subproblem is shown to have a smaller effect than the other tiers. Further research into solution approaches to the PLAP can therefore be explored.

In Chapter 4 a novel mathematical formulation was introduced to assign DBNs to picking lines. The size of the maximal SKU was used as an estimator for total walking distance in a picking line and the sum of the sizes of the maximal SKUs within each picking line was minimised. This formulation is not solvable for problem instances with more than four picking lines. A greedy insertion approach proposed by Martello & Toth [2] for generalised assignment problems was adapted for the PLAP and tested. It was shown that in many cases a feasible solution to the problem was not found as all the DBNs were not assigned to a picking line. A new phased greedy insertion approach (GP) was therefore introduced in an effort to achieve feasible solutions for all problem instances.

The overall walking distance decreased by on average 22% compared to the historical assignments using the GP approach. It was further shown that a trade off exists between the walking distance of pickers, the number of small cartons produced and the size of the picking lines. Using the GP approach yields a poor number of small cartons produced and an undesirable number of picking lines requiring large volumes of stock. These results illustrated the scope for improved efficiency in the DC. Moreover, this was the first time that the concept of a maximal SKU was tested as a proxy estimator for distance in the PLAP. It was shown that if the maximal SKU is used as a proxy for distance the actual distance would be reduced by minimising the sum of the sizes of the maximal SKUs within each picking line. It was proposed that the size of the maximal SKU be used as an estimator of the actual walking distance of the pickers in a picking line. Analysis of the correlation between the size of the maximal SKU and the total walking distance of pickers revealed that minimising the size of the maximal SKU reduces the total walking distance for unidirectional picking lines and can be used.

Phased greedy insertion approaches using correlation measures were introduced in Chapter 5 to improve the number of small cartons produced. Two novel measures which aggregated correlations between assigned DBNs were included in these correlation measures. These approaches were shown to reduce the number of small cartons produced with a slight increase in walking distance. It was also shown that it was better to calculate correlations between a candidate DBN and all the assigned DBNs collectively rather than using the sum of the correlations between the individual assigned DBNs.

A novel segmentation approach (SEG_C) for DBN assignments making use of mathematical formulations is introduced in Chapter 6. Picking lines are first assigned to clusters. DBNs are then assigned to the clusters before being assigned to individual picking lines using smaller subproblems. This segmentation approach reduced the problem size and showed comparable results to the GP approach in terms of walking distance. This suggests that the constraints imposed on the problem due to segmentation does not significantly reduce solution quality. Additional volume capacity constraints to control the size of each picking line was further introduced. This approach also introduces the possibility of adding additional constraints which is usually not possible for greedy insertion approaches. Additional constraints may be added should additional capacities be applied to a picking line such as a limit to the number of bulky items discussed in Chapter 1. This approach was solvable for all problem instances and both the walking distance and picking line sizes were reduced compared to the historical assignments.

A novel hybrid approach (HAS_C) was introduced which merged the ADT and SEG_C approaches. This approach yielded the best trade-off between the thee goals as it took advantage of correlation measures in instances where capacity constraints were not binding. The hybrid approach had significantly smaller walking distances compared to the historical assignments and had a satisfactory number of small cartons produced. Moreover, picking lines requiring large volumes of stock were generated when compared to the historical assignments.

As a result of testing and analysing solution approaches to the SLP and PLAP several further contributions were made. A representative set of historical DBN to picking line assignments were developed for use in future studies. Moreover, a test framework is presented for future studies which uses this data. Finally, a methodology to integrate these techniques into the current WMS at PEP is proposed. This methodology takes into account human interaction and allows flexibility during the decision making process.

Bibliography

[1] HAGSPIHL R & VISAGIE S, 2014, *The number of pickers and stock-keeping unit arrangement on a uni-directional picking line*, The South African Journal of Industrial Engineering, **25(3)**, pp. 169–183.

[2] MARTELLO S & TOTH P, 1990, *Knapsack Problems Algorithms and Computer Implementations*, Johhn Wiley & Sons, West Sussex England.

APPENDIX A

Minimising the maximal SKU additional results

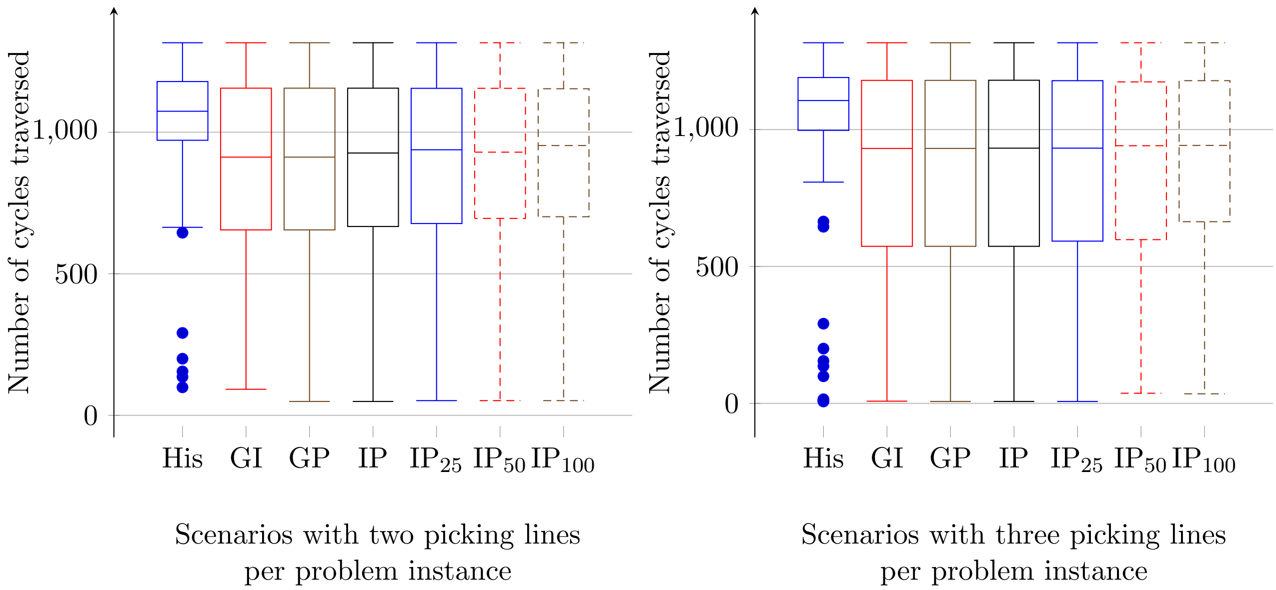
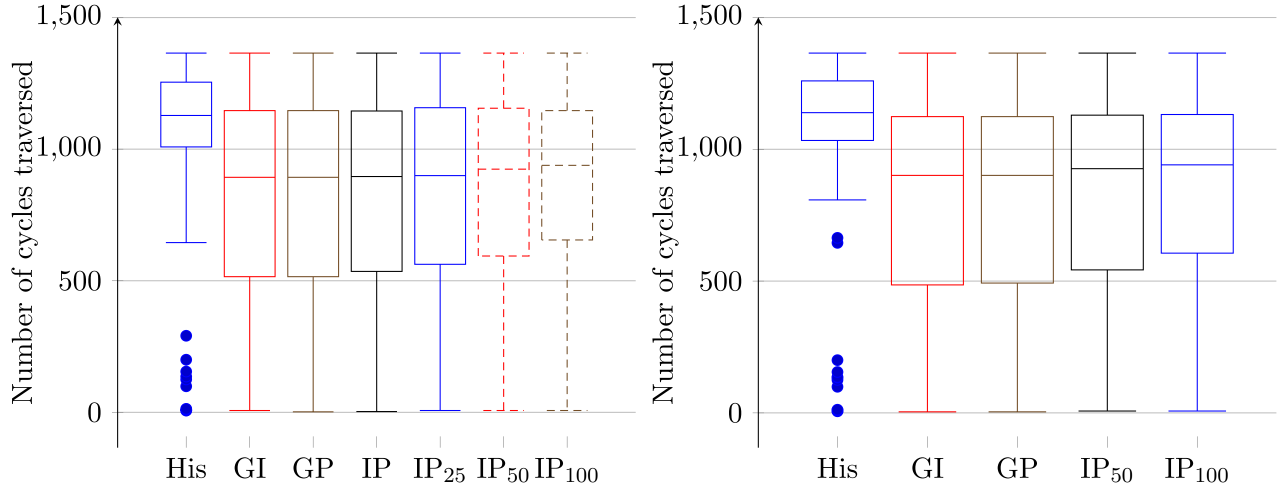


