Order picking optimisation on a unidirectional cyclical picking line

Flora Hofmann

Dissertation presented in fulfilment of the requirements for the degree of

Doctor of Philosophy (Operations Research)

in the Faculty of Economic and Management Sciences at Stellenbosch University

Promoter: Prof SE Visagie

December 2020

Declaration

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

Date: December 1, 2020

Abstract

The order picking system in a company's distribution centre is the biggest contributor to the operational cost within the DC. Optimisation should thus aim at running this activity as efficiently as possible. The order picking process consists of three main activities, namely walking to the stock, picking stock in fulfilment of a customer order and handling the picked stock for further processing. While the total amount of work for the picking and handling activities remain constant, the minimisation of walking distance becomes the main objective when minimising the total picking effort. The minimisation of walking distance can be translated into a reduced overall picking time which can lead to a decrease in the total cost of operating the picking system.

The main objective of this dissertation is to optimise the order picking system on a unidirectional cyclical picking line. Order batching is introduced to the picking system, since it is an effective methodology that minimises walking distance in operations research literature. Order batching has been introduced to the standard single block parallel-aisle warehouse layout, but not to the specific layout of a unidirectional cyclical picking line. Additionally, the unidirectional cyclical picking line can offer two configuration options that change the physical set up and thereby influence the way in which pickers walk during the order picking process.

Order batching is introduced to the unidirectional cyclical picking line through picking location based order-to-route closeness metrics. These metrics are further extended by taking the characteristics of the layout into account. The distribution centre of a prominent South African retailer provides real life test instances. Introducing the layout specific stops non-identical spans metric in combination with the greedy smallest entry heuristic results in a reduction of 48.3% in walking distance.

Order batching increases the pick density which may lead to higher levels in picker congestion. In a discrete event simulation, the reduction of the overall picking time through a decrease in walking distance is thus confirmed. On tested sample picking waves, the overall picking time can be reduced by up to 21% per wave. A good number of pickers in the picking system is dependent on the pick density. The pick density, amongst other explanatory variables, can also be used to predict the reduction in picking time.

The effects of different structural options of the unidirectional cyclical picking line, namely the U- and Z-configuration, are investigated. This results in four decision tiers that have to be addressed while optimising the order picking system. The first decision tier assigns stock to picking lines, the second arranges stock around a picking line, the third chooses the configuration and the last sequences the orders to be picked. Order batching is added as an additional layer. An increase in pick density benefits the reduction of walking distance throughout the decision tiers and supports the choice of the U-configuration after evaluating different test instances. The total completion time of a picking wave can thus be reduced by up to 28% when compared to benchmark instances. The dissertation is concluded by suggesting further research directions.

Opsomming

Die opmaak van bestellings op 'n uitsoeklyn in 'n onderneming se distribusiesentrum is die grootste bydraer tot die bedryfskoste van 'n distribusiesentrum. Dit is dus belangrik om hierdie aktiwiteit so doeltreffend moontlik te maak. Die proses om bestellings op te maak bestaan uit drie hoofaktiwiteite, naamlik stap na die voorraad, uitsoek (kies en bymekaarsit) van die voorraad vir 'n bestelling en die pak van die gekose voorraad in kartonne vir verdere verwerking en verspreiding. Omdat die totale hoeveelheid werk vir die uitsoek- en hanteringsaktiwiteite konstant bly, word die vermindering van loopafstand die hoofdoelwit om die totale koste van hierdie proses te minimeer. Die minimering van loopafstand lei tot 'n vermindering in totale tyd om bestellings op te maak, wat op sy beurt weer lei tot 'n afname in die totale koste van die stelsel om bestellings op te maak.

Die hoofdoel van hierdie proefskrif is om die stelsel vir die uitsoek van bestellings op 'n eenrigting sikliese uitsoeklyn te optimeer. Metodes vir die samevoeging of groepering (Eng.: *batching*) van bestellings (om gelyktydig opgemaak te word) word ontwikkel vir hierdie uitsoekstelsel aangesien operasionelenavorsingsliteratuur aantoon dat groepering van bestellings 'n effektiewe metode is om loopafstand te verminder. Groepering van bestellings is reeds gedoen vir die standaard blokuitleg van distribusiesentra, maar nie vir hierdie spesifieke uitleg van 'n eenrigting sikliese uitsoeklyn nie. Daarbenewens het die eenrigting sikliese uitsoeklyn twee konfigurasie-opsies wat die fisiese opstelling verander en sodoende die manier beïnvloed waarop werkers tydens die uitsoekproses loop.

Die groepering van bestellings word ontwikkel vir 'n eenrigting sikliese uitsoeklyn deur middel van 'n plek-gebaseerde maatstaf wat die nabyheid van bestellings se roetes meet. Hierdie maatstaf word verder uitgebrei deur die eienskappe van die uitleg in ag te neem. Regte voorbeelde van die probleem uit 'n distribusiesentrum van 'n prominente Suid-Afrikaanse kleinhandelaar word gebruik vir toetsing. Die ontwikkeling en implementering van 'n uitlegspesifieke stop-nie-identiese-strek-maatstaf in kombinasie met die gulsige kleinste-invoegingsheuristiek lei tot 'n vermindering van 48.3% in stapafstand.

Die groepering van bestellings verhoog die digtheid van plekke waar werkers stop vir voorraad, wat kan lei tot hoër vlakke van kongestie vir werkers. 'n Diskrete-gebeurtenis-simulasie bevestig dat 'n afname in loopafstand ook 'n vermindering van die totale voltooiingstyd tot gevolg het. Met behulp van werklike historiese data kon die totale tyd vir die uitsoek van bestellings met tot 21% per golf verminder word. 'n Goeie aantal werkers in die uitsoekstelsel is afhanklik van die uitsoekdigtheid. Die uitsoekdigtheid en andere verklarende veranderlikes, kan ook gebruik word om die vermindering in totale tyd om bestellings op te maak, te voorspel.

Die invloed van verskillende strukturele opsies van die eenrigting sikliese uitsoeklyn, naamlik die U- en Z-konfigurasie, word ook ondersoek. Dit het tot gevolg dat vier besluitnemingsvlakke aangespreek moet word om die uitsoekstelsel te optimeer. Die eerste besluitnemingsvlak ken voorraad aan die uitsoeklyne toe, die tweede rangskik voorraad binne die uitsoeklyn, die derde

kies die konfigurasie van die lyn en die laaste kies die volgorde waarin die bestellings uitgesoek word. Groepering van bestellings word bygevoeg as 'n addisionele vlak. 'n Toename in werksdigtheid bevoordeel die vermindering van loopafstand deur die besluitvlakke en bevoordeel die U-konfigurasie na evaluering van verskillende toetsdata. Die totale voltooiingstyd van 'n uitsoekgolf kan dus verminder word met tot 28% in vergelyking met eweknie voorbeelde. Die studie word afgesluit deur verdere navorsingsmoontlikhede voor te stel.

Acknowledgements

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

- Prof SE Visagie for all his guidance and support from my master's thesis to the compilation of this dissertation (and a special thanks for the amazing research workshops in Glentana);
- My labmates Chesme Messina, Kurt Marais and Hein von Stein for all the good times in (and out of) our research lab;
- The Graduate School of Economic and Management Sciences and especially Dr Jaco Franken for his support and the curation of the GEM programme;
- PEP and especially Hennie Serdyn for his support and instant data supply;
- My parents Helga & Hans-Peter and my sister Viola for their long-distance support from Germany;
- Lydon Carter, Bella & the Dorringtons, Stephan "Junior" Du Toit, Dr Matthew Mayne, Dr Sabrina Kumschick, and Vanessa & Kudakwashe "Byron" Bimha for all adventures during my PhD journey;
- The Department of Logistics at Stellenbosch University for the use of their computing facilities and office space;
- The Stack Overflow community for the answers to my coding questions;

Any opinions or findings in this thesis are those of the author and do not necessarily reflect the view of Stellenbosch University.

Contents

List of figures

List of tables

List of acronyms

A-GS	Spans-ratio greedy smallest entry batching combination
ANOVA	Analysis of variance
B-GS	Non-identical stops-spans metric greedy smallest entry batching combination
C-GS	Stops-spans ratio metric greedy smallest entry batching combination
CI	Confidence interval
CO	Combination heuristic
CPU	Central processing unit
CW	Clarke and Wright-algorithm
D-GS	Non-identical minimum span metric greedy smallest entry batching combination
DBN	Distribution
DC	Distribution centre
DM	Desirability measure
DES	Discrete event simulation
E-GR	Non-identical span greedy random batching combination
F-CO	Stops list-spans metric combination
FIFO	First-in-first-out
G-CO	Minimum spans list-spans ratio metric combination
GD	Great deluge
GBU	Greedy bottom-up heuristic
GI	Greedy insertion heuristic
GIDM	Greedy insertion heuristic using a desirability measure
GOF	Goodness of fit test
GR	Greedy random heuristic
GRA	Greedy random assignment approach
GRL	Greedy random arrangement
GS	Greedy smallest entry heuristic
GSL	Greedy sequential arrangement
GTD	Greedy top-down heuristic
H-CO	Stops list-spans ratio metric combination
HA	Historical data
ID	Identifier
ILS	Iterated local search
IP	Integer program
K-CO	Minimum spans list-spans metric combination
L-CO	Minimum spans list-non-identical minimum span metric combination
MA	Marginal analysis
MILP	Mixed integer linear program

N-GR	Non-identical stops metric greedy random batching combination
NE	Nearest end heuristic
NP	Non-deterministic polynomial-time
O-CO	Minimum spans list-non-identical span metric combination
OBP	Order batching problem
OSP	Order sequencing problem
P-CO	Stops list-non-identical minimum span metric combination
PMA	Pattern mining approach
Q-CO	Stops list-non-identical span metric combination
QM	Quick match algorithm
R-GR	Stops ratio metric greedy random batching combination
S-GR	Spans metric greedy random batching combination
SA	Simulated annealing
SKU	Stock keeping unit
SAP	SKU arrangement problem
SCP	System configuration problem
SPLAP	SKU assignment problem
TS	Tabu search
T-GR-GD	Stops metric greedy random great deluge batching combination
U-CO	Minimum spans list-stops metric combination
V-CO	Minimum spans list-non-identical stop metric combination
VND	Variable neighbourhood descent
VNS	Variable neighbourhood search
VRS	Voice recognition system
VV	Verification and validation
W-CO	Minimum spans list-stops ratio metric combination
WMS	Warehouse management system
Z-GS	Stops non-identical spans metric greedy smallest entry batching combination

CHAPTER 1

Introduction

Contents

Before a product reaches a retailer's shelf to be bought by an end customer it has to go through several value adding and non-value adding, but essential processes. Different organisations and business entities perform these processes. Supply chain management strategically manages the different processes and relationships between various business entities to maintain competitiveness [77].

The concept of supply chain management first entered the business vocabulary during the 1990s [25]. According to Beamon [10] it can be defined as an integrated process in which different business entities work together to acquire raw materials, turn them into products and deliver these products to a retailer. While there is a forward flow of materials, it also includes a backward flow of information. Coyle *et al.* [25] describes this integrated supply chain concept in Figure 1.1 as the effective and efficient flow of products, services, information and finances through different business entities to the end-customer.

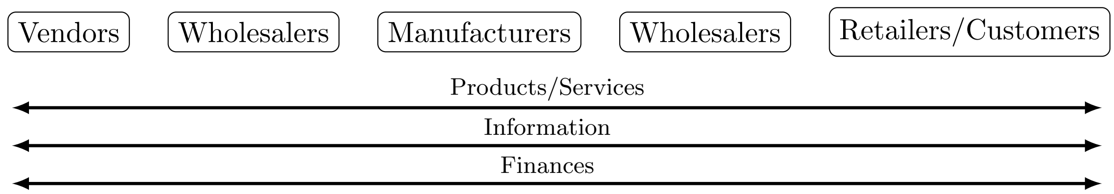


FIGURE 1.1: *A schematic representation of an integrated supply chain representing the products/services, information and material flow. Source: Coyle et al. [25].*

The physical movement of products between business entities is a challenge and carried out by the logistics network of the integrated supply chain. In Figure 1.2 a logistics channel is depicted. Raw material is sent to manufacturers and the finished products are then sent to distribution centres (DCs). [1] In DCs the products are consolidated and distributed to retailers to be put on their shelves.

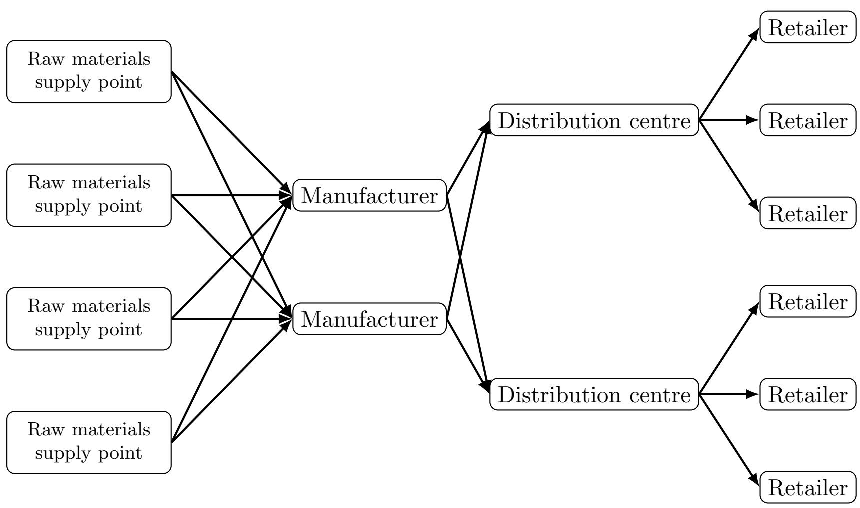


FIGURE 1.2: *A schematic representation of a logistics network. Source: Coyle et al. [25].*

DCs connect manufacturers and retailers in the logistics network of an integrated supply chain. DCs are unlikely to vanish from the current business landscape as the trend towards a greater product variety and shorter response times are increasing. The operation of the nodes (for example DCs) of a logistics network mainly determines the network's efficiency and effectiveness. Therefore, optimisation efforts focused on this part of the supply chain might result in an increase in both efficiency and effectiveness [98].

1.1 Distribution centres

DCs tend to buffer variations between supply and demand since they can hold products in their inventory. Consequently, DCs have a strong focus on consolidation and accumulation of various products from suppliers [77]. Normally products from different suppliers arrive in bulk at DCs. The bulk stock is then reworked or repacked into orders for delivery to customers [113].

Frazelle [35] divides DC activities into eight different functions that are common despite the type of DC. The areas in which these functions take place are depicted in Figure 1.3. Inbound

[1] In this dissertation no distinction is made between warehouses and distribution centres.

processes are receiving, put away and storage. Outbound processes are order picking, packaging and pricing; sortation and accumulation; and unitising and shipping [7].

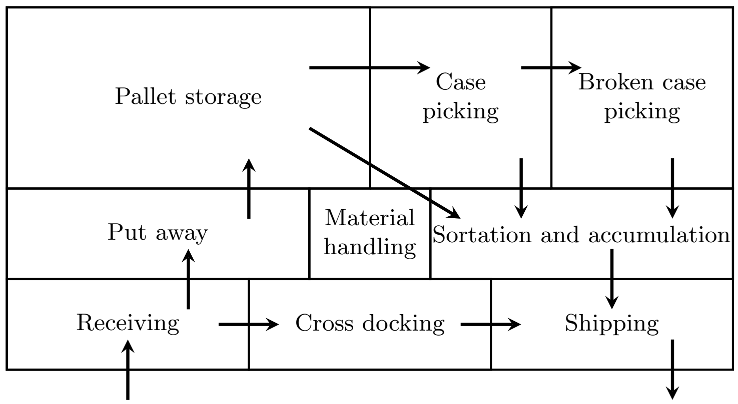


FIGURE 1.3: *A schematic representation of the DC's functional areas and the stock movement governed by material handling and indicated through arrows. Source: Frazelle [35].*

Receiving begins with the advanced notification that goods are arriving. Therefore, the unloading can be coordinated efficiently to match the other activities within the DC. It includes the collection of all goods coming into the DC and a quality and quantity check according to the purchase order. Any exceptions will be noted. Scanning the goods registers their arrival and dispatches payments accordingly. If no processing is required, goods are pushed through to the cross-docking area. The goods are distributed to storage or taken to an area for follow up activities [7]. In a typical DC, the reception of goods accounts for about 10% of the operational cost, since goods usually arrive in large quantities and the handling is thus less labour-intensive [35].

The activity of pre-packaging is performed if goods are received in bulk, but need to be packaged in quantities that are mechanisable or put together with other items to form kits and assortments. Depending on the storage requirements, and whether the products are part of kits and assortments, the processing can either happen over time or immediately. Pre-packaging is an optional activity of material handling. Therefore, no general percentage of operational cost can be assigned [35].

An appropriate storage location has to be determined before a product can be put away. How quickly and how costly the process of retrieval is, depends to a large extent on the storage location. Information about the storage locations must be available at all times. After handling the material, the storage location has to be verified by scanning to record the placement of the good. Pick lists for order picking are generated from this information [7]. Put away comprises approximately 15% of the DC's operational cost as goods may have to be moved significant distances to reach their storage location [35].

The activity of storage is the physical containment of goods while awaiting customer demand. The size, quantity and the characteristics of the product or its container determine the storage method. There is no operational cost, only rent expense, since the product is waiting to be processed further [35].

Once a customer request or order is in the system, the DC checks the inventory for availability. Then the DC produces pick lists to process order picking. Order picking and shipping have to be scheduled and shipping documents need to be issued. Most of these activities are performed by the software coordinating all activities within a DC which is called the warehouse management system (WMS). Order picking is the basic service of a DC. Order picking is a labour-intensive process and thus accounts for approximately 55% of the operational cost of a DC [110]. Therefore, all processes, as well as the layout of the DC, centre around the activity of order picking. Additionally, the goods have to be replenished to sustain order picking. The restocker retrieves goods from bulk storage and prepares pallets, cases or broken cases for picking. In general a restock is thus more expensive than a pick [7].

Packaging and pricing are optional at this point. Comparable to pre-packaging, individual goods or kits and assortments are containerised to provide flexibility in the use of on-hand inventory. Additionally, pre-pricing at the manufacturer leads to repricing as price lists may change while the goods are stored. Sometimes picking tickets and price stickers are combined to reduce material handling [35].

Sortation of batch picks to individual orders or accumulation of distributed picks to one order can be considered for efficiency. In most cases these activities are combined in the order picking process. Therefore, no separate cost of operation is accounted for [35].

Unitising and shipping involves combining packages, checking order completeness, and loading shipping containers. Order accuracy is important due to the high cost of returns. This part of the process can be rather labour-intensive although there is little walking. If shipping documents have not been processed yet, they have to be prepared. Additionally, during this phase, order sizing and weighing can be done to determine shipping costs. Products are scanned again to register the customer order available for shipping. Shipping is less labour-intensive, since larger units are processed. Partial shipments must be staged to accumulate all orders by outbound carrier. However, staging results in double-handling as goods have to be loaded on the truck again afterwards. In some cases the loading of the truck is part of the shipping activity, but in most cases loading falls under the responsibility of the carrier [7]. The shipping activity accounts for about 20% of the operational cost [110].

Labour accounts for most of the expenses in a typical DC [35]. The order picking process is the most labour-intensive and normally accounts for about 60% of the total operational cost. It is thus the most expensive activity [113]. Therefore, most research on DCs focuses strongly on this activity. Research should result in fast, simple, intuitive and reliable methods to facilitate practical applications and thereby increase the cooperation between academia and industry [42]. Analytic models combined with simulation models can analyse the interactions with other DC activities [43].

1.2 The order picking process

Retrieving products from storage areas to fulfil customer requests describes the main activity of order picking. Clustering and scheduling orders, assigning stock to order lines, releasing orders to be picked, the actual picking of products from storage locations, and the distribution of the products are included in the detailed process steps [29]. Order picking accounts for 50% to 65% of the total operating expenses in a DC and is often the most expensive activity [7, 35, 110, 113]. According to Frazelle [35] order picking itself can be broken down into 55% of travelling, 15% of searching, 10% of extracting, and 20% of paperwork and other activities. Travelling is thus the most resource consuming activity as no value is added to the product. Therefore, in the design and operation of order picking systems, the reduction of travelling should be the main focus [77].

DCs differ in regards to the customers they serve, and also with regards to the size and quantity of products they distribute. Therefore, the order picking operation is unique to each company's DC. Customer orders consist of order lines with each line containing a unique item or stock keeping unit (SKU). Order lines can be split into case picks and broken case picks based on the quantity and the product carrier of the SKU [29]. The number of order lines picked per day, the number of items, and the average size of a customer order, influence the choice of the order picking system [27].

The level of automation also induces differences in the picking operations. While humans physically pick items in manual order picking, automated order picking is carried out by machines. Most DCs rely on manual order picking systems, due to the diversity in the size of the products and the velocity in the product portfolio that has to be picked [49]. Automated order picking is mainly used when SKUs are small and uniform, as for example in the pharmaceutical industry [77].

In manual order picking there are two major systems, namely picker-to-parts and parts-to-picker. In the picker-to-parts system the picker walks or drives along the aisles to collect items. An automated storage and retrieval system is used in a parts-to-picker system, sending the items to the picker. Pickers pick requested items from ground level storage racks travelling along the aisles in low-level picking or using lifts or cranes to retrieve items from high storage racks in high-level picking.

Different variants of the picker-to-parts system are batch picking (picking by SKU) or discrete picking (picking by order). In batch picking, orders for multiple customers can be picked simultaneously, and sorting can be done immediately (sort-while-pick) or after picking (pick-and sort). Zone picking splits storage areas into multiple parts with different pickers. In progressive zoning, orders are passed on successively from one zone to the next, while in synchronised zoning, zones are picked in parallel [29].

Parts-to-picker systems mainly incorporate storage and retrieval systems in which one or multiple unit loads are retrieved by a crane and brought to the picking position. The crane can work under single, dual, and multiple command cycle modes. In other systems, modular vertical lifts or carousels are used to send unit loads to the pickers [29].

According to De Koster *et al.* [29], the majority of picking systems worldwide are low-level picker-to-parts systems employing human pickers, with 80% of that particular type in Western Europe. Therefore, an investigation in these systems is of practical interest. Van Gils *et al.* [116] suggests to combine order picking planning problems as all relations among the various problems are statistically significant.

1.3 Operations at a prominent South African retailer

In this dissertation, a prominent South African retailer is considered. This retailer (referred to in this dissertation as "the Retailer") is the biggest single brand retailer in South Africa with around 2 000 stores [77]. The company is a lifeline, mainly selling essential products (such as clothing, homeware and airtime vouchers) in rural and remote areas. The Retailer runs the largest clothing factory in South Africa, but also buys merchandise from local and international suppliers. There are four types of nodes to the logistics network at the Retailer: suppliers, distribution centres, transport hubs, and retail outlets. This supply chain is depicted in Figure 1.4. The distribution system consists of three DCs with the largest situated in Durban and two smaller ones that are located close to Cape Town and Johannesburg. The DC in Durban additionally sends stock for picking to the DC in Johannesburg [76]. The supply chain of the Retailer has to run efficiently to keep prices low, since it serves the low income part of the population.

The customer served by a DC primarily categorises the type of DC. In the case of the Retailer, a

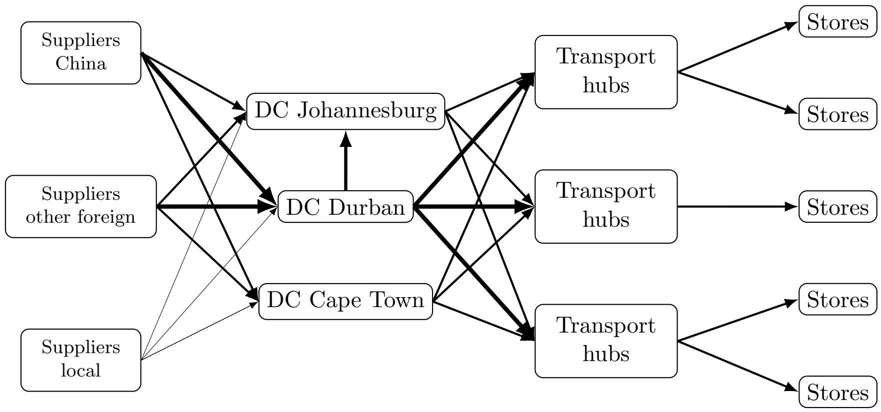


FIGURE 1.4: *A schematic representation of the Retailer's supply chain. Three DCs distribute to* 13 *transport hubs which serve a mutually exclusive set of stores. Stock movement is indicated by an arrows and the line thickness represents the relative stock movement. Source: Matthews [77].*

retail store is a regular customer of the DC, receiving shipments on a scheduled basis. Bartholdi and Hackman [7] describe the characteristics of a retail DC. High volumes of hundreds or even thousands of different products comprise a typical order. The changes in season, fashion, customer taste and marketing campaigns lead to a continuously changing composition of products. The Retailer is influenced by the nature of the products, since clothing arrives in a variety of bulky items which in most cases require large storage areas. Each branch has a different product profile depending on the market segment and location, resulting in a set of non-uniform orders that needs to be handled by the DC on a daily basis. The product profile changes constantly due to fashion trends and the seasonal nature of clothing products [76].

However, the biggest difference to retailers in the clothing industry is the Retailer's philosophy of central planning. The required stock for each store is defined by a central planning department, removing control of stock order from local stores and limiting the number of decisions made by local management [76]. This process is carried out for a set of SKUs called distributions (DBNs). DBNs consist of the same product type (for example white T-shirts), but incorporate different sizes (for example small, medium, or large) and quantities that should go to each store. Each size is identified by a unique SKU. The planners in the central planning department decide on the number of SKUs for each store upon their availability and issue DBN instructions to the DC. The DC then selects a subset of DBNs to be picked in a single picking wave. The SKUs within the DBNs present in a wave define the orders that must fulfil the store requirements. In this dissertation, an *order* is the SKUs that are required for a particular store in a picking wave. A wave is processed on a picking line in the DC. A single SKU is assigned to a unique location on a specific picking line for each wave. The activities of populating the line with SKUs, the actual picking process and removing excess stock from the line comprise a *picking wave* [77].

The order picking system of the Retailer with its overall processes, layout and physical picking process is influenced by the central planning approach. The DC continuously supplies a set list of stores (customers) and thus focuses on all store requirements for an individual item. Therefore, one operation entails picking and shipping, for all stores, for that item [76].

This dissertation will focus on the Retailer's DC in Durban, South Africa. The DC in Durban has the most flexible operations and processes the most products [76]. The DC in Cape Town and its specifications will also be investigated. Even though the data from the DC in Johannesburg is not explicitly used in this dissertation, its operations are similar to the other two DCs and thus the results and findings provide a holistic optimisation approach to all DCs.

The picking operation at the Retailer starts with the arrival of physical stock at the DC and the scheduling of DBNs to be sent to stores and the assignment of an out-of-DC date to each SKU by the central planning department. Even though all of the Retailer's DCs have a specific layout, each of them uses the same fundamental order picking framework. The Durban DC is the largest DC and will thus be described in detail with the main difference to Cape Town pointed out in the picking line area. A schematic layout is depicted in Figure 1.5. The main functional areas are receiving, storage, order picking, and distribution. The order picking facilities are situated left to the storage area. The storage and picking operations account for approximately 62 200 m^2 and are adjacent to the shipping operation with approximately 42 776 m^2 [76].

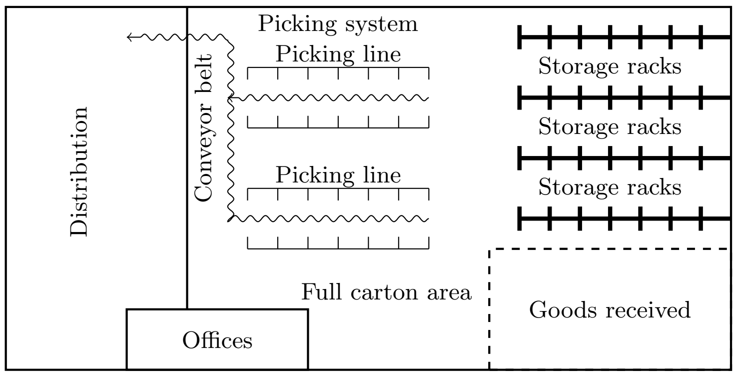


Figure 1.5: *A schematic layout of the Retailer's storage and picking operations in the Durban DC. Source: De Villiers [30].*

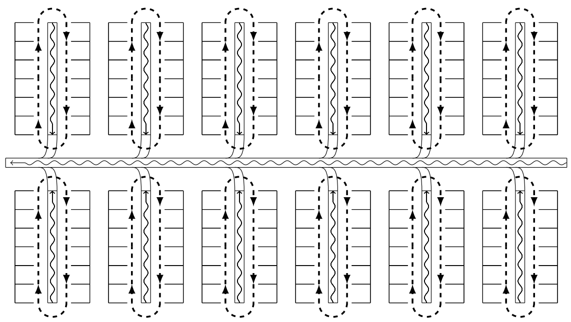
Stock arrives at one of the 15 loading bays and is loaded into the goods received area. After all quality checks have been completed, the loaded pallets are either moved to the floor or rack storage. Pallets stored in the full carton area will be picked in a carton picking operation, while stock in storage racks is intended for piece picking. Pallets destined for the Johannesburg DC are directly reloaded onto delivery vehicles and shipped off for further processing.

The storage area has 23 aisles that are serviced by five high lifts. Forklifts and pump trolleys are used to move stock in the floor storage spaces [77].

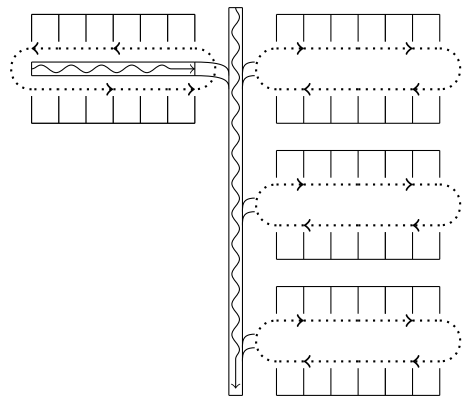
The Retailer utilises 12 unidirectional picking lines in the Durban DC as illustrated schematically in Figure 1.6(a). There are 56 locations for five pallets of an identical SKU per picking line [77]. In the Cape Town DC, the picking system includes one picking line on the floor with up to 144 locations, and a module with three picking lines on top of each other. This layout is depicted in Figure 1.6(b). Each picking line in the module consist of 64 to 76 locations. While the picking lines in the Durban DC and on the floor have conveyor belts in the middle, the picking lines in the module do not [57].

After the picking process, packed cartons are placed on the conveyor belt and arrive at the distribution area. As the size of the carton and the volume of stock varies, the cartons have to be resized. A quality control check is carried out on a sample. Closed cartons are then placed in buffer areas that are designated to specific transport hubs. A delivery vehicle is scheduled and stock is loaded as soon as a buffer area has reached a sufficient volume of stock. Typically, a store would receive three deliveries from a transport hub each week [77].

This dissertation focuses on the optimisation of the order picking system in the Retailer's DC. Therefore, the picking process in the specific picking line layout of the Retailer, namely the unidirectional cyclical picking line will be described in detail. The unidirectional carousel – a similar system discussed in literature – will be compared.



(a) *A schematic representation of a picking system with 12 picking lines in the Durban DC. Source: Matthews [77].*



(b) *A schematic representation of a picking system with four picking lines in the Cape Town DC.*

FIGURE 1.6: *On the left, a schematic representation of the picking system in Durban is presented. On the right, a schematic representation of the picking system in Cape Town is presented. In Cape Town, the picking line on the left is on the ground floor and has a conveyor belt. The three picking lines next to it are without conveyor belts. They are on top of each other and connected by a tunnel.*

1.3.1 Unidirectional cyclical picking lines

The unidirectional cyclical picking line is illustrated in Figure 1.7. At the Retailer's DC in Durban a picking line is made up of m locations with a conveyor belt that is placed in the middle. There are two gates to access the set up.

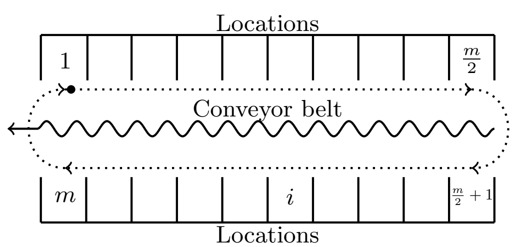


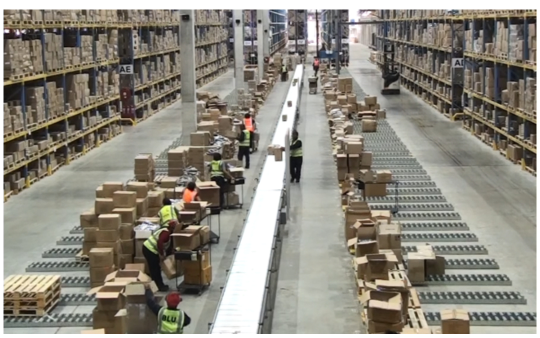
FIGURE 1.7: *A schematic representation of a picking line with m locations including a conveyor belt. Source: De Villiers [30].*

From the storage area, SKUs are assigned to each location of the picking line. Therefore, each SKU has a unique location on the picking line. The number of SKUs to be picked for all stores are known prior to the start of a picking wave. Each location has the storage capacity of up to five pallets of a single SKU. If additional stock is needed, it is kept on the floor space between the picking lines in a staging area to avoid stock outs. The restocking of picking lines can be eliminated from the considerations, since stock does not have to be replenished during a single wave of picking.

Pickers move around the conveyor belt in a clockwise direction picking the orders. Voice recognition software (VRS) guides pickers around the conveyor belt. Before starting an order, an empty carton is prepared by sticking a unique barcode to it and registering it with the VRS. The pickers reuse empty cartons from suppliers, stacking them on their trolleys. The picking process is thus not influenced by the availability of cartons. However, the capacity of an order may require more than one carton. The VRS directs a picker around the conveyor belt in a clockwise direction to fulfil the requirements of a single store in the shortest distance. The underlying picking strategy can thus simply be summarised as pick the closest SKU in a clockwise direction. A picker does not have to walk longer than one cycle around the conveyor belt to complete an order. Each order is completed by one picker, picking one order at a time. Space conditions allow for pickers to pass each other during the picking process. After picking, full cartons and finished orders are placed on the conveyor belt for further processing in the distribution area of the DC. Any excess stock is removed from the picking line after the picking wave is completed. This picking system used by the Retailer can be categorised as a discrete picker-to-parts system. [76, 77].

1.3.2 Comparison to carousel picking systems

A picking system from literature that is closely related to a unidirectional cyclical picking line is the unidirectional carousel system. In Figure 1.8 the two systems are shown. In the unidirectional cyclical picking line, pickers rotate around a conveyor belt. In the unidirectional carousel system, on the other hand, a set of shelves in which products are stored is fitted on a closed-loop rail. Upon request of a product, the carousel rotates on the horizontal axis until the shelf reaches the picker and thus the product can be retrieved by the picker. One or multiple pickers have to pick from one or multiple carousels in practice. Carousels are mainly used for small and medium-sized products, since they can easily fit into shelves [73].



(a) *The unidirectional cyclical picking line in the Cape Town DC.*

(b) *A unidirectional vertical carousel illustration. Source: Nicolas et al. [87].*

FIGURE 1.8: *Comparison between the unidirectional cyclical picking line and the vertical carousel.*

Similar to the picking process in the Retailer's DC, the picker picks products into a number of bins according to the number of orders. Therefore, each bin corresponds to only one order. The picker retrieves all items of the current order from the first shelf of the first carousel. Afterwards, the picker moves to the second carousel to pick all items of the order from the first shelf of the second carousel. The picker continues moving from one carousel to the next until all items from the first shelf of each carousel are collected. During this time, the first carousel rotates to present the next shelf in the carousel's opening. The picker then returns to the the first carousel to pick items from the second shelf. Similar to the initial collection step, the picker then moves from carousel to carousel to gather items from the second shelf. This process is repeated until all items of the order are picked. The case in which the items of the order have to be collected from only one carousel resembles the set up of the unidirectional cyclical picking line. The total completion time of the picking process can be divided into waiting time and picking time. The waiting time is dependent on the total number of items per order that have to be collected and on the time the carousel needs to rotate from one location to the next. The shelves that have to be visited determine the rotation time. Minimising the total completion time is thus equivalent to minimising the total travel distance of the carousel, since the total time for picking all orders and the time required for the rotation from one shelf to another are constant. Therefore, the objective is to minimise the total distance travelled by each carousel for that particular order [87, 88].

The main difference between this system and the unidirectional cyclical picking line is the presence of wave picking. This implies that all the SKUs on the line are at least picked once and all the orders are known *a priori* [78]. New orders may be added in the carousel system during the picking operation as more information about the orders becomes available. Therefore, carousel systems work with mixes of orders that are based on historical data. Bidirectional carousels are more common in practice than unidirectional carousels [73]. Additionally, a carousel system can only be operated by one picker, whereas multiple pickers pick on a unidirectional cyclical picking line [46].

1.4 Problem description

The main aim of this dissertation is to optimise order picking on a unidirectional cyclical picking line. In a picking line the main activities of a picker are walking to the next requested SKU, perform the picking process and handle the carton in which the SKU is placed. The total amount of time spent on picking and handling SKUs is constant during a picking wave. This renders the minimisation of travel time as the main objective when minimising the total picking effort. Furthermore, the minimisation of the total travel time is equivalent to the minimisation of the total length of all picker tours [64]. Therefore, minimising the walking distance is the focus of the optimisation problem, and the question this dissertation attempts to answer becomes:

Do changes in the organisation of the picking system increase the efficiency of a unidirectional picking line, and if so, to what extend?

According to Van Gils *et al.* [116] there are several methodologies available in literature that aim at minimising overall picking time such as routing, batching, storage location assignment, job assignment and zone picking. In their review of 62 articles, Van Gils *et al.* [116] observed that 42 articles focused on the topic of routing, 41 articles on batching, 30 articles on storage location assignment, 14 articles on job assignment, and 6 articles on zone picking with some articles combining several order picking planning problems.

In an attempt to optimise order picking on a unidirectional cyclical picking line, the methodology of routing in the form of sequencing orders in this cyclical set up has been investigated by Matthews and Visagie [78]. Order batching has not been introduced to a unidirectional cyclical pickling line system, but seems to be an effective way to minimise walking distance according to Van Gils *et al.* [116]. In order batching multiple orders are picked simultaneously by one picker [50]. Theoretically, this approach can divide the walking distance by the number of orders that are batched together. The only restriction becomes the capacity of the picking device that should accommodate all orders in the batch. The decrease in walking distance can be translated into a reduction in total picking time. Therefore, this methodology is introduced to a unidirectional cyclical picking line. Different structural configurations of the unidirectional cyclical picking line are investigated, since they influence the walking distance of pickers directly.

A known methodology (order batching) will be applied to a new picking environment. Additionally, the effect of order batching on different configurations of this picking system will be investigated extensively for the first time.

This research will only focus on the effects of the order batching implementation in the picking system. Therefore, other functional areas of the DC will not be addressed. The layout of the DC is fixed and will also not be investigated. Performance measures are productivity related, meaning the goal is to reduce the total picking time for a wave and seeing what the impact of this is on the DC's economic goals.

In the wave picking environment, Matthews [77] addressed sequential decision tiers that have to be tackled by the DC on a daily basis. Each decision tier will be referred to as a specific Tier. A picking module in the DC allows to change the configuration of the unidirectional cyclical picking line from a U-configuration to a Z-configuration since there is no conveyor belt in the middle of the line. This option will be included as an additional decision tier. Order batching is added as an extra layer to these decision tiers as depicted in Figure 1.9.

Each Tier defines an NP complete problem. Since they cannot be combined to solve simultaneously as it would become too computationally expensive, the four intractable problems are addressed sequentially. The results of the sequential optimisation approach by Matthews [77] show an improvement in walking distance.

In Tier 1, DBNs have to be assigned to picking lines in the order picking system. This is defined as the SKU to picking line allocation problem (SPLAP). After DBNs are allocated to a picking line, the SKUs have to be arranged such that each SKU is assigned to its own location on that picking line for that wave of picking. Tier 2 is referred to as the SKU arrangement problem (SAP) [76]. The configuration in which to operate the picking line, either U- or Z-configuration, has to be determined in Tier 3. This is the system configuration problem (SCP). Finally, the VRS needs to establish the sequence of orders that will be assigned to each picker before the picking operation starts [58]. Tier 4 is thus referred to as the order sequencing problem (OSP). Each decision tier is defined by the previous tier, as these decisions are made in sequence.

Even though the decisions are made and solved in this succession, optimisation models have to be developed in reverse order, since the lower tier is the objective function of the upper tier in Figure 1.9. For example, the effect of the SKU arrangement (Tier 2) on walking distance can only be evaluated once the configuration is chosen (Tier 3) and the orders are sequenced (Tier 4) [77].

Each decision tier will be described in more detail with the aim of minimising picking time by reducing walking distance. The extra layer of order batching is added to the decision flow and influences each tier directly or indirectly.

1.4.1 Tier 1: SKU to picking line assignment

The central planning department issues DBNs according to available stock at the DCs and the needs of the customers (stores). At the beginning of each day, these DBNs will be assigned to available picking lines. Thereby all SKUs of the same DBN will be assigned to the same picking line so that all sizes of a product arrive at the store at the same time. DBNs are ranked according to their out-of-DC dates. After the scheduling of DBNs, they have to be assigned to picking waves on the available picking lines. The stock is then brought to the respective picking line. As the store requirements for each DBN is known *a priori*, sufficient stock for all stores is brought to the line. Therefore, restocking during the picking wave is not required [77].

This forms Tier 1 or the SPALP. A set of DBNs containing SKUs has to be assigned to a set

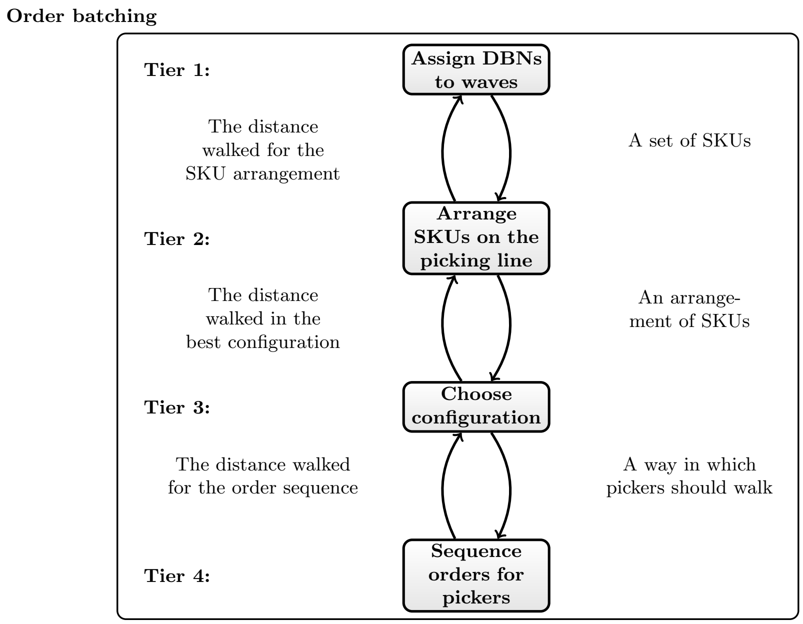


FIGURE 1.9: *A schematic representation of the information flow between the four decision tiers of a picking wave.*

of available picking lines to minimise walking distance and thus reduce the overall picking time. Orders can only be defined after they are assigned to a picking wave. Order batching approaches can thus not be applied directly, but previous experience concerning store requirements may support order batching decisions.

1.4.2 Tier 2: SKU arrangement

SKUs are arranged around the picking line during Tier 2. SKUs from the same DBN do not need to be assigned to locations adjacent to each other. Therefore, any SKU can be assigned to any location on the picking line. However, if the volume of the SKU is bigger than the capacity of the location multiple adjacent locations are assigned, but the VRS treats them as a single location [77].

This process forms Tier 2 or the SAP. A set of SKUs assigned to a picking line has to be allocated to available locations on the line to minimise walking distance. Orders have been formed at this stage of the decision process. The location of SKUs and thereby the number of locations that have to be passed to pick all items of the order are not defined yet. Order batching can not be introduced directly, since the necessary distance information is not available. Matthews and Visagie [82] have shown that the influence of the last decision tier outweighs the impact of Tier 2.

Tier 1 and 2 aim to minimise walking distance, but may influence the order batching problem indirectly. In Figure 1.10 the interactions between Tier 1 and 2 are illustrated.

1.4.3 Tier 3: System configuration

The orders that have been formed in Tier 1 together with the SKUs that have been assigned to unique locations in Tier 2 have to be picked. Before the start of the picking process, the configuration of the picking system has to be chosen in Tier 3. The unidirectional cyclical picking line may be placed in a picking module and without conveyor belt in the middle. Therefore,

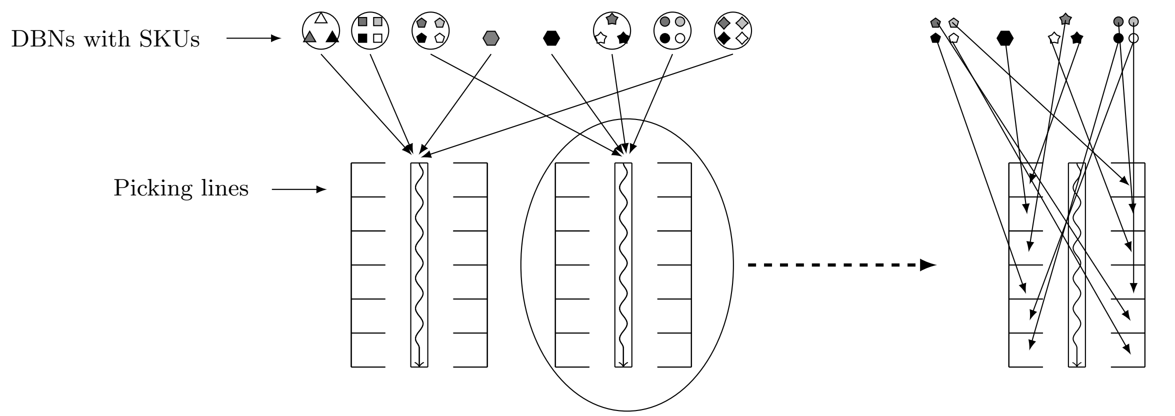
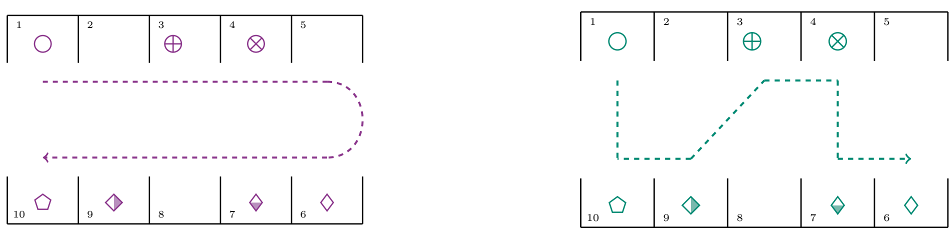


FIGURE 1.10: *A schematic representation of the Tiers 1 and 2. Tier 1 is shown on the left and Tier 2 is shown on the right. A DBN is represented by a shape. Each shaded shape represents a SKU. Different shades are SKUs of the same DBN. For display purposes only, shapes are grouped in Tier 1. Source: Matthews [77].*

pickers can either pick according to the U-configuration as depicted in Figure 1.11(a) or the Z-configuration as shown in Figure 1.11(b). Both configurations differ in the way in which pickers walk either along the locations (U) or with crossing the aisle (Z). This influences the walking distance of the pickers significantly.



(a) *A schematic representation of the U-configuration in the module.*

(b) *A schematic representation of the Z-configuration in the module.*

FIGURE 1.11: *Comparison of picker walking in U- and Z-configurations.*

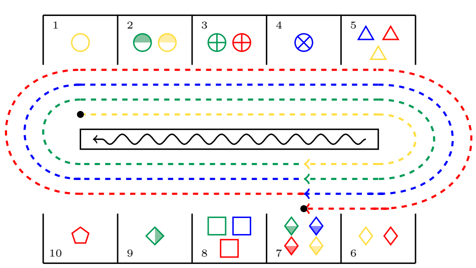
Tier 3 forms the SCP. At this point, SKUs have been assigned to locations. From the order information, distance approximations (relating to SKU locations) can be determined since they are not influenced by the picking configuration. However, note that walking distance approximations are dependent on the configuration. Therefore, the configuration choice may influence order batching directly.

1.4.4 Tier 4: Order sequencing

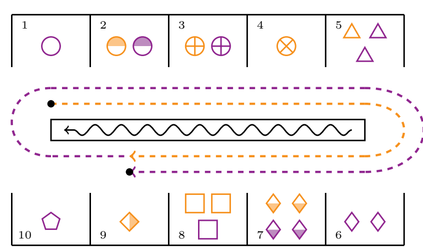
In Tier 4 pickers are guided to SKU locations by the VRS. The picking configuration influences how that movement takes place. Before moving on to the next order, a picker has to finish picking the current order. Adding the distances from start to end location of each picked order can be used to measure the total walking distance of pickers. However, there are several complexities in computing the walking distance. The end position of the last order determines the starting position of the next order thus influencing the length of the next order. Therefore, the walking distance to the next order is influenced by all preceding orders that are passed to a picker. It gives picking a stochastic nature. The presence of multiple pickers, and the dynamic addition or removal of pickers from the line, add to the complexities of the picking system. Therefore, the VRS must be able to dynamically adjust the assignment of orders to pickers while ensuring that the total walking distance is minimised [77].

Tier 4 forms the OSP. The overall picking time for a given picking wave with fixed SKU positions has to be minimised by sequencing all orders for the pickers. All the information requirements for including order batching, such as orders, SKU locations and configuration, are given. Depending on the number of orders that can be combined to a batch, order batching could directly influence walking distance.

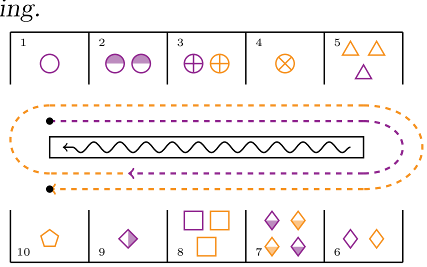
In an extreme case, batching two orders may reduce the walking distance by up to 50%. For example, four orders are picked in a U-configuration and, as illustrated in Figure 1.12(a), are indicated by the colours green, blue, yellow and red. If the orders are sequenced starting with yellow, then green, then blue and finally red, a picker would walk past 37 locations. If yellow and red are combined as a purple batch, and blue and green as an orange batch, then a picker would only have to pass 19 locations as depicted in Figure 1.12(b). Similarly, if yellow and green are combined to form a purple batch, and red and blue as the orange batch, as illustrated in Figure 1.12(c), then the purple batch would be collected followed by the orange batch, resulting in 20 locations. If yellow and blue are batched together as purple, and red and green as the orange batch, as depicted in Figure 1.12(d), then only 18 locations have to be passed by the picker. The length of a picking path depends on the composition of batches and their sequencing. Order batching may thus have a direct influence on the last decision tier. Similar orders should form batches to minimise walking distance. Therefore, the similarity between orders in terms of walking distance has to be determined to introduce order batching to the unidirectional cyclical picking line.



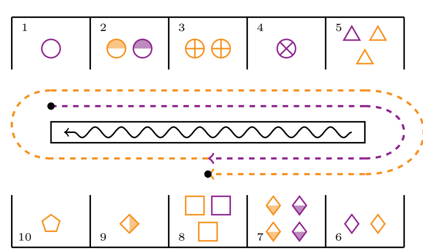
(a) *A schematic representation of single order picking.*



(b) *A schematic representation of batch picking yellow and red.*



(c) *A schematic representation of batch picking yellow and green.*



(d) *A schematic representation of batch picking yellow and blue.*

FIGURE 1.12: *Comparison of single order picking and batch picking for different compositions of batches.*

Tiers 3 and 4 may have a direct influence on order batching, since the configuration of the system and the order sequence influence walking distance significantly. Order batching requirements such as orders and SKU locations are given at this point in the decision making process.

1.5 Objectives

This research aims to optimise the order picking system of a unidirectional cyclical picking line. The methodology of order batching is introduced to the unidirectional cyclical picking line to further reduce walking distance and thus to minimise overall picking time. Furthermore, the influence of the configuration choice on the walking distance is investigated. All solution approaches are tested on data that has been provided by the Retailer and organised in a test framework. The research question translates into the following six main objectives that are further supported by subobjectives. Each objective is part of the structure of the dissertation.

Objective I: Investigate the order picking system on a unidirectional picking line:

a Describe the layout and operations of the Retailer's DCs to comprehend the broader context of the problem;

b Describe the order picking system in detail to emphasise the characteristics of a unidirectional cyclical picking line;

c Describe the different configurations of the order picking system.

Objective II: Perform a literature study:

a Describe the optimisation approaches on a unidirectional cyclical picking line;

b Describe the standard order batching problem and its solution approaches;

c Identify the differences (in layout) to a unidirectional cyclical picking line.

Objective III: Apply order batching to a unidirectional cyclical picking line:

a Model order batching on a unidirectional cyclical picking line;

b Emphasise the specific layout in the order batching approach.

Objective IV: Build a simulation of the order picking system to measure picking time:

a Build a tool that can measure picking time by simulating the U-configuration of the unidirectional cyclical picking line;

b Build a tool that can measure picking time by simulating the Z-configuration of the unidirectional cyclical picking line;

c Identify a predictor for picking time;

c Determine an indicator for the selection of configuration.

Objective V: Apply order batching as an extra layer to all four decision tiers:

a Apply order batching to Tier 1;

b Apply order batching to Tier 2;

c Apply order batching to and define Tier 3;

d Apply order batching to Tier 4.

OBJECTIVE VI: Propose directions for future research:

a Summarise the contributions of this dissertation;

b Give recommendations to practitioners;

c Propose further optimisation ideas for a unidirectional cyclical picking line.

1.6 Research methodology

The research question of this dissertation addresses the optimisation of an order picking system on a unidirectional cyclical picking line. Data from the DCs of a prominent South African retailer will be processed. Therefore, the order picking system on a unidirectional cyclical picking line will be evaluated on a quantitative basis to answer the research questions. The order batching methodology from literature is introduced as the main optimisation approach on a unidirectional cyclical picking line. It is chosen as it claims to reduce walking distance and thus decreases picking time significantly [116].

In the secondary phase of this dissertation, material of different resources such as academic journals, books and case studies dealing with the topic of order batching are collected, analysed, categorised, and evaluated. Thereby, the current body of scientific knowledge is described, and the characteristics of the Retailer's order picking system are described. A simulation model of the order picking system is built to investigate the order picking system in detail in the primary research. Historical data provided by the Retailer will be used as input. The best order picking performance in terms of the shortest overall picking time can be determined by these simulation experiments.

The research design divides the introduction of order batching to a unidirectional cyclical picking line into three subproblems. In the first subproblem, order batching approaches are introduced to the new layout. The characteristics of the unidirectional cyclical picking line are used to define distance approximations for this specific set up. Mathematical modelling is used to determine good batches thereby reducing walking distance. In the second subproblem, a simulation model of the unidirectional cyclical picking line is used to investigate the interactions between the main entities in the system. The simulation model helps to confirm the assumption that a reduction in walking distance decreases picking time. It can be used as a tool for both configuration options, since the reduction is measured in time and not in distance which is configuration dependent. The third subproblem models each decision tier and investigates the inclusion of order batching. The first subproblem results in two articles, the second subproblem generates two articles and the third subproblem concludes the topic in one article. Each article reviews the literature on the particular problem, models the problem, and discusses the results of the model. Optimisation strategies will include changes in the organisation of the picking system. Finally, a comparison to a benchmark scenario on the effectiveness of these changes answers the research question. The Retailer's input data for each model is described in the following section.

1.7 Data and test framework

Different test instances are needed to evaluate the introduction of order batching to the unidirectional cyclical picking line. The decision tiers have different time horizons in which decisions for each tier have to be made. For example, the SPALP is solved daily, while the SAP, SCP and OSP have to be solved for each picking wave. Therefore, the dataset for each problem will be described separately.

All algorithms of this dissertation are provided in pseudocode. The Python code for each algorithm can be found in the GitHub repository [54].

1.7.1 Test instances for order batching

Order batching is resolved in the last decision tier and aims to develop order proximity measurements that include the specific picking line layout to minimise walking distance. Therefore, the information needed are the store requirements (orders) and the locations of the SKUs on the picking line. A variety of numbers of orders and SKUs has to be provided by several picking waves to test the developed order proximities for average and extreme cases.

The DC in Durban is the biggest and most flexible of the Retailer's DCs. Sample picking wave data for this DC is available online [79]. All problem instances are derived from this historical dataset. Therefore, 50 sample picking waves are extracted from the online dataset and categorised according to the number of orders and SKUs.

The large dataset includes picking waves with more than 1 000 orders and is split into a category of a large number of SKUs ($54 - 56$), a medium number of SKUs ($49 - 52$) and a small number of SKUs ($43 - 47$). The medium dataset consists of $101 - 1\,000$ orders and has between $55 - 56$ SKUs. The smallest dataset only consists of 100 orders or less and is split into a large number of SKUs ($50 - 52$), a medium number of SKUs ($41 - 42$) and a small number of SKUs ($36 - 37$). Table 1.1 shows the number of sample waves per test instance with the bulk of test instances in the large dataset representing average cases and more extreme cases in the medium and small dataset.

Dataset	**Number of orders**	**Number of SKUs**	**Number of sample waves**
Large dataset	1 001 – 1 500	54 – 56	14
Large dataset	1 001 – 1 500	49 – 52	12
Large dataset	1 001 – 1 500	43 – 47	12
Medium dataset	101 – 1 000	55 – 56	6
Small dataset	6 – 100	50 – 52	2
Small dataset	6 – 100	41 – 42	2
Small dataset	6 – 100	36 – 37	2

Table 1.1: *The composition of test instances from historical data of the Durban DC available online.*

1.7.2 Test instances for picking time simulation

The Cape Town DC does not only have a unidirectional cyclical picking line on the floor, but also an order picking module. While the picking line on the floor corresponds to the one in Durban (even though the picking line in Cape Town can accommodate with up to 144 SKUs more SKUs than Durban), the picking lines in the module do not have a conveyor belt in the middle. Without the conveyor belt, a different option to the U-configuration, namely the Z-configuration in which pickers can move across aisles to get to the next locations, can be applied. Therefore, 30 sample picking waves including 10 waves for each configuration (U-configuration with conveyor belt, U-configuration without conveyor belt, Z-configuration without conveyor belt) have been recorded as test instances.

As depicted in Table 1.2 the three configurations, namely unidirectional cyclical picking line on the floor (U floor), unidirectional cyclical picking line in the module (U module) and Z-configuration in the module (Z module) mainly have test instances including a large number of orders (average cases) and some smaller numbers of orders to test the influence on extreme cases.

Configuration	Dataset	Number of orders	Number of sample waves
U floor	Large dataset	1 701 – 2 000	5
U floor	Medium dataset	1 101 – 1 700	2
U floor	Small dataset	101 – 1 100	3
U module	Large dataset	1 581 – 2 000	5
U module	Medium dataset	1 551 – 1 580	3
U module	Small dataset	101 – 1 550	2
Z module	Large dataset	1 581 – 2 000	5
Z module	Medium dataset	1 551 – 1 580	3
Z module	Small dataset	101 – 1 550	2

TABLE 1.2: *The composition of different configurations from historical data of the Cape Town DC.*

1.7.3 Test instances for all decision tiers

Data concerning DBN information (that can be assigned to two or more picking lines) needs to be available to test solutions that introduce order batching. Multiple days of picking waves are used to test the effect of DBN assignment on Tier 1. The store requirements are used to define orders and assign SKUs to locations on the available picking lines in Tier 2. The test instances should be comparable to historical assignment methods.

The Cape Town DC allows for different configurations of the picking lines and can thus address Tier 3. Therefore, all picking waves of one month are used to test the influence of the holistic optimisation approach including order batching and Tier 4. This results in 66 picking waves that are tested in different scenarios.

The DC in Cape Town has between two to four picking lines available to run in parallel. Therefore, three scenarios according to the number of available picking lines have been generated and are depicted in Table 1.3. The number of orders is not displayed, since it is dependent on the solution of the SPALP. These scenarios will be used to test a holistic optimisation approach addressing all four decision tiers and including the additional layer of order batching (directly or indirectly).

Scenario	Number of lines	Number of days	Total DBNs	Total SKUs	Number of sample waves
1	2	27	767	2 887	30
2	3	21	708	2 510	24
3	4	9	330	1 283	12

TABLE 1.3: *The composition of three test scenarios from historical data of the Cape Town DC.*

1.8 Dissertation organisation

The chapter outline will correspond to the research objectives of the dissertation. The core of this dissertation comprises of 5 papers which are in various stages of the publication process – from submitted for review to published. Due the the main chapters (Chapters 3 to 7) being standalone papers as well, necessary overlap in the introduction and background sections of these chapters exist.

Besides this introductory chapter, Chapter 2 provides an in-depth literature review on the topic of order batching. The standard order batching problem is discussed and solution approaches from literature are presented. The difference in layout between a single-block parallel aisle warehouse environment and the unidirectional cyclical picking line are pointed out. The related unidirectional carousel system and its order batching approaches are described in detail.

In Chapter 3 order batching is introduced to the unidirectional cyclical picking line for the first time. Therefore, distance approximation measures are adapted. Three picking location based metrics are suggested and several heuristic and metaheuristic combination approaches are tested on historical data from the Retailer's DC in Durban. This chapter is published in the Operations Research Society of South Africa's journal ORION [55].

The specific layout of the unidirectional cyclical picking line is taken into account in Chapter 4. Similar to the picking location metrics, three route overlap based addition metrics are developed. The route overlap includes the information about the span of an order which includes all the locations stopped at or passed while collecting items for the order. The picking location and route overlap addition metrics are compared on test instances of the Retailer's Durban DC. This chapter has been submitted to a peer reviewed scholarly journal.

In Chapter 5 the unidirectional cyclical picking line is modelled as a discrete event simulation. With the help of the simulation it can be determined if order batching can also decrease picking time in addition to reducing walking distance. A time study was carried out in the Retailer's DC in Cape Town. The historical data of the Cape Town DC is used to verify and validate the simulation model. Additionally, a good number of pickers for the picking system is determined. Different pick density measures are also introduced and tested to predict picking time. This chapter is accepted for publication in the International Journal of Logistics Systems and Management [59].

The two different configurations U and Z are introduced to the unidirectional cyclical picking line in Chapter 6. The Cape Town DC's picking module does not have a conveyor belt in the middle of the system and thereby allows for a configuration change. Therefore, the discrete event simulation is extended to incorporate the option to simulate the Z-configuration. The configurations can easily be compared, since both measure picking time. Additionally, an indicator to help decide which configuration to choose is suggested. This chapter has been submitted and is under review at a peer reviewed scholarly journal.

In Chapter 7 all four decision tiers are addressed with the extra layer of order batching on a unidirectional cyclical picking line. A recommendation for the holistic picking system optimisation is provided. This includes when and how to introduce order batching. All solution approaches are tested and compared against a benchmark scenario using the data provided by the Retailer's DC in Cape Town. The chapter will also be submitted for publication.

Chapter 8 summarises the dissertation and draws conclusions from the results of the experiments. This leads to recommendations for practitioners working with a unidirectional cyclical picking line. This chapter also makes recommendations for future research in this area.

Order batching reduces walking distance and has the potential to decrease picking time. It has been applied in the standard layout of a single-block warehouse with parallel aisles. The current body of scientific knowledge on solution approaches in the standard environment is presented and an outlook on the introduction of order batching to a unidirectional cyclical picking line is provided in the next chapter.

CHAPTER 2

Literature review

Contents

A review of 62 order picking optimisation articles by Van Gils *et al.* [116] showed that articles aiming at minimising the overall order picking time focus on the methodologies of routing (42 articles), order batching (41 articles) followed by storage location assignment (30 articles) or a combination of multiple solution approaches.

Optimisation approaches on a unidirectional cyclical picking line have first been developed by Matthews and Visagie [78, 80, 81, 82] and focus on order sequencing (routing in the cyclical set up) and storage location assignment. In this dissertation, order batching is introduced as it efficiently minimises picking time according to Van Gils *et al.* [116].

Combining customer orders into picking orders describes the process of order batching. The number of orders that can be batched is restricted by the capacity of the picking device. Splitting orders is usually prohibited, since it would require additional sorting [50].

According to Wäscher [120] order batching answers the question of how customer orders should be grouped into picking orders to minimise the total length of all picking tours necessary to collect all items while no customer order is split, given a set of customer orders consisting of a number of items, an assignment of items to storage locations in the order picking system of a DC, and the capacity of a picking device.

In the following chapter optimisation approaches on a unidirectional cyclical picking line are described in Section 2.1. This is followed by an in depth discussion of different solution approaches to the order batching problem (OBP). The challenges of introducing order batching to a unidirectional cyclical picking system are pointed out. An exact solution is provided in Section 2.2, heuristics in Section 2.3, and metaheuristics in Section 2.4. Section 2.5 concludes this chapter and points out potentials for order batching on a unidirectional cyclical picking line.

2.1 Optimisation approaches on a unidirectional picking line

Matthews and Visagie [78] introduced the optimisation idea of routing to a unidirectional cyclical picking line. Since pickers walk in one direction pushing their picking devices around the conveyor belt, the route is predetermined in this set up. Therefore, they investigated the problem of sequencing orders for pickers for a single picking wave, given a fixed arrangement of SKUs. They assumed that the time required to pick stock and to handle cartons is fixed rendering walking distance the minimisation objective. They proposed a maximal cut approach which yields in the shortest walking distance in the test instances of their data set. Furthermore, they suggested a nearest ending heuristic as an easily implementable alternative.

The arrangement of SKUs on a picking line for a wave of picking addresses the methodology of storage location assignment. Matthews and Visagie [82] tested two heuristics which are optimal in a carousel system environment and two heuristics which incorporate SKU correlations. Their experimental results showed that addressing the order sequencing problem outweighs the gain in walking distance from addressing the SKU arrangement problem on a unidirectional cyclical picking line.

Matthews and Visagie [80] also introduced the optimisation idea of storage location assignment to the level of assigning DBNs to picking lines. They applied the maximal SKU measure as an approximation of the actual cycles traversed. Minimising the sum of the sizes of the maximal SKUs for each picking line, they suggested a phased greedy insertion approached that resulted in the shortest walking distance.

To mitigate negative effects on volume distribution and the number of small cartons, Matthews and Visagie [81] incorporated four correlation measures in the phased greedy insertion approach to assign DBNs to picking lines. Based on the experimental results, they recommended the desirability score that considers the number of stores required by the candidate DBN and that requires at least one DBN that has already been assigned to the picking line.

2.2 Exact solution approaches to order batching

The formulation of the standard OBP was introduced by Gademann and Velde [36] for the layout of a single-block warehouse with parallel aisles. Assigning orders to batches simultaneously and calculating a picking tour for each batch to optimise an objective function describes the OBP. The objective function can either reduce the total travel time (proximity batching) or increase due-date performance (time-window batching). The OBP to minimise total travel time in a parallel aisle warehouse layout thus addresses the fundamental proximity batching problem.

The picking system considered by Gademann and Velde [36] is presented in Figure 2.1. The rectangular single-block DC contains parallel aisles of equal length and width. Aisles are connected via cross-aisles at the front and back of each vertical aisle. Therefore, every picking tour starts and ends at an input or output station that is located at the horizontal aisle in front. There are storage locations on both sides of the vertical picking aisles.