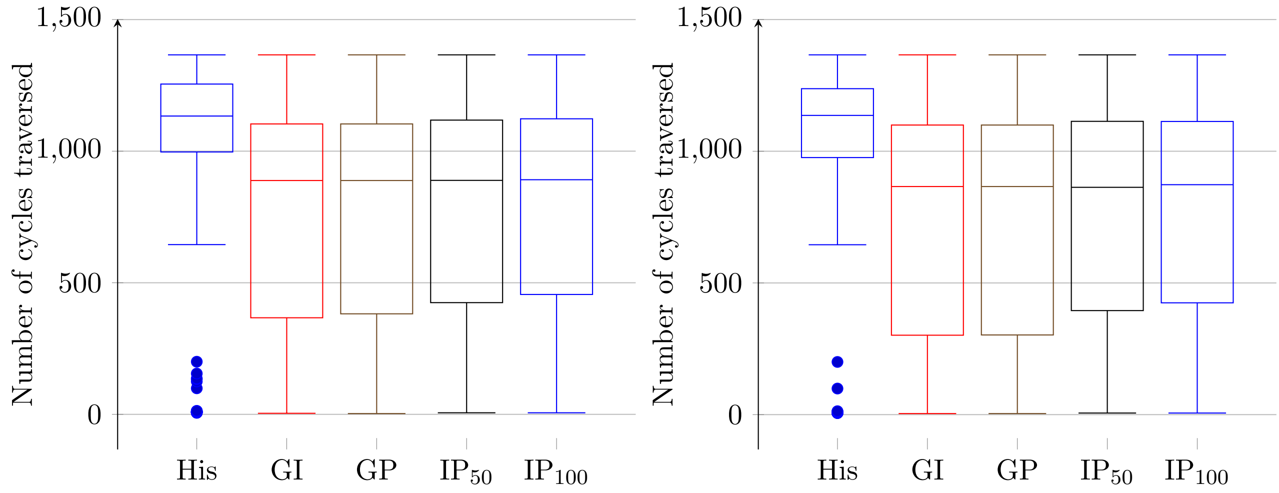
Figure A.1: *A graphical box-plot representation of the distribution of the number of cycles traversed for each line after scheduling scenarios with two and three picking lines per problem instance using the greedy insertion (GI), the phased greedy insertion (GP), the integer programming formulations (IP$_\alpha$) as well as the historical assignment (His).*



Scenarios with four picking lines per problem instance

Scenarios with five picking lines per problem instance

Figure A.2: *A graphical box-plot representation of the distribution of the number of cycles traversed for each line after scheduling scenarios with four and five picking lines per problem instance using the greedy insertion (GI), the phased greedy insertion (GP), the integer programming formulations (IP$_\alpha$) as well as the historical assignment (His).*



Scenarios with six picking lines per problem instance

Scenarios with seven picking lines per problem instance

Figure A.3: *A graphical box-plot representation of the distribution of the number of cycles traversed for each line after scheduling scenarios with six and seven picking lines per problem instance using the greedy insertion (GI), the phased greedy insertion (GP), the integer programming formulations (IP$_\alpha$) as well as the historical assignment (His).*

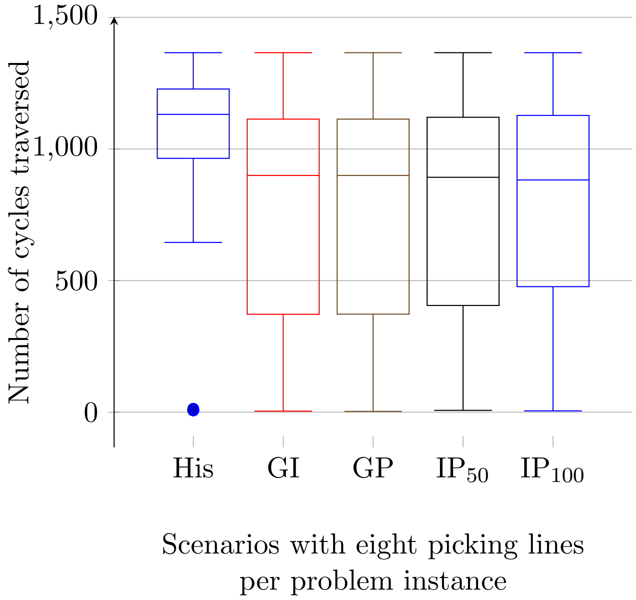


Figure A.4: *A graphical box-plot representation of the distribution of the number of cycles traversed for each line after scheduling scenarios with eight picking lines per problem instance using the greedy insertion (GI), the phased greedy insertion (GP), the integer programming formulations (IP$_\alpha$) as well as the historical assignment (His).*

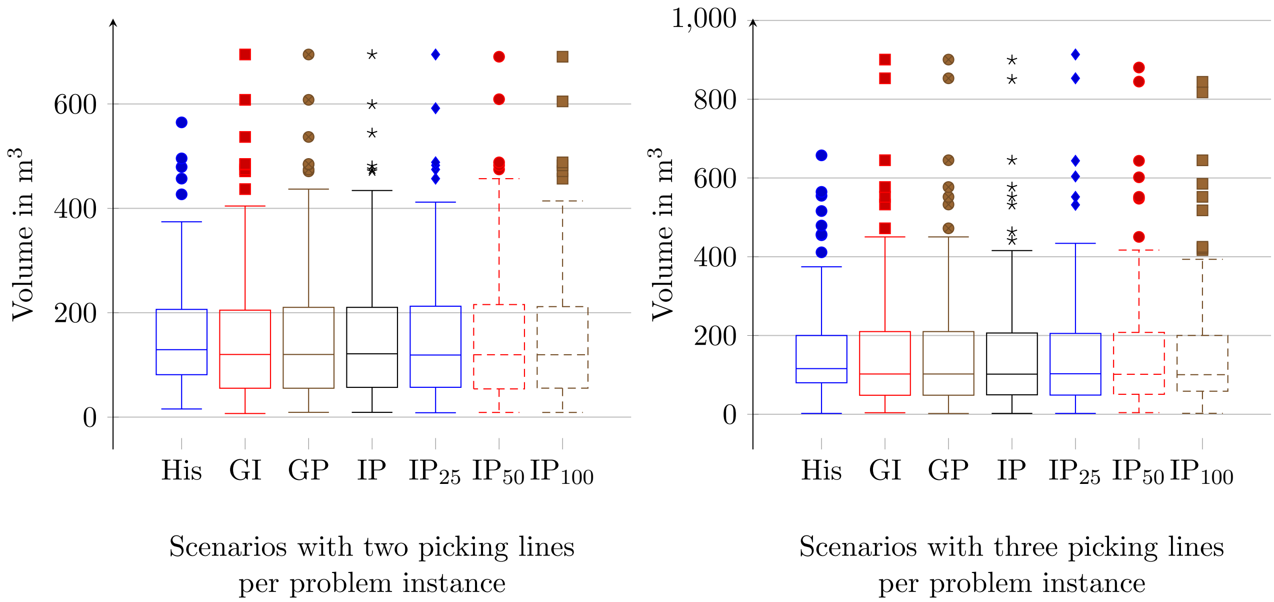
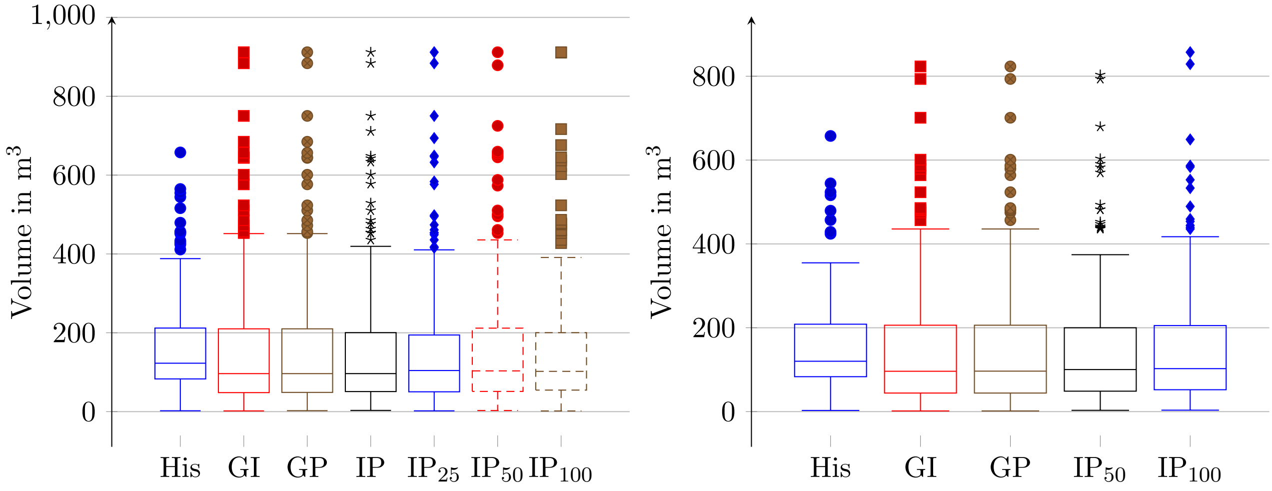


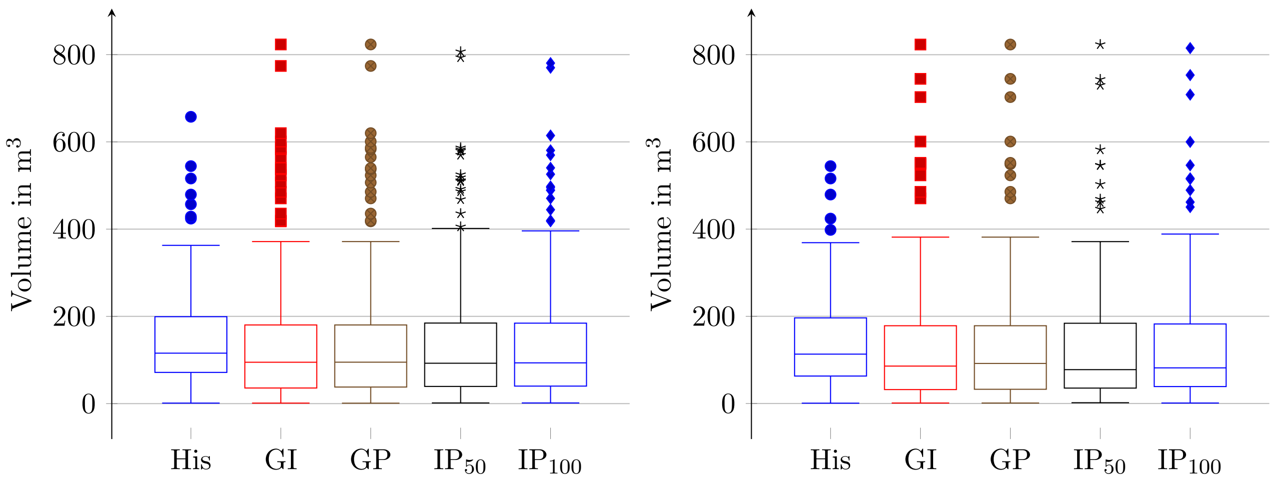
Figure A.5: *A graphical box-plot representation of the distribution of the total volume for each line after scheduling scenarios with two and three picking lines per problem instance using the greedy insertion (GI), the phased greedy insertion (GP), the integer programming formulations (IP$_\alpha$) as well as the historical assignment (His).*