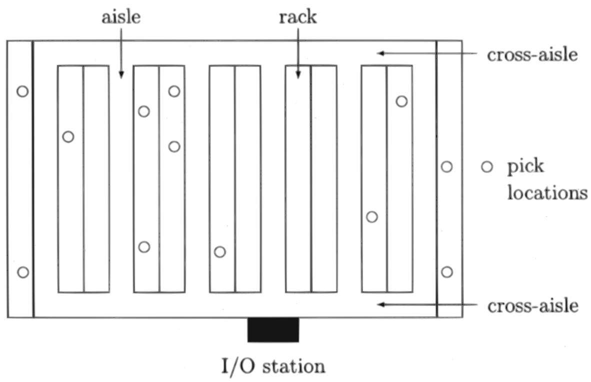


FIGURE 2.1: *A schematic layout of a single-block warehouse with parallel aisles. Source: Gademann* and *Velde [36].*

Pickers pick items manually into separate bins on their picking devices according to order lists. Each bin belongs to a single order in this case. It is assumed that a complete order fits into a bin. Hence the strategy of sorted-while-picked is applied. The number of orders in a batch are limited by the number of bins that can fit onto a picking device. Picking time includes travel time from one picking location to the next, and an extraction time for administering the order. The underlying routing strategy has to be taken into consideration while calculating the travel distance in a single-block warehouse with parallel aisles environment. The extraction time is assumed to be independent of the batch composition, forming a constant that can thus be omitted from solving the optimisation problem. The velocity of a picker is assumed to be constant resulting in a linearity between travel time and distance. Therefore, a tour of minimum length is the optimal tour for a given batch [36].

Gademann and Velde [36] proved that the OBP is NP-hard in the strong sense. However, if no batch contains more than two orders the problem can be solved in polynomial time. If the OBP is formulated as a set partitioning problem, as proposed by Gademann and Velde [36], let

- $\mathcal{J}$ be the set of orders with elements $j = 1, 2, \ldots, j, \ldots, J$,
- $\mathcal{B}$ be the set of feasible batches with elements $1, 2, \ldots, b, \ldots, B$,
- $\boldsymbol{y}_b$ be the J dimensional vector representing a feasible batch, with $\boldsymbol{y}_b = (y_{b1}, \ldots, y_{bj}, \ldots, y_{bJ})$, where $y_{bj} = 1$, if, and only if, order j is included in batch b,
- λ_b be the total distance to pick all items of orders in batch b,

and define the set of variables as

$$x_b = \begin{cases} 1 & \text{if a batch } b \text{ is chosen,} \\ 0 & \text{otherwise.} \end{cases}$$

The objective is then to

$$\text{minimise} \sum_{b \in \mathcal{B}} \lambda_b \, x_b \tag{2.1}$$

subject to

$$\sum_{b \in \mathcal{B}} y_{bj} \, x_b = 1 \qquad j \in \mathcal{J}, \tag{2.2}$$

$$x_b \in \{0, 1\} \qquad b \in \mathcal{B}. \tag{2.3}$$

The total travel distance is minimised by the objective function (2.1). Constraint (2.2) and the integrality condition of constraint (2.3) ensure that each order is allocated to only one batch.

The branch-and-prize algorithm developed by Gademann and Velde [36] solved test instances of up to 32 orders to optimality in 2005. The objective of their algorithm is to minimise the total distance to be travelled to pick all items of the orders in one batch. Therefore, distance measurements, depending on the probabilities of a location being visited (determined by the storage policy) were employed. Additionally, Henn and Wäscher [51] solved instances of up to 40 orders. The routing strategies S-shape and largest-gap are considered in their study. Their solution approach required a time-consuming preprocessing of all feasible batches. Muter and Öncan [86] applied traversal, return, and midpoint as routing strategies in the parallel aisles warehouse layout. Their objective is defined by minimising the total travel cost per picker so that each order forms part of exactly one batch. Instances of up to 100 orders were solved to optimality with their tailored column-generation based algorithm. In practice the number of customer orders is likely to exceed the test instances, but the exact algorithms proposed so far are not able to consistently solve larger instances. Additionally, the order batching should allow for batch sizes bigger than two orders. Thereby, the development of heuristics to solve practice-oriented problems is justified.

The layout of a single-block warehouse to which the standard OBP has been applied is significantly different from the Retailer's DC with unidirectional cyclical picking lines. The vertical carousel system applied in Nicolas *et al.* [87] shows similarities in terms of cyclical movement to the picking line set up investigated in their study. Additionally, the location of SKUs resemble the shelves of a carousel. Therefore, the introduction of the OBP to a vertical carousel system by Nicolas *et al.* [87] is described.

The OBP can be formulated as a mixed integer linear program (MILP), as proposed by Nicolas *et al.* [87]. Therefore, to retrieve all orders of a batch from one bidirectional carousel, let

$\mathcal{B}$	be the set of all batches with elements $1, 2, \ldots, b, \ldots, B$,
B_{max}	be the maximum batch size,
$\mathcal{J}$	be the set of all orders with elements $j = 1, 2, \ldots, j, \ldots, J$,
$\mathcal{Q}$	be the set of all shelves with elements $q = 1, 2, \ldots, q, \ldots, Q$,
Q^b	be the number of shelves to be visited for batch b,
$\mathcal{Q}^b(h)$	be the number of the h-th shelf that the carousel has to visit with $h = 1, \ldots, Q^b$,
$\boldsymbol{z}_b$	be the vector of size Q^b including the numbers of the shelves visited during batch b in increasing order such as $z_b(1) < z_b(2) < \ldots < z_b(Q^b)$,
λ^b	be the minimum distance travelled by the carousel in order to collect items from all shelves that have to be visited for batch b,
λ^b_q	be the distance between shelf q and the closest shelf with a lower number than q,
λ^b_h	be the distance between the h-th shelf and the shelf of $h-1$,
I_{qj}	be an input parameter of 1, if, and only if, order j visits shelf q,

and define the set of variables as

$$x_j^b = \begin{cases} 1, & \text{if an order } j \text{ is included in batch } b \\ 0, & \text{otherwise,} \end{cases}$$

and

$$V_q^b = \begin{cases} 1, & \text{if shelf } q \text{ is visited by batch } b \\ 0, & \text{otherwise.} \end{cases}$$

The objective is then to

$$\text{minimise} \sum_{b \in \mathcal{B}} \lambda^b \tag{2.4}$$

subject to

$$\sum_{b \in B} x_j^b = 1 \qquad j \in \mathcal{J}, \tag{2.5}$$

$$\sum_{j \in J} x_j^b \leq B_{max} \qquad b \in \mathcal{B}, \tag{2.6}$$

$$V_q^b = \sum_{j \in J} x_j^b \cdot I_{qj} \qquad b \in \mathcal{B}, q \in \mathcal{Q}, \tag{2.7}$$

$$\lambda^b = Q - \max_q \lambda_q^b \qquad b \in \mathcal{B}, \tag{2.8}$$

$$\text{if } Q^b = \sum_{q \in Q} V_q^b = 1 \quad \rightarrow \quad \lambda^b = 0 \qquad b \in \mathcal{B}, \tag{2.9}$$

$$\text{if } Q^b = \sum_{q \in Q} V_q^b > 1 \quad \rightarrow \quad \lambda_h^b = z_b(h) - z_b(h-1) \qquad b \in \mathcal{B}, q \in \mathcal{Q}, h = 2, \cdots, Q^b, \tag{2.10}$$

$$\lambda_1^b = Q - z_b(Q^b) + z_b(1), \tag{2.11}$$

$$x_j^b \in \{0, 1\} \qquad b \in \mathcal{B}, j \in \mathcal{J}, \tag{2.12}$$

$$V_q^b \in \{0, 1\} \qquad b \in \mathcal{B}, q \in \mathcal{Q}. \tag{2.13}$$

The objective function (2.4) minimises the total distance travelled by the carousel. Each order is assigned to only one batch through constraint (2.5). The maximum batch size is set by constraint (2.6). Equation (2.7) computes V_q^b if at least one order of batch b has to visit shelf q. The minimal travel distance is calculated by equations (2.8) – (2.11) for each shelf. In the case of all orders being located on one shelf, no distance needs to be travelled by the carousel. This is expressed in constraint (2.9). However, if the number of shelves to be visited is bigger than one, all the shelves containing requested products have to be visited and returned as in constraints (2.10) and (2.11) accounting for the total travel distance. The binary conditions of the two sets of variables are described in (2.12) and (2.13).

The exact solution approach of Nicolas *et al.* [87] shows considerable savings in completion time. Nevertheless, the batching of only 50 orders has to be stopped after 30 minutes, since it would take too much time to solve otherwise. Heuristics could help to reduce the computational time and thus make an industry application feasible.

The number of possible batches and thus the binary decision variables increase exponentially with the number of customer orders. Therefore, an exact solution approach involving the application of a commercial linear programming (LP) solver to an explicitly formulated optimisation problem as in (2.1) or (2.4) only covers a limited range of problem instances with regards to the number of customer orders. Heuristics and metaheuristics, that will be discussed in the following sections, are applied to the OBP to solve real world problem instances [50].

2.3 Heuristics

Larger problem instances can only be solved using heuristics. Therefore, heuristics like priority rule-based, seed algorithms, savings algorithms, and data mining approaches have been introduced to solve the OBP. The different algorithms are described in the following section, while specifications of the set up of the applications are pointed out.

2.3.1 Priority rule-based algorithms

In priority rule-based algorithms priorities are assigned to customer orders which are allocated to batches, while ensuring that the capacity constraint is met. The rules can include availability, distance metrics, and sequential or simultaneous assignment of orders to batches.

A straight-forward heuristic is the first-in-first-out method that groups together the first entries of the list of orders as close as possible to the predetermined maximum batch size. Then the next orders are grouped using a similar logic until all orders form part of a batch [28, 37].

Gibson and Sharp [37] consider different distance measurements such as the two-dimensional space filling curve, where a point on a unit circle is continuously mapped onto a unit square. Therefore, items which are close to each other can generate values which are close in magnitude, making relative proximity assessments of groups of possible items. The sorted list of orders is then used as an input to generate batches close to the desired batch size. The four-dimensional space filling curve is applied to define the smallest rectangle that encloses all items of an order. Except for the inverse mapping that takes points in four-dimensional space in one step onto the unit circle, the idea of the four-dimensional is similar to the two-dimensional space filling curve [9]. The sequential minimum distance batching heuristic is applied to measure the distances between the item locations of one order to the item locations of another order. Factors that influence the evaluation of batching are: the Euclidean, rectilinear, Chebyshev, and aisle travel metrics and their respective tour sequencing; the continuous and discrete storage region representation; the uniform and skewed assignment of item locations; the fixed and variable number of items per order; and the number of problem instances. Gibson and Sharp [37] found that in a typical DC layout with aisle metric the sequential minimal distance heuristic is the most effective in terms of tour length and execution time.

Ruben and Jacobs [99] investigated several batch heuristics that assign orders according to the best- or first-fit rule. The first-fit-decreasing algorithm begins with sorting the list of orders in a non-increasing order according to the number of items per order. Initially a first batch is created and the first order is assigned to this batch. The list of previously created batches is then scanned by every other order until a batch with sufficient capacity is found to accommodate the order. This process terminates when all orders are allocated to batches. The first-fit envelope based batching heuristic incorporates order location and order size information. The term *order envelope* describes the minimum and maximum number of aisles in which an item of an order is located. The distinct order envelopes, whose numbers depend on the number of aisles in the DC, are indexed and considered sequentially according to the first-fit mechanism. Additionally, the first-fit based batching works similar to the first-fit envelope based algorithm with the exception that the orders are partitioned into classes. The classes correspond to different areas in the DC emphasising the capacity utilisation aspect of the optimisation problem. The experimental results of Ruben and Jacobs [99] showed that the first-fit envelope based batching heuristic performs best especially with regards to cost effective order picking and full utilisation of picking device capacity. On the other hand, the next-fit rule assigns orders sequentially to batches. Therefore, orders are added to the batch until the capacity of the picking device is reached and a new batch is started [50].

2.3.2 Seed algorithms

Seed algorithms start by initiating batches, then allocating orders to batches and terminate through a stopping rule when a batch has been completed. The objective is to minimise the total travel distance for collecting all orders.

De Koster *et al.* [28] evaluated seed algorithms in a parallel aisle warehouse layout proposing the seed selection rules of the farthest storage location with respect to the drop-off location, the largest number of aisles a picker has to enter to collect the order, the longest travel time, and the largest aisles range, which is determined by the absolute difference between the number of the aisles entered from the left and the number of aisles entered from the right. Time saving for which the reduction in travel time is maximal when added to the batch is combined with the possible order addition rules. De Koster *et al.* [28] found that their experimental setup generated the best batches when there was a large number of aisles, or a large number of orders, or conversely, a small number of aisles that could facilitate the addition of new orders. In general, seed algorithms performed best with a S-shape travel metric and a large picking device capacity.

Ho and Tseng [53] investigated several location or aisles-based seed algorithms in the standard OBP environment of a single-block parallel aisle warehouse. They proposed the following additional seed selection rules. A seed can be selected by the smallest or greatest location-aisles ratio, which is deducted by the number of locations to be visited over the number of aisles to be entered, or the aisle-weight sum. The aisles-weight is equal to the aisles-index and thus increases with distance from the drop-off location. The minimum number of additional picking locations or aisles, the maximum number of identical picking locations or aisles, the greatest picking location or aisle similarity ratio, and the greatest identical location to additional aisle ratio can be applied to add another order to the batch. The cumulative seeding rule updates the seed at each step. Additionally, the route planning methods of largest-gap and largest-gap with simulated annealing are considered and two different sets of problem instances are generated to test the influence of the aisle-picking frequency. Ho and Tseng [53] concluded that the combination of the smallest number of picking aisles seed rule and the smallest number of additional picking aisles addition rule generate the best results in their experimental set up. In an extension, Ho *et al.* [52] proposed further distance- and area-based selection rules. Therefore, seed selection rules include the smallest or greatest rectangular-covering area by which all storage locations of an order are covered, and the shortest average rectangular, Euclidean or aisle distance to the drop-off location. The smallest or greatest overlapping covering area, the smallest or greatest additional covering area, the ratio between overlapping area to total cover, the ratio between identical pick-up locations to additional covering area, and the shortest average mutual-nearest-rectangular, -Euclidean or -aisles distance, and the smallest weighted aisle-index difference are taken into consideration for adding an order to a batch. Ho *et al.* [52] showed that the combination of the smallest rectangular covering area rule with the distance based rule for order addition performs best in their experiments.

In their comparative study Pan and Liu [92] evaluated four initial seed selection and four order addition rules, since most seed algorithms are composed of the major steps of selecting seeds and adding orders to a batch until the capacity limit is reached. In this case, the OBP is investigated in an automated storage and retrieval system consisting of a single storage rack with equally sized storage locations, serviced by a single storage and retrieval machine.

Elsayed and Stern [33] started with no orders in a batch initially and then introduced the cumulative and the single seeding rule. While the single seeding rule stays with the initial seed until the batch is completed, all the orders in the batch are part of the new seed that is thus updated by the cumulative seeding rule. Their results showed that the cumulative seeding rule shows better results in minimising travel distance.

One of the following selection rules chooses a seed and defines the seed as the first order of the batch in the study of Pan and Liu [92]. An order with the largest number of items, an order with the greatest total weight of items, an order with the largest economic convex hull among all orders, or an order with a six-dimensional space filling curve value that has the smallest deviation from zero can be selected as the seed. Hwang and Lee [62] defined the region in which a storage and retrieval machine moves (without increasing undesirable travelling time) as the economic convex hull. For each order the boundary points of the associated convex hull are defined by a different similarity coefficient. This coefficient is either the maximum order cardinality or the maximum order quantity rule. In the six-dimensional space filling curve method the concentration degree of item locations was taken into consideration as an extension to the two- and four-dimensional space filling curve methods. After the seed order has been selected, the remaining orders are assigned to the batch according to one of the following addition rules: the order with the highest number of locations in the common seed; the order with the minimum total distance (determined by a comparison of the total distance between its item location and the closest item location of the seed order); the order with the largest similarity coefficient (defined by the proportion of the area of intersection to the area of union of the economic convex hull); or the order with the smallest deviation from the six-dimensional space filling curve. Any of the above mentioned rules can be applied to the seed order. After adding the chosen order to the batch, the seed is updated to include the new order in compliance with the cumulative seeding rule.

Pan and Liu [92] recommended the economic convex hull algorithm of Hwang and Lee [62] to generate the most efficient batches in terms of travel time for small and large capacity picking devices. Their study focused on an automated storage and retrieval system single aisles layout applying the travelling salesman algorithm with the Chebyshev metric to determine the travel time for this specific set up.

2.3.3 Savings algorithms

The idea of savings algorithms is based on the Clarke and Wright-algorithm (CW) for the vehicle routing problem that has been adapted for the OBP. Clarke and Wright [24] developed an iterative procedure that enables the rapid selection of optimal or near-optimal routes from a large number of options of a fleet of trucks with varying capacities from a central depot to a number of delivery points. Savings algorithms in the OBP are based on time saving that can be obtained by comparing the collection of orders in one route to individual picking [28].

The minimum additional aisles heuristic proposed by Rosenwein [97] starts with calculating a score for each pair of orders in terms of additional aisles to pick for cases that orders are picked simultaneously or separately. Therefore, their algorithm is capable of generating fewer but shorter picking tours when compared to Gibson and Sharp [37]. The smallest centre of gravity describes the average number of aisles and is an additional distance metric that pairs orders according to their absolute difference between the centre of gravity of an order and a batch.

The first OBP version of the algorithm (CW I) described by De Koster *et al.* [28] computes savings for each combination of orders. The saving can be determined through tour length reduction by collecting the items of both orders simultaneously. The pair of orders providing the highest saving opens a new batch if none of the orders has been assigned to a batch yet. In the case where one order is already allocated, the other order is added if the capacity constraints allow it, otherwise the next pair of orders is considered [50].

In the second version of the algorithm (CW II), each time a new combination of orders to batches has been determined, the savings are recalculated according to De Koster *et al.* [28]. The formed clusters are considered as new orders and excluded from these calculations. Therefore, the clusters become larger and the savings matrix shrinks with every order addition [50]. Bozer and Kile [15] improved this version of the algorithm by introducing a normalised time saving value for each pair of orders. Therefore, the travel time saved by picking two orders simultaneously is divided by the time needed to pick the orders separately.

Each time an order has been assigned the initial savings matrix is modified, resulting in a third version (CW III) proposed by De Koster *et al.* [28]. This was implemented because the initial versions of the algorithms may result in solutions where there are too many batches. Where orders have not been allocated to a batch, the modification reduces the savings of pairs by a constant value. Therefore, the algorithm prefers the selection of orders when one order has already been allocated to a batch and thus new batches will not be formed too fast [50]. De Koster *et al.* [28] found that savings algorithms perform best in combination with largest gap strategies in parallel aisle warehouse layouts and for cases where the picking device capacity is small.

Elsayed and Unal [34] proposed a small and large algorithm. Before orders are assigned to batches, they are either classified as small or large according to a predefined value. The value is deducted as a certain percentage of the picking device capacity. According to Elsayed and Unal [34] near-optimal results are obtained if the value for small batches ranges between 10% and 35% of the picking device capacity. Large orders are separated and assigned to batches applying the EQUAL algorithm. Their EQUAL algorithm calculates and compares the travel time and saving of every possible order combination using a seed algorithm principle. The pair of orders with the highest saving in travel time functions as the seed. The remaining orders which increase the savings of the combination by the largest amount in travel time is added to the tour until the capacity limit of the picking device is reached. Orders that would exceed the capacity constraints if combined with another order are processed individually. Afterwards, small orders with the largest number of items are assigned to one of the batches. The batch which gains the biggest increase in travel time savings by the combination is chosen to be added to the small order. If an order cannot be added to any batch it will form an isolated tour. The process is repeated until all orders are allocated.

2.3.4 Data mining approaches

An order batching approach that is based on data mining and integer programming was presented by Chen and Wu [21] and further extended by Chen *et al.* [20]. Similarities between orders are detected with the help of association rules as introduced by Agrawal *et al.* [3]. A support value, that acts as a correlation measure between two orders, is determined for each order pair. Orders are then clustered into batches by an integer programming approach to maximise the sum of all support values. The batching approach was tested in a DC with a parallel aisle layout. However, the determination of the association rules is independent of the layout.

A data mining approach to generate similarity measures between SKUs was provided by Chiang *et al.* [22]. Each available storage location and new SKU thus gets an association index. A general assignment model is then used to minimise the number of aisles based on their index resulting in order batches.

Priority-based algorithms will serve as a benchmark to measure the performance of the order batching approach for the unidirectional cyclical picking line presented in this dissertation. In general, seed algorithms save more central processing unit (CPU) time than more complex algorithms. Therefore, the focus will be on seed algorithms to develop a solution approach that is applicable to real world problem instances. Additionally, clustering of orders with a data mining approach could be successfully applied to bigger sized batches without prior information about the layout.

2.4 Metaheuristics

Metaheuristics are algorithms that are designed to solve a wide range of hard optimisation problems. The Greek prefix *meta* indicates that these algorithms are higher level heuristics. Therefore, they do not have to adapt deeply to each problem. Hard optimisation problems are problems that cannot be solved to optimality with an exact solution approach in a reasonable amount of time [14].

In a real world application the OBP can be considered a hard optimisation problem. The focus of this dissertation is on batching two orders together (due to the picking device capacity), thus the OBP can be formulated as an assignment problem. Single-solution based metaheuristics can be applied to this type of problem. A single initial solution forms the starting point that the metaheuristics move away from describing a trajectory in the search space [14]. In this section the following algorithms are discussed: iterated local search, variable neighbourhood search, tabu search, simulated annealing, great deluge, and hybrid approaches. In the case of batching more than two orders population-based metaheuristics like ant colony optimisation or a genetic algorithm could be applied, but are not considered here.

2.4.1 Iterated local search

In an iterated local search (ILS) the embedded heuristic builds a sequence of solutions iteratively. Thereby, better solutions are generated than using random trials of the heuristic. According to Lourenço *et al.* [74] an iterated local search is characterised by following a single chain and also by the output of a black-box heuristic that defines a reduced solution space. The components of the algorithm are a cost function that has to be minimised for the combinatorial optimisation problem at hand and a local search routine that moves from an initial solution to the bottom of the corresponding basin of attraction. Perturbations are applied to escape a current local minimum. Additionally, a random restart to repeat the searching procedure from another starting solution is the easiest way to improve on the cost found by the local search. However, it has to be taken into consideration that as the instance size increases, the probability of finding a significantly lower cost by random sampling decreases. This can be achieved by including a cost history into the acceptance criterion of a new solution. The difference in cost between the newly generated solution and the current best solution defines the most basic acceptance criteria that can be applied in an iterated local search. The four components that influence the computational time are the generation of the initial solution, the local search, the perturbation of the solution, and the acceptance criterion [74].

Stützle [108] introduced ILS in his dissertation as an unifying framework of algorithms that rely on the same basic idea. These algorithms are addressed as large step Markov chains, chained local optimisation or iterated descent that introduce the ILS principle. The biggest differences between these algorithms are in the main components, namely the local search procedure, the perturbation, and the acceptance criterion of the specific algorithm [14].

Henn *et al.* [49] applied the ILS to manual order picking systems in the standard OBP. The initial solution is generated using a FIFO approach. In the local search procedure two sets of batches are referred to as neighbours if one solution can be generated from the other solution only by swapping two orders from the two different batches or by shifting an order to another batch. In the swapping move, the order changes to the same position in the sequence of orders. In the shifting move, the order is assigned to the end of the sequence. Depending on the success of a swap move, another swap move is carried out until unsuccessful in terms of improvement and thus changes to a shift move. In the perturbation procedure two batches are randomly selected and the first two orders are exchanged. The acceptance criterion allows a new solution as the best solution if the value of the objective function is lower. Experiments carried out by Henn *et al.* [49] showed that the solution quality of the ILS is superior to the FIFO benchmark in the environment of a parallel aisles warehouse.

2.4.2 Variable neighbourhood search

The variable neighbourhood algorithm successively explores a set of predefined neighbourhood structures to improve a solution. The exploration of the neighbourhoods can either be systematic or random, depending on the type of algorithm. The variable neighbourhood descent (VND) and the variable neighbourhood search (VNS) can be applied as a solution approach to the OBP [109].

The VND constitutes the basis for the variable neighbourhood search, since it is a deterministic version of the algorithm. As the name suggests, successive neighbourhoods in descent are used to determine a local optimum. In the beginning, a set of neighbourhood structures $\mathcal{H}_h$ with $h = 1, \ldots, h_{max}$ is defined and a solution initiated. After a local search in the neighbourhood structure, the search is restarted in the first neighbourhood $\mathcal{H}_1$. If no improvement can be made on the solution, the neighbourhood is changed from $\mathcal{H}_h$ to $\mathcal{H}_{h+1}$ [109]. The neighbourhoods have to be designed to be complementary in terms of a local optima in one neighbourhood while not being a local optima in another neighbourhood to successfully implement the VND. Different characteristics and properties of the search space should be exploited by the chosen neighbourhoods. Therefore, the design of the algorithm is mainly influenced by the neighbourhood structures. Larger neighbourhoods increase the runtime of the algorithm [13].

The VNS consists of the three steps shaking, local search, and move. Adding the shaking step to the VND, where at each iteration an initial solution $b^{'}$ is randomly generated from the current neighbourhood, makes the algorithm stochastic. Afterwards a local search is applied to find $b^{'*}$. The algorithms remains in the neighbourhood if the current solution $b^{'*}$ is better than the best solution b. If there is no improvement, the algorithm moves to the next neighbourhood comparable to the procedures in the VND. The work in the local search and the shaking phase are adjusted to balance intensification and diversification of the algorithm [109]. The selection of the neighbourhoods in the shaking phase mainly influences the performance of the VNS. Providing a good starting point for the local search is the objective of the shaking phase [13].

Mladenović and Hansen [85] first proposed the VNS metaheuristic by examining the change of neighbourhood within a local search algorithm. A basic scheme of the algorithm was presented which can implement any local search subroutine. Its effectiveness was tested through an application in the travelling salesman problem.

Albareda-Sambola *et al.* [4] applied a VNS to the OBP in a parallel aisles warehouse layout. A variable neighbourhood descent was implemented and three neighbourhoods were defined. The first neighbourhood transfers an order from one batch to another, the second neighbourhood transfers at most two orders from one batch to others, and the third neighbourhood transfers at most three orders. Therefore, the neighbourhoods are from a nested set of neighbourhood structures. The VNS is a competitive alternative solution approach in the OBP.

2.4.3 Tabu search

Tabu search (TS) incorporates the history of the search not only to escape local optima, but also to further explore the search space. A basic version of tabu search uses a best improvement local search and a short term memory to avoid cycling. This short-term memory is represented by a tabu list that keeps track of the most recently visited solutions and forbids moving towards them. Therefore, the neighbourhood of a solution $\mathcal{H}$ is restricted to solutions outside the tabu list creating an allowed set. The current solution is the best solution chosen from the allowed set at each iteration. This current solution is added to the tabu list while another solution from the tabu list is removed. This dynamic restriction of allowed solutions classifies the TS as a dynamic neighbourhood search technique. The algorithm either stops when a termination criterion is met or when the allowed set is empty [13]. The tabu list is the main feature of the tabu search. The memory of the search process is thus controlled by the length of the tabu list. A long tabu list forces to explore larger regions, while a shorter list concentrates the search in a smaller area. Storing solution attributes such as components of solutions or differences between solutions is more efficient, but can also result in an information loss [14].

Glover [38] first introduced the basic ideas of the TS. Further explanations on concepts and the method of TS were followed by Glover and Laguna [39]. The main characteristic of this algorithm is inspired by the human memory mechanism.

Henn and Wäscher [51] applied TS in a manual parallel aisles warehouse layout. A classic TS with a tabu list that records moves from previous iterations to forbid them in the current iteration was introduced to the OBP. In a numerical experiment the TS showed superior results compared to existing solution approaches.

2.4.4 Simulated annealing

Solving a large combinatorial optimisation problem can be compared to simulating the evolution of a solid in a heat bath to thermal equilibrium. Within this process called annealing, the metropolis algorithm generates a sequence of states for the solid. If a current state i of the solid with energy E_i is given, a perturbation mechanism, that transforms the current state into the next state j (by for example the replacement of a particle). Therefore, the energy of the next state is denoted as E_j. State j is accepted with a certain probability as the new current state, if the energy difference, namely $E_j - E_i$, is greater than zero. This acceptance or so-called

Metropolis criterion is given by

$$\exp\left(\frac{E_i - E_j}{k_B \cdot F}\right), \tag{2.14}$$

where the temperature of the heat bath is denoted by F and κ_B is a physical constant known as the Boltzmann constant. The solid can reach thermal equilibrium at each temperature if the temperature is lowered sufficiently slow. In simulated annealing the metropolis algorithm can be iterated to generate a sequence of solutions for a combinatorial optimisation problem [1].

In simulated annealing (SA) the process of annealing is transposed to the solution of an optimisation problem. Similar to the energy of the material, the objective function is minimised by controlling a fictitious temperature that is represented by a parameter in the algorithm [31]. Different approaches on determining an adequate cooling schedule are available. Van Laarhoven and Aarts [117] divided them into empirical and theoretically based cooling schedules. The programme of annealing mainly influences the convergence speed of the SA algorithm. According to Dréo *et al.* [31] it must thus specify the values of the parameters including the initial temperature, the change criterion, the law of decreasing the temperature and the program termination criterion.

Kirkpatrick *et al.* [66] and independently Černỳ [19] first introduced the SA algorithm. Low energy configurations of disordered magnetic material posed optimisation problems with several valleys of unequal depth that were approached by using the technique of annealing.

The great deluge (GD) is a variant of the SA. The main difference is the way in which the solution is accepted. Additionally, the algorithm is easier to execute, since the GD only requires one parameter that has to be calibrated. The algorithm is inspired by the idea of a hiker who wants to keep her feet dry and thus visits the peaks in the search space while the water level rises [31].

Dueck [32] introduced the GD and tested it on instances of the travelling salesman problem. The tours generated by the algorithm are sparse, dense and regular. He also included a comparison to the simulated annealing approach.

Matusiak *et al.* [83] applied SA to batch more than two customer orders together. The results show only a small deviation from an optimal approach on batching and a good alternative to other heuristic approaches. Their study was performed in a standard DC layout.

Nicolas *et al.* [88] extended their optimal solution approach for vertical lift modules (carousels) that are closely related to the unidirectional cyclical picking line, since order numbers exceeding 60 require a significant amount of computational time. Therefore, they tested a SA approach, a TS and a genetic algorithm with the SA approach obtaining the best results. The implementation of the metaheuristics outperformed the optimal solution approach in their study.

2.4.5 Hybrid algorithms

Hybrids benefit from synergies. Therefore, different algorithms are combined to exploit the complementary characteristics between various optimisation strategies. However, certain hybrids might work well on some problems, but not on others making generalisation a difficult task [12].

Öncan [90] suggested an ILS with a tabu threshold for the OBP in a parallel aisles warehouse layout. They reported that the hybrid preforming diversification and intensification steps simultaneously achieved outstanding results in accuracy and efficiency on a standard and on a randomly generated test set.

Žulj *et al.* [122] developed a hybrid of an adaptive large neighbourhood search and a TS and applied it to the OBP in a rectangular single-block warehouse layout with a low-level picker-to-parts order picking system. Thereby, they combined the diversification capabilities of the adaptive large neighbourhood search with the intensification capabilities of the tabu search. Their hybrid outperformed previously published algorithms in terms of average solution quality and runtime, especially for large number of orders.

2.5 Order batching on a unidirectional cyclical picking line

Typically, batching heuristics choose a seed order and then expand the batch by adding orders with proximity until the capacity of the picking device is reached. Therefore, the primary issue of batching heuristics is to define a measure of proximity between orders [20]. The OBP for picker-to-parts order picking systems is generally applied in single-block parallel aisles warehouse layouts. In defining proximity measures between orders, travel distance is affected by the DC layout [52]. Therefore, Valle and Beasley [112] introduced distance approximations incorporating aisles and cross-aisles distances. However, the layout used in the standard OBP differs significantly from the unidirectional cyclical picking line.

The vertical lift module (carousel) investigated by Nicolas *et al.* [87, 88] shows similarity to the unidirectional cyclical picking line. However, their system is an automated parts-to-picker system. They defined the number of trays visited for different batches as a measure of proximity between orders [88]. This is not possible on the unidirectional cyclical picking line.

No proximity measures between orders for a picker-to-parts order picking system incorporating a unidirectional cyclical picking line have been developed in literature. Therefore, the batching rules that have been introduced need to be adapted and extended for the specific layout and characteristics of the unidirectional cyclical picking line. Additionally, each picking wave in the Retailer's order picking system processes around 1 800 orders. The solution approach for the OBP thus has to be fast to meet the requirements of the picking system.

According to Van Gils *et al.* [116] multiple order picking planning problems should be combined to efficiently manage order picking systems. Planning problems that are combined should be within a similar time frame, because the OBP has to be solved by a DC on a daily basis. Analytical and simulation models can serve as decision support tools for DC managers taking interactions in the picking system into account. These models can be used to address practical problems such as the dynamics between multiple pickers resulting in picker blocking and congestion. The planning problems should also be solved with fast solution approaches to make the application practical [17]. The importance of the product properties, may lead to data mining approaches in the order batching operation [18].

Order batching is introduced to a unidirectional cyclical picking line in the next chapter. Different distance approximations are developed and combined with greedy heuristics and metaheuristics to determine the combination that minimises walking distance most effectively.

CHAPTER 3

Picking location metrics for order batching on a unidirectional cyclical picking line

Contents

In existing operations research literature the order batching problem (OBP) has not been introduced to a unidirectional cyclical picking line. In most cases order batching takes place in a single-block warehouse with parallel aisles environment. Therefore, proximity measures are adapted and extended to meet the requirements of the specific set up of this dissertation, since they are dependent on the underlying layout. The idea that the introduced proximity measures can be used to approximate walking distance is supported by a regression analysis. Combination algorithms that generate batches, reducing the walking distance of pickers in a short amount of time, are identified.

3.1 Introduction

The need for greater product variety and shorter response times emphasises a company's ability to establish efficient logistical operations. This efficiency is, amongst others, determined by the operation of its warehouses or distribution centres (DCs) that define the nodes in the distribution network [98]. In a supply chain, connecting production plants with end customers warehousing cannot be eliminated. The role of warehouse operations is constantly changing and thus has to remain flexible. A DC focuses on the consolidation and accumulation of numerous products from various suppliers and delivering those to customers. Products from different suppliers arrive in bulk at the DC. The stock is then turned into customer or store orders for delivery to the DC's customers [35].

Van den Berg and Zijm [113] subdivide DC activities into four categories, namely receiving, storage, order picking and shipping. From the total logistics cost of a company, approximately 25% can be attributed to the cost of operating a warehouse [49]. Therein, the basic service of a DC is order picking that makes up for 50% to 65% of the operational cost, accounting for labour, capital and supporting activities [113]. De Koster *et al.* [29] define the process of order picking as retrieving products from storage or buffer areas and turning them into specific orders in response to a customer request. The design of the order picking system is crucial to its performance. The number of orders, together with the items per order that are picked in a day, the average size of an order, and the layout of the storage racks are key parameters to set up an efficient order picking system [27].

The order picking system of a prominent South African retailer (referred to as the Retailer) with around 2 000 outlets or stores is considered here. A set of non-uniform orders is the result of a large number of stores with different sizes and various customer profiles that needs to be handled by the DC on a daily basis. A key characteristic of the Retailer's operations is the central planning of the inventory kept at store level. Instead of stock requested by store managers, planners at a central planning department assign stock that is available at the DC to stores. Thereby, the central planning department allocates store keeping units (SKUs) to stores. Planners in the central planning department decide on the number of SKUs destined for each store after the arrival of the SKUs at the DC. Planners issue instructions to the DC about the SKUs and the stores where they should go. The DC then selects a subset of SKUs to be picked in a single picking wave to satisfy all store requirements for that set of SKUs. In this dissertation, the term *order* will thus refer to the set of store requirements for a single store for all SKUs selected to be processed in a wave. A wave is processed on a picking line in the DC. A single SKU is assigned to a single location on a specific picking line for each wave. The activities of populating the line with SKUs, the actual picking process and removing excess stock from the line comprise a *picking wave* [78].

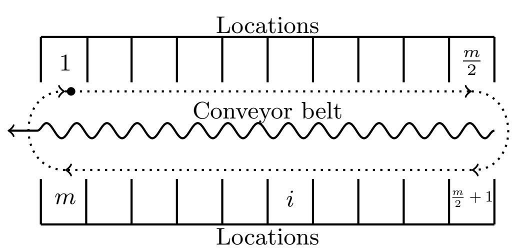


Figure 3.1: *A schematic representation of a picking line with m locations. Source: De Villiers [30].*

A schematic representation of a picking line with m locations is depicted in Figure 3.1. At the Retailer's DC in Durban a picking line consists of 56 locations with a conveyor belt placed in the middle that has two access gates. From the storage area, a single SKU is assigned to a single location on the picking line. The number of items of each SKU to be picked for all stores are known prior to the start of a picking wave. Each location has the storage capacity of five pallets of an identical SKU. Additional stock is kept on the floor space between different picking lines for easy refill, avoiding stock-outs. Therefore, stock does not have to be replenished from storage racks during a picking wave. Pickers move around the conveyor belt in a clockwise direction picking all orders. A voice recognition system (VRS) guides the pickers through the picking process by sending a picker to the closest required SKU. Therefore, the underlying picking strategy can be summarised as pick the closest SKU in a clockwise direction. The picking system is categorised as a picker-to-parts system. Before the picking of an order is started, an empty carton is placed on the trolley and registered through a barcode with the VRS. Each order is connected to a unique carton (or set of cartons depending on the size of the order). After picking, full cartons and finished orders are placed on the conveyor belt for further processing in the dispatch area of the DC [76, 77].

The set up of the picking system at the DC shows many similarities to unidirectional carousel systems, if the SKUs are viewed as moving relative to a static picker. A carousel is an automated warehouse system that can be used for picking small and medium products. It is designed as a rotatable circuit of shelving holding multiple SKUs. A picker, who remains at a fixed location during the picking process, operates the system. In literature this picking system is referred to as a parts-to-picker system, since the storage location travels to the picker instead of the picker travelling to the storage location. A bidirectional carousel can rotate in both directions to bring the required SKU in front of the picker, whereas a unidirectional carousel can only move in one direction [7]. The cyclical set up of the picking line, the assumptions that one picker processes one order, and the automation of pick sequencing by the VRS resembles a unidirectional carousel. However, the presence of wave picking is the main difference in the Retailer's system to the carousel systems studied in literature. Not all SKUs may be picked and new orders may be added to the set during the picking operation in typical carousel systems. Therefore, optimisation approaches make use of expected SKU mixes in orders that are derived from historical data [73]. The deterministic nature of the orders (because all orders are known and fixed when a wave of picking starts) is unique to the Retailer's picking system in operations research literature [78].

Planning for order picking includes the order batching problem. Orders that are requested are subsequently released for fulfilment. This set of orders needs to be picked and accumulated for packing and shipping in a picking wave [42]. The combination of customer orders into picking orders describes the process of order batching [50]. According to Wäscher [120] order batching answers the question of how customer orders should be grouped into picking orders to minimise the total length of all picking tours necessary to collect all items while no customer order is split, given (1) a set of customer orders consisting of a number of items, (2) an assignment of items to storage locations in the order picking system of a warehouse, and (3) the capacity of a picking device. Tours that efficiently shorten the picking time can be generated, since several orders are picked simultaneously leading to a reduction in labour cost [28].

While there has been intensive research on the order batching problem in single-block warehouses with parallel aisles, according to Nicolas *et al.* [87] little research has been published on carousel systems. The literature on carousels can be categorised into either determining the pick sequence on a carousel or finding the best storage location for a product in a carousel. No study on order batching applied to a manual unidirectional carousel systems as implemented by the Retailer could be found in literature.

This chapter aims at answering the question if the OBP can be applied to the specific layout of a unidirectional cyclical picking line. The orders will be batched before the pickers are routed. However, walking distance can not be determined *a priori* and thus a realistic approximation for distance has to be provided. Different location-based metrics are developed to estimate walking distance, since calculating walking distance is too time consuming. Different algorithms that correlate with these metrics to solve the order batching problem are tested. A suggested combination of metric and algorithm to minimise the total distance travelled by a picker to collect all orders during a picking wave concludes this chapter.

A brief overview on the literature of order-batching in single-block and automated storage and retrieval warehouse set ups is given in Section 3.2. The model will be described in Section 3.3 including assumptions, measurements and an integer programming formulation of the problem. The location-based order-to-route closeness metrics will be proposed in Section 3.4. In Section 3.5 different heuristic approaches which include greedy heuristics and metaheuristics are introduced to solve the OBP in reasonable time. A case study with real life data from the Retailer is used to test the performance of the combinations of metrics and algorithms in Section 3.6. In Section 3.7 a summary and an outlook on future research opportunities is provided.

3.2 Literature review

The formulation of the standard OBP was introduced and proved to be NP-hard in the strong sense by Gademann and Velde [36] for a single-block warehouse with parallel aisles environment. However, if no batch contains more than two orders the problem can be solved in polynomial time. The branch-and-prize algorithm developed by Gademann and Velde [36] solved test instances of up to 32 orders to optimality. Additionally, Henn and Wäscher [51] solved instances of up to 40 orders. Nevertheless, their solution approach required a time-consuming preprocessing of all feasible batches. Muter and Öncan [86] solved instances of up to 100 orders with their specially tailored column-generation based algorithm. In practice the number of customer orders is likely to exceed these test instances, but the exact algorithms proposed so far are not able to consistently solve larger instances. Nicolas *et al.* [87] introduced the OBP to a vertical carousel system that is closely related to the layout of the unidirectional cyclical picking line. The OBP is formulated as a mixed integer linear program (MILP). Nevertheless, the MILP has to be stopped after 30 minutes due to time constraints for test instances over 50 orders. In the Retailer's DC a picking line can service up to 1 500 orders. Larger problem instances must be solved using heuristics. Therefore, construction heuristics like seed algorithms and saving algorithms have been introduced to solve the OBP.

A straight-forward heuristic is the first-in-first-out method that groups together the first entries of the list of orders as close as possible to the predetermined maximum batch size. Then the next orders are grouped using a similar logic until all orders form part of a batch [37].

Seed algorithms start by initiating batches, then allocating orders to batches and terminate through a stopping rule when a batch has been completed. The objective is to minimise the total travel distance for collecting all orders. De Koster *et al.* [28] evaluated seed algorithms in a parallel aisle warehouse layout proposing the seed selection rules of the farthest storage location. Ho and Tseng [53] investigated several location- and aisles-based seed algorithms in the standard OBP environment of a single-block parallel aisle warehouse. Ho *et al.* [52] extended this study with further distance- and area-based selection rules in this environment. In their comparative study Pan and Liu [92] evaluated four initial seed selection and four order addition rules investigating the OBP in an automated storage and retrieval system that consists of a single storage rack with equally sized storage locations serviced by a single storage and retrieval machine.

Saving algorithms in the OBP are based on the Clarke and Wright-algorithm (CW) [24] and thus the time saving that can be obtained by comparing the collection of orders in one route to individual picking [28]. The minimum additional aisles heuristic proposed by Rosenwein [97] starts with calculating a score for each pair of orders in terms of additional aisles to pick from in the cases that orders are picked simultaneously or separately. Therefore, their algorithm is capable of generating fewer but shorter picking tours when compared to Gibson and Sharp [37]. The first OBP version of the algorithm (CW I) described by De Koster *et al.* [28] computes savings for each combination of orders. Each time a new combination of orders to batches has been determined the savings are recalculated in the second version of the algorithm (CW II). Bozer and Kile [15] improved this version of the algorithm by introducing a normalised time saving value for each pair of orders. Each time an order has been assigned the initial savings matrix is modified resulting in a third version (CW III) proposed by De Koster *et al.* [28]. Additionally, Elsayed and Unal [34] proposed a small and large algorithm. Therefore, orders are either classified as small or large according to a predefined value before they are assigned to batches. In general seed algorithms are more central processing unit (CPU) time saving than saving algorithms. The order-to-route closeness metric is central to all types of heuristics [42].

Metaheuristics are designed to solve a wide range of hard optimisation problems and can thus also be applied to the OBP. An iterated local search that explores a neighbourhood to identify a new solution with a smaller objective function value was applied to the OBP by Henn *et al.* [49]. Albareda-Sambola *et al.* [4] defined three neighbourhoods to introduce the OBP to a variable neighbourhood search. A tabu search that simulates the human memory processes including a tabu list was applied to the OBP by Henn and Wäscher [51]. Matusiak *et al.* [83] introduced a simulated annealing approach that simulates the cooling process of metal to the OBP. Öncan [90] introduced a combination of an iterated local search with a tabu threshold to the OBP. A hybrid of a large adaptive neighbourhood search and a tabu search was developed by Žulj *et al.* [122] based on the findings of Henn and Wäscher [51] and Öncan [90].

3.3 Model formulation

The problem environment of this study is a DC of the Retailer. The DC is made up of several unidirectional cyclical picking lines functioning in parallel. Thus one such picking line will be at the centre of attention. This layout differs significantly from the parallel-aisle or the single-aisle automated storage and retrieval warehouses that are frequently studied in academic literature.

The following assumptions are made while modelling the Retailer's order picking system.

1. The orders that need to be picked during a wave and the SKUs and their location in the picking line will be fixed *a priori*.

2. For movability the picking devices are small. However, SKUs are generally bulky. Therefore, the capacity of the picking devices is currently restricted to accommodate two orders at a time restricting the batch size to two.

3. A picker must complete the entire order before starting the next, packing stock directly onto the picking device. When an order is completed or the carton is full, it is placed on the conveyor belt for further processing.

4. It is assumed that a picker walks at a constant speed and that the time taken to pick and pack a SKU or switch between orders is constant.

The error potential of adding the wrong item to an order through the introduction of order batching was mitigated by the design of the picking devices with two shelves accommodating two orders. Furthermore, the orders are colour-coded corresponding to the colour-coded cartons (Order A is blue and picked into the blue carton, while Order B is red and picked into the red carton).

3.3.1 Measurements

The cycles traversed measurement, that was introduced by Matthews and Visagie [78], counts the number of cycles that have to be traversed to pick all items requested during a picking wave as a measure of distance travelled to pick all orders. This measurement counts the total number of cycles required to pick and link all orders.

In the system currently in use by the Retailer each order is processed by one picker completing one order at a time. Orders are sequentially assigned to the next available picker based on a fixed list of orders. In effect each order is assigned randomly, since the assignment does not take the previous order or the current position of the picker into consideration. Therefore, each picker picks a random sequence of orders. Matthews and Visagie [78] proposed a nearest end heuristic as an easily implementable option to determine a sequence of orders that minimises the distance that pickers have to travel during a wave. The nearest end heuristic sequentially selects the order with the nearest ending position from the current picking position. Therefore, it simultaneously considers the order sequence and the item sequence within the order. An example of a unidirectional cyclical picking line with 10 locations and 10 SKUs is depicted in Figure 3.2. Four different orders are indicated by the colours green, blue, red and yellow. The different shapes represent the SKUs and locations respectively. If the nearest end heuristic is applied to this example, Order 4 (yellow) would be picked first, starting at the first location and ending at location seven. This would be followed by the picking of Order 1 (green), then Order 3 (red) and finally Order 2 (blue). In total a picker has to traverse four cycles to collect all orders of the picking wave. The nearest end heuristic is easily reproducible. Therefore, it is used in determining an order sequence to test the effectiveness of the various batching logics in terms of number of cycles traversed.

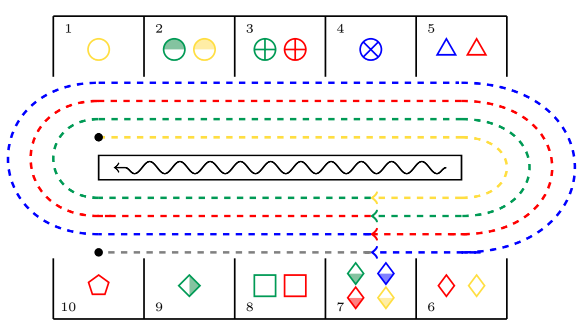


FIGURE 3.2: *Schematic representation of single order picking in a picking line with* 10 *SKUs and* 10 *locations.*

This chapter focuses on location-based order-to-route-closeness metrics. Each picking location defines where a picker has to stop at a location to pick items for an order. Different picking location or so-called stop metrics will be developed to identify compatible orders in terms of number of stops in common. The assumption is that a good overlap in stops may lead to a reduction in total walking distance.

3.3.2 Exact formulation

Minimising the completion time is equivalent to minimising the travel distance to collect all orders in a picking wave. Therefore, the objective is to minimise the incompatibility in terms of distance between orders that are described by a picking location metric to obtain the smallest number of cycles that have to be traversed. Orders must be combined in batches of size two. This can be formulated as an integer programming model. Let

n be the number of orders,
c_{ij} be the cost in terms of incompatible stops to batch order i and order j,

and define the set of variables as

$$x_{ij} = \begin{cases} 1, & \text{if order } i \text{ and order } j \text{ are in the same batch,} \\ 0, & \text{otherwise.} \end{cases}$$

The objective is to

$$\text{minimise} \sum_{i=1}^{n} \sum_{j=i+1}^{n} c_{ij}\, x_{ij} \tag{3.1}$$

subject to

$$\sum_{k=1}^{i-1} x_{ki} + \sum_{j=i+1}^{n} x_{ij} = 1 \qquad i = 1, \ldots, n \tag{3.2}$$

$$x_{ij} \in \{0, 1\} \qquad \begin{cases} i & = 1, \ldots, n \\ j & = i+1, \ldots, n. \end{cases} \tag{3.3}$$

The incompatibility between orders in terms of stops expressed by a picking location metric is minimised by objective function (3.1). The set of triangular inequality constraints (3.2) and the binary condition (3.3) assign each order to only one other order in the case of a symmetric cost matrix.

The problem formulated in (3.1) – (3.3) is in essence an assignment problem. Another option to solve the assignment problem is the quick match algorithm that was introduced by Orlin and Lee [91]. It is based on the successive shortest path algorithm and combines a forward Dijkstra with a reverse Dijkstra algorithm. Heuristics are included to speed up its performance. Therefore, this algorithm will also be included for testing purposes.

3.4 Picking location metrics

The simplest way of introducing order batching to a unidirectional cyclical picking line would be to implement a first-in-first-out (FIFO) approach. According to the FIFO rule the first entries of a list of orders are grouped together until the predetermined maximum batch size is reached [37]. FIFO is often used in literature as a benchmark to compare different batching methods. It will also be used here to compare the metrics that measure the incompatibility of orders in terms of stops. An example of the picking locations of four orders is illustrated in Table 3.1. Employing the nearest end heuristic as a routing strategy (Section 3.3.1), a picker has to traverse four cycles to collect all four orders. If FIFO is applied as the random batching strategy and a batch is only allowed to contain two orders as illustrated in Figure 3.3, then the first two orders would form Batch 1 (in orange) and Batch 2 would be composed of Order 3 and Order 4 (in purple). The circles represent the movement of a picker showing that only two cycles have to be traversed. The number of cycles traversed will indicate the walking distance to compare different metrics.

Locations:	1	2	3	4	5	6	7	8	9	10
Order 1		◓	⊕				◈	□	◈	
Order 2				⊗	△		◈			
Order 3			⊕		△	◇	◈	□		⬠
Order 4	○	◓				◇	◈			

TABLE 3.1: *The picking locations of the four orders in the example.*

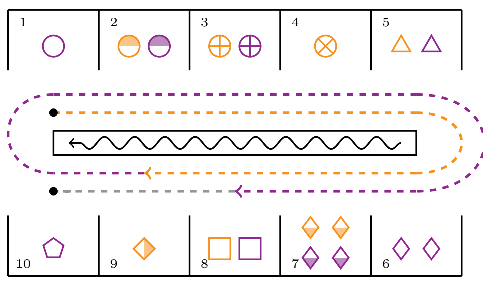


FIGURE 3.3: *Schematic representation of order batching in a picking line with* 10 *SKUs and* 10 *locations.*

In the remainder of this section, the stops metric, the non-identical stops metric and the stops ratio that approximate the compatibility between orders are introduced. Additionally, a small example on the basis of Table 3.1 will be provided to explain each metric.

The stops metric in matrix $\mathbf{T}$ with general element t_{ij} is a location-based batching rule. It results in the orders with the smallest total number of picking locations forming a batch. The logic of this metric combines the seed selection rule of choosing the smallest number of non-overlapping picking locations introduced by Elsayed and Stern [33] with the smallest number of additional picking locations rule proposed by Ho and Tseng [53]. The number of picking locations of each order when combined with every other order is calculated first. This results in the total number of picking locations that a picker has to visit if those two orders are batched. The orders with the smallest number of combined picking locations are then selected to form a batch. Therefore, each location at which a picker has to stop to pick for one of the orders is counted. On the contrary, counting all locations a picker passes not stopping to collect items for any order may result in the combination of orders which only have a small number of picking locations in common. This option is thus not further investigated.

The stops metric t_{ij}, with the sets $\mathcal{S}_i$ and $\mathcal{S}_j$, containing all stops for orders i and j may be calculated as

$$t_{ij} = |\mathcal{S}_i| + |\mathcal{S}_j| - |\mathcal{S}_i \cap \mathcal{S}_j|. \tag{3.4}$$

Batching Order 1 with $\mathcal{S}_1 = \{2, 3, 7, 8, 9\}$ and Order 2 with $\mathcal{S}_2 = \{4, 5, 7\}$ results in the stops entry $t_{12} = 5 + 3 - 1 = 7$. Calculating the stops metric for all orders in the example results in a symmetric matrix as illustrated in Table 3.2(a). Combining the orders with the exact solution approach, as described in Section 3.3.2, batches the orders with the smallest number of picking locations. Therefore, the first batch could contain Order 1 and 3 and the second batch could be composed of Order 2 and 4 in this example.

$$\mathbf{T} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} - & 7 & 8 & 7 \\ 7 & - & 7 & 6 \\ 8 & 7 & - & 8 \\ 7 & 6 & 8 & - \end{bmatrix} \end{array}$$

(a) *The* **T** *matrix.*

$$\mathbf{N} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} - & 6 & 5 & 5 \\ 6 & - & 5 & 5 \\ 5 & 5 & - & 6 \\ 5 & 5 & 6 & - \end{bmatrix} \end{array}$$

(b) *The* **N** *matrix.*

$$\mathbf{R} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} - & 0.86 & 0.63 & 0.71 \\ 0.86 & - & 0.71 & 0.83 \\ 0.63 & 0.71 & - & 0.75 \\ 0.71 & 0.83 & 0.75 & - \end{bmatrix} \end{array}$$

(c) *The* **R** *matrix.*

TABLE 3.2: *The* **T**, **N** *and* **R** *matrices for the picking line example.*

An additional location-based rule is the non-identical stops metric in matrix **N** with general element n_{ij}. Batches are formed by combining orders with the smallest number of non-identical picking locations as a result of the application of this metric. Ho and Tseng [53] proposed the greatest number of identical picking locations rule. The logic of n_{ij} is contrary to their rule. The picking locations for combining each order with every other order are obtained in the first step. However, the order with the smallest number of non-identical picking locations between orders becomes the accompanying order in this case. Therefore, each location where a picker stops to collect for only one of the orders is counted to identify non-identical picking locations.

The non-identical stops matrix **N**, with element n_{ij}, can be calculated as

$$n_{ij} = |\mathcal{S}_i| + |\mathcal{S}_j| - 2\,|\mathcal{S}_i \cap \mathcal{S}_j|. \tag{3.5}$$

In the example Order 1 with stops $\mathcal{S}_1 = \{2, 3, 7, 8, 9\}$ and Order 3 with $\mathcal{S}_3 = \{3, 5, 6, 7, 8, 10\}$ result in the non-identical stops entry $n_{13} = 5 + 6 - 6 = 5$. Table 3.2(b) results from calculating the non-identical stops metric for all orders in the example. Using the exact solution approach the combinations of orders with the smallest number of non-identical picking locations could result in the combination of Order 1 and 4, and also Order 2 and 3.

The stops ratio metric in matrix **R** with general element r_{ij} combines the non-identical stops with the combined picking locations of the orders. This metric is also based on the location of items in an order. The non-identical stops metric is divided by the stops metric and thus results in the stops ratio metric. The closer the stops ratio is to zero, the lower the incompatibility of orders in terms of walking distance.

Formulating the stops ratio metric can be achieved by dividing n_{ij} by t_{ij}. Therefore, the stops ratio is calculated as

$$r_{ij} = \frac{n_{ij}}{t_{ij}}. \tag{3.6}$$

For batching Orders 1 and 4 with stops $\mathcal{S}_1 = \{2, 3, 7, 8, 9\}$ and $\mathcal{S}_4 = \{1, 2, 6, 7\}$, the stops ratio metric $r_{14} = 5/7$ would result in 0.71. In Table 3.2(c) all stops ratios for all orders from the example are illustrated. Order 1 and 4 (and thus Order 2 and 3) could be combined when applying the exact solution formulation.

Even in this small example as described in Table 3.1 the picking location metrics measure the incompatibility of orders in terms of distance differently and thus suggest different batch combinations ($1, 2$ and $3, 4$ in FIFO, $1, 3$ and $2, 4$ in T, $1, 4$ and $2, 3$ in N and R). With bigger data sets the differences in using different metrics become more evident. Furthermore, picking waves with a size of up to 2 000 orders have over a million different possible combinations in determining the best match between two. Applying an exact solution approach would make checking all possible combinations very time consuming. Therefore, heuristic and metaheuristic solution approaches that reduce computational time are described in the following section.

3.5 Heuristic solution approaches

After the orders have been measured using one of the proposed picking location metrics, different algorithms to combine the orders in batches of size two for a unidirectional cyclical picking line in the Retailer's DC are described in this section. Four greedy heuristics and six metaheuristics are introduced.

3.5.1 Greedy heuristics

The four different variations of greedy heuristics include a greedy top-down, a greedy bottom-up, a greedy random and a greedy smallest entry approach. All four variations search through the symmetric matrices generated by applying the stop metrics for minimum entries.

The greedy top-down (GTD) starts searching the matrix from the first row until it reaches the last row in a top-down fashion. Thereby, the tuple corresponding to the minimum entry m_{kq} of row k is recorded in a set $\mathcal{B}$ to indicate that orders k and q are batched. Then both rows and columns k and q are removed. This process continues until all rows and columns of the matrix are eliminated. At this time set $\mathcal{B}$ has the cardinality n, where n is the size of the problem. The greedy bottom-up (GBU) and the greedy random (GR) progress in a similar way. The GBU starts at the last row and searches until it reaches the first row. The GR searches the rows in a random sequence. This is displayed in Algorithm 1. Furthermore, a greedy algorithm that globally searches for the smallest entry (GS) in the matrices was developed. This algorithm then eliminates rows and columns in the same way as the other greedy heuristics. There is no difference between searching rows or columns, since the picking location metrics generate symmetric matrices [30].

Algorithm 1: `Greedy random heuristic (GR)`

Input: A picking location metric consisting of a $n \times n$ matrix M with entries m_{ij}, an empty solution set $\mathcal{B}$
Output: The solution set $\mathcal{B}$ as a list of batched orders

$\mathcal{B} \leftarrow \emptyset$
while $|\ \mathcal{B}\ | < n/2$ **do**
    $k \leftarrow$ random row from M
    $m_{kq} = \min\limits_{j}[m_{kj}]$
    $\mathcal{B} \leftarrow \mathcal{B} \cup (k, q)$
    Remove both rows and columns k and q from $\mathbf{M}$
end while
Return $\mathcal{B}$

3.5.2 Metaheuristics

Only single-solution based methods have been chosen for the six metaheuristics, since the objective is to generate batches of size two. These methods start with a single initial solution, then moving away from it to describe a trajectory in the search space [13]. The algorithms are described in this section, while the parameters for each metaheuristic controlling the trade-off between intensification and diversification have been calibrated specifically for the problem statement of the unidirectional cyclical picking line in the Retailer's DC. For each metaheuristic starting from parameter values found in literature the number of configurations used to calibrate differ between algorithms, due to their ability to find a solution within a limited number of iterations that allows for an overall optimisation of the picking process. Including a range in number of orders and SKUs 12 sample picking waves are used to determine the configuration for each metaheuristic that provides the lowest number of total cycles traversed.

The iterated local search (ILS) generates a new starting solution for the following iteration by perturbing the local optimum found at the current iteration. Therefore, the local search procedure is not repeatedly applied to a randomly generated starting solution, but to the best solution found in the previous iteration. The underlying assumption is that the perturbation mechanism, which is the key feature of the ILS, is able to provide a solution that lies in the basin of attraction of a better local optimum [14]. The ILS and its framework was first defined by Stützle [108]. In Algorithm 2 the pseudocode of the ILS for the OBP is displayed. The acceptance criterion incorporates a restart of the search after a predefined non-acceptance counter thus incorporating a simple form of history. By means of parameter calibration 12 configurations with different termination criteria and non-acceptance counters under limited time have been tested on the sample picking waves. The lowest numbers of total cycles traversed were found incorporating the configuration with a termination criterion t_I of 5 and a non-acceptance counter a_I of 3.

Algorithm 2: `Iterated local search (ILS)`

Input: An initial solution b_a, a cost function $c(\cdot)$ as well as a non-acceptance counter a_I and a termination criterion t_I
Output: The best solution b as a list of batched orders

$a_I, t_I \leftarrow 0$
$b_a \leftarrow$ initiate a starting solution
$b \leftarrow b_a$
$b^* \leftarrow$ perform a local search on b
while the termination criterion t_I is not met **do**
  $b^{'} \leftarrow$ perturb b^*
  $b^{*'} \leftarrow$ perform a local search $b^{'}$
  if $c(b^{*'}) \leq c(b^*)$ **then**
    $b^* \leftarrow b^{*'}$
    $a_I = 0$
  else
    $a_I = a_I + 1$
    if non-acceptance counter a_I is not met **then**
      $t_I = t_I + 1$
    else
      $b^* \leftarrow$ restart search
      $t_I = 0$
    end if
  end if
  $b \leftarrow b^*$
end while
Return b

The variable neighbourhood search (VNS) explores the solution space by dynamically changing neighbourhoods around a given solution. After an initial solution is introduced, the main cycle of the VNS consists of the steps shaking, local search, and move. In the shaking step a solution $b^{'}$ is randomly selected in the h^{th} neighbourhood of the current solution. If the local search produces a better solution $b^{*'}$ than b, the solution is updated and the algorithm stays in the first neighbourhood. Otherwise, the algorithm moves to explore the next neighbourhood [14]. Mladenović and Hansen [85] introduced the structure of this algorithm. The pseudocode displayed in Algorithm 3, applies the VNS to the OBP. In this study three neighbourhoods are defined after testing different configurations in numerical experiments. In the first neighbourhood the order with the highest cost according to the picking location metric is swapped with the lowest, then the second highest and lowest are swapped, and finally, the third highest and lowest are interchanged respectively. The termination criterion t_V has been set to 5 through parameter calibration that included six different configurations of termination criteria, and were tested on the sample picking waves under time restriction.

Algorithm 3: `Variable neighbourhood search (VNS)`

Input: An initial solution b_a, a set of neighbourhood structures $\mathcal{H}$ with $h = 1, 2, 3$, a cost function $c(\cdot)$ as well as a termination criterion t_V

Output: The best solution b as a list of batched orders

$t_V \leftarrow 0$
$\mathcal{H} \leftarrow$ generate a set of neighbourhood structures with $h = 1, 2, 3$
$b_a \leftarrow$ initiate a starting solution
$b \leftarrow b_a$
while the termination criterion t_V is not met **do**
    $h \leftarrow 1$
    while in one of the defined neighbourhood structures **do**
        $b^{'} \leftarrow$ select a random solution in the h^{th} neighbourhood $\mathcal{H}_h(b)$ of b
        $b^{*'} \leftarrow$ perform a local search on $b^{'}$
        if $c(b^{*'}) \leq c(b)$ **then**
            $b \leftarrow b^{*'}$
            $h = 1$
        else
            $h = h + 1$
        end if
    end while
    $t_V = t_V + 1$
end while
Return b

The variable neighbourhood descent (VND) is a variation of the variable neighbourhood search. The VND is a deterministic version of the VNS excluding the shaking step [109]. The search steps of the VND are illustrated in Algorithm 4. The neighbourhood structure stays the same as in the VNS apart from the termination criterion t_D that is set to 50 through parameter calibration by testing 10 configurations with varying termination criteria on the sample picking waves under limited time. The VND is able to find a feasible solution faster than the VNS in this study, thus more configurations can be tested.

In a tabu search (TS) the history of the search is used to escape from local optima. The search history is also used to implement an exploitative strategy. A defined number of previously encountered solutions is recorded in a tabu list and thus forbidden to be revisited. The list can be described as a short term memory that prevents the algorithm from endless cycling and forces the search to explore different solution spaces. The length of the tabu list thus controls the memory of the search process. Glover [38] first introduced the tabu search algorithm. The mechanisms of the human memory inspired the main characteristic of this algorithm. Its application to the OBP is illustrated in Algorithm 5. The configuration with its tabu list length φ set to 0.8 relative to the size of the problem and its termination criterion t_T set to 3 provides the lowest number of total cycles traversed when testing 32 configurations with different length of tabu lists and termination criteria on the sample picking waves under time restriction.

Inspiration for the simulated annealing (SA) algorithm comes from the annealing technique used by metallurgists. Material is heated up to a high temperature and then lowered down slowly to obtain a well ordered state of minimum energy. Applying this technique, the objective

Algorithm 4: `Variable neighbourhood descent (VND)`

Input: An initial solution b_a, a set of neighbourhood structures $\mathcal{H}$ with $h = 1, 2, 3$, a cost function $c(\cdot)$ as well as a termination criterion t_D

Output: The best solution b as a list of batched orders

$t_D \leftarrow 0$
$\mathcal{H} \leftarrow$ generate a set of neighbourhood structures with $h = 1, 2, 3$
$h \leftarrow 1$
$b_a \leftarrow$ initiate a starting solution
$b \leftarrow b_a$
while in one of the neighbourhood structures and the termination criterion t_D is not met **do**
    $b^{'} \leftarrow$ perform a local search on b in the h^{th} neighbourhood $\mathcal{H}_h(b)$
    if $c(b^{'}) \leq c(b)$ **then**
        $b \leftarrow b^{'}$
        $h = 1$
    else
        $h = h + 1$
    end if
    $t_D = t_D + 1$
end while
Return b

Algorithm 5: `Tabu search (TS)`

Input: An initial solution s_b, a neighbourhood $\mathcal{H}(b)$, a tabu list length φ, a cost function $c(\cdot)$ as well as a termination criterion t_T

Output: The best solution b as a list of batched orders

$t_T \leftarrow 0$
`TabuList` $\leftarrow \emptyset$
$b_a \leftarrow$ initiate a starting solution
$b \leftarrow b_a$
while the termination criterion t_T is not met **do**
    $b^{'} \leftarrow$ select the best solution in $\mathcal{H}(b) \setminus$ `TabuList`
    if $c(b^{'}) \leq c(b)$ **then**
        $b \leftarrow b^{'}$
        Update `TabuList`
        $t_T = 0$
    else
        $t_T = t_T + 1$
    end if
end while
Return b

function of the optimisation problem is minimised similar to the energy of the material [14]. The starting solution and the annealing scheme for the temperature decrease are initiated. At each iteration, a new solution is accepted with a certain probability determined by the Metropolis criterion [1]. The SA algorithm was introduced by Kirkpatrick *et al.* [66] and by Černỳ [19] independently. In Algorithm 6 the structure of the SA is illustrated. The annealing scheme is crucial to the performance of the algorithm [31]. Therefore, three different approaches in determining the annealing scheme have been analysed for this application. A constant lowering of the temperature, dynamically changing the temperature according to the acceptance of a number of perturbations, and restarting to the initial solution after a number of non-acceptances of the new solution have been tested. Thereby, reheating is incorporated in two variations in the second and third configuration of the algorithm. Numerical experiments showed that using the acceptance of a perturbation performs best in the application of this study. Additionally, the initial temperature τ_a was set to 1.0, the alpha value α to 0.9, the acceptance counter a_S

to 12, and the termination criterion t_S to 3. This was done by fine-tuning these parameters for 120 different configurations under limited time.