Scenarios with four picking lines per problem instance

Scenarios with five picking lines per problem instance

Figure A.6: *A graphical box-plot representation of the distribution of the total volume for each line after scheduling scenarios with four and five picking lines per problem instance using the greedy insertion (GI), the phased greedy insertion (GP), the integer programming formulations (IP$_\alpha$) as well as the historical assignment (His).*



Scenarios with six picking lines per problem instance

Scenarios with seven picking lines per problem instance

Figure A.7: *A graphical box-plot representation of the distribution of the total volume for each line after scheduling scenarios with six and seven picking lines per problem instance using the greedy insertion (GI), the phased greedy insertion (GP), the integer programming formulations (IP$_\alpha$) as well as the historical assignment (His).*

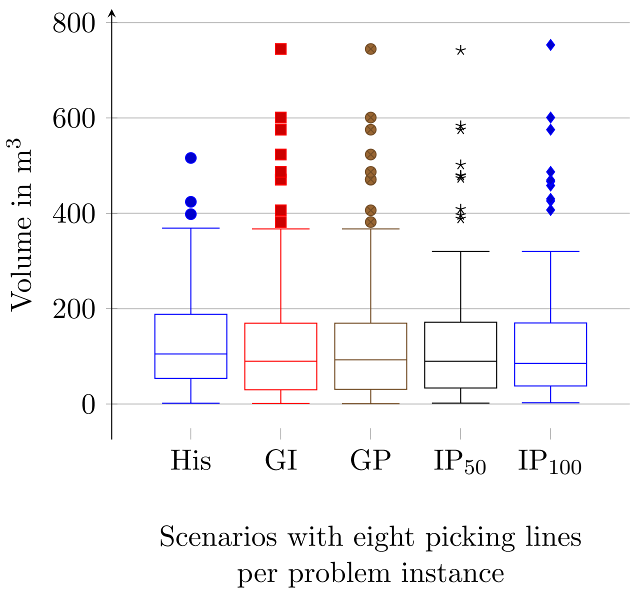


Figure A.8: *A graphical box-plot representation of the distribution of the total volume for each line after scheduling scenarios with eight picking lines per problem instance using the greedy insertion (GI), the phased greedy insertion (GP), the integer programming formulations (IP_α) as well as the historical assignment (His).*

APPENDIX B

Correlation assignments additional results

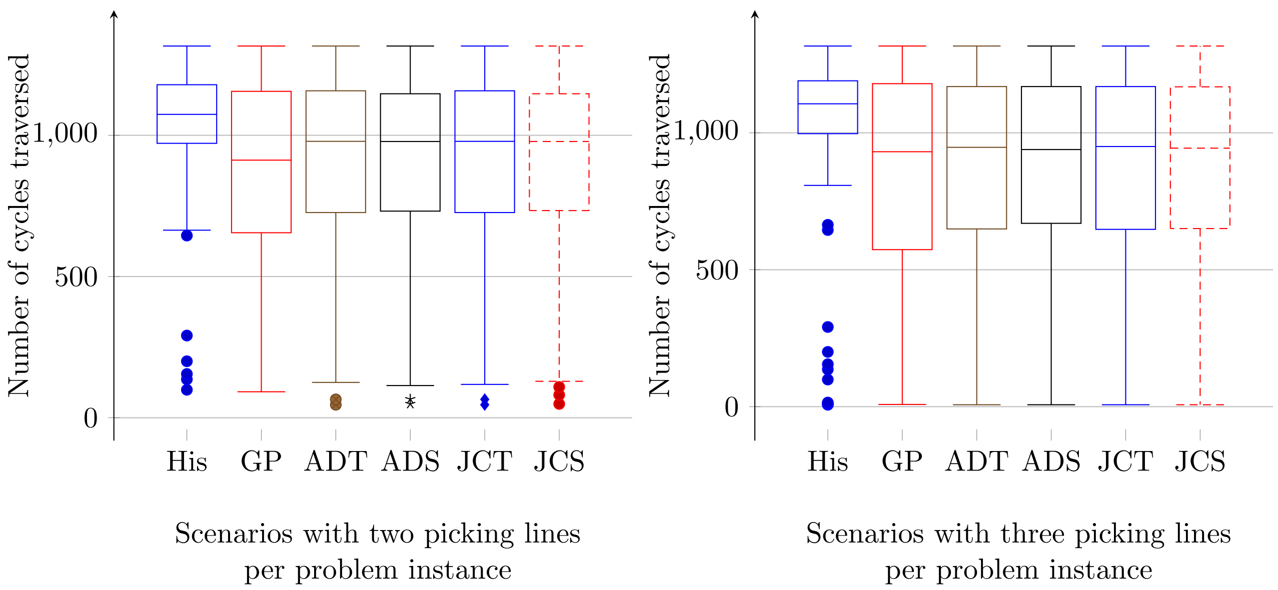
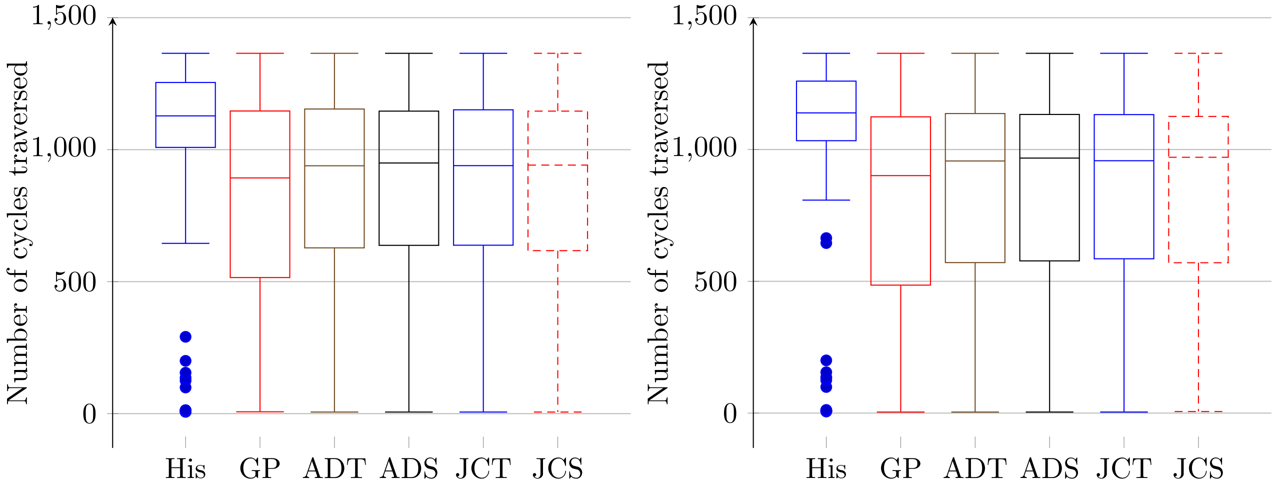


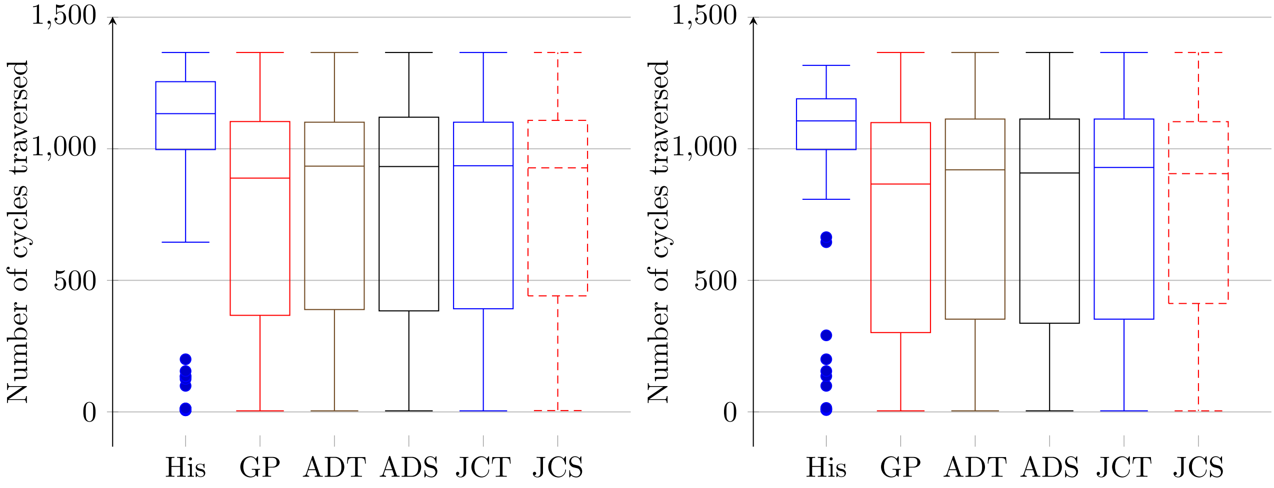
Figure B.1: *A graphical box-plot representation of the distribution of the number of cycles traversed for each line after scheduling scenarios with two and three picking lines per problem instance using the ADT, ADS, JCT and JCS correlation based heuristics as well as the phased greedy insertion (GP) heuristic and historical results (His).*