Algorithm 6: `Simulated annealing (SA)`

Input: An initial solution b_a, a neighbourhood $\mathcal{H}(b)$, an initial temperature τ_a, a alpha value α, a cost function $c(\cdot)$ as well as an acceptance counter a_S and a termination criterion t_S
Output: The best solution b as a list of batched orders

$a_S, t_S \leftarrow 0$
$\tau \leftarrow \tau_a$
$b_a \leftarrow$ initiate a starting solution
$b \leftarrow b_a$
while the termination criterion t_S is not met **do**
  $b^{'} \leftarrow$ randomly select a solution in $\mathcal{H}(b)$
  if $c(b^{'}) \leq c(b)$ **or** accept $b^{'}$ with probability $\exp(-\frac{c(b^{'})-c(b)}{\tau})$ **then**
    $b \leftarrow b^{'}$
    $a_S = a_S + 1$
    $t_S = 0$
  else
    $a_S = 0$
    $t_S = t_S + 1$
  end if
  if the thermodynamic equilibrium is reached through acceptance counter a_S **then**
    $\tau = \tau \cdot \alpha$
    $a_S = 0$
  end if
end while
Return b

The great deluge (GD) algorithm is a variation of the SA algorithm. It differs in the acceptance of solutions and is easier to apply, since it only has one parameter that needs to be determined. The metaphor this algorithm uses is that of a hiker that tries to keep her feet dry while visiting the peaks of an unexplored area under a slowly rising water level ω [31]. Dueck [32] proposed this algorithm and it is applied to the OBP as illustrated in the pseudocode of Algorithm 7. The parameters have been calibrated to a rain speed ρ of 0.5 and a termination criterion t_G that is set to 5 with 40 different configurations for rain speed and termination criteria tested on the sample picking waves under time restriction.

3.6 Experimental results

The proposed picking location metrics that approximate the distances to pick orders before picker routing are tested on 50 sample picking waves. Also tested on the 50 sample waves are the algorithms that combine the orders into batches. Thereby, the best combinations of metric and algorithm are determined and the best combination to introduce order batching to a unidirectional cyclical picking line is identified. Information about the dataset, the experiment and the statistical analysis of the results are provided in the following sections.

All algorithms were implemented in Python 3.6 [94] utilising the C-based libraries Numpy [89] and Pandas [93]. These implementations were run on a Dell Optiplex 5050 with a Intel Core i7-7700 CPU at 3.6 GHz, 1x8GB 2400MHz DDR4 RAM, a 2.5" 256GB SSD class 20 drive, and the Microsoft Windows 10 Enterprise 2016 LTSB operating system [84]. The integer programmes were solved by means of Lingo 11.0.1.3. [72]. Additionally, IBM SPSS Statistics 25 [63] was used for the statistical analysis of the generated results.

Algorithm 7: `Great deluge (GD)`

Input: An initial solution b_a, a neighbourhood $\mathcal{H}(b)$, a rain speed ρ, a cost function $c(\cdot)$ as well as a termination criterion t_G

Output: The best solution b as a list of batched orders

$t_G \leftarrow 0$
$b_a \leftarrow$ initiate a starting solution
$b \leftarrow b_a$
$\omega \leftarrow c(b)$
while the termination criterion t_G is not met **do**
  $b^{'} \leftarrow$ randomly select a solution in $\mathcal{H}(b)$
  if $c(b) \leq c(b^{'})$ **then**
    $t_G = t_G + 1$
  end if
  if $c(b^{'}) < \omega$ **then**
    $s \leftarrow s^{'}$
    $\omega \leftarrow$ recalculate with $\omega - \frac{(\omega - c(b^{'}))}{\rho}$
    $t_G = 0$
  end if
end while
Return b

3.6.1 Data

A set of real life historical data were obtained from the Retailer. These data were made publicly available by Matthews and Visagie and can be accessed online [79]. For reporting purposes, 50 sample picking waves were randomly selected and divided into large data sets with more than 1 000 orders, medium data sets with 400 − 600 orders and small data sets with less than 100 orders. Within these data sets the picking waves are subdivided with respect to the number of SKUs into picking waves with a large, medium or small number of SKUs. The variety in size and location of the different retail stores together with the seasonality of the product portfolio lead to a set of non-uniform orders that is processed by the DC on a daily basis.

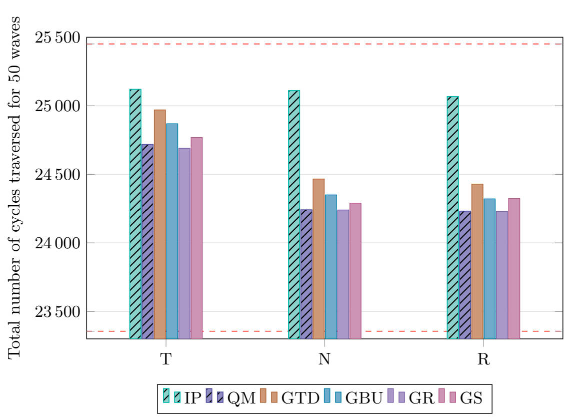
3.6.2 Computational results

The integer programming formulation (IP) in (3.1) − (3.3) was employed to combine orders based on the picking location metrics. The symmetric matrix resulting from applying one of the metrics was turned into an upper triangular matrix already containing all information needed to solve the problem more efficiently. Nevertheless, most test cases had to be stopped after the time limit proposed by Nicolas *et al.* [87] of 30 minutes. An average computational time of over 11 minutes per sample picking wave without a guaranteed solution within 30 minutes is not feasible in a real life application as planning of the lines need to be completed in seconds. The picking waves that can be solved to optimality do not justify the reduction in walking distance, since only the picking location metric is solved optimally. Solving to optimality is computationally expensive, thus implementing the IP on a stronger solver to get a guaranteed solution does not yield a return on time investment.

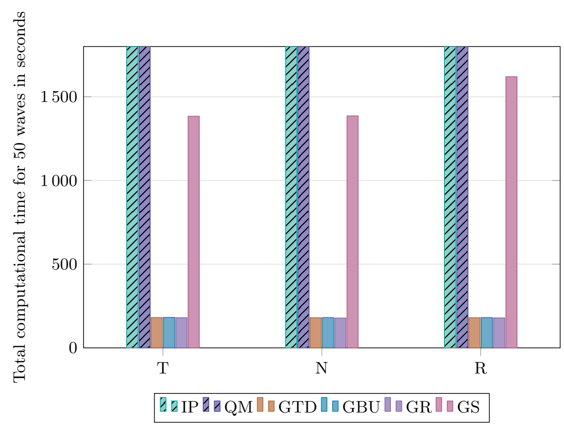
Applying the quick match algorithm (QM), seems to increase performance since the total number of cycles traversed is lower for QM than for IP as illustrated in Figure 3.4(a). Nevertheless, QM is also not able to guarantee a solution within the set time frame and has to be terminated before exceeding the time limit.

If an exhaustive optimisation of the picking system is the aim, two additional decision tiers have to be solved for each picking wave to determine where a SKU should be placed on the picking line and on which picking line a SKU should be picked. Additionally, the configuration in which the picking line is run needs to be determined. In consultation with the management of the Retailer a time frame of about 30 seconds per picking wave for combining all orders to batches would be feasible. Thereby, line managers do not have to wait longer than a minute to solve all decision tiers for a picking wave.

Besides the time constraint, an upper and a lower bound help to evaluate the performance of metric and algorithm combination in terms of solution quality. The worst case would be to not introduce order batching. The upper bound would thus be at 46 711 total cycles traversed. The best case would be to reduce the walking distance by 50%, thus resulting in an absolute lower bound of 23 356 total cycles traversed. Therefore, if no batching was introduced the cycles traversed could be halved and put next to each other. However, this would not result in a feasible solution as orders that overlap do not have the same starting and end location. Linking up the starting and end locations would result in a longer walking distance than the application of one of the picking location metrics. Additionally, the benchmark incorporating a FIFO approach as discussed in Section 3.4 will give a guideline of 25 451 cycles to evaluate the performance of the combinations compared against a random batching approach. The lower bound and the FIFO benchmark will be indicated by a red dashed line in Figure 3.4(a).



(a) *Cycles traversed per greedy heuristic.*



(b) *Computational time per greedy heuristic.*

FIGURE 3.4: *Total number of cycles traversed and computational time per greedy heuristic per metric.*

Greedy heuristics choose the next best combination of orders in a greedy fashion and thus speed up performance. All four greedy heuristics, namely the GTD, GBU, GR and GS, are tested per metric on the 50 sample picking waves. For five test runs, the GR performs best in terms of minimum number of cycles traversed on all 50 sample picking waves. It also performs slightly better than the QM. This is depicted in Figure 3.4(a). For 5 configurations, the standard deviation in total number of cycles traversed for the stops metric is 28.96, for the non-identical stops is 33.07 and for the stops ratio is 18.62. Calculating the coefficient of variation by division of the mean shows that all metrics have a low variation in relation to their means with stops at 0.12%, non-identical stops at 0.14% and stops ratio at 0.08%. As illustrated in Figure 3.4(b), all greedy heuristics have a comparable total computational time of approximately 180 seconds, because of the similar mechanism of the algorithms.

Metaheuristics are higher level heuristics that do not have to be adapted deeply to a specific problem, but that are able to escape local optima. Therefore, metaheuristics might get even closer to the optimal solution of the minimum number of cycles traversed within the predetermined time frame. All six metaheuristics ILS, VNS, VND, TS, SA and GD are combined with the three picking location metrics. Test runs showed that using GR to generate an initial solution performs better than initialising a random solution. Therefore, a hybrid of GR with each metaheuristic is analysed. Each algorithm was run five times with the same greedy starting solution for each run. However, the computational time restrictions resulted in the metaheuristics only being able to improve on the number of cycles traversed in combination with the stops metric. This is depicted in Figure 3.5. For the stops metric, the GD followed by the SA shows slightly better results in terms of minimum number of total cycles traversed and computational time than the other four metaheuristics.

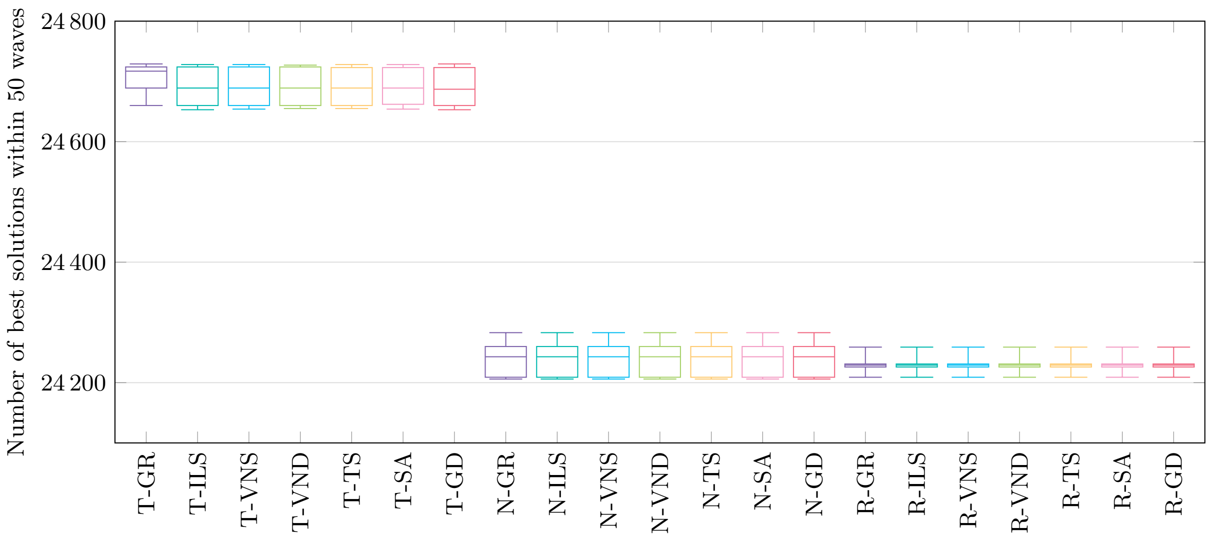


FIGURE 3.5: *Box and whisker plots of the total cycles traversed per metaheuristic per metric.*

All computational times per algorithm per metric for all 50 sample picking waves are illustrated in Figure 3.6. For all metrics the GR is the fastest, since all other metaheuristics are hybrids including the time of the GR algorithm. Nevertheless, the GR is followed by the GD- and SA-hybrid, while the VNS followed by the ILS and TS take the longest to get to a solution.

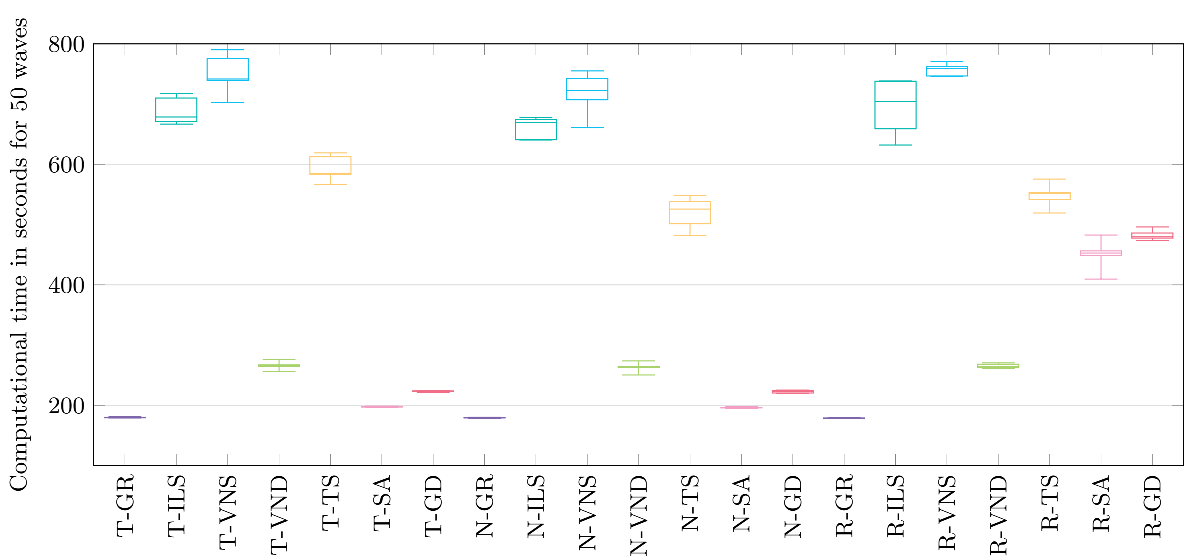


FIGURE 3.6: *Box and whisker plots of the computational time per metaheuristic per metric.*

The hybridisation between metaheuristics may also lead to a better solution quality and an improvement in computational time. Therefore, the combination of metaheuristics has been tested on the sample waves, but has shown no constant improvement under the time restriction.

Metaheuristics could not improve the solution quality under the time restriction for two out of three picking location metrics. This raises the question whether the solution found by GR is close to optimal (and thus few improvements exist) or the metaheuristics are incapable of finding improvements on a poor solution found by the GR. To evaluate this question a sample picking wave in which the IP was able to find the optimal solution within the time frame is analysed. For the stops metric, possible combinations on a sample picking wave with 1 336 orders and 55 SKUs are investigated. The IP is able to generate the best solution of 620 cycles that have to be traversed to pick all orders in 137.73 seconds. While the GR algorithm is able to get to 631 cycles in approximately 5 seconds, the GD- and the SA-hybrid are able to reach 629 cycles in almost the same time. The solutions are within 98.55%, and very close to the best solution. These results indicate that the more computational time a metaheuristics is allowed to use, the closer it can get to the best solution. However, if the walking distance could be reduced by 50% the lowest bound would be at 594 cycles. A sample picking wave with 1 356 orders and 51 SKUs is analysed for the non-identical stops metric. It takes 169.72 seconds for the IP to generate the best solution of 599 cycles that have to be traversed to collect all orders. None of the metaheuristics is able to improve on the GR solution within the given time restrictions. Nevertheless, the GR solution is within 99.50% the best solution for this metric, while the lower bound for this example would be at 573 cycles. For the stops ratio metric a sample picking wave with 1 354 orders and 56 SKUs is investigated. The IP generates the optimal solution of 583 cycles in 118.91 seconds that have to be traversed. No metaheuristic is able to improve on the GR solution within the predetermined time restriction. However, the solution is within 99.49% also very close to the best solution and the lower bound for this example is at 555 cycles.

Figure 3.5 suggests the combination of the stops metric with GR-GD-hybrid, the non-identical stops metric with GR, and the stops ratio metric with GR, because of the minimum number of total cycles traversed generated by these combinations. This is supported by the lowest total computational times for these algorithms as illustrated in Figure 3.6.

3.6.3 Statistical analysis

A regression analysis per location metric supports the application of location-based metrics as approximations for walking distance before picker routing. The correlation between the objective value per metric and the number of cycles traversed is evaluated. This results in $R^2 = 0.783$ for stops, $R^2 = 0.784$ for non-identical stops and $R^2 = 0.776$ for stops ratio. This shows a strong correlation between all metrics and the final walking distance. While the approximation purpose of each metric is validated, the regression analysis does not provide a basis for the comparison of metrics, since the objective values of all metrics are expressed in different units.

Further inferential statistics are necessary to investigate the influence of metrics and algorithms on the solution. Chiarandini *et al.* [23] propose either a univariate model for analysing solution quality and computational time, or a bivariate model combining both measurements. This analysis will focus on the solution quality in terms of total number of cycles traversed for all 50 picking waves, since the computational time has shown to be suitable for an application in the operations of the Retailer's DC. Therefore, A Welch-ANOVA with a Games Howell *post-hoc* test, that is robust to the equality of means, is used to analyse if there is a statistical difference by applying the three picking location metrics. A two-way ANOVA investigates the additional influence of the algorithm [101].

In Table 3.3, the Welch-ANOVA is statistically significant ($F(2, 102) = 3\,419.293, p = 3.3682E$-

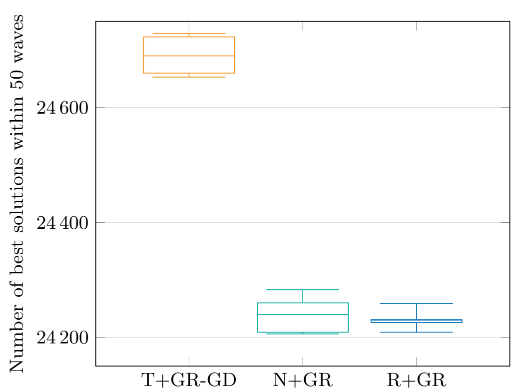
94) emphasising the influence of each metric on the number of cycles traversed. The *post-hoc* Games Howell test shows that the difference in cycles traversed is significantly lower when using N ($24\ 240 \pm 29.993, p = 5.1011E$-9) and R ($24\ 230 \pm 16.90, p = 5.1012E$-9) compared to T ($24\ 693 \pm 30.937$) [68, 102]. Nevertheless, the two-way ANOVA reveals no significant influence on the number of cycles traversed through the interaction between metrics and algorithms ($F(12, 84) = 0.047, p = 1.00$) [69, 104].

	Sum of squares	*df*	Mean square	*F*	*p*
Welch-ANOVA					
Between metrics	4 883 348.648	2	2 441 674.324	3 419.293	$3.3682E-94$**
Within metrics	72 836.914	102	714.087		
Two-way ANOVA					
Metrics	4 883 348.648	2	2 441 674.324	2 844.117	$6.9709E-78$**
Algorithms	241.962	6	40.327	0.047	1.00
Metrics × algorithms	480.952	12	40.079	0.047	1.00
Error	72 114.00	84	858.50		

Note: Two asterisks indicate significance at the 5% level or below.

Table 3.3: *Welch-ANOVA on metrics and two-way ANOVA on metrics and algorithms.*

Descriptive statistics are used to compare the best combinations of picking location metrics and their corresponding algorithms based on the homogeneous unit of total number of cycles traversed for all 50 sample picking waves. These combinations are depicted in Figure 3.4(a). In general, N and GR, and also R and GR combined perform better in terms of minimum number of cycles traversed when compared to the T and GR-GD hybrid combination.



(a) *Box-whisker plot for each combination.*

	T+GR-GD	**N+GR**	**R+GR**
Mean	24 693	24 240	24 230
Median	24 687	24 243	24 229
Range	76	77	52
Std. Dev.	30.937	29.993	16.900
Variance	957.104	899.558	285.600

(b) *Descriptive statistics.*

Table 3.4: *Comparison between best combinations of metric and algorithm.*

The stops ratio shows lower measures of central tendency than non-identical stops with a mean of 24 230 compared to 24 240 as illustrated in Table 3.4(b). Comparing the measurements of variability between R and N, the range between maximum and minimum total number of cycles traversed is 32.5% smaller for R. The standard deviation of R is almost half the size of N resulting in smaller variance and thus providing a metric-algorithm combination that seems more stable than N. Therefore, this comparison suggests choosing the combination of stops ratio metric and greedy random heuristic to introduce order batching to a unidirectional cyclical picking line.

3.7 Conclusion

These best combinations were then compared in terms of minimum number of cycles traversed for all 50 picking waves. The stops ratio and greedy random combination showed the lowest average of 24 230 cycles within the five test runs. Therefore, this combination is recommended to be applied if order batching is introduced to a unidirectional cyclical picking line. Compared to the FIFO benchmark of 25 451 cycles through applying random batching, this is 4.80% less walking distance. If batching is not introduced to the unidirectional cyclical picking line using this combination, pickers have to walk approximately 48.13% further. Additionally, this combination is only 3.74% higher than the absolute lower bound of reducing 50% of the walking distance. The reduction in walking distance through the combination of stops ratio and greedy random heuristic can be translated into time savings leading to a direct decrease of picking cost.

The analysis of the generated results indicate two findings. Firstly, the metric applied plays a more important role than the algorithm used to group the orders into batches. Secondly, the more information about the location of the items of an order is available, the better the results in terms of minimum number of cycles traversed.

3.8 Chapter summary

In this chapter order batching was introduced to a unidirectional cyclical picking line as deployed in the layout of a South African retailer's DC. Three location-based order-to-route closeness metrics were proposed to approximate the walking distance before picker routing and thus identify compatible orders. The picking location metrics include stops counting all locations a picker has to stop, non-identical stops counting the locations a picker has to stop to pick only one order, and stops ratio describing the ratio between non-identical and combined stops. The assignment problem of grouping orders in batches of two was solved with exact, greedy and metaheuristic solution approaches. The exact solution approaches take too much computational time to allow for an integrated optimisation approach. Therefore, four greedy heuristics including a top-down, bottom-up, random and smallest entry search approach, and also six metaheuristics, namely an iterated local search, a variable neighbourhood search, a variable neighbourhood descent, a tabu search, a simulated annealing and a great deluge algorithm were applied to 50 sample picking waves. The picking waves were recorded from real life data and vary in size by number of orders and SKUs. The algorithms terminated after a reasonable computational time restriction to allow for a real life application. All metrics and algorithms were tested in different combinations to identify the best combinations reducing the total number of cycles traversed. This results in the best combinations of stops and greedy random-great deluge, non-identical stops and greedy random, and also stops ratio and greedy random. The recommended combination is the stops ratio and greedy random that generated the lowest average of cycles traversed.

The stop based distance approximation metrics derived in this chapter showed promising savings in the total walking distance of pickers. In an attempt to improve on these savings, route or span based measures that incorporate more information about the specific layout of a unidirectional cyclical picking line are considered in the next chapter.

CHAPTER 4

Route overlap metrics for order batching on a unidirectional picking line

Contents

The proximity measures incorporated in the batching of orders seem to influence the reduction of walking distance significantly. So far, the batching metrics that have been suggested to introduce order batching to the unidirectional cyclical picking line are based on the picking location. Route overlap distance approximations are developed to add information about route similarity before picker routing. These route overlap metrics include information about the span of each order, meaning the section of the picking line that has to be traversed to pick an order. The best combinations of metric and algorithm are identified in a numerical experiment using real life data from the Retailer's DC. The results confirm the usability of route overlap metrics as approximations for walking distance before picker routing.

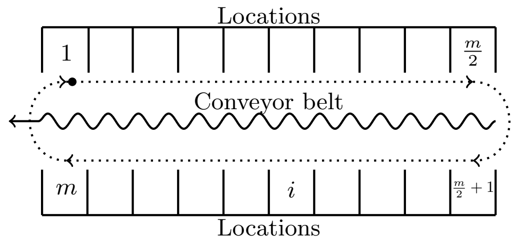
4.1 Introduction

A distribution centre (DC) forms a link to better match changing demand with reacting supply. Various factors influence the supply and demand of a product. Some of these factors include: product seasonality, product life span, transport networks, and delivery efficiency [98]. Even though a DC requires resources such as space, labour, equipment and information it is unlikely to vanish from the present economic environment. A DC and other logistical services are interdependent. Their aim is to improve customer service by delivering products efficiently while keeping transport costs to a minimum [35]. A DC collects products from various suppliers then delivers those products to different end customers. Bulk supplies arrive at the DC and are then turned into specific customer orders [113].

Rouwenhorst *et al.* [98] divide DC activities into three different functions, namely receiving, storage, order picking, and shipping. Warehousing is expensive and accounts for 2% to 5% of a company's cost of sales. Therefore minimising the cost of operation of a DC has become a focus point. However, the emphasis on customer services limits the extend to which cost can be minimised [35]. The basic service of a DC is order picking that makes up for 50% to 65% of the operational cost [113]. Most DCs rely on manual order picking systems because of the diversity in product size and the velocity of product portfolio that need to be picked. This results in high labour cost [49]. According to De Koster *et al.* [29] order picking describes the process of retrieving items from storage or buffer areas in response to a specific customer request. The key parameters that are of importance in a picking system is the size of the orders and the number of orders that are picked on a daily basis [27].

The order picking system of a prominent South African retailer (referred to as the Retailer) is considered in this study. Its DC needs to handle a variety of orders for a large number of stores with different sizes and various customer profiles on a daily basis. The central planning of the inventory at store level is a key characteristic of the Retailer's operations. Planners at a central planning department assign available stock from the DC to the stores instead of store managers requesting stock directly. This means that the central planning department allocates all available stock keeping units (SKUs) to the stores. At the central planning department the planners decide on the number of SKUs destined for each store upon the arrival of the SKUs at the DC. Planners issue instructions to the DC about SKUs and the stores where they should go. A subset of SKUs to be picked in a single picking cycle, called a wave, is then selected by the DC to satisfy store requirements for that set of SKUs. Therefore, the term *order* will refer to the store requirements for a single store for all the SKUs present in a wave. A picking line processes a picking wave in the DC. For each wave, a single SKU is assigned to a single location on a specific picking line. A *picking wave* involves populating the line with SKUs, the actual picking process, and removing excess stock from the line [78].

A representation of a picking line with m locations is depicted in Figure 4.1(a), while a photo of an actual picking line without SKUs is shown in Figure 4.1(b). The picking line at the Retailer's DC in Durban comprises of 56 locations with a conveyor belt that is placed in the middle and that has two gates for access. A single SKU from the storage area is assigned to a single location, thereby having a specific location on the picking line. Prior to the start of the picking wave, the orders for all stores are known, which means that the number of SKUs to be picked for each store is known. Each location can store up to five pallets (depicted on the rollers of Figure 4.1(b)) containing a particular SKU. Additional stock is kept on the floor (between the picking lines) for easy access. Therefore no stock has to be replenished from storage racks during a picking wave. Pickers move in a clockwise direction around the conveyor belt to pick all orders. They are guided through the picking processes by a voice recognition system (VRS) that sends the



(a) *Schematic representation of a picking line with m locations. Source: De Villiers [30].*

(b) *Empty picking line at the DC.*

FIGURE 4.1: *Representation of a unidirectional cyclical picking line.*

picker to the closest required SKU. The underlying picking strategy can thus be summarised as pick the closest SKU in a clockwise direction. This picking system falls under the category of a picker-to-parts system. An empty carton is placed on a trolley and registered by a barcode with the VRS before an order is started. Full cartons and finished orders are placed on the conveyor belt after picking. Cartons are then moved to the dispatch area for further processing [76, 77].

For the unidirectional cyclical picking line, Matthews and Visagie [78] introduced the *number of cycles* that have to be traversed to pick all items requested during a picking wave as a measure of distance travelled to pick all orders. This measure adds up the number of cycles required by each individual order and the number of cycles required to link all orders and can be used to compare the effectiveness of different methodologies introduced to this specific picking system.

The distance that has to be covered to collect all items of an order, given a starting location, is called a *span* as defined by Matthews and Visagie [78]. For order i a span may be represented by $P_i^a = \langle a, e_i^a \rangle$, with e_i^a the closest ending location of order i starting at position a. For every order, the shortest span, called the *minimum span*, is calculated. This is done by searching for the biggest gap, which refers to the number of locations that a picker passes without stopping to pick items. The end of this gap will be the start of the minimum span, while the beginning of the gap is the end of the minimum span. Consider the example in Figure 4.2. This picking line consists of 10 locations. Order 1 needs SKUs from Locations $2, 3, 7, 8$ and 9. Figure 4.2(a) illustrates all spans of Order 1 starting at each location that accommodates a SKU. The first gap includes Location $4, 5$ and 6, while the second gap covers Location 1 and 10. The first gap can be identified as the biggest gap in this example, since it includes three locations. The minimum span to collect all SKUs for Order 1 starts at Location 7 and ends at Location 3 when the biggest gap is taken into consideration. The minimum span is depicted by the red arrow in Figure 4.2(b). The length of the minimum span of this order is denoted by $|P_1^{\min}| = \langle 7, 3 \rangle = 7$ locations.

If the SKUs are viewed as moving relative to a static picker, this layout shows similarities to unidirectional carousel systems. Small- and medium-products are usually picked with the help of automated carousel warehouse systems. A carousel consists of a rotatable circuit of shelving holding multiple SKUs. A picker, who operates the system, remains at a fixed location throughout the picking process. This picking system is referred to as a parts-to-picker system in literature, since the storage location moves to the picker instead of the picker moving to the storage location. A carousel that can rotate in both directions to turn the required SKU towards the picker is called a bidirectional carousel, while a unidirectional carousel only turns in one direction [7]. The cyclical set up of the picking line is similar to a unidirectional carousel. One order is processed by one picker at a time. Also, picker movement is directed by a VRS which resembles the automated sequencing of orders in a carousel. However, the main difference

(a) *All possible spans of Order* 1.

(b) *The minimum span of Order* 1*, denoted as* $P_1^{min} = \langle 7, 3 \rangle$.

FIGURE 4.2: *Schematic representation of spans for an order in a picking line. The shape indicates a specific SKU, while the colour indicates which order requires that SKU to be picked.*

in the Retailer's picking system is the presence of wave picking. In literature on carousel systems, optimisation approaches make use of expected SKU mixes derived from historical data, since not all SKUs may be picked and (new orders may be added to the set during the picking operation). Research and industry implement bidirectional carousel systems more frequently, because they are generally more efficient [73]. However, in this case, bidirectional movement of pickers (corresponding to a bidirectional carousel) cannot be used because it would cause picker congestion. The deterministic nature of the orders in the Retailer's picking system is unique in operations research literature [78].

The order batching problem (OBP) forms part of the planning for a wave of order picking. Orders are determined by the set of SKUs included in a picking wave. All the orders (that are unique to a wave) have to be picked during that picking wave [42]. The two fundamental approaches in manual order picking are single order or batch picking. While one picker is only responsible to pick one order at a time in single order picking, in batch picking multiple orders are picked simultaneously by the same picker. Orders can either be sorted-while-picked or picked-and-sorted after the batch picking process to release individual orders. Efficient tours that shorten the total picking time can be generated in this way, since several orders are picked simultaneously. The decrease in picking time can be translated into a reduction of labour cost [28]. The number of orders that can be batched is restricted by the capacity of the picking device [120]. Splitting of orders is usually prohibited, since it would result in an additional sorting effort [50].

While there has been intensive research in single-block warehouses with parallel aisles, according to Nicolas *et al.* [87] little research has been published on the OBP in automated carousel systems. Hofmann and Visagie [55] introduced order batching to the specific layout of the unidirectional cyclical picking line. They developed picking location metrics to approximate walking distance between orders, since calculating walking distance before picker routing would be very time consuming. However, basing the metrics on the picking locations where a picker stops to collect stock, excludes the locations the picker simply passes to get to the next required SKU. The idea to add spans to the metrics aims at overcoming this limitation by including more information about each order. This chapter searches for the best combination of span-based metrics in terms of total distance travelled by pickers to collect all orders during a picking wave.

The literature on the OBP in warehouses with a single-block parallel aisles or automated storage and retrieval layout is reviewed in Section 4.2. In Section 4.3, the route overlap metrics are developed comparing different approaches while using greedy heuristics to combine orders into batches. In Section 4.4, a case study with real life data from the Retailer is used to test the performance of the route overlap metrics against the picking location metrics proposed by Hofmann and Visagie [55]. Thereby the option generating the shortest overall walking distance to pick all orders in a wave is identified. In Section 4.5 a summary of this chapter and an outlook on future research opportunities is provided.