Scenarios with four picking lines per problem instance

Scenarios with five picking lines per problem instance

Figure B.2: *A graphical box-plot representation of the distribution of the number of cycles traversed for each line after scheduling scenarios with four and five picking lines per problem instance using the ADT, ADS, JCT and JCS correlation based heuristics as well as the phased greedy insertion (GP) heuristic and historical results (His).*



Scenarios with six picking lines per problem instance

Scenarios with seven picking lines per problem instance

Figure B.3: *A graphical box-plot representation of the distribution of the number of cycles traversed for each line after scheduling scenarios with six and seven picking lines per problem instance using the ADT, ADS, JCT and JCS correlation based heuristics as well as the phased greedy insertion (GP) heuristic and historical results (His).*

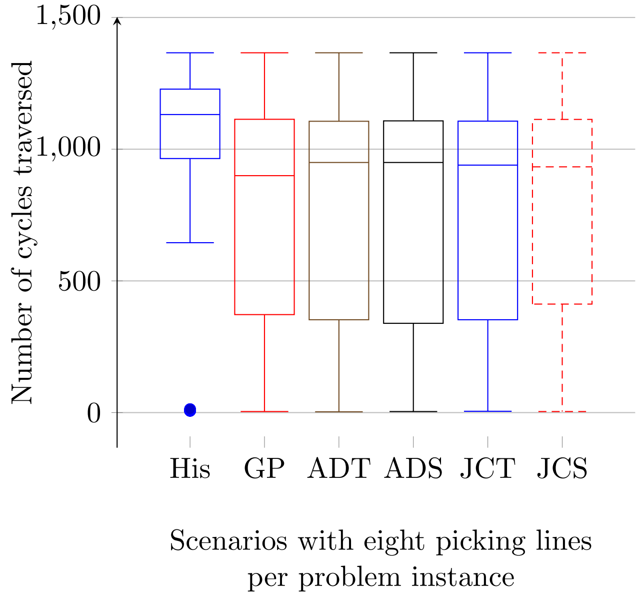


Figure B.4: *A graphical box-plot representation of the distribution of the number of cycles traversed for each line after scheduling scenarios with eight picking lines per problem instance using the ADT, ADS, JCT and JCS correlation based heuristics as well as the phased greedy insertion (GP) heuristic and historical results (His).*

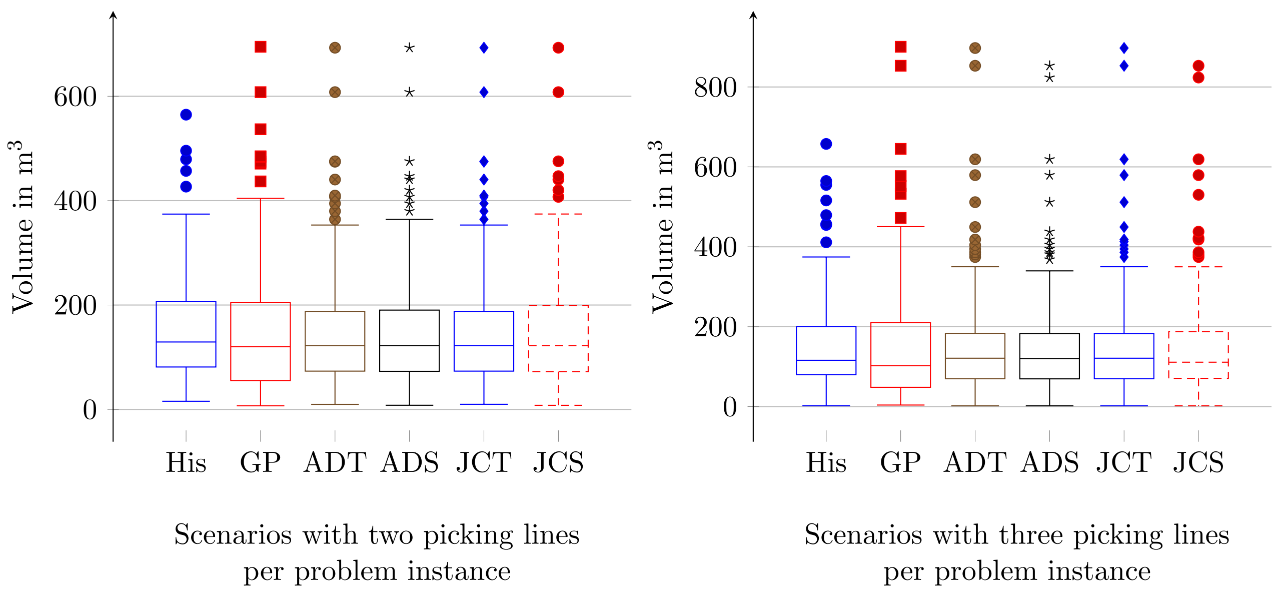
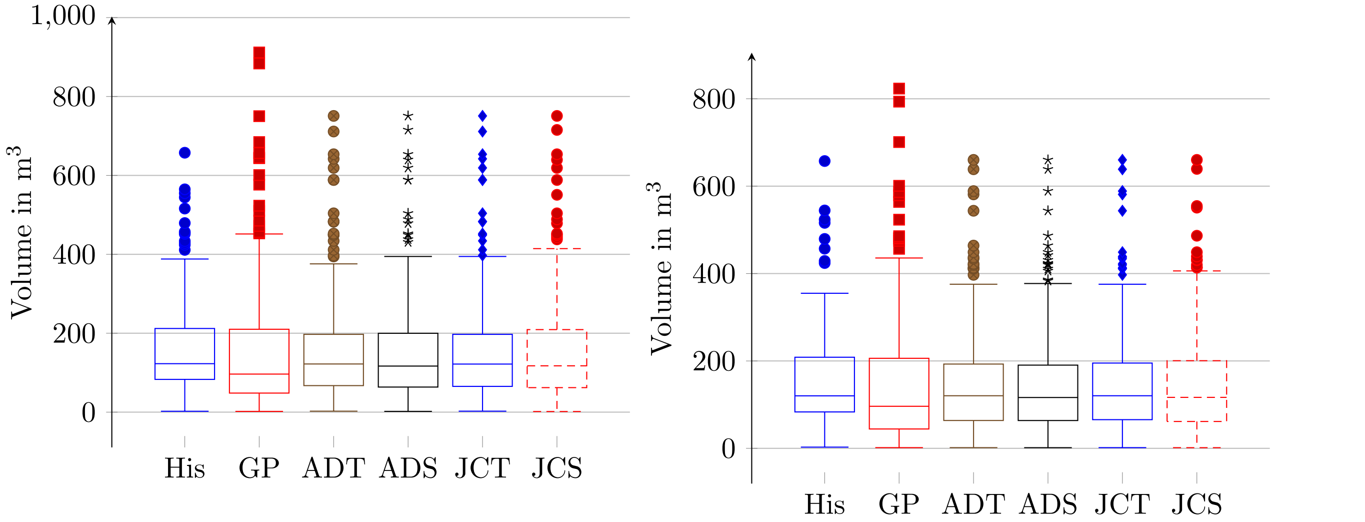


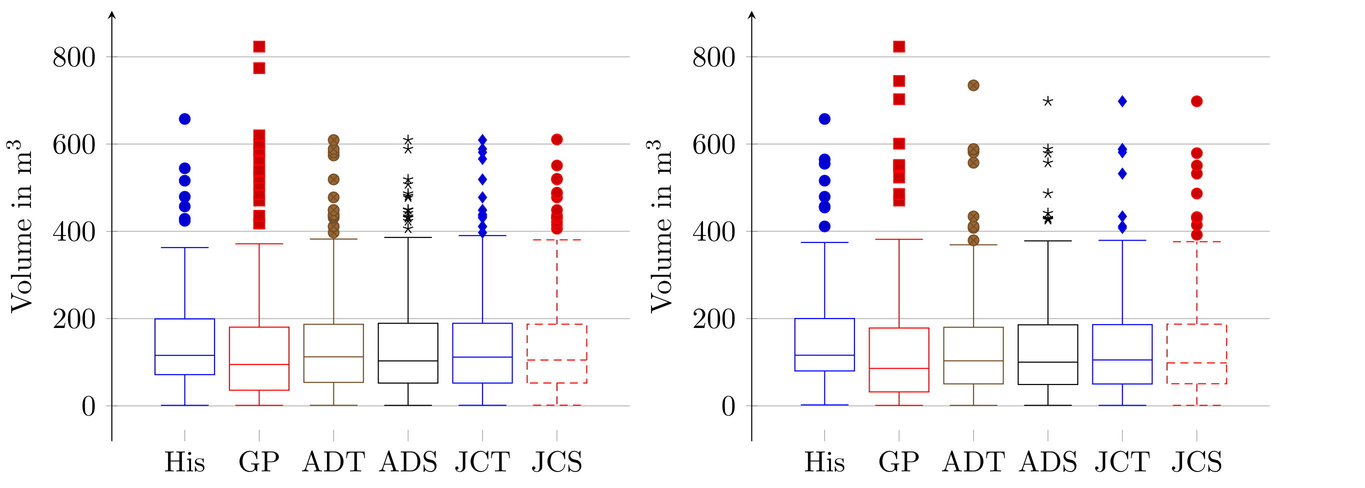
Figure B.5: *A graphical box-plot representation of the distribution of the total volume for each line after scheduling scenarios with two and three picking lines per problem instance using the ADT, ADS, JCT and JCS correlation based heuristics as well as the phased greedy insertion (GP) heuristic and historical results (His).*



Scenarios with four picking lines per problem instance

Scenarios with five picking lines per problem instance

Figure B.6: *A graphical box-plot representation of the distribution of the total volume for each line after scheduling scenarios with four and five picking lines per problem instance using the ADT, ADS, JCT and JCS correlation based heuristics as well as the phased greedy insertion (GP) heuristic and historical results (His).*



Scenarios with six picking lines per problem instance

Scenarios with seven picking lines per problem instance

Figure B.7: *A graphical box-plot representation of the distribution of the total volume for each line after scheduling scenarios with six and seven picking lines per problem instance using the ADT, ADS, JCT and JCS correlation based heuristics as well as the phased greedy insertion (GP) heuristic and historical results (His).*

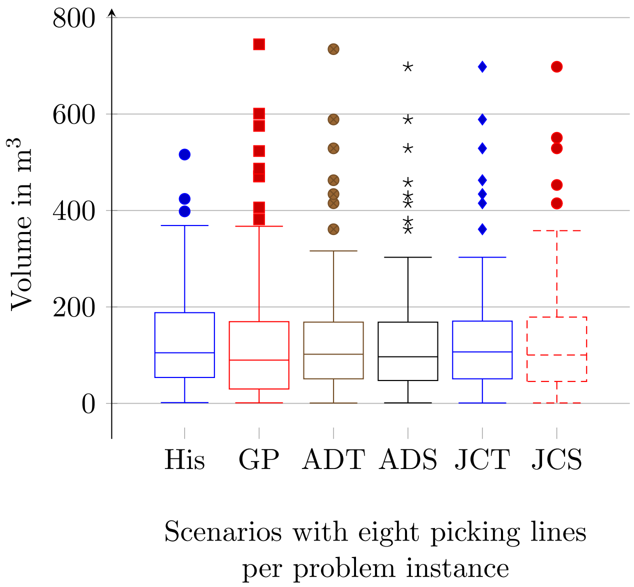
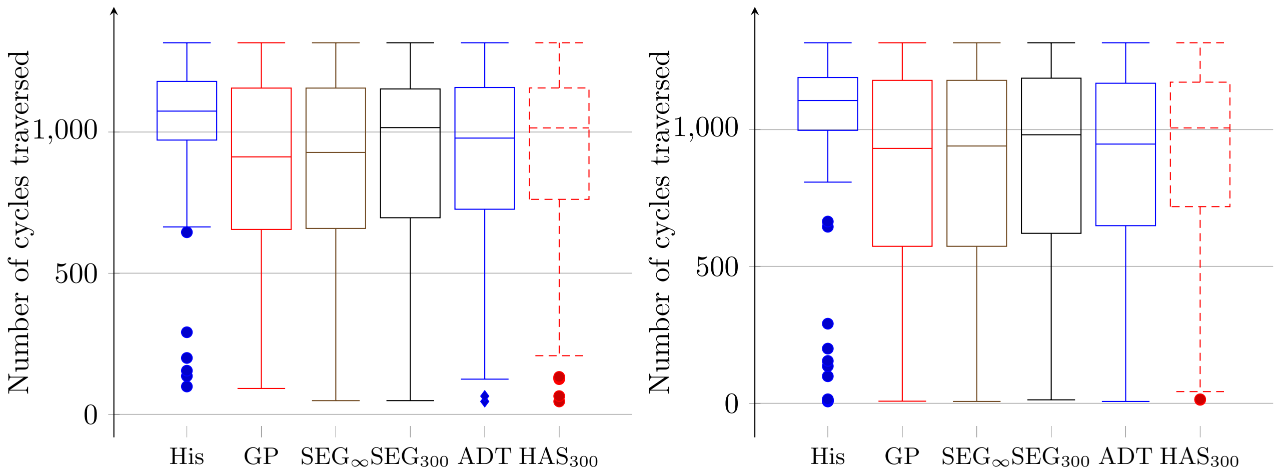


Figure B.8: *A graphical box-plot representation of the distribution of the total volume for each line after scheduling scenarios with eight picking lines per problem instance using the ADT, ADS, JCT and JCS correlation based heuristics as well as the phased greedy insertion (GP) heuristic and historical results (His).*

APPENDIX C

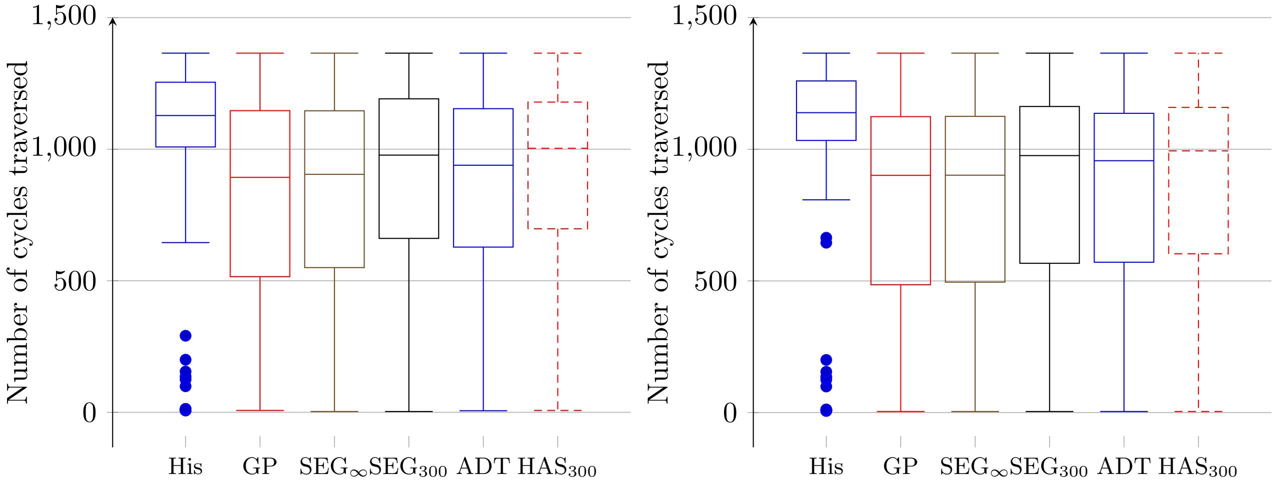
Capacity constraint assignment additional results



Scenarios with two picking lines per problem instance

Scenarios with three picking lines per problem instance

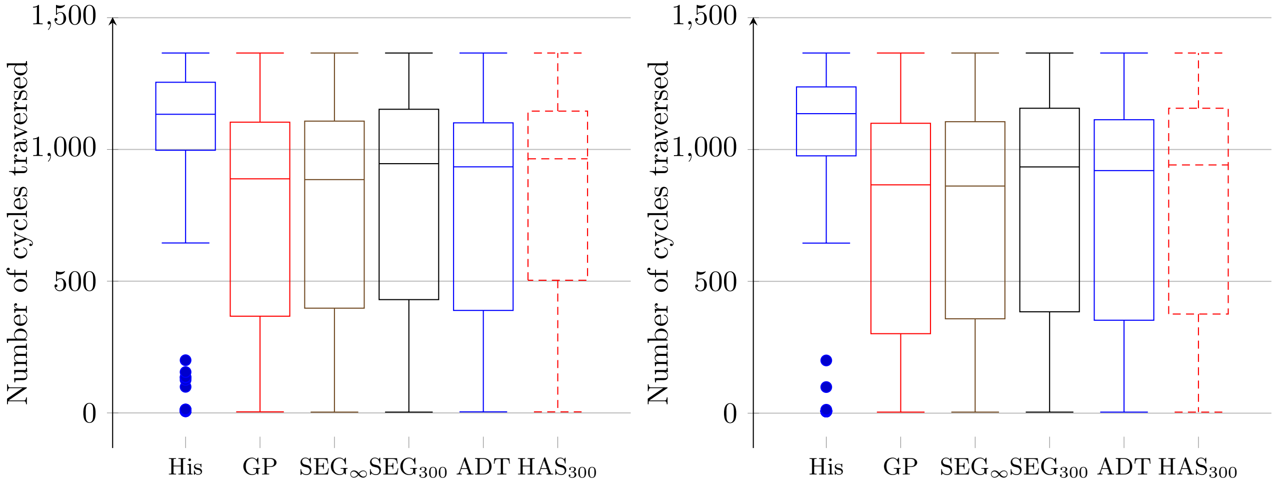
Figure C.1: *A graphical box-plot representation of the distribution of the number of cycles traversed for each line after scheduling scenarios with two and three picking lines per problem instance using the SEG_{∞}, SEG_{300} and HAS_{300} segmentation and hybrid heuristics as well as the phased greedy insertion (GP) heuristic, the adjacencies (ADT) heuristic and historical results (His).*



Scenarios with four picking lines per problem instance

Scenarios with five picking lines per problem instance

Figure C.2: *A graphical box-plot representation of the distribution of the number of cycles traversed for each line after scheduling scenarios with four and five picking lines per problem instance using the* SEG_∞*,* SEG_{300} *and* HAS_{300} *segmentation and hybrid heuristics as well as the phased greedy insertion (GP) heuristic, the adjacencies (ADT) heuristic and historical results (His).*



Scenarios with six picking lines per problem instance

Scenarios with seven picking lines per problem instance

Figure C.3: *A graphical box-plot representation of the distribution of the number of cycles traversed for each line after scheduling scenarios with six and seven picking lines per problem instance using the* SEG_∞*,* SEG_{300} *and* HAS_{300} *segmentation and hybrid heuristics as well as the phased greedy insertion (GP) heuristic, the adjacencies (ADT) heuristic and historical results (His).*