4.2 Literature review

Gademann and Velde [36] introduced the formulation of the standard OBP for a single-block warehouse with parallel aisles environment and proved that it is NP-hard in the strong sense. Nevertheless, the problem can be solved in polynomial time if no batch contains more than two orders. Test instances of up to 32 orders were solved to optimality with the branch-and-prize algorithm developed by them. Additionally, test instances of up to 40 orders were solved by Henn and Wäscher [51]. However, a time-consuming preprocessing to determine all feasible batches is required in their solution approach. Test instances of up to 100 orders were solved by Muter and Öncan [86] with their tailored column-generation based algorithm. Nicolas *et al.* [87] introduced the OBP to a vertical carousel system that is closely related to the layout of a unidirectional cyclical picking line. They formulated the OBP as a mixed integer linear program (MILP). However, due to time constraints, the MILP has to be stopped after 30 minutes and could thus not solve test instances over 50 orders.

In a practical application considered in this paper the number of orders is likely to exceed the number of these test instances as a picking line at the Retailer's DC can service in excess of 1 500 orders in a single wave of picking. Construction heuristics such as seed and saving algorithms have been introduced to solve larger problem instances of the OBP.

Seed algorithms first initiate batches, then allocate orders to batches and terminate by fulfilling a stopping criterion when the batches are completed. The objective is to minimise the total travel distance that is needed to collect all orders. Different seed algorithms were evaluated by De Koster *et al.* [28] in the standard single-block parallel aisles warehouse layout. They proposed the farthest storage location as a seed selection rule. Several location- and aisle-based seed algorithms have been investigated by Ho and Tseng [53]. Ho *et al.* [52] extended this study with additional distance- and area-based selection rules. They suggested the smallest number of picking locations, the smallest number of picking aisles, and the smallest rectangular covering area selection rules. A comparative study conducted by Pan and Liu [92] evaluated four initial seed selection and four order addition rules recommending the economic convex hull method selection rule. They investigated the OBP in an automated storage and retrieval system that consists of a single storage rack with equally sized storage locations serviced by a single storage and retrieval machine. All investigated layouts differ from the unidirectional cyclical picking line implemented by the Retailer.

In the OBP, saving algorithms are based on the Clarke and Wright-algorithm (CW) [24]. Therefore time saving is obtained by comparing the collection of orders in one route to the collection of individual orders [28]. Rosenwein [97] proposed the minimum additional aisles heuristic that begins with calculating a score for each pair of orders with regards to additional aisles to pick. Their algorithm in comparison to Gibson and Sharp [37] is thus capable of generating fewer but shorter picking tours. De Koster *et al.* [28] describes the first OBP version of the Clarke and Wright-algorithm (CW I) that computes savings for each combination of orders. Each time a new combination of order batches are determined the second version of the algorithm (CW II) recalculates the savings. This version of the algorithm was improved by Bozer and Kile [15] who introduced a normalised time saving value for each pair of orders. The initial savings matrix is modified each time an order has been assigned. This resulted in the third version of the algorithm (CW III) introduced by De Koster *et al.* [28]. Additionally, a small and large algorithm was proposed by Elsayed and Unal [34]. According to a predefined value orders are either classified as small or large before they are assigned to batches. Because of the dynamic batch comparison, saving algorithms are in general more complex and thus demands more CPU time than seed algorithms.

Central to all types of heuristics is the order-to-route closeness metric [42]. Hofmann and Visagie [55] introduced the order batching problem to the layout of a unidirectional cyclical picking line. The objective to minimise the walking distance of pickers was achieved by developing three picking location-based (also called stop-based, meaning the locations where pickers must stop to pick an order) order-to-route closeness metrics to approximate walking distance between orders before picker routing. The stops, non-identical stops, and stops ratio metrics are used to combine similar orders to batches through exact solution approaches, greedy heuristics, and metaheuristics. In terms of minimum number of total cycles traversed and computational time, the combination of stops ratio metric and greedy random heuristic generates the lowest number of cycles traversed. Nevertheless, a limitation of these metrics is that the information about the locations that are passed on the picking line are not included when calculating the metrics.

4.3 Batching metrics

This paper considers the same picking set up as in Matthews and Visagie [78] and Hofmann and Visagie [55]. For completeness, the main characteristics are summarised here. The batching metrics are developed, and include information about the order spans in the spans metric, the combination metrics, and the addition metrics.

The DC of the Retailer is made up of several unidirectional cyclical picking lines. All picking lines function in the same manner. A single picking line is thus considered for the remainder of this chapter.

While modelling the Retailer's order picking system the following assumptions are made.

1. The SKUs and locations of an order that will be picked during a wave are fixed *a priori.*
2. The trolleys are small to increase moveability of pickers, even though SKUs are generally bulky. Therefore, the capacity of a trolley and thus the batch size is restricted to two orders per batch.
3. A picker must complete the entire batch that is currently processed before a next batch can commence.
4. A picker is assumed to walk at a constant speed, since the aisles are wide enough for pickers to pass.
5. Each batch currently has to start and end at the same location as the orders it incorporates.

Currently, each order is picked by one picker that completes one order at a time. The nearest end heuristic, as proposed by Matthews and Visagie [78], is an easily implementable option to determine a sequence of orders that minimises the distance pickers have to travel during a wave. The order with the nearest ending position from the picker's current position is sequentially selected in the nearest end heuristic. It considers the order sequence and the item sequence within the order simultaneously.

An example of a unidirectional cyclical picking line with 10 locations and 10 SKUs is considered. The locations of the orders in this small picking line are presented in Table 4.1.

Locations:	1	2	3	4	5	6	7	8	9	10
Order 1		◒	⊕				⬗	□	⬗	
Order 2				⊗	△		⬗	□		
Order 3			⊕		△	◇	⬗	□		⬠
Order 4	○	◒			△	◇	⬗			

TABLE 4.1: *The locations of the orders in the small picking line.*

In Figure 4.3 the colours green, blue, red and yellow represent four different orders, while the shapes indicate SKUs in various locations. Applying the nearest end heuristic to this example, Order 4 (yellow) would be picked first starting at Location 1 and ending at Location 7. This would be followed by Order 1 (green), Order 2 (blue) and Order 3 (red). Thereby a picker picking all orders of the picking wave would have to traverse four cycles. The nearest end heuristic can be used to compare the effectiveness of other batching techniques in terms of cycles traversed, since it is proven to produce good solutions and is easily reproducible [78].

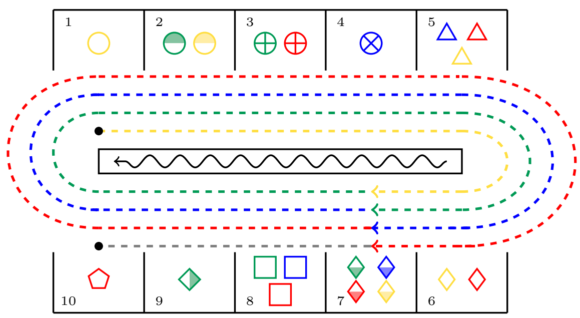


FIGURE 4.3: *Schematic representation of order picking in the small picking line with 10 SKUs and 10 locations. The shape indicates a specific SKU, while the colour indicates which order requires that SKU to be picked.*

Implementing a first-in-first-out approach (FIFO) would be the easiest way to introduce order batching to a unidirectional cyclical picking line. The first entries of a list of orders are grouped together until a predetermined maximum batch size is reached by the FIFO rule [37]. This rule is used as a benchmark to evaluate different batching methods. Applying the FIFO rule as the random batching strategy with a maximum batch size of two, the first two orders would make up Batch 1 and Orders 3 and 4 would form Batch 2. The two batches are depicted in Figure 4.4 in orange and purple respectively. Illustrated by the circles representing the picker movement, only two cycles have to be traversed now. Hence, the different batching metrics are compared with regards to the number of cycles traversed that indicates the total walking distance.

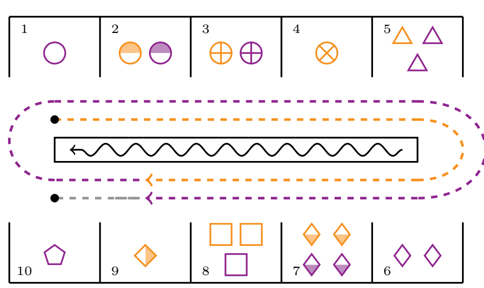


FIGURE 4.4: *Schematic representation of order batching with FIFO applied to the small picking line.*

For this chapter, the focus is to include information about the minimum spans in metrics that approximate distance before routing, so that the total walking distance can be minimised. Similarities between orders can be evaluated by comparing their minimum spans, since the minimum span of an order is order specific, and provides the shortest route in which all items of an order can be collected. For example, the minimum span of Order 4 in Figure 4.3 is $P_4^1 = \langle 1, 7 \rangle$ with a length of $|P_4^{\min}| = 7$ locations is the same as the span in which it is collected using the nearest end heuristic. However, Order 1 has to start from Location 8 thus resulting in a span of $P_4^8 = \langle 8, 7 \rangle$ with a length of $|P_4^8| = 10$ locations, whereas the minimum span would have been $P_1^7 = \langle 7, 3 \rangle$ with a length of $|P_1^{\min}| = 7$ locations. Orders can be collected in different spans depending on their starting location, but their length can never exceed one cycle. However, the information on the most favourable span in which items of an order should be picked is only available after routing. Therefore, the following order-to-route closeness metrics (which include information about the locations passed) are based on the definition of the minimum span specific to the layout of a unidirectional cyclical picking line.

The same solution approaches to combine orders into batches *a priori* picker routing that have been described by Hofmann and Visagie [55] are applied to the route overlap metrics developed here. Metaheuristics did not improve the solution quality of the order batching problem. Also, solving the picking location metrics to optimality did not show improvements in walking distance in their study. Greedy heuristics provided the best results in terms of solution quality and computational time [55]. Therefore the greedy random heuristic (GR) and the greedy smallest entry heuristic (GS) will be used in this chapter and are described next.

The GR and GS heuristic only differ in their search mechanism, while the elimination process stays the same. This is displayed in Algorithm 8. The GR heuristic searches matrix $\mathbf{M}$ with entries m_{ij} (resulting from the batching metric) according to a random permutation. The GS heuristic, on the other hand, starts by searching for the row with the smallest entry. The global smallest entry in matrix $\mathbf{M}$ is the entry that has the biggest difference between smallest and second smallest entry. Therefore this entry must be batched next. Indicating that orders k and q are batched, the tuple that corresponds to the minimum entry m_{kq} of row k is recorded in set $\mathcal{B}$. Both rows and columns k and q are then removed from $\mathbf{M}$. These steps are repeated for both algorithms until set $\mathcal{B}$ has the cardinality of $n/2$, with the problem size n.

Algorithm 8: `Greedy heuristic (GR or GS)`

Input: A batching metric comprising a $n \times n$ matrix $\mathbf{M}$ with entries m_{ij}, an empty solution set $\mathcal{B}$
Output: The solution set $\mathcal{B}$ as a list of batched orders

```
1: B ← ∅
2: while | B | < n/2 do
3:     k ← search mechanism of GR or GS
4:     m_kq = min_j [m_kj]
5:     B ← B ∪ (k, q)
6:     Remove both rows and columns k and q from M
7: end while
8: Return B
```

The batching metrics aim at minimising the incompatibility between orders in terms of distance travelled between orders to minimise walking distance before picker routing. The combination of metrics and algorithms are thus tested on a representative set of real life historical data to determine the total number of cycles traversed for different order combinations. This data was obtained from the Retailer and made publicly available by Matthews and Visagie [79]. For reporting purposes 50 sample picking waves have been recorded and divided into large data sets with more than 1 000 orders, medium data sets with 100 – 1 000 orders and small data sets with less than 100 orders. The picking waves are subdivided with respect to the number of SKUs into picking waves with a large, small or medium number of SKUs within these datasets. A set of non-uniform orders that has to be processed by the DC on a daily basis is the result of the variety in size and location of the different retail stores together with the seasonal product portfolio. In the experimental set up of this chapter, the total number of cycles traversed for the 50 sample picking waves is added up to determine the effectiveness of each batching metric in reducing walking distance.

All algorithms are implemented in Python 3.6 [94] utilising the C-based libraries Numpy [89] and Pandas [93]. These implementations were run on a Dell Optiplex 5050 with a Intel Core i7-7700 CPU at 3.6 GHz, 1x8GB 2400MHz DDR4 RAM, a 2.5" 256GB SSD class 20 drive, and the Microsoft Windows 10 Enterprise 2016 LTSB operating system [84]. IBM SPSS Statistics 25 [63] was used for the statistical analysis of the results.

In the next sections, the batching metrics based on the definition of the minimum span are introduced and compared to the picking location metrics proposed by Hofmann and Visagie [55]. The best performing picking location metrics from Hofmann and Visagie [55] are the stops and great deluge (T-GR-GD), the non-identical stops and greedy random (N-GR), and the stops ratio and greedy random (R-GR) combination. These three metrics will be compared to the route overlap metrics introduced here. The logical extension from the stop metrics is the development of span metrics. The spans lead to combination metrics which finally result in route overlap addition metrics.

The small example described in Table 4.1 and illustrated in Figure 4.3 is used to provide a matrix of the values for each of the batching metrics as an illustration of that metric. Only the upper triangular matrices are given as each metric produces a symmetric matrix with no entries on the diagonal (as an order cannot be batched with itself). The integer programming formulation (IP) described by Hofmann and Visagie [55] may be used to batch orders of this small example and thus provide optimal (in terms of that metric) batches. The batches obtained by the IP formulation are emphasised by frames around the matrix entry for that metric. The frame indicates which pair of orders should be batched together before routing the pickers.

4.3.1 Stop metrics

For completeness, the picking locations metrics from Hofmann and Visagie [55] are summarised here. If the sets $\mathcal{S}_i$ and $\mathcal{S}_j$ contain all stops for orders i and j respectively, the stops metric (T) can be calculated as

$$t_{ij} = |\mathcal{S}_i| + |\mathcal{S}_j| - |\mathcal{S}_i \cap \mathcal{S}_j|, \text{ with } i \neq j. \tag{4.1}$$

Similarly the smallest number of non-identical stops (N) can be calculated as

$$n_{ij} = |\mathcal{S}_i| + |\mathcal{S}_j| - 2\,|\mathcal{S}_i \cap \mathcal{S}_j|, \text{ with } i \neq j. \tag{4.2}$$

Finally, the stops ratio (R) can be calculated as

$$r_{ij} = \frac{n_{ij}}{t_{ij}}, \text{ with } i \neq j. \tag{4.3}$$

Applying each picking location metric to the small picking line example results in Table 4.2 with the batched orders emphasised accordingly.

4.3.2 Span metrics

In the first step, three different span-based metrics are developed along a similar logic to the picking location metrics introduced by Hofmann and Visagie [55]. However, the basis of the

$$\mathbf{T} = \begin{array}{c c} & \begin{array}{ccc} 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 7 & \boxed{8} & 7 \\ - & 7 & \boxed{6} \\ - & - & 8 \end{bmatrix} \end{array}$$

(a) *The* **T** *matrix.*

$$\mathbf{N} = \begin{array}{c c} & \begin{array}{ccc} 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 6 & 5 & \boxed{5} \\ - & \boxed{5} & 5 \\ - & - & 6 \end{bmatrix} \end{array}$$

(b) *The* **N** *matrix.*

$$\mathbf{R} = \begin{array}{c c} & \begin{array}{ccc} 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 0.86 & 0.63 & \boxed{0.71} \\ - & \boxed{0.71} & 0.83 \\ - & - & 0.75 \end{bmatrix} \end{array}$$

(c) *The* **R** *matrix.*

TABLE 4.2: *The* **T**, **N** *and* **R** *matrices for the small picking line. The best batching of orders for each metric is indicated by the framed numbers in that matrix.*

span metrics is not the number of stops, but the order specific minimum span described by $P_i^{\min}$ and $P_j^{\min}$ for order i and j respectively. The notation $|P_i^{\min} \Cap P_j^{\min}|$ indicates the length of the overlap in number of locations between $P_i^{\min}$ and $P_j^{\min}$. The complement of $P_i^{\min}$, denoted as $\widetilde{P}_i^{\min}$ represents the biggest gap of order i. Adding $P_i^{\min}$ and $\widetilde{P}_i^{\min}$ would thus result in one complete cycle.

The minimum spans metric s_{ij} is similar to the stops metric containing all picking locations and passed locations that make up the minimum span of order i and order j. Therefore the minimum spans matrix **S** with element s_{ij} can be calculated as

$$s_{ij} = (|P_i^{\min}| + |P_j^{\min}| - |P_i^{\min} \Cap P_j^{\min}|), \text{ with } i \neq j. \tag{4.4}$$

The non-identical spans metric has two options. On the one hand, non-identical minimum span can only take the part of a minimum span that is not the same between two orders. This is comparable to the number of non-identical stops metric. The matrix **D** with element d_{ij} can be calculated by

$$d_{ij} = (|P_i^{\min}| + |P_j^{\min}| - 2|P_i^{\min} \Cap P_j^{\min}|), \text{ with } i \neq j. \tag{4.5}$$

On the other hand, non-identical span can also consist of the overlap in biggest gap between two orders as in the e_{ij} metric. The logic is thus similar to the spans metric. The matrix **E** with element e_{ij} can thus be calculated by

$$e_{ij} = (|\widetilde{P}_i^{\min}| + |\widetilde{P}_j^{\min}| - |\widetilde{P}_i^{\min} \Cap \widetilde{P}_j^{\min}|), \text{ with } i \neq j. \tag{4.6}$$

The spans ratio a_{ij} is similar to the stops ratio metric, where the ratio between the non-identical part of the minimum span and the combined minimum span for two orders is calculated. The matrix **A** with element a_{ij} can be calculated as

$$a_{ij} = \frac{d_{ij}}{s_{ij}}, \text{ with } i \neq j. \tag{4.7}$$

Applying the spans metric to the small picking line example would result in matrix **S** as depicted in Table 4.3(a). Batching Order 1 with $P_1^3 = \langle 7, 3\rangle$ with $|P_1^{\min}| = 7$ locations and Order 2 $P_2^8 = \langle 4, 8\rangle$ with $|P_2^{\min}| = 5$ locations results in the spans metric $s_{12} = (7 + 5 - 2) = 10$. Therefore Order 1 and 3 (and thus Order 2 and 4) would form a batch. The non-identical minimum span metric results in matrix **D** described in Table 4.3(b) with Order 1 and Order 4 (and thus Order 2 and 3) batched. Order 1 and 3 (and thus Order 2 and 4) are batched according to matrix **E** in Table 4.3(c) resulting from an application of the non-identical span metric. Applying the spans ratio metric results in matrix **A** as depicted in Table 4.3(d). Order 1 and 4 (and thus Order 2 and 3) would form a batch.

The combination of the S-GR produced the lowest number of cycles traversed for the 50 sample picking waves for the spans metric. Combining D-GS as well as E-GR generates the lowest

$$\mathbf{S} = \begin{array}{cc} & \begin{array}{ccc} 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 10 & \boxed{10} & 10 \\ - & 8 & \boxed{8} \\ - & - & 10 \end{bmatrix} \end{array}$$

(a) *The* **S** *matrix.*

$$\mathbf{D} = \begin{array}{cc} & \begin{array}{ccc} 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 8 & 5 & \boxed{6} \\ - & \boxed{3} & 4 \\ - & - & 5 \end{bmatrix} \end{array}$$

(b) *The* **D** *matrix.*

$$\mathbf{E} = \begin{array}{cc} & \begin{array}{ccc} 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 8 & \boxed{5} & 6 \\ - & 5 & \boxed{6} \\ - & - & 5 \end{bmatrix} \end{array}$$

(c) *The* **E** *matrix.*

$$\mathbf{A} = \begin{array}{cc} & \begin{array}{ccc} 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 0.8 & 0.5 & \boxed{0.6} \\ - & \boxed{0.4} & 0.5 \\ - & - & 0.5 \end{bmatrix} \end{array}$$

(d) *The* **A** *matrix.*

TABLE 4.3: *The matrices* **S**, **D**, **E** *and* **A** *for the small picking line, with the best batching of orders for each metric shown by the framed numbers in that matrix.*

number of total cycles traversed for the non-identical spans metrics. The lowest number of cycles traversed for the spans ratio metric in the 50 sample picking waves is generated by A-GS. Nevertheless, none of the span-based metrics show lower numbers of total cycles traversed than the picking location metrics as depicted in Figure 4.5. Exclusively using the minimum span of each order does not generate better results, since longer individual spans can sometimes be used to combine two orders resulting in a shorter combined span. Only counting the picking locations where a picker stops allows for different spans and thus leads to a lower number of total cycles traversed. This result confirms the findings of Matthews and Visagie [78] on picker routing in a unidirectional cyclical picking line. Their travelling salesman approach outperforms the shortest spanning interval linking in number of cycles traversed, since it allows for different (longer) span options rather than just the minimum spans. Therefore, another approach to combine stops and spans in a batching metric is investigated.

4.3.3 Combination metrics

In this section stops and spans are combined in a single metric. Orders are sorted in a list ϕ. They are batched with an accompanying order according to matrix $\mathbf{M}$ produced by one of the batching metrics. All combination configurations are identified in the first column of Table 4.4.

Metric	List ϕ	Matrix M
F-CO	ϕ_s	S
P-CO	ϕ_s	D
Q-CO	ϕ_s	E
H-CO	ϕ_s	A

(a) *Stops-spans combinations.*

Metric	List ϕ	Matrix M
K-CO	ϕ_p	S
L-CO	ϕ_p	D
O-CO	ϕ_p	E
G-CO	ϕ_p	A

(b) *Spans-spans combinations.*

Metric	List ϕ	Matrix M
U-CO	ϕ_p	T
V-CO	ϕ_p	N
W-CO	ϕ_p	R

(c) *Spans-stops combinations.*

TABLE 4.4: *Route overlap combination metric configurations. The name of the combination metric is in the first column, the second column describes the logic of the list according to which the orders are sorted, and the distance approximation metric is in the third column of each table.*

The combination heuristic (CO) displayed in Algorithm 9 is used to merge all configurations. Thereby short orders are batched first to generate a lower number of total cycles traversed.

The first combination configuration starts with calculating the number of picking locations per order. The orders are sorted from their smallest to largest number of stops in list ϕ_s. Algorithm 9 selects the first order for batching according to ϕ_s and the second order according to a matrix provided by one of the span metrics from Section 4.3.2. All combination configurations of the stops-spans combinations are depicted in Table 4.4(a) and result in the combinations F-CO, P-CO, Q-CO and H-CO.

The minimum span can also be used for sequencing orders. Therefore orders are sorted from the shortest minimum span to the longest in a list ϕ_p. The second combination configuration selects the first order according to list ϕ_p, while the second order of the batch is selected from one of the span metrics. In Table 4.4(b) all configurations of the spans-spans combinations are described resulting in K-CO, L-CO, O-CO and G-CO.

Algorithm 9: `Combination heuristic (CO)`

Input: An ascending list ϕ, a batching metric comprising a $n \times n$ matrix $\mathbf{M}$ with entries m_{ij}, an empty solution set $\mathcal{B}$
Output: The solution set $\mathcal{B}$ as a list of batched orders

$\mathcal{B} \leftarrow \emptyset$
while $|\mathcal{B}| < n/2$ **do**
  $k \leftarrow$ first order in ϕ
  $m_{kq} = \min_{j}[m_{kj}]$
  $\mathcal{B} \leftarrow \mathcal{B} \cup (k, q)$
  Remove both rows and columns k and q from $\mathbf{M}$
  Remove the orders corresponding to k and q from ϕ
end while
Return $\mathcal{B}$

In the third combination configuration, the first order for batching is selected according to list ϕ_p and the second order is selected according to a matrix provided by the stop metrics from Section 4.3.1. The combination metrics U-CO, V-CO and W-CO result from the configurations of the spans-stops combinations as depicted in Table 4.4(c).

Using the shortest minimum span as a basis leads to better results in terms of total number of cycles traversed, since it includes more information about the specific locations of the items of an order. Starting the search with the shortest minimum span and then checking a matrix of picking location metrics as in the spans-stops combination results in the lowest number of total cycles traversed when all combination metrics are compared.

In Figure 4.5 the combination metrics cut down the number of cycles traversed when compared to the span metrics. The combination of spans and stops improves the results compared to the stops metrics of the picking location metrics. However, it still does not provide results as good as the non-identical stops or the stops ratio metric. Sorting the stops or spans from smallest to largest in a list leads to multiple orders with the same number of stops or the same length of spans. This can lead to multiple order candidates that could be included in the next batch. Changing the sequence of a list does not improve or worsen the results significantly. To overcome this specific location information with regards to both picking locations and passed locations is incorporated in the next batching metric.

4.3.4 Addition metrics

In the third extension, span metrics are added to picking location metrics. The route overlap addition metrics will therefore contain information about both the spans and the stops. All different possible combinations between the stops, non-identical stops, stops ratio, minimum span, non-identical minimum span, non-identical span, and the spans ratio metric have been tested.

The non-identical span is added to the stops metric resulting in the stops non-identical spans metric in matrix $\mathbf{Z}$ with general element z_{ij}. It merges the number of picking locations for two orders with the combined span that is not part of a minimum span for those two orders. The first part of this metric is adopted from Hofmann and Visagie [55] and described in equation (4.1). The idea of the second part originates from the smallest additional covering area introduced by Ho *et al.* [52], but is adjusted for this system. This second part is added according to equation (4.6). Thereby the orders with the smallest number of combined picking locations and shortest non-identical span are selected to form a batch.

The stops non-identical spans metric z_{ij} is calculated by

$$z_{ij} = (|\mathcal{S}_i| + |\mathcal{S}_j| - |\mathcal{S}_i \cap \mathcal{S}_j|) + (|\widetilde{P}_i^{\min}| + |\widetilde{P}_j^{\min}| - |\widetilde{P}_i^{\min} \Cap \widetilde{P}_j^{\min}|), \text{ with } i \neq j. \tag{4.8}$$

In the second addition metric, the number of non-identical picking locations for two orders are merged with the biggest gap for two orders. This makes up the non-identical stops-spans metric in matrix $\mathbf{B}$. The first part of this metric is the total number of non-identical picking locations as developed by Hofmann and Visagie [55] provided in equation (4.2). The second part is an adaptation to the specific layout of the picking system according to equation (4.6). Batches are formed by the orders with the smallest number of non-identical picking locations and the shortest non-identical span between them.

The non-identical stops-spans matrix $\mathbf{B}$ with element b_{ij} can be formulated as

$$b_{ij} = (|\mathcal{S}_i| + |\mathcal{S}_j| - 2\,|\mathcal{S}_i \cap \mathcal{S}_j|) + (|\widetilde{P}_i^{\min}| + |\widetilde{P}_j^{\min}| - |\widetilde{P}_i^{\min} \Cap \widetilde{P}_j^{\min}|), \text{ with } i \neq j. \tag{4.9}$$

Finally, the stops-spans ratio with general element c_{ij} in matrix $\mathbf{C}$ can be calculated by adding the stops-spans ratio (as developed by Hofmann and Visagie [55] and defined in equation (4.3)) and the spans ratio (defined in equation (4.7)). The length of the non-identical part of the minimum span is divided by the length of the combined minimum span. This is added to the non-identical picking locations divided by the combined picking locations. Therefore the closer this metric gets to zero, the lower is the incompatibility between orders in terms of walking distance.

The stops-spans ratio matrix $\mathbf{C}$ with element c_{ij} can be calculated as

$$c_{ij} = \frac{n_{ij}}{t_{ij}} + \frac{d_{ij}}{s_{ij}}, \text{ with } i \neq j. \tag{4.10}$$

In Table 4.5(a) the symmetric matrix that results from calculating the stops non-identical spans metric for all orders in the small picking line example is displayed. The IP would batch Order 1 and 3 in the first batch and Order 2 and 4 in the second batch. Calculating the non-identical stops-spans metric for all orders in the example leads to Table 4.5(b). Batch 1 could be made up of Order 1 and Order 4 and Batch 2 would be composed of Order 2 and Order 3. All stops-spans ratios for the example are illustrated in Table 4.5(c). Order 1 and 4, or Order 2 and 3 could be batched together.

$$\mathbf{Z} = \begin{array}{cc} & \begin{array}{ccc} 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 15 & \boxed{13} & 14 \\ - & 12 & \boxed{13} \\ - & - & 13 \end{bmatrix} \end{array}$$

(a) *The* $\mathbf{Z}$ *matrix.*

$$\mathbf{B} = \begin{array}{cc} & \begin{array}{ccc} 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 13 & 10 & \boxed{12} \\ - & \boxed{9} & 11 \\ - & - & 10 \end{bmatrix} \end{array}$$

(b) *The* $\mathbf{B}$ *matrix.*

$$\mathbf{C} = \begin{array}{cc} & \begin{array}{ccc} 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 1.51 & 1.13 & \boxed{1.35} \\ - & \boxed{0.95} & 1.21 \\ - & - & 1.13 \end{bmatrix} \end{array}$$

(c) *The* $\mathbf{C}$ *matrix.*

TABLE 4.5: *The matrices* $\mathbf{Z}$, $\mathbf{B}$ *and* $\mathbf{C}$ *for the small picking line, with the best batching of orders for each metric shown by the framed numbers in that matrix.*

All three measurements improve in terms of minimum number of cycles traversed when solved 10 times compared to the picking location metrics proposed by Hofmann and Visagie [55]. This is shown in Figure 4.5. Applying the GR heuristic results in the best average of c_{ij} with 24 222 versus the best average of r_{ij} with 24 230 in total number of cycles traversed for the Retailer's 50 sample picking waves. This is followed by b_{ij} with 24 236 versus n_{ij} with 24 240 and z_{ij} with 24 332 versus t_{ij} with 24 693 total number of cycles traversed. The descriptive statistics for the combination of batching metrics and GR are illustrated in Table 4.6.

	T-GR-GD	N-GR	R-GR	Z+GR	B+GR	C+GR
Mean	24 693	24 240	24 230	24 332	24 236	24 222
Median	24 687	24 243	24 229	24 336	24 236	24 223
Range	76	77	52	51	87	43
Std.Dev.	30.937	29.993	16.900	16.068	21.541	15.521
Variance	957.104	899.558	285.600	258.178	464.011	240.900

TABLE 4.6: *Descriptive statistics for the total number of cycles traversed to pick all the orders for all 50 picking waves using the GR heuristic.*

However, the GS heuristic provides even lower numbers of total cycles traversed for the route overlap addition metrics. The algorithm searches greedily for the global minimum in the metric and then uses that to batch. It is deterministic and was thus only solved once. For the route overlap addition metrics it results in 24 189 cycles for c_{ij}, 24 166 for b_{ij}, and 24 149 for z_{ij}. Therefore the combination of route overlap addition metrics and GS is recommended.

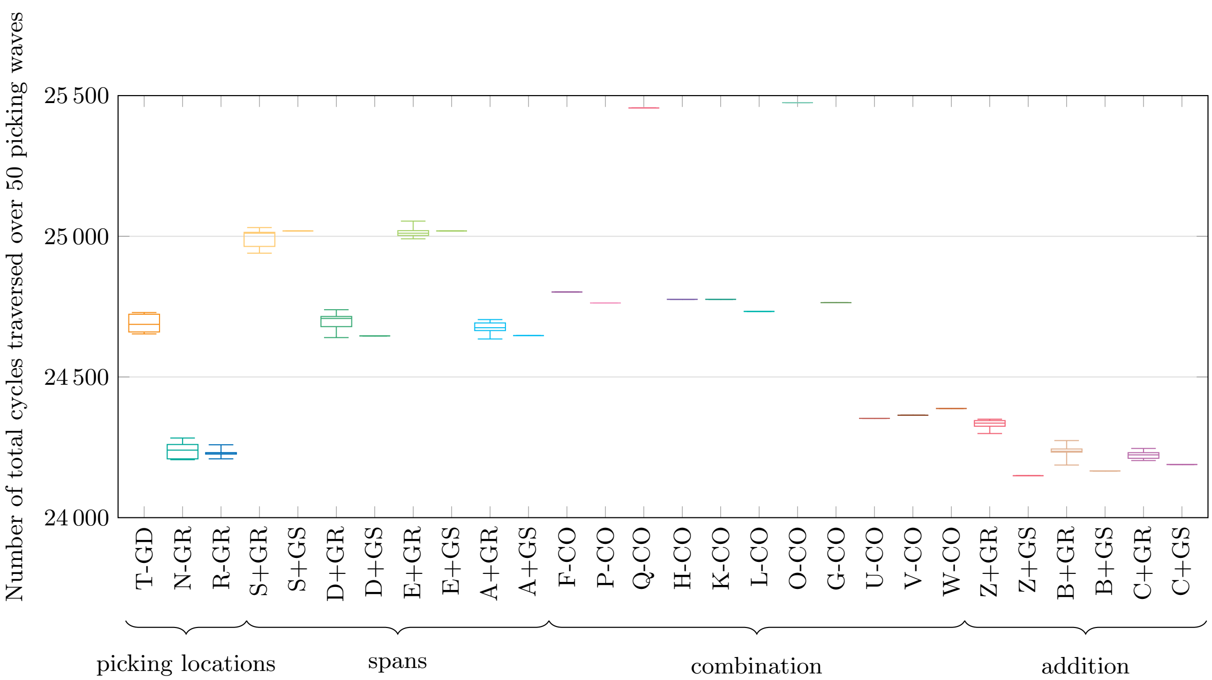


FIGURE 4.5: *Box and whisker plots of the total cycles traversed per algorithm per route overlap metric. All instances with GR were solved 10 times.*

4.4 Discussion of experimental results

The combination of route overlap addition metrics generate the lowest number of total cycles traversed for the 50 sample picking waves. Therefore their performance will be evaluated against the picking location metrics proposed by Hofmann and Visagie [55]. These results will also be analysed statistically.