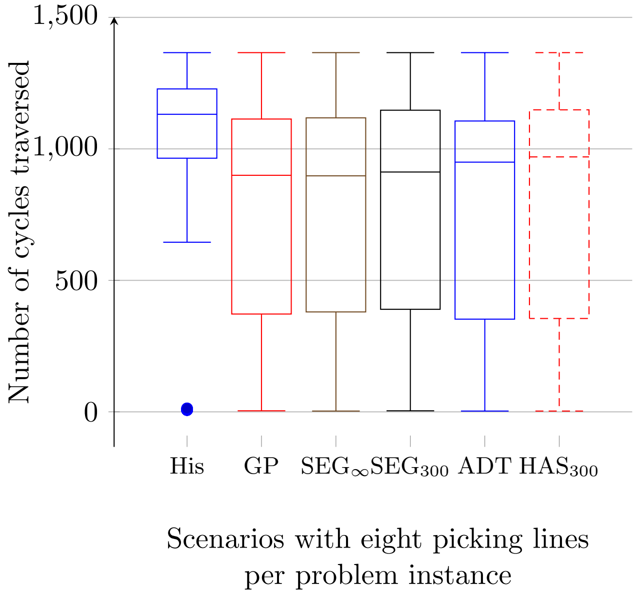
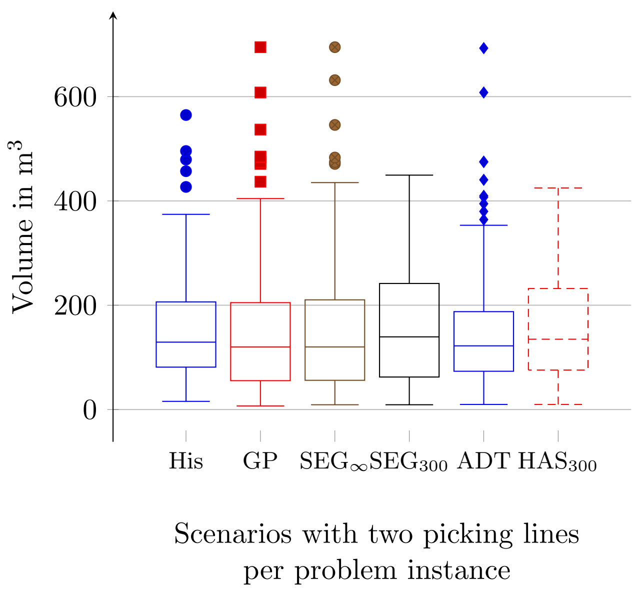


Figure C.4: *A graphical box-plot representation of the distribution of the number of cycles traversed for each line after scheduling scenarios with eight picking lines per problem instance using the* SEG_∞*,* SEG_{300} *and* HAS_{300} *segmentation and hybrid heuristics as well as the phased greedy insertion (GP) heuristic, the adjacencies (ADT) heuristic and historical results (His).*



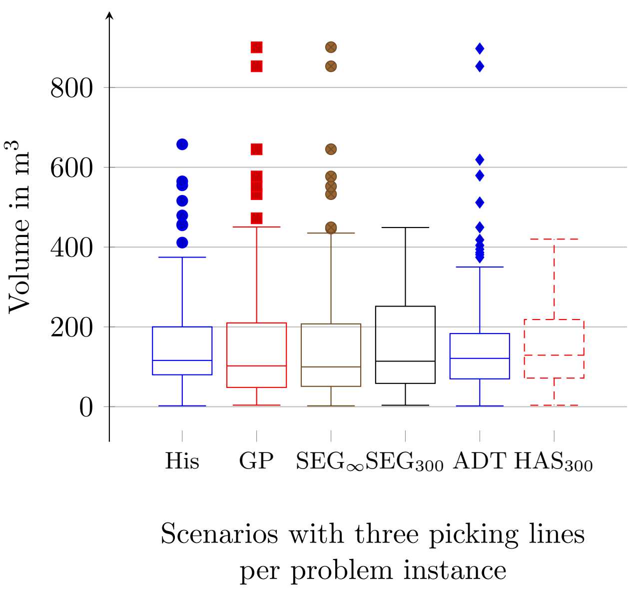


Figure C.5: *A graphical box-plot representation of the distribution of the total volume for each line after scheduling scenarios with two and three picking lines per problem instance using the* SEG_∞*,* SEG_{300} *and* HAS_{300} *segmentation and hybrid heuristics as well as the phased greedy insertion (GP) heuristic, the adjacencies (ADT) heuristic and historical results (His).*

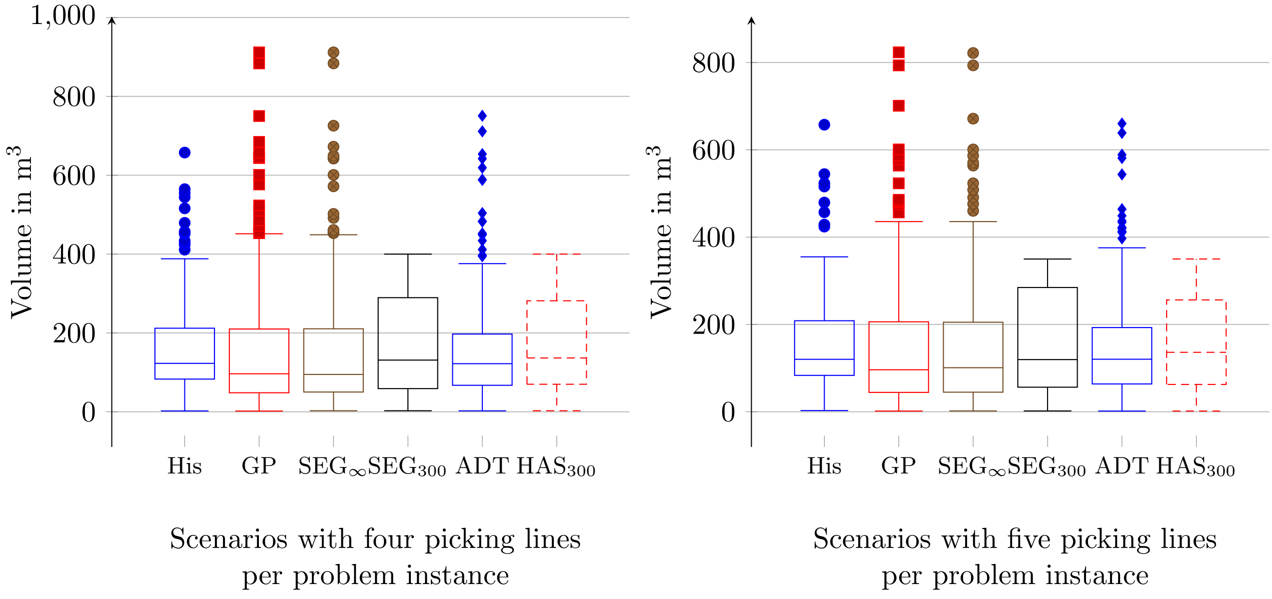


Figure C.6: *A graphical box-plot representation of the distribution of the total volume for each line after scheduling scenarios with four and five picking lines per problem instance using the SEG_∞, SEG_{300} and HAS_{300} segmentation and hybrid heuristics as well as the phased greedy insertion (GP) heuristic, the adjacencies (ADT) heuristic and historical results (His).*

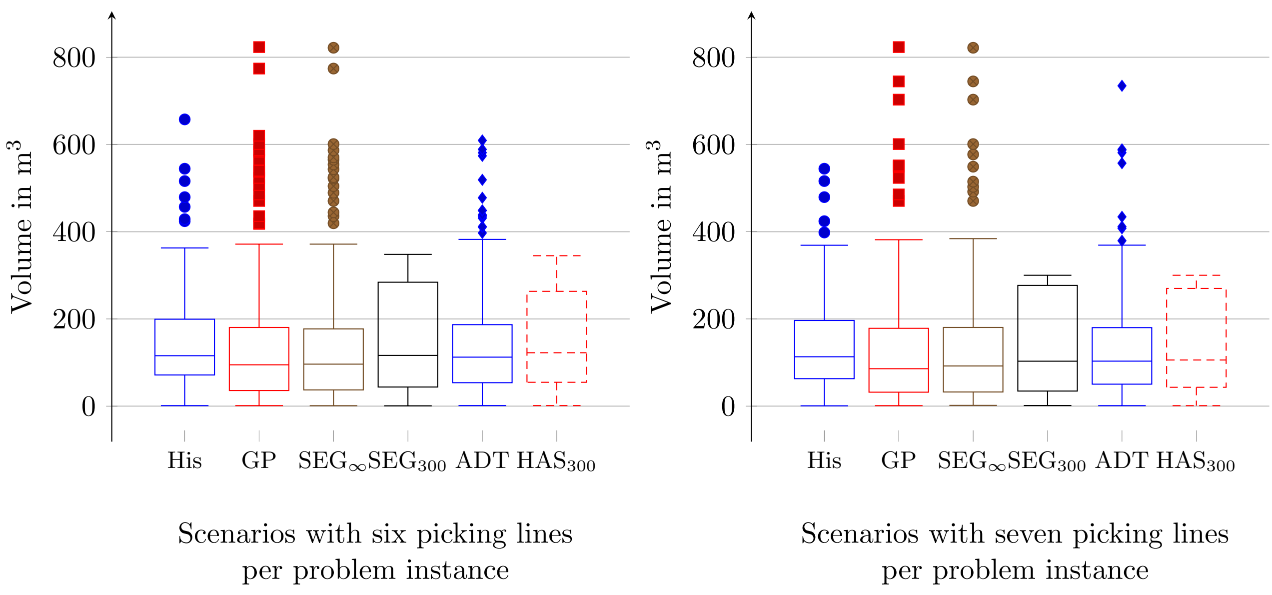


Figure C.7: *A graphical box-plot representation of the distribution of the total volume for each line after scheduling scenarios with six and seven picking lines per problem instance using the SEG_∞, SEG_{300} and HAS_{300} segmentation and hybrid heuristics as well as the phased greedy insertion (GP) heuristic, the adjacencies (ADT) heuristic and historical results (His).*

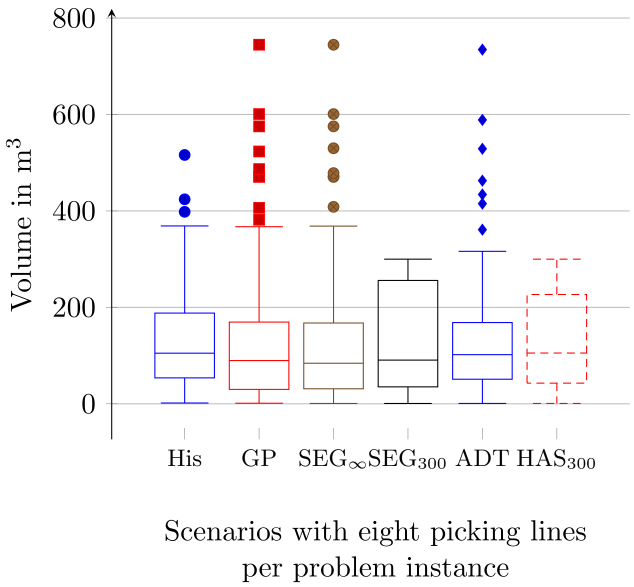


Figure C.8: *A graphical box-plot representation of the distribution of the total volume for each line after scheduling scenarios with eight picking lines per problem instance using the SEG_∞, SEG_{300} and HAS_{300} segmentation and hybrid heuristics as well as the phased greedy insertion (GP) heuristic, the adjacencies (ADT) heuristic and historical results (His).*

APPENDIX D

Scatter plot additional results

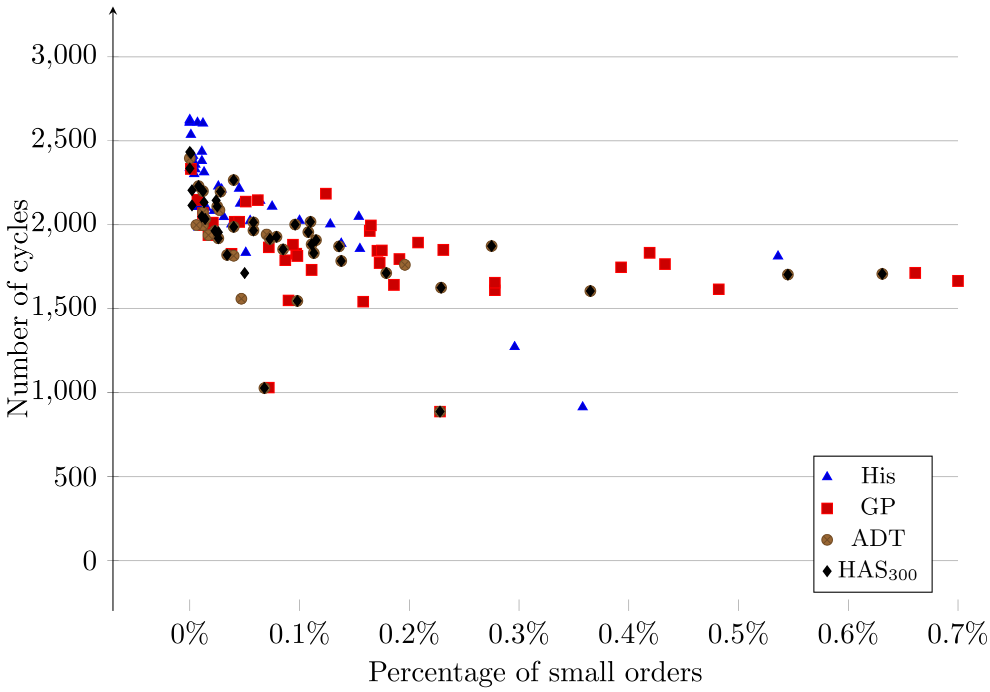


Figure D.1: *A scatter plot comparing the GP, ADT and HAS_{300} approaches to the historical assignments in terms of the number of cycles traversed and the percentage of the total picked volume attributed to small orders. Each marker represents a single problem instance with two picking lines.*

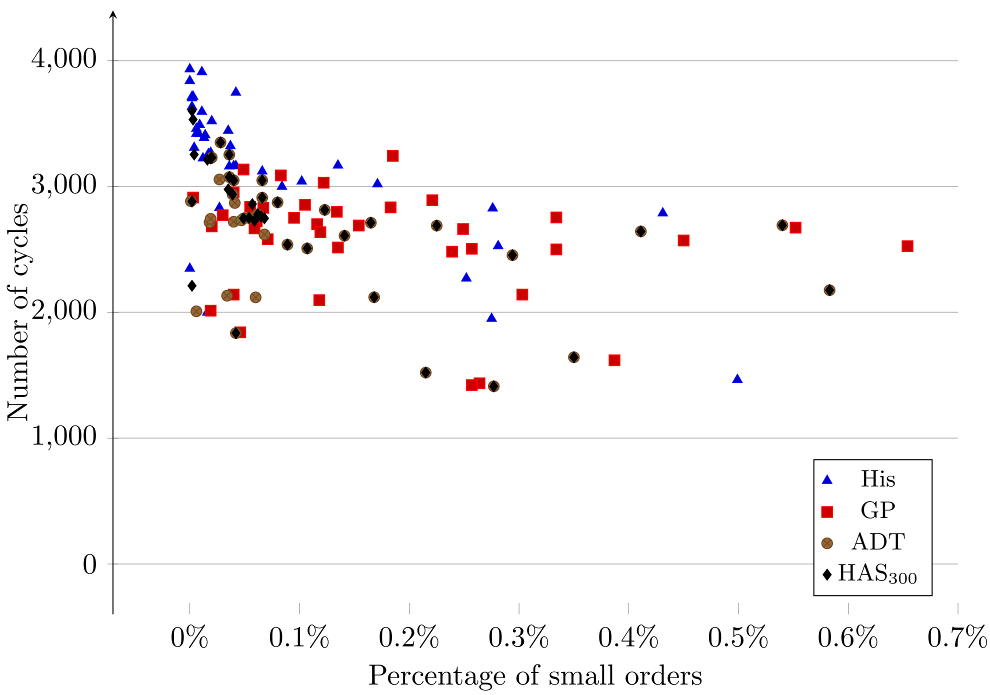


Figure D.2: *A scatter plot comparing the GP, ADT and HAS_{300} approaches to the historical assignments in terms of the number of cycles traversed and the percentage of the total picked volume attributed to small orders. Each marker represents a single problem instance with three picking lines.*

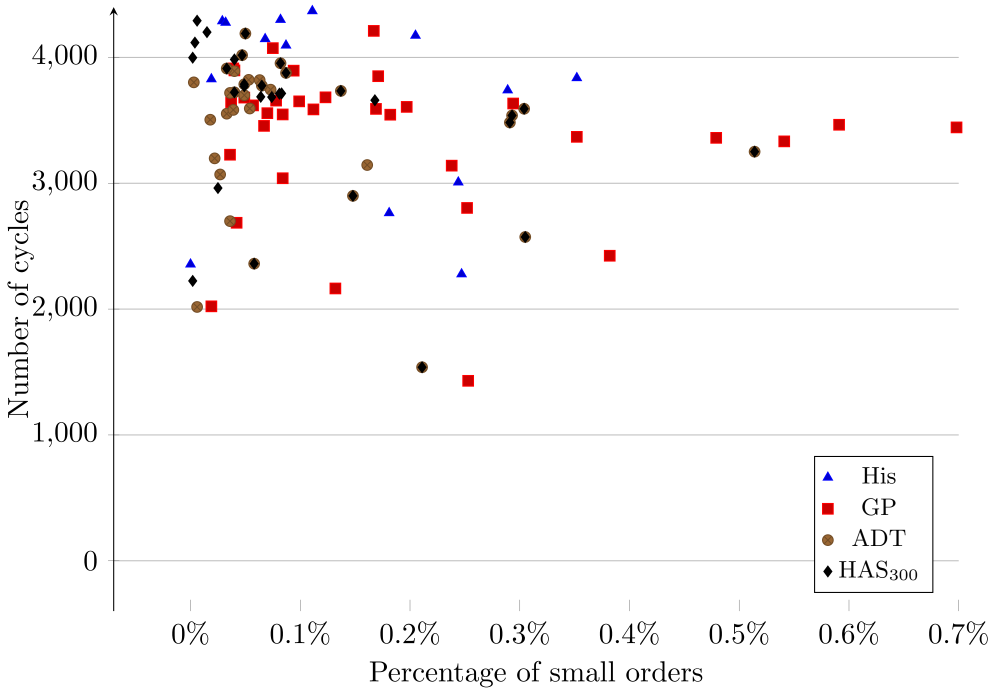


Figure D.3: *A scatter plot comparing the GP, ADT and HAS_{300} approaches to the historical assignments in terms of the number of cycles traversed and the percentage of the total picked volume attributed to small orders. Each marker represents a single problem instance with four picking lines.*