4.4.1 Route overlap versus picking location metrics

The following combinations produce the best results for picking location metrics (meaning the lowest number of cycles traversed for the 50 sample picking waves): The stops metric (T-GR-GD) and the great deluge metaheuristic (using the greedy random heuristic as an initial starting solution); the non-identical stops metric (N-GR) and the greedy random heuristic; and the stops ratio (R-GR) and the greedy random heuristic. For route overlap metrics, the best results are generated by: The stops non-identical spans (Z-GS) and the greedy smallest entry heuristic; the non-identical stops-spans (B-GS) and the greedy smallest entry heuristic; and the stops-spans ratio (C-GS) and the greedy smallest entry heuristic. This is pointed out by a comparison of all metric-algorithm combinations in Figure 4.5.

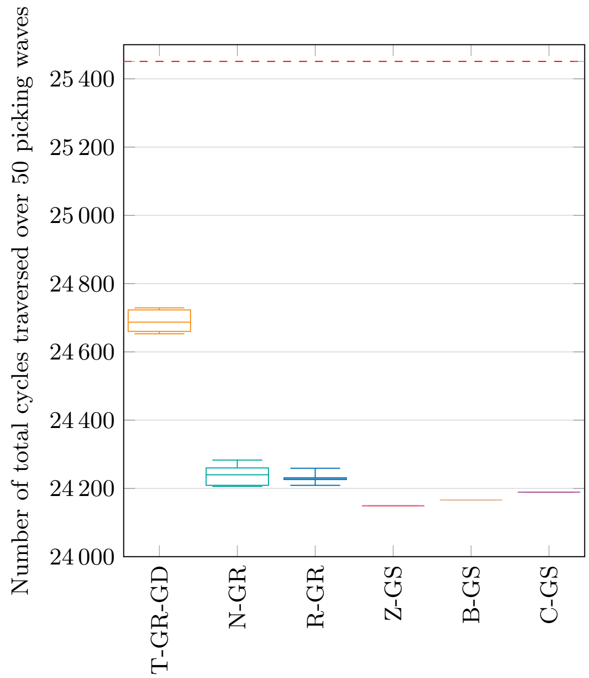
The best performing picking location and route overlap metrics are evaluated in terms of solution quality and computational times in Figure 4.6. An upper and a lower bound might contextualise the performance of the different metrics in terms of the total number of cycles traversed to pick all orders. In the worst case, order batching would not be introduced. The *maximal SKU* per picking wave describes the maximum number of orders needed to pick a SKU. It can therefore give an idea about the minimum number of cycles that has to be traversed for a picking wave [78]. For the 50 sample picking waves, the maximal SKU results in 40 361 cycles. In the best case, walking distance could be reduced by 50%. Therefore the maximal SKU provides a strict lower bound of 20 181 cycles. However, this is a poor bound as it incorporates two relaxations. Firstly, it does not include the walking distance needed to link up orders. Secondly, it does not take into account that spans of orders from different batches may not overlap. An approximation for comparison is provided by the nearest end heuristic that is known to produce good solutions. However, it cannot be used as a bound [78]. Applying the nearest end heuristic without batching results in 46 711 cycles. These cycles could be cut in two and put next to each other resulting in 23 356 cycles. But again, taking into account that orders have to be linked so that different batches may not overlap, this would not be a feasible solution. Nevertheless, the 23 356 cycles are an estimate that can be aimed at while evaluating performance as it may be closer to the optimal solution than the 20 181 cycles determined by the maximal SKU. Another benchmark used in literature employs a FIFO rule, as described in Section 4.3, that results in 25 451 total cycles traversed. This provides a guideline to compare the performance of the batching metrics against a random batching approach. The benchmark is depicted by a red dashed line in Figure 4.6(a).

All six metrics outperform the benchmark used in literature. Within the picking location metrics, the R-GR combination produces the lowest number of total cycles traversed with a mean of 24 230 cycles. This is higher than the C-GS combination of 24 189 cycles which in turn is the highest number of total cycles traversed within the route overlap metrics. Therefore the route overlap metrics perform better in terms of solution quality compared to the picking location metrics as depicted in Figure 4.6(a).

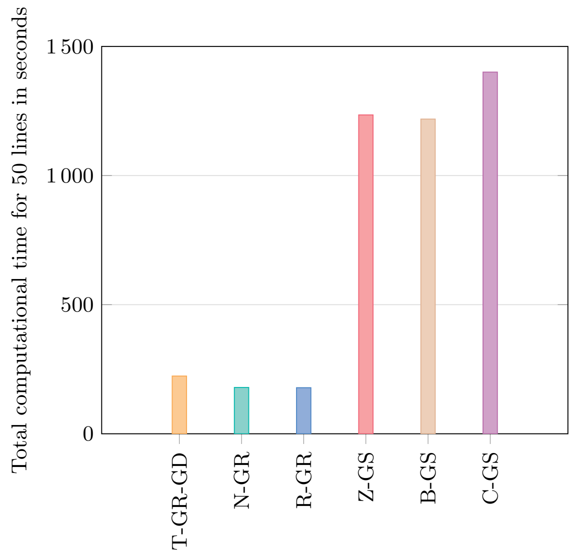
The stops non-identical spans metric Z produces 24 149 cycles – the lowest number overall, since the smallest number of combined stops between two orders allows for different spans. The order specific locations are dominant in this metric.

Two additional decision tiers of allocating and placing a SKU have to be solved for each picking wave and a choice of configuration if an exhaustive optimisation of the picking system is the aim. A time frame of about 30 seconds per picking wave to form batches was determined by the Retailer. All decision tiers can then be solved in less than a minute making a real life application feasible.

In terms of computational time as illustrated in Figure 4.6(b), the combinations N-GR and R-GR solve the fastest, while the hybrid of GR-GD with the T metric takes slightly longer. The



(a) *Total cycles traversed per metric.*



(b) *Computational time per metric-algorithm combination.*

FIGURE 4.6: *Comparison in terms of total cycles traversed between best performing picking location and route overlap metrics.*

route overlap metrics with the GS heuristic take longer than the picking location metrics. The C metric takes longer to solve than then B or Z metric as the entries of this matrix produced by a ratio are less distinct. Nevertheless, the route overlap metrics are still within the set time limit and thus produce practically implementable times.

4.4.2 Statistical analysis

The application of picking location metrics as approximations for walking distance before picker routing has been validated [55]. That is why the application of route overlap metrics is also validated by a regression analysis. The correlation between each metric's objective function value and the number of cycles traversed is thus investigated. For stops non-identical spans this evaluation results in $R^2 = 0.782$, for non-identical stops-spans in $R^2 = 0.786$, and for the stops-spans ratio in $R^2 = 0.785$. Thereby a strong correlation between all route overlap metrics and the final walking distance is indicated. However, the regression analysis does not provide a basis for comparison between metrics as the objective value of each metric is expressed in different units.

Inferential statistical tests are necessary for the comparison of the influence of the different metrics on the number of cycles traversed. Chiarandini *et al.* [23] propose either a univariate model focusing on one of the measurements or a bivariate model focusing on both measurements combined. The main focus of this analysis will be on the solution quality in terms of number of cycles traversed for all 50 picking waves, since the time requirement has shown to be suitable for a real life application. Therefore a one-way ANOVA with a Tuskey's HSD *post-hoc* test will be used to analyse the influence of the six metrics [101]. The one-way ANOVA shows a statistical significance ($F(5, 24) = 475.24, p = 3.5727 \times 10^{-23}$) in Table 4.7, pointing out the influence of the metrics on the total number of cycles traversed. With the help of the Tuskey's HSD *post-hoc* test it can be concluded that there is a statistically significant difference when comparing picking location metrics to route overlap metrics [102, 103].

One-way ANOVA	Sum of squares	*df*	Mean square	*F*	*p*
Between metrics	1 054 364.30	5	210 872.86	475.24	3.5727×10^{-23}**
Within metrics	10 649.20	24	443.72		

Note: Two asterisks indicate significance at the 5% level or below.

Table 4.7: *One-way ANOVA on metrics.*

4.5 Conclusion

Since the combination of stops non-identical spans and greedy smallest entry algorithm shows the lowest number of 24 149 cycles traversed for all 50 sample picking waves, it is the recommended combination for applying order batching to a unidirectional cyclical picking line. Applying random batching leads to 25 451 total cycles traversed and results in 5.4% additional walking distance. If batching is introduced pickers walk 48.3% shorter.

Even though metrics including information about the specific layout of the picking system have shown to be valid for approximating distance, the route overlap metrics still remain an approximation before picker routing. In general, the results indicate that the more information about the specific layout is incorporated, the lower the total number of cycles traversed. Therefore a dynamic batching idea is suggested for future research. However, the same start and end of a batch incorporating two orders increases the walking distance. To overcome this, the philosophy of starting and ending the orders at the same location might be challenged.

4.6 Chapter summary

In this chapter order batching was introduced to a unidirectional cyclical picking line as utilised in a South African retailer's DC. Different stop-span-based order-to-route closeness metrics that approximate walking distance before picker routing were developed. These route overlap metrics include the following: the stops non-identical spans that combines the picking locations and non-identical spans; the non-identical stops-spans that connects the non-identical picking locations and the non-identical spans; and finally the stops-spans ratio that incorporates the non-identical picking locations, the combined picking locations, the non-identical spans and the combined spans. The greedy smallest entry algorithm in combination with these route overlap metrics produced the lowest number of total cycles traversed when tested on 50 sample picking waves of different sizes in terms of orders and SKUs. The combination of stops non-identical spans and greedy smallest entry algorithm is recommended. The route overlap metrics extend the picking location metrics that have been proposed by Hofmann and Visagie [55] by incorporating more information about the specific layout of a unidirectional cyclical picking line including the order specific measurement of a minimum span.

The combinations that produce the lowest number of total cycles traversed on the 50 sample picking waves for the picking location metrics and the route overlap metrics were compared. The Z-GS route overlap metric traversed 81 cycles less than the R-GR picking location metric.

The span based measures developed in this chapter reduced the total walking distance even further. This reduction in walking distance may be translated into time savings. However, picking multiple orders simultaneously will increase the time a picker has to stop at a location which may reduce walking speed through congestion. The next chapter aims at simulating the unidirectional cyclical picking line including order batching to measure time savings and thereby quantify a potential decrease in operational cost.

CHAPTER 5

The effect of order batching on a unidirectional cyclical picking line's completion time

Contents

Order batching reduces walking distance on a unidirectional cyclical picking line, but does it also minimise overall picking time? Before picker routing, batches are formed by means of proximity measures and combination heuristics. Additionally, order batching increases pick density which might lead to congestion and also influences the number of pickers in the system. This picking system is modelled as a discrete event simulation, which can measure the total time savings and also give an idea of the best number of pickers after order batching. Historical input data and the simulation model output is verified and validated. The potential of using pick density as a predictor for the picking time reduction is investigated.

5.1 Introduction

Distribution centres (DCs) form a link between supply and demand in supply chains. Approximately 25% of a company's total logistics cost can be attributed to DCs [49]. Within a DC the highest contributor to operational cost, about $50 - 65\%$, is the order picking system that transforms inbound bulk stock into outbound customer orders [29, 113]. Therefore, decreasing the total picking time is crucial when improving the order picking system that will ultimately minimise total logistics cost. A warehouse designer has to address the questions of physical configuration of the DC and order picking policy to minimise operational cost [44].

In this dissertation the order picking system of a prominent South African retailer (referred to as the Retailer) is considered. Serving about 2 000 stores with varying customer profiles the DC has to handle a large number of non-uniform orders on a daily basis. A key characteristic of the Retailer is the central planning of stock that goes to each store. Instead of store managers requesting stock, the planners at a central planning department allocate stock keeping units (SKUs) available at the DC to the stores. Central planners issue details on the specific SKUs (and their quantities) that have to be distributed to each store. The DC then selects a subset of these instructions to be picked in a single picking wave to fulfil the store requirements. Such a picking wave is processed on a picking line in the DC. On a specific picking line a single SKU is assigned to a unique location for each wave. A *picking wave* thus includes the following activities: populating the line with the selected SKUs; the actual picking process; and finally, the removal of excess stock from the line back to storage [78].

In Figure 5.1 a schematic representation of the layout of a picking line with m locations is shown. The unidirectional cyclical picking line is a picker-to-parts system. A picking line at the Retailer's DC in Cape Town consists of 144 locations with a conveyor belt in the middle. The conveyor belt has three access gates: one at each end, and one in the middle. Consequently, only a part of the picking line is in use, and the line has 68 to 76 available locations. Each SKU has a unique location on the picking line. Five pallets of the identical SKUs can be stored at each location. Additional stock is kept on the floor space between different picking lines. Therefore stock does not have to be replenished from storage racks during a picking wave. The number of units of each SKU to be picked for each store is known prior to the start of the picking wave giving orders a deterministic nature within the wave. All orders are picked by pickers moving around the conveyor belt in a clockwise direction. Each picker pushes a trolley with enough space for two cartons – one carton for each order. The pickers are guided through the picking process by a voice recognition system (VRS) that sends them to the closest required SKU. An empty carton is placed on the trolley and registered with the VRS before the picking of an order starts. Once the picking of the order is completed or the carton is full the carton is placed on the conveyor belt for further processing in the dispatch area of the DC [76, 77].

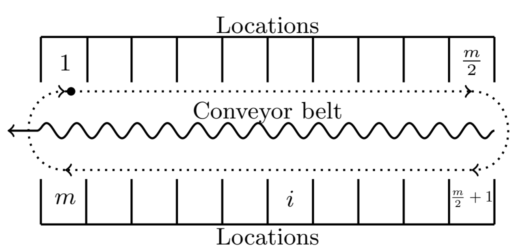


Figure 5.1: *A schematic representation of a picking line with m locations. Source: De Villiers [30].*

The layout of the unidirectional cyclical picking line shows similarities to unidirectional carousels, but in the case considered here pickers move relative to static SKUs. Automated warehouse carousel systems are usually used for picking small- or medium-sized products. A picker operating the system remains at a fixed location during picking while the carousel rotates to present stock to the picker. Therefore a typical carousel is a parts-to-picker system. The cyclical layout of the picking line with one picker per order guided by an automated VRS is similar to a unidirectional carousel. However, the main difference between the Retailer's system and carousel systems studied in literature is the presence of wave picking with all orders known and fixed before the start of a specific wave of picking. In most bidirectional carousels only one picker is used thus investigating the number of pickers in the system is omitted in literature. It is possible to insert multiple pickers in this set up to reduce the overall time of picking orders in a wave.

Tompkins *et al.* [110] categorised three picking policies, namely discrete order picking in which the picker picks all items for a single order in a tour, batch picking in which all items for multiple orders are collected in a tour, and zone picking in which a picker is allocated to a specific zone in the DC and is assigned to the orders in that zone. Walking to, from and in between picking locations account for approximately 50% of the picker's time in discrete order picking. A reduction of walking distance can be achieved by the introduction of order batching.

Hofmann and Visagie [55, 56] introduced order batching by means of different metrics and showed a reduction in walking distance in the unidirectional cyclical picking system (measured in cycles traversed, as introduced by Matthews and Visagie [78]). However, order batching may result in picker congestion that consequently influences the overall completion time. Batching orders decreases the walking distance, but increases pick density which in turn influences picker congestion. This has an effect on the number of pickers in the system and raises the question of how many pickers should be used during a wave of picking to minimise the total picking time. Existing literature does not focus on the layout of a unidirectional cyclical picking line [73]. Van Gils *et al.* [114] show a gap between academic research and practice in addressing these real life issues such as picker blocking. Therefore, if only travel distance is taken into account, managers might choose inefficient policies.

This chapter aims at answering the question of how much time is saved by the introduction of order batching to a unidirectional cyclical picking line using the batching metrics by Hofmann and Visagie [55, 56]. The influence of the increased pick density and congestion on the total picking time is thus studied in the remainder of this chapter. A discrete event simulation approach is used to determine the total completion time of picking waves. The simulation can thus be used to check if a reduction in walking distance reduces the picking time, and if it does, by how much.

After a brief introduction to discrete event simulation (DES) the proposed simulation model is introduced in Section 5.2. The data gathering process is described in Section 5.2.3. In Section 5.2.4 the model is verified and validated. Different batching metrics are tested and compared in terms of picking time in Section 5.3 and explanatory variables are evaluated to account for picking time reduction. Section 5.4 provides a conclusion and an outlook on future research opportunities.

5.2 Simulation of a pick wave

The unidirectional cyclical picking line is modelled in a DES to compare picking times generated by applying different batching metrics. Simulation has proven to be a practical and implementable tool in optimising warehouse operations [41]. Simulation models address picker speeds and other conditions more realistically than analytical models [44]. Van Gils *et al.* [114] uses a DES to explore real life issues in a parallel aisles warehouse layout. The random distributions for walking velocity, picking speed, and setup time are fitted and the results of the simulation model are validated against real life data. Van Gils *et al.* [115] emphasise the importance of accounting for these real life issues in bringing together theory and practice. Therefore, after a brief literature review on DES modelling, the input data is extracted from real historical picking data, while the output data is verified and validated against historical data from the Retailer's warehouse management system (WMS). Picker blocking is addressed by looking at a good number of pickers that would allow for maximal picking efficiency, while still avoiding heavy congestion by influencing pick density.

5.2.1 Background

According to Law [70], DES models a system evolving over time. Thereby, the simulation represents the instantaneous change of state variables at discrete points in time. An event, which occurs at these points in time, is defined as an instant with the potential to change the state of the system. DES represents the internal processes of a system, its components, and its interactions. It is thus dynamic when compared to mathematical models, descriptive models and statistical models [5].

The collection of information that is needed to define what is happening within a system is called system state variables. In a discrete system these variables remain constant and only change at event times. Therefore, the picking system at the Retailer is modelled as a discrete system as it only changes states upon picker action. An object that requires explicit definition is referred to as an entity. In this simulation each picker is a dynamic entity. Each entity has attributes such as walking velocity, picking speed, and handling speed. A resource is an entity that provides service to the dynamic entity. Each picking location has the attribute of accommodating a unique SKU of a certain quantity to pick from and can thus be described as a resource of which a picker can request one or more items. The state of the location can be idle or blocked by another picker. Requests from entities are processed according to the list of sequenced orders. The orders have been sequenced by a nearest end heuristic that chooses the order with the nearest ending location from a given starting location to be picked next [78]. Events initiate the beginning of an activity, the ending of an activity, or a delay. The duration of an activity is known prior to its start and describes the length of time in which entities engage. In this simulation the activities walking, picking, and handling draw random times from statistical distributions. Delays are caused by a combination of different system conditions and can have indefinite durations. In this simulation, a delay may occur if more than one picker wants to pick from the same location at the same time. A mechanism that moves simulated time forward, in this case driven by the advance of the next event, conducts the DES [5].

5.2.2 Simulation model

The environment in which the simulation takes place is a *picking system* in the Retailer's DC as depicted in Figure 5.2. The locations in the picking line are the resource entities. Pickers push a trolley with the capacity of two cartons (for two orders). Pickers request a unique SKU at a location when they arrive there. All locations can be either unoccupied or occupied by a picker requesting SKUs.

The dynamic entities of the simulation model are the *pickers*. Each picker can carry out one of three activities, namely walk to a SKU, pick a SKU for one or two orders, and handle the carton by preparing and packing it. All activities are depicted in Figure 5.3. Each picker's

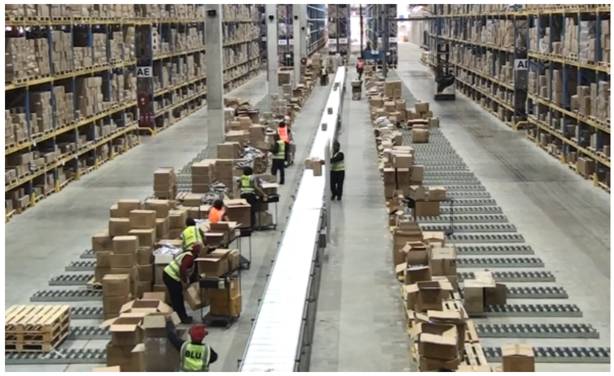


FIGURE 5.2: *Picking line in the Retailer's DC in Cape Town, South Africa.*

attributes are the speeds at which these activities are carried out. If a picker finds a location being occupied by another picker then the picker is delayed.

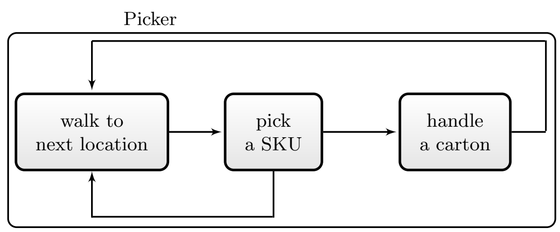


FIGURE 5.3: *Picker entity with its three different activities during a picking wave.*

The following assumptions are made while extracting input data for the picker speeds to model the Retailer's order picking system. These assumptions align the simulation with the real life set up to model constraints that are currently applied at the Retailer.

1. Pickers complete an order before starting the next order.
2. Picking and handling times are independent of SKU and location.
3. The proportion of time spent allocated to the three activities does not vary over pickers.

The logic of a picker in the simulation model is illustrated in a flow chart in Figure 5.4. In the picking system environment multiple pickers carry out the order picking process. This process starts with a new order. The picker prepares a carton and walks to a picking location in the picking line. If the location is occupied by another picker, the picker has to wait until the previous picker finishes picking. When the location is unoccupied, the picker picks the SKU. If the current carton is full and the SKU is the last SKU requested by the order, the carton is packed and placed on the conveyor belt. The picker will then start a new order. If the carton is not full and the last SKU of the current order is reached the carton is also packed and placed on the conveyor belt. However, if the carton is not full and the last SKU of the current order is not reached the picker walks to the next SKU for the current order. The process is repeated until all orders of a picking wave are picked.

This simulation was implemented in Python 3.6 [94] utilising the C-based libraries Numpy [89] and Pandas [93]. The implementation was run on a Dell Optiplex 5050 with a Intel Core i7-7700 CPU at 3.6 GHz, 1x8GB 2400MHz DDR4 RAM, a 2.5” 256GB SSD class 20 drive, and the

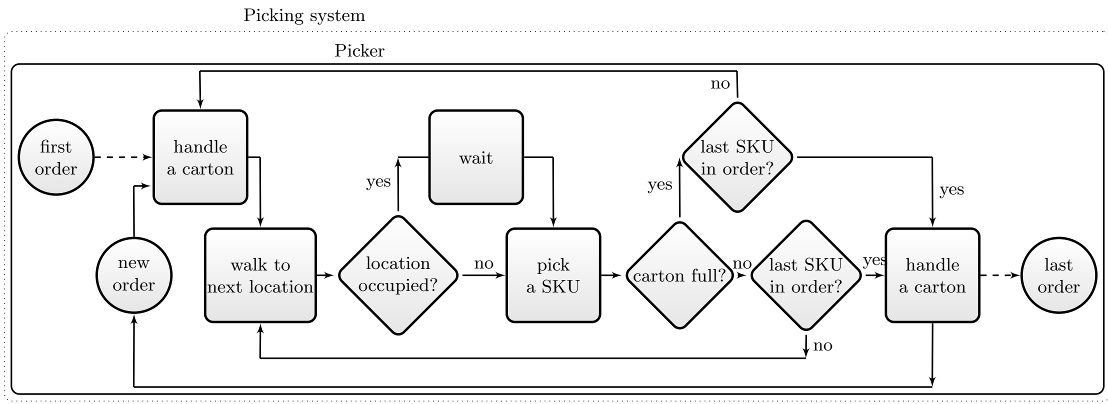


FIGURE 5.4: *A schematic representation of the logical flow chart of the simulation model.*

Microsoft Windows 10 Enterprise 2016 LTSB operating system [84]. For the statistical analysis of the input data and the generated output data R [95] was used.

5.2.3 Input data capturing

Historical time stamp data recorded by the WMS for the picking process were received from the Retailer. However, the time stamp data only indicate the time when a pick was completed. They do not include the times for handling the carton, walking to the SKU and the duration of a pick. Thus this data do not provide walking velocities nor picking and handling distributions which is critical for the verification and validation process. Therefore a time study of the picking process on the unidirectional cyclical picking line in the DC was performed. The time spent walking, picking, and handling could thus be calculated for every picker present in the picking line. In Table 5.1 an example of the data provided by the time study is displayed.

	Walking			**Picking**			**Handling**	
Start	Stop	Duration	Start	Stop	Duration	Start	Stop	Duration
08 : 29 : 09	08 : 29 : 13	3.99	08 : 29 : 13	08 : 29 : 15	1.56	08 : 29 : 15	08 : 29 : 21	5.75
08 : 29 : 21	08 : 29 : 25	3.96	08 : 29 : 25	08 : 29 : 29	3.58			
08 : 29 : 29	08 : 29 : 30	0.57	08 : 29 : 30	08 : 29 : 37	6.89			
08 : 29 : 37	08 : 29 : 39	1.79	08 : 29 : 39	08 : 29 : 45	5.87			
08 : 29 : 45	08 : 29 : 47	1.88	08 : 29 : 47	08 : 29 : 57	9.62	08 : 29 : 57	08 : 30 : 00	2.99
08 : 30 : 00	08 : 30 : 03	2.68	08 : 30 : 03	08 : 30 : 10	6.66			
08 : 30 : 10	08 : 30 : 13	2.74	08 : 30 : 13	08 : 30 : 19	5.99	08 : 30 : 19	08 : 30 : 31	11.52
08 : 30 : 31	08 : 30 : 36	4.98	08 : 30 : 36	08 : 30 : 40	3.72			
08 : 30 : 40	08 : 30 : 42	1.93	08 : 30 : 42	08 : 30 : 50	7.51			
08 : 30 : 50	08 : 30 : 52	1.94	08 : 30 : 52	08 : 30 : 57	4.61			
	sum	**26.46**		**sum**	**56.01**		**sum**	**20.26**
	% of total	**25%**		**% of total**	**55%**		**% of total**	**20%**

TABLE 5.1: *An example of the recorded times spent walking, picking, and handling. Start and stop times are measured in (hh:mm:ss). Duration is measured in (s).*

A proportion of time spent walking, picking, and handling could be obtained from the time study. The proportions are shown in Table 5.2(a).

The average picking times, handling times, and walking velocities are calculated by combining the extracted time proportions from the Retailer's WMS data. In Table 5.2(b) an example of the time stamp data is provided which gives information on a picker, order, location, SKU identity number, packslip code, date and time.

Walking	27.95%
Picking	48.05%
Handling	24.00%

(a) *The proportion of time spent by the pickers per activity.*

Picker	Order	Location	SKU	Packslip	Date	Time
Picker 1	6759	052	910141	891974178	06/05/2019	08 : 32 : 59
Picker 1	6427	056	918535	891974179	06/05/2019	08 : 33 : 10
Picker 1	6427	057	918536	891974179	06/05/2019	08 : 33 : 16

(b) *An example of the time stamp data. Time is measured in (hh:mm:ss).*

TABLE 5.2: *Proportions and time stamp data to obtain walking, picking, and handling speeds.*

At least 10 cycles were considered to calculate the average walking velocity, the average picking time, and the average handling time. Cycles with excessive time delays and congestion were disregarded [45]. Excessive time delays between two picks can occur due to breaks or congestion. For each picker, over these at least 10 cycles, the number of locations that were passed, the number of picks that were made, the number of cartons that were handled, and the total time spent picking can be calculated.

The calculation of walking velocities, picking times, and packing times, follows from the information about the number of locations, the picks, and the cartons. The time stamp data of one picker completing a minimum of 10 full cycles is provided as an example. There are 144 locations per picking cycle, but this picker has to pass 1 681 locations if the starting locations of the first order (128 locations) and the ending locations of the last order (113 locations) of the 10 sample cycles are taken into account. This picker completed 548 picks and prepared and packed 43 cartons. The picker spent a total of 6 297 seconds in the picking process. The time spent in each process step can be calculated using the total times and the proportion. The walking velocity is given by the time it takes a picker to pass the length of one location. Passing the picking locations in 1 760 seconds leads to an average walking velocity of 1.0472 seconds per location. The average time per pick can be calculated by dividing the total picking time by the number of picks. For this example it was calculated as 5.5212 seconds. The average handling time was calculated as 35.1413 seconds per carton. The data for this example is presented in Table 5.3.

	Proportion	Total time	Events	Average rates
Walking	27.95%	1 760	1 681 locations	1.0472 s/location
Picking	48.05%	3 026	548 picks	5.5212 s/pick
Handling	24.00%	1 511	43 cartons	35.1413 s/carton

TABLE 5.3: *An example of the table to calculate walking velocity, picking time, and packing time for a specific picker. Time is measured in (s).*

For any specific picker the triangular distribution for walking velocity, average picking time, and average handling time were used [46]. The smallest value over all pickers was defined as the lower bound, while the largest value over all pickers was defined as the upper bound of the distribution.

The pick density measure in the unidirectional cyclical picking line plays an important role in this simulation because it has an influence on the total completion time of a picking wave. Hagspihl and Visagie [46] measured pick density as the average number of locations that are empty between every pick, if one picker completes the entire picking wave. In a picking line with high pick density a picker stops and picks more frequently. Therefore, a low value for this measure describes a small number of passed bays per gap and thus represents a high pick density. A low value for this measure describes a small number of passed bays per gap and thus represents a high pick density. The average gap size d_{g} with the unit of passed locations per gaps can be calculated as

$$d_{\mathrm{g}} = \frac{1}{g}\sum_{j=1}^{g}\gamma_j, \tag{5.1}$$

where γ_j is the length of the j-th gap and g is the total number of gaps. Observations on the floor, confirmed that pickers walk faster in a picking line with low density, since there is more distance between locations to pick up speed. Numerical experiments, in which time stamp data were fitted to time study data, determined the criteria for low, medium and high levels of pick density measure. A large d_{g} for batched orders lies above 10 passed locations per gap (low pick density measure) and the corresponding triangular distributions are shown in Table 5.4(a). As displayed in Table 5.4(b) walking velocity decreases in a picking line with a medium d_{g} of $3-9$ passed locations per gap as compared to a line with a big d_{g}. Moreover, picking and handling increases slightly as the proportion of number of picks and cartons to locations increases. A small d_{g} above zero but below two passed locations per gap (high pick density measure) decreases the walking velocity even further, due to frequent stops. Picking lines with high pick densities are often made up of heavier and slightly bulkier SKUs, this increases the time per pick but decreases the time to pack a carton [114]. The speeds for picking lines with high densities are depicted in Table 5.4(c). The problem with this measure is that the density is picker route specific. The density can only be calculated once the pickers are routed. It would thus be good to consider density measures that are not route specific.

	Lower	Central	Upper
Walking	0.4603	0.8376	1.2327
Picking	2.7937	5.1983	6.8197
Handling	9.4149	26.0441	47.7112

(a) *Distributions with a big gap size (low density).*

Lower	Central	Upper
1.0472	1.3883	1.8852
4.0486	5.5418	6.8197
21.9297	34.4901	47.7112

(b) *Distributions with a medium gap size (medium density).*

Lower	Central	Upper
1.8852	2.7076	3.2327
6.8197	16.1065	23.5212
9.4150	17.0853	21.9297

(c) *Distributions with a small gap size (high density).*

TABLE 5.4: *Triangular distributions for picking lines with different gap sizes.*

The general distance pick density measure d_{m} is defined without prior routing within the picking system. Therefore, it includes the total number of stops s to describe where a picker has to stop, the number of locations m available in the system and the maximal SKU v. The maximal SKU is needed by the maximum number of orders in a picking wave. It thus provides a minimum number of cycles that has to be traversed to pick all orders [78]. Pick density measure d_{m} can only take on values from $0-1$ with a high value representing a high pick density. The pick density measure d_{m} with the unit stops per locations can be calculated as

$$d_{\mathrm{m}} = \frac{s}{m \cdot v}. \tag{5.2}$$

The Retailer's definition of pick density measure d_{r} is also independent of picker routing. It includes the total number of stops s and the total number of orders n multiplied with the total number of SKUs u which describes the maximum number of stops possible. However, this definition of pick density does not include any approximation of distance. Therefore, a fairly low number of stops may indicate that the pick density is low even though SKUs are located closely to each other. The pick density measure d_{r} of the Retailer with the unit of actual stops per potential stops can be formulated as

$$d_{\mathrm{r}} = \frac{s}{n \cdot u}. \tag{5.3}$$

The density can also be expressed as the ratio between the total number of stops s and the total number of orders n as pick density measure d_n. This definition is independent of picker routing but not ideal. Since only averages are considered in this equation, it may describe a picking wave in which the majority of the orders have a few picks with a similar density to a picking wave that has an even spread of picks per order.

The pick density measure d_n with the unit of stops per orders can be calculated by

$$d_\mathrm{n} = \frac{s}{n}. \tag{5.4}$$

All four pick density measure definitions will be compared in the real world example. Thereby the best way of describing pick density in a unidirectional cyclical picking line can be identified.

An exponential distribution was fitted to the varying picking and handling times, since it can be directly attributed to each SKU picked or each carton handled [45]. The Kolmogorov-Smirnov test [101] and Anderson-Darling test [26] were used to measure the goodness of fit (GOF) of the exponential distribution for the picking and handling time distributions. The test statistic for the exponential distribution of the picking and handling time distributions was not rejected for a 99% confidence level as displayed in Table 5.5.

Exponential distribution: $\lambda = 0.212542$					
Kolmogorov-Smirnov					
Size					3 196
Statistic					0.0117
P-value					0.7715
Alpha	0.2	0.1	0.05	0.02	0.01
C-value	0.019	0.022	0.024	0.027	0.029
Reject?	no	no	no	no	no
Anderson-Darling					
Size					3 196
Statistic					0.4757
P-value					0.7717
Alpha	0.2	0.1	0.05	0.02	0.01
C-value	0.816	1.162	1.321	1.591	1.959
Reject?	no	no	no	no	no

(a) *GOF for the picking distribution.*

Exponential distribution: $\lambda = 0.159236$					
Kolmogorov-Smirnov					
Size					1 225
Statistic					0.0227
P-value					0.5519
Alpha	0.2	0.1	0.05	0.02	0.01
C-value	0.031	0.035	0.039	0.043	0.047
Reject?	no	no	no	no	no
Anderson-Darling					
Size					1 225
Statistic					0.7766
P-value					0.4982
Alpha	0.2	0.1	0.05	0.02	0.01
C-value	0.816	1.162	1.321	1.591	1.959
Reject?	no	no	no	no	no

(b) *GOF for the handling distribution.*

Table 5.5: *Goodness of fit test for picking and handling distributions.*

5.2.4 Verification and validation of output data

The verification and validation (VV) of the simulation model is important, because it confirms that the model produces realistic results. VV is thus used to increase the confidence in the simulation, not to prove its absolute accuracy [96]. Model verification ensures that the real world system has been translated into a computer model accurately and that its implementation is correct. Model validation substantiates the model as having sufficient accuracy for the intended application of the simulation [100]. Robinson [96] suggests four strategies for performing the VV of a simulation model: conceptual model validation; data validation and verification; white-box validation; and black box validation.