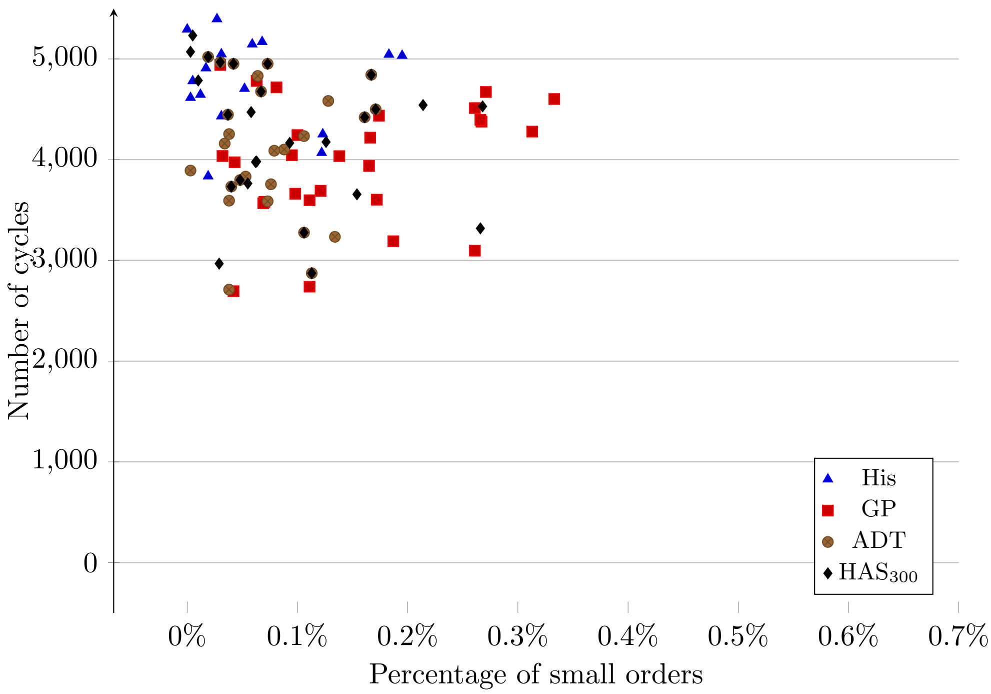


Figure D.4: *A scatter plot comparing the GP, ADT and HAS_{300} approaches to the historical assignments in terms of the number of cycles traversed and the percentage of the total picked volume attributed to small orders. Each marker represents a single problem instance with five picking lines.*

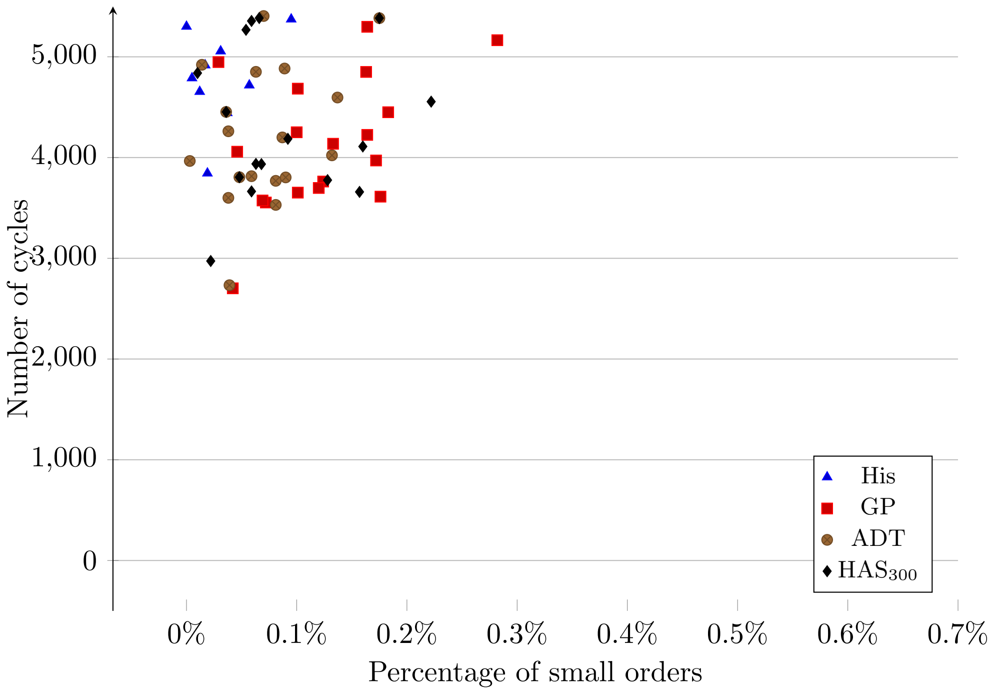


Figure D.5: *A scatter plot comparing the GP, ADT and HAS_{300} approaches to the historical assignments in terms of the number of cycles traversed and the percentage of the total picked volume attributed to small orders. Each marker represents a single problem instance with six picking lines.*

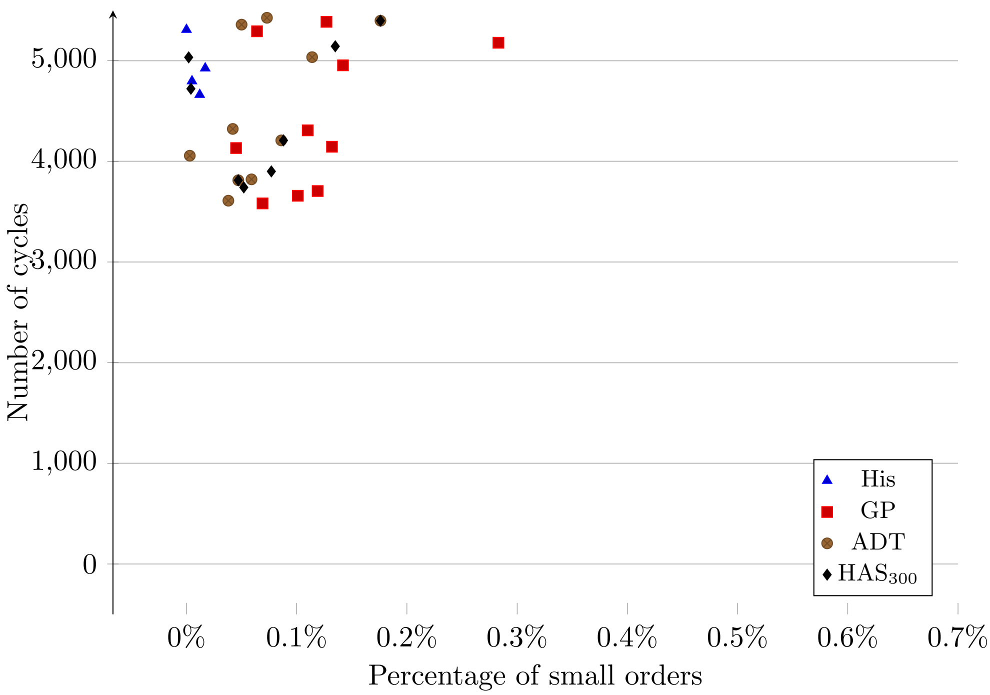


Figure D.6: *A scatter plot comparing the GP, ADT and HAS_{300} approaches to the historical assignments in terms of the number of cycles traversed and the percentage of the total picked volume attributed to small orders. Each marker represents a single problem instance with seven picking lines.*

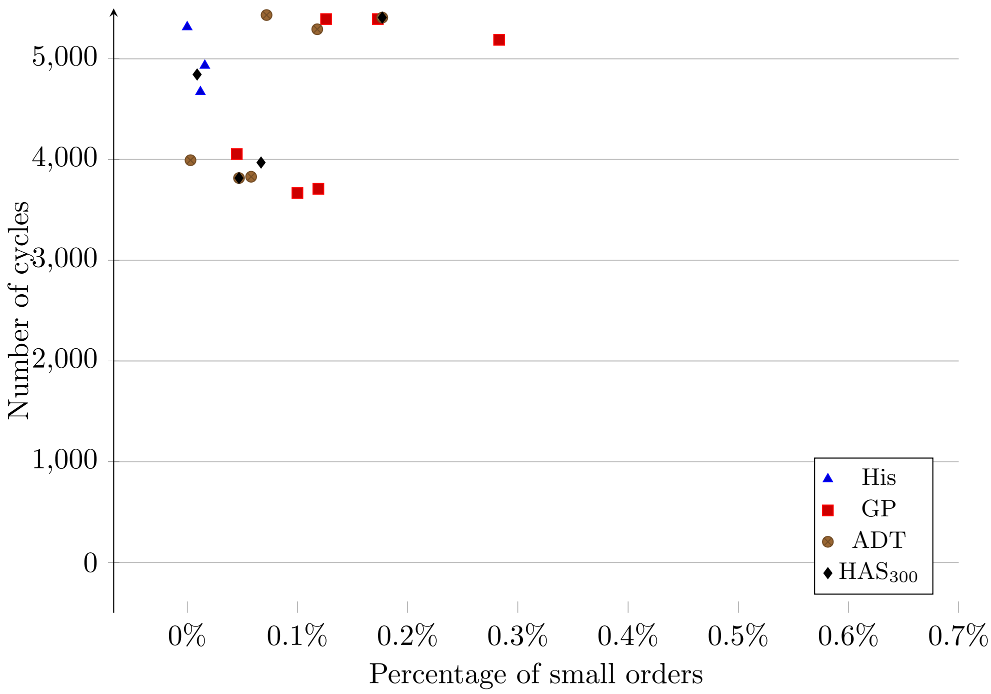


Figure D.7: *A scatter plot comparing the GP, ADT and HAS_{300} approaches to the historical assignments in terms of the number of cycles traversed and the percentage of the total picked volume attributed to small orders. Each marker represents a single problem instance with eight picking lines.*