Conceptual model validation determines whether the simulation contains all necessary detail to meet the objectives proposed for their study [96]. The DES aims at answering the question of how much picking time was saved by introducing order batching to the unidirectional cyclical picking line. The detail involved in the simulation is sufficient as it includes the three activities of walking, picking, and handling. It also addresses the occurrences of congestions that may arise from multiple pickers picking at the same location. The properties of the model have been discussed with management at the Retailer who confirmed that enough detail is present in the simulation model. Enough detail is thus included to determine the overall savings in picking time with the help of this simulation model.

Data validation determines if the data used to build, experiment and validate the model are sufficiently accurate [96]. The input data for this simulation has been obtained by a time study of the picking system and combined with historical data recorded by the WMS. An exponential distribution has been fitted to the picking and handling times. Kolmogorov-Smirnov and Anderson-Darling tests have not been rejected for a 99% confidence level. Therefore the input data is sufficiently accurate. The walking velocity defines how long it takes for a picker to pass one location, while congestion causes a delay before the next pick event.

Verification ensures that the simulation is consistent with the conceptual model. The *white-box validation* determines whether each part of the model accurately represents their real world counterpart [96]. In this VV step, it is checked that the picker velocity, the picking time, and handling time are aligned with the actual data. It was ensured that the picker picks orders according to the right sequence, thus walks in the right direction and exhibits the right behaviour when encountering another picker. A single picker was modelled in the simulation and the activities were checked in the code. Analysis of the events showed that the picker always walks in the correct direction, picking the right orders, and handling the right cartons. If more than one picker was inserted into the simulation, the model reflects the anticipated behaviour of the picker slowing down and waiting when destined locations were occupied. The walking velocities, picking speeds, and handling speeds of the simulation output were assessed for correctness.

Black-box validation determines whether the simulation represents the real world at a sufficiently aggregate level [96]. In this VV step the model output is compared to historical data. Therefore the data of 10 sample picking waves are considered and displayed in Table 5.6. These 10 picking waves were selected by the Retailer as representative of the picking line setup.

Wave ID	**Orders**	**SKUs**	**Cartons**	d_g	d_m	d_r	d_n	**Picks per distance**	**Cartons per distance**	**SKU [kg]**	**SKU [m^3]**
Large data set											
2405	1 788	74	2 556	5.44	0.21	0.22	16.10	0.43	0.04	0.553	0.003
2784	1 784	68	2 349	4.01	0.46	0.39	26.37	0.78	0.04	0.289	0.003
2433	1 780	74	2 771	6.24	0.36	0.26	18.98	0.51	0.04	0.619	0.003
3010	1 779	74	2 060	5.48	0.34	0.20	14.44	0.39	0.03	0.388	0.002
3318	1 742	74	2 655	7.59	0.24	0.22	16.11	0.44	0.04	0.604	0.004
Medium data set											
3483	1 691	141	2 463	17.90	0.16	0.13	18.16	0.26	0.02	0.406	0.003
2580	1 557	68	1 620	4.40	0.40	0.25	17.02	0.50	0.03	0.287	0.002
Small data set											
2848	1 028	6	1 231	1.43	0.53	0.41	2.04	0.86	0.51	2.051	0.004
2856	691	9	1 190	1.77	0.56	0.19	1.70	0.42	0.42	6.032	0.013
2765	490	6	526	1.43	0.58	0.24	1.44	0.67	0.50	1.398	1.41

TABLE 5.6: *Historical data for* 10 *sample picking waves.*

The data set includes the picking line (or wave) identifier (ID), the number of orders, the number of SKUs, the number of cartons, the pick density measures, the ratio of number of picks to locations, the ratio of number of packs to location, the average weight in kg per SKU per picking wave, and the average volume in m^3 per SKU per picking wave. The dataset is divided into large picking waves of more than 1 699 orders, medium sets between 1 100 − 1 699 orders and small sets of less than 1 100 orders with a range in the number of SKUs from $6 - 141$. The picking wave data of $2405, 2784, 2433, 3010, 3318$ and 2580 fall under medium gap size d_g. Picking wave 3483 has a large gap size d_g, while picking waves $2848, 2856$ and 2765 have a small gap size d_g. High density levels for the small data sets are pointed out by pick density measure d_m. However, d_r fails to identify the higher density clearly. The pick density measure d_n manages to show an increase in density for the small data sets.

The accuracy of the model is influenced by the number of simulation replications. According to Burghout [16] an equation that is derived from a statistical $1 - \alpha$ confidence level t-test on the simulated mean as extracted from a more general procedure described by Law [70] can determine this number. Let

$N(w)$	be the number of replications,
$\bar{X}(w)$	be the estimate of the real completion time,
$S(w)$	be the standard deviation,
ϵ	be the percentage error of the simulated mean and
$t_{w-1,1-\alpha/2}$	be the two-tailed t-distribution critical value for $w - 1$ degrees of freedom at an α significance level.

Then the necessary number of replications can be calculated by

$$N(w) = \left(\frac{S(w) t_{w-1,1-\alpha/2}}{\bar{X}(w)\epsilon} \right)^2 . \tag{5.5}$$

The total completion time cannot be inserted into equation (5.5), since pickers starting a wave rarely finish it. Therefore this calculation uses the completion times of specific pickers with a sample size of 24. For example, for 20 replications the standard deviation of the total completion times is 970 which is 2% of the simulated mean of the completion times (which is 49 054 seconds). The historical completion time for a specific picker was 49 143 seconds. The acceptance of an error of a minute per hour results in $\epsilon = 0.016$. The t-distribution critical value for 19 degrees of freedom is 2.093. The number of necessary replications calculated by substituting these values in equation (5.5) results in approximately 6.69 replications. Although seven replications are sufficient to provide adequate results, the simulation was replicated 10 times for each picking wave.

The model accuracy was measured by comparing the historical data of an isolated picker's path (for a certain distance), to a run that simulates that same path. This was done multiple times with different pickers. The total completion time cannot be used as pickers rarely start and finish the same picking wave in real life. Therefore, the picker specific historical data have been summed up and cleaned from excessive time delays between two picks. Outliers (occurring from the *ad hoc* addition or removal of pickers from the line) have been removed from the historical completion times.

A difference between the historical completion times and simulated completion times can be computed [45]. If the calculated differences are normally distributed with a mean close to zero, it implies that the simulation times are accurate. The standard deviation describes the spread of the values around the mean. Illustrated in Table 5.7(a) the mean of 10 replications falls between the 95% confidence intervals for each picking wave.

A normal distribution was fitted to the differences in total completion times. The Kolmogorov-Smirnov test [101] and Anderson-Darling test [26] were used to measure the GOF of the normal distribution for the time differences. The example of a picking wave in Table 5.7(b) shows that the test statistic was not rejected for a 99% confidence level.

Wave ID	**Mean errors**	**Standard deviation of errors**	**Confidence intervals**
2405	−25.74	475.52	[−321, 269]
2784	110.11	297.78	[−75, 295]
2433	138.19	347.71	[−77, 354]
3010	−6.94	154.41	[−103, 89]
3318	−67.64	521.25	[−391, 255]
3483	32.37	481.30	[−266, 331]
2580	61.34	197.74	[−61, 184]
2848	53.16	139.95	[−34, 140]
2856	43.04	156.09	[−54, 140]
2765	−29.97	120.62	[−105, 45]

(a) *CI for the time differences. Time in (s).*

Normal distribution: $\mu = -25.74, \sigma = 475.52$					
Kolmogorov-Smirnov					
Size					10
Statistic					0.2391
P-value					0.5404
Alpha	0.2	0.1	0.05	0.02	0.01
C-value	0.323	0.369	0.409	0.457	0.489
Reject?	no	no	no	no	no
Anderson-Darling					
Size					10
Statistic					0.2624
P-value					0.6188
Alpha	0.2	0.1	0.05	0.02	0.01
C-value	0.509	0.631	0.752	0.873	1.035
Reject?	no	no	no	no	no

(b) *GOF for the time differences on picking wave* 2405.

TABLE 5.7: *Differences in total completion times between historical data and simulation data for* 10 *sample picking waves.*

The VV results show that the simulation has an acceptable accuracy level and can thus be used to simulate the picking time of the Retailer's real world unidirectional cyclical picking system.

5.2.5 Number of pickers

Inserting more than one picker in the picking wave can decrease the overall completion time of the wave, but can also lead to congestion if pickers have to pick items from the same location. With the introduction of order batching pick density increases, influencing congestion levels. This raises the question of how many pickers should be used on a picking wave.

Unfortunately there is no data available on congestion times. A picker that arrives at an occupied location has to wait until the previous picker completes the pick and the location becomes unoccupied again.

The aim of determining a good number of pickers for the picking wave takes two factors into consideration. While keeping congestion at a controllable level, the total completion time is minimised. Three techniques are considered to accomplish this: the absolute minimum approach, the critical limit approach, and marginal analysis.

In the absolute minimum approach the number of pickers is increased iteratively. Research by Hagspihl and Visagie [46] has shown that the total completion time decreases as more pickers are inserted into the picking system until the congestion becomes so high that the total completion time increases again. In their research, 45 pickers could be inserted into a wave before the total completion time increased (indicating congestion influences). The model output of this chapter confirms the same pattern. The DC management normally uses about eight pickers per picking wave. However, if there is little congestion the management might insert another picker. As confirmed by management, no more than 20 pickers will ever be inserted into the picking wave. For all large picking waves the allowed 20 pickers generate the lowest total completion time. However, this results in very high congestion levels of $36 - 66\%$. These levels of congestion lead to excessive labour costs, low morale amongst pickers, and labour inefficiencies.

The critical limit approach introduces a limit to the time a picker should be waiting for another picker to finish the picking during the total completion time. The DC aims at efficient picker utilisation thus keeping the percentage of picker congestion low. Therefore a good number of pickers is chosen by setting a limit on picker congestion. The Retailer suggests the critical limits between 15% and 17.5% to determine a good number of pickers. For medium picking waves with a medium density, this would result in 11 pickers. A large picking line with low density results in 12, while a short picking line with high density in 7 pickers.

The total completion time approach produced unacceptable levels of congestion, while the critical limit approach provided realistic results. But choosing the acceptable level of congestion is difficult and at best an educated guess. Therefore the third approach considers total completion time and congestion simultaneously. In marginal analysis (MA) the input of an extra unit brings additional benefits and costs that are weighed up against each other. An extra unit is added until the additional benefit is less than the additional cost. At this point the previous number of units can be determined as optimal for the system [121]. In this case, the benefit of adding one picker to the system lies in the marginal decrease of total completion time. Thereby answering the question of how much the total completion time decreases if one extra picker is inserted. The cost is determined by the increase in total congestion if one extra picker is added. Both values are then weighed against each other.

An example of a MA performed on a large picking wave is depicted in Figure 5.5. In this picking wave, the cost in congestion is higher than the benefit of decreasing total completion time when inserting picker number 10. Therefore a good number of pickers for this picking wave is 9. A good number of pickers over the whole dataset is 9 pickers, which is close to the number of pickers the Retailer has determined by trial and error.

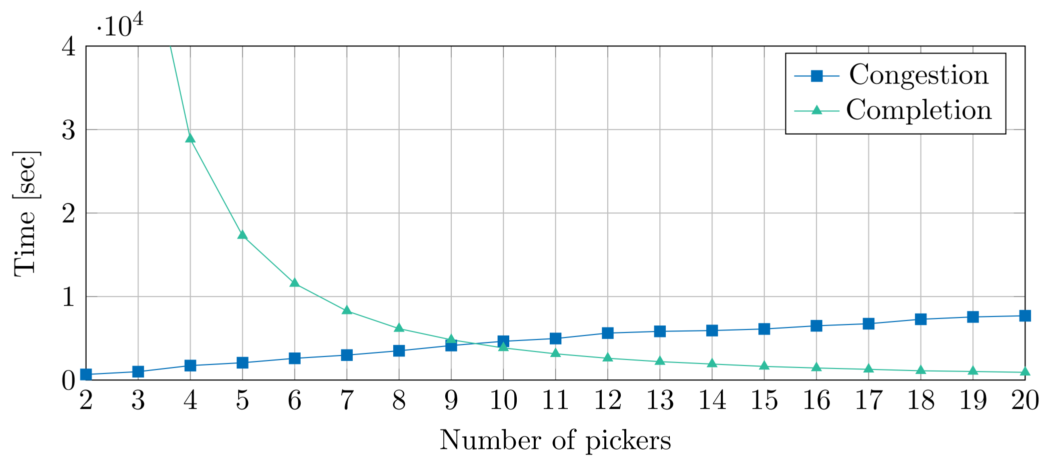


FIGURE 5.5: *A plot of the marginal times gained and lost with the increase in the number of pickers for picking wave* 2405.

According to Gue *et al.* [44] pick density has a direct influence on congestion in a parallel narrow aisles warehouse layout. The more pickers in the picking system, the more congestion arises. This is enhanced by higher picker speeds. Passing another picker, as it is allowed in the Retailer's picking system, lowers congestion levels. Gue *et al.* [44] observed little blocking under a low pick density, while congestion increases with higher pick densities. However, as pick density continues to increase it results in less congestion, as pickers travel less and pick more. Only while travelling pickers can be blocked, so less picker blocking occurs if pickers are busy picking the majority of the time. According to Hagspihl and Visagie [46] there is a correlation between pick density and a good number pickers. The dataset used in this chapter supports these findings. A large picking wave with low density shows the shortest total completion time operating with 10 pickers. Medium picking waves with medium pick densities result in $8-9$ pickers, while short waves with high densities are most efficient with7 pickers (or the maximum number of pickers possible). The DC management often inserts $8-9$ pickers per picking wave per day thus validating the results.

5.3 Comparison between batching metrics

Different batching metrics were tested to answer the question of how much picking time can be saved by introducing order batching to the unidirectional cyclical picking line. Therefore batching metrics based on the picking locations, batching metrics based on the section of the picking line that is traversed to pick an order (route overlap), and a combination of both are simulated.

5.3.1 Experimental results

The simulation of completion times per picking wave is carried out on the 10 sample picking waves. Each experiment is replicated 10 times and the average in picking time per wave is compared. For evaluation purposes picking times without batching and a benchmark with a first-in-first-out random approach (FIFO) from literature are included in the analysis. The following approaches were shown to give good results in reducing walking distance. Therefore they are used to investigate possible time savings.

The stops ratio batching metric combined with a greedy random heuristic (R-GR) was introduced by Hofmann and Visagie [55] and is based on the number of picking locations that a picker has to stop to collect items for an order. The stops ratio batching divides the non-identical stops (the number of stops that are not shared between two orders) by the identical stops (the number of stops that are similar between two orders). The lower the incompatibility between two orders the closer this metric is to zero. The greedy random heuristic combines orders with a low level of incompatibility according to a random permutation to form batches.

Matthews and Visagie [78] defined a span of an order as the distance that has to be covered to pick all items of that order given a starting location. Searching for the biggest gap between the items of an order leads to the so-called minimum span of the order that starts at the end of the biggest gap. This minimum span idea forms the basis of the spans ratio batching metric that is combined with a greedy smallest entry heuristic (A-GS) as developed by Hofmann and Visagie [55]. This approach divides the non-identical minimum span (the part of the minimum span that is not the same between two orders) by the minimum span (the part of the minimum span that is similar between two orders). The greedy smallest entry combines orders with a low level of incompatibility into batches by searching for the smallest entry globally.

The stops non-identical spans metric combined with the greedy smallest entry heuristic (Z-GS) suggested by Hofmann and Visagie [55] adds the minimum span idea to the stops metric. Thereby leading to the smallest number of cycles traversed for all 10 sample picking waves as shown in Table 5.8.

The total completion time for all 10 sample picking waves is the lowest when applying order batching through the Z-GS metric (see Table 5.8). A lower number of total cycles traversed consistently leads to shorter times, thus showing that order batching reduces picking time.

Hong [60] suggests that the batching algorithm has a significant effect on the pick density level. In other words, congestion levels impact the total completion times of picking waves. The gap size d_{g} is depicted in Table 5.8 for each of the 10 sample picking waves and d_{m}, d_{r} and d_{n} for the batching metrics that result in the lowest picking times. In most cases the Z-GS metric results

Wave ID	No batch			Random (FIFO)			Stops ratio (R-GR)						Spans ratio (A-GS)			Stops non-identical spans (Z-GS)					
	Cycles	Time	d_g	Cycles	Time	d_g	Cycles	Time	d_g	d_m	d_r	d_n	Cycles	Time	d_g	Cycles	Time	d_g	d_m	d_r	d_n
2405	1 788	49 645	5.44	894	38 809	4.73	894	38 797	4.50	0.28	0.29	21.23	894	38 727	4.63	894	38 708	4.26	0.29	0.30	22.14
2784	1 643	57 684	4.01	865	48 127	3.76	848	47 971	3.39	0.50	0.46	31.24	860	47 952	3.43	847	47 939	3.37	0.51	0.47	31.96
2433	1 540	53 393	6.24	864	42 683	5.47	825	42 583	5.21	0.39	0.33	24.57	852	42 578	5.31	807	42 548	5.07	0.42	0.35	24.58
3010	1 394	44 769	5.48	785	34 855	4.47	747	34 738	4.25	0.38	0.26	19.44	767	34 701	4.38	741	34 700	4.05	0.39	0.27	20.27
3318	1 698	46 392	7.59	864	36 125	6.96	863	36 307	6.78	0.29	0.26	19.37	870	36 097	6.77	857	36 021	6.74	0.30	0.27	20.16
3483	1 542	42 138	17.90	819	33 390	17.84	797	32 559	13.77	0.22	0.18	24.77	817	32 590	17.65	799	32 573	17.63	0.21	0.17	24.16
2580	1 364	39 782	4.40	738	31 403	3.56	718	31 343	3.31	0.43	0.32	22.01	723	31 369	3.25	710	31 342	3.24	0.44	0.33	22.77
2848	707	9 735	1.43	447	8 937	1.61	365	8 661	1.43	0.53	0.41	2.05	422	8 855	1.51	353	8 642	1.42	0.54	0.42	2.09
2856	366	13 930	1.77	271	13 334	1.81	188	12 931	1.73	0.56	0.19	1.75	249	13 166	1.95	187	12 912	1.71	0.57	0.20	1.80
2765	238	24 123	1.43	167	22 854	1.78	123	22 186	1.42	0.25	0.59	1.47	156	22 902	1.65	123	21 979	1.41	0.60	0.26	01.49
	12 280	**381 580**		**6 714**	**310 436**		**6 368**	**308 097**					**6 610**	**308 879**		**6 318**	**307 360**				

Table 5.8: *Comparison of different batching metrics. Walking distance is measured in cycles traversed on the picking line. Time is measured in (s). The three different density measurements have units according to their definitions.*

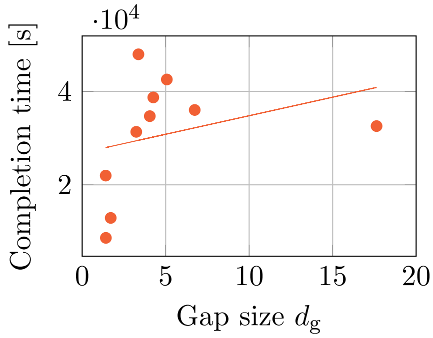
in the highest pick density making pickers pick more and travel less. Congestion increases with batching, but when batching is incorporated efficiently with a good number of pickers congestion can be lower than in a system without batching.

The Pearson correlation coefficient [101] is used to further analyse the interaction between walking distance, picking time and pick densities using order batching through the Z-GS metric. The correlation coefficient of 0.84 shows a strong positive correlation between completion time and total number of cycles traversed, confirming that less cycles correlate with shorter times. The correlation between picking time and pick densities is calculated to test the influence of density on picking speed. Thereby the correlation coefficient of 0.30 between completion time and gap size d_{g} indicates a positive moderate correlation, emphasising that a picker can gain speed in longer gaps as depicted in Figure 5.6(a). Route independent pick density measure d_{m} shows a strong positive correlation between completion time and density with a correlation coefficient of 0.78. This relationship is illustrated in Figure 5.6(b). Depicted in Figure 5.6(c), pick density measure d_{r} has a positive moderate correlation coefficient of 0.25 as an increase in value seems to indicate an increase in density (comparable to d_{m}). Even though pick density measure d_{n} has a strong positive correlation coefficient of 0.93 which is illustrated in Figure 5.6(d), it only considers averages and thereby may not be able to distinguish between different compositions of orders. For example d_{n} favours FIFO in all sample picking waves even though long picking times are produced by this batching metric. Therefore, the pick densities may not explain picking time solely and the influence of pick densities on a reduction of picking time has to be investigated further.

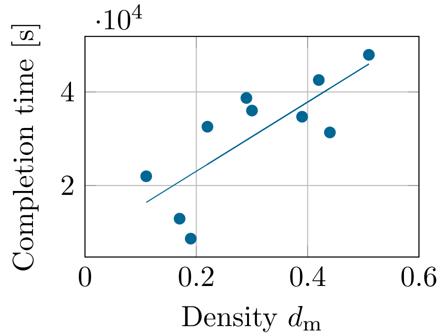
5.3.2 Explanatory variables

The differences in picking time reduction can be investigated with the help of explanatory variables such as number of orders, maximal SKU, number of locations, same locations, number of picks, and number of cartons. The four different pick densities will be tested on their ability to predict the magnitude of picking time reduction.

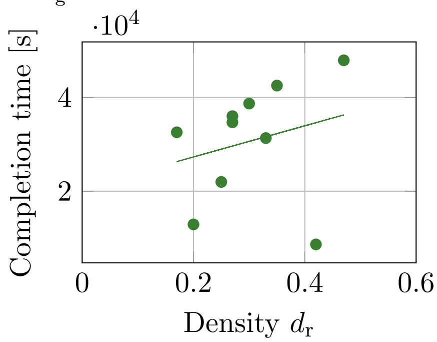
In Table 5.9 the explanatory variables for introducing order batching by applying Z-GS are depicted. In general, larger picking waves with more SKUs (and thus more locations, such as for example wave 3483 with 141 locations) seem to reduce picking time in accordance with the introduction of order batching more so than smaller picking waves (such as for example waves 2848, 2856 and 2765 with 5, 9 and 6 locations respectively). Picking waves with a higher number for maximal SKU seem to result in a higher picking time reduction through batching. If there are a lot of picks and a lot of SKUs needed from the same location (for example on wave 2784) a lower picking time reduction (due to congestion) is observed. The picking time reduction slowly decreases with many cartons to pack (for example on wave 2433).



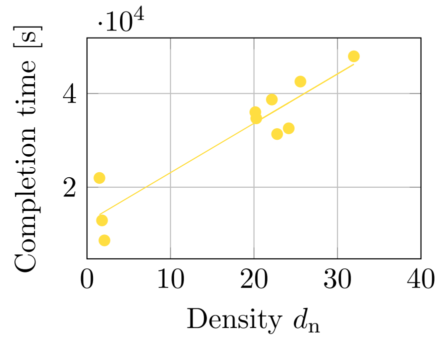
(a) *Correlation between time and gap size* d_g.



(b) *Correlation between time and density* d_m.



(c) *Correlation between time and density* d_r.



(d) *Correlation between time and density* d_n.

FIGURE 5.6: *Correlation between completion time and different measures of pick density, namely* d_g, d_m, d_r *and* d_n.

Wave ID	Pick time reduction	Orders	Max SKUs	Locations	Same location	Picks	Cartons
2405	22.09%	1 788	1 788	74	1 596	28 793	2 556
2784	16.89%	1 784	1 507	68	2 106	47 047	2 349
2433	20.31%	1 780	1 243	74	1 772	33 777	2 771
3010	22.49%	1 779	982	74	1 361	25 696	2 060
3318	22.36%	1 742	1 526	74	1 472	28 055	2 655
3483	22.73%	1 691	1 298	141	881	30 702	2 463
2580	21.21%	1 557	986	68	1 767	26 498	1 620
2848	11.22%	1 028	663	5	739	2 093	1 231
2856	7.31%	691	326	9	153	1 173	1 190
2765	8.86%	490	202	6	0	708	526

TABLE 5.9: *Explanatory variables for* 10 *sample picking waves.*

All variables can be predetermined by the number and quantity of SKUs per order or the set up of the picking wave except for the same location variable. The same location shows how often pickers had to stop and wait at the same location at which another picker was busy picking thus leading to congestion in the wave. Therefore, it can only be determined after a simulation run.

For each of the four pick densities a linear regression is carried out to determine which measure of pick density can best predict picking time reduction. In a stepwise linear regression, significant explanatory variables with a variance influence factor below 10 are combined with each pick density measure to generate a multiple regression model per pick density measure. The test data set of the 10 sample picking waves is extended to 60 waves by including the 50 sample picking waves from the Retailer's DC in Durban to increase the sample size.

The Retailer runs another DC in Durban, South Africa. The picking lines in this DC are made up of 56 locations with the conveyor belt in the middle. All processes are carried out similar. The 50 sample picking lines tested in Hofmann and Visagie [55, 56] are from the Durban DC, thus the simulation is run on these lines and results are presented in Appendix A.1 for completeness.

The results of a multiple regression model that includes the gap size d_{g} and the statistically significant explanatory variables (orders, locations, picks, and cartons) are shown in Table 5.10. The regression model results in an adjusted R^2 of 0.67. The coefficient estimate for d_{g}, and the locations is positive, indicating that if the gap size increases pickers walk faster and the picking time reduces. The more picks or cartons (as indicated by the negative coefficient estimates) the less time can be saved. However, the influence of picks and cartons on the picking time reduction is small.

Coefficients	Estimate	Std. error	t-value	p-value
Intercept	9.132	1.650	5.534	$9.4E-07$
Gap size d_{g}	0.251	0.055	4.604	$2.6E-05$**
Orders	0.006	0.001	6.596	$1.9E-08$**
Locations	0.121	0.029	4.133	$1.3E-04$**
Picks	-0.0002	0.0001	-3.015	$3.9E-03$**
cartons	-0.0013	0.0006	-2.248	$2.9E-02$**
Residual standard error:	3.082 on 54 degrees of freedom			
Adjusted multiple R^2:	0.6735			
F-statistic:	25.35			
p-value:	$4.639E-13$			

Note: Two asterisks indicate significance at the 5% level or below.

TABLE 5.10: *Multiple regression model summary on gap size d_{g} and explanatory variables.*

The results of a multiple regression model that includes the route independent pick density measure d_{m} and the statistically significant explanatory variables (orders, locations, and picks) are shown in Table 5.11. The regression model results in an adjusted R^2 of 0.58. The density d_{m}, orders and locations have a positive influence on the picking time reduction, thus the higher the density the more time can be saved. From all coefficient estimates the estimate for density d_{m} is the highest. Therefore, density measure d_{m} is the best indicator of picking time reduction.

Coefficients	Estimate	Std. error	t-value	p-value
Intercept	8.290	2.121	3.909	$2.6E-04$
Density d_{m}	10.250	4.574	2.241	$2.9E-02$**
Orders	0.006	0.001	5.853	$2.8E-07$**
Locations	0.164	0.032	5.133	$3.9E-06$**
Picks	-0.0004	0.0001	-7.592	$4.1E-10$**
Residual standard error:	3.5 on 55 degrees of freedom			
Adjusted multiple R^2:	0.579			
F-statistic:	20.99			
p-value:	$1.192E-10$			

Note: Two asterisks indicate significance at the 5% level or below.

TABLE 5.11: *Multiple regression model summary on pick density measure d_{m} and explanatory variables.*

Unfortunately, a linear regression of the pick densities d_{r} and d_{n} is not statistically significant. Therefore, a multiple regression would only incorporate the explanatory variables orders, picks and locations and the regression model would result in an adjusted R^2 of 0.55.

From the route independent pick density measures, only the multiple regression model of d_{m} is significant. Even though the adjusted R^2 of the d_{m} model is slightly lower than that of the d_{g} model, pick density measure d_{m} is a better predictor for picking time reduction, because it can be used independently of picker routing. A decrease in walking distance results in the average reduction of 17.54% in picking time in the dataset with 10 sample picking waves (the average reduction for the entire test data set of 60 sample picking waves is 20.91%). Therefore, it can be confirmed that a reduction in walking distance through order batching reduces the overall picking time.

5.4 Conclusion

The experiments showed that a reduction in walking distance by implementing a stops non-identical spans batching metric leads to up to 21% picking time reduction per picking wave. The route independent pick density measure d_{m} showed a strong correlation coefficient of 0.78 with total completion time. A multiple regression model that included d_{m} and the statistically significant explanatory variables (orders, locations, and picks) resulted in an adjusted R^2 of 0.58. Therefore, these variables can be used as a predictor for the possible reduction in completion time.

The picking lines of the Durban DC are run most efficiently with up to nine pickers per picking line under the current performance indicators. As suggested by Hofmann and Visagie [55] the stops non-identical spans batching metric is able to reduce the walking distance measured in cycles traversed to a minimum. The reduced walking distances resulted in the shortest overall picking time, thus confirming that a reduction in walking distance increases pick density and leads to a shorter picking time in the Durban DC.

Van Gils *et al.* [116] emphasises the importance of combining different planning problems to increase the overall efficiency in warehouses. As this chapter aims at introducing order batching to the order sequencing decision tier on the picking line, the decision tiers of where to place the SKU on the line and on which line to pick the SKU should be addressed.

5.5 Chapter summary

This chapter showed that order batching on a unidirectional cyclical picking line decreases not only the walking distance, but also leads to a reduction in picking time. The picking system of a Retailer's DC was modelled in a discrete event simulation. Input for the walking velocity, picking and handling distributions was retrieved through a time study carried out at the Retailer's DC. The output of the simulation was verified and validated against historical data from the Retailer's WMS. Different batching metrics, based on the picking location, the route overlap, or a combination of these were tested in an experimental set up.

DCs may incorporate a picking module that contains picking lines without conveyor belts in the middle. Managers can thus decide whether to run a unidirectional cyclical picking line or employ a Z-configuration which allows pickers to cross the aisle. The DES can be used as a tool to compare a unidirectional cyclical picking line set up to a Z-configuration, since it measures the picking time and not the walking distance. This will be the topic of the next chapter.

CHAPTER 6

Configuration selection on a unidirectional cyclical picking line

Contents

In this chapter, a new configuration to a unidirectional cyclical picking line is introduced. If the structural set up is changed by removing the conveyor belt in the middle, the picking line can operate either in a U- or Z-configuration. The question arises which configuration minimises walking distance and thereby picking time the most. Therefore, a worst case scenario, average case scenario, and a real world simulation are analysed. Both configurations are combined with adaptations of the order batching metrics to reduce walking distance. An indicator of when to switch between configurations is also investigated.

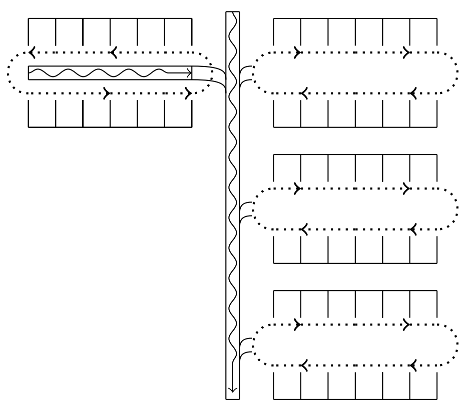
6.1 Introduction

The order picking system in the distribution centre (DC) of a prominent South African retailer (referred to as the Retailer) is considered in this chapter. The picking system transforms bulk stock into customer requests and is the main activity within a DC. Generally, order picking contributes the most to the operational cost in a DC [29, 113]. Therefore, optimising the order picking system has the potential to improve the efficiency of the entire supply chain of the Retailer.

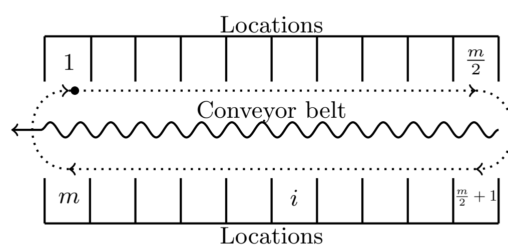
On a daily basis the Retailer serves about 2 000 stores with a large number of non-uniform orders due to varying store profiles. Mainly selling apparel to the lower income part of the population in South Africa, the Retailer needs to reach this market with stores all across the country. The Retailer has adopted a central inventory planning approach to keep operational cost low, while managing this large number of stores. Planners at the planning department of the central office allocate stock keeping units (SKUs) to stores, instead of store managers placing orders individually. This centralised approach drives the design of the order picking system. Once the central planner releases all the store requirements of a certain SKU to the DC, that SKU can be picked in a picking wave. A *picking wave* is processed on a picking line, where each SKU is assigned to a unique location. A picking wave consists of the following activities: populating the picking line with SKUs, picking the SKUs, and removing excess stock from the picking line [78].

The Retailer operates three DCs. While all DCs have structural differences, the fundamental layout of the picking system is the same. This chapter's findings can thus be applied to the picking systems in all of the Retailer's DCs.

The DC in this study runs a picking line with a conveyor belt in the middle and a module with three floors having a similar set up. However, the conveyor belt has been taken out in the module to allow for different walking configurations for pickers. The picking system is depicted in Figure 6.1(a). The three floors are connected by a main conveyor belt on to which full boxes are placed.



(a) *A schematic representation of the picking system layout with four picking lines.*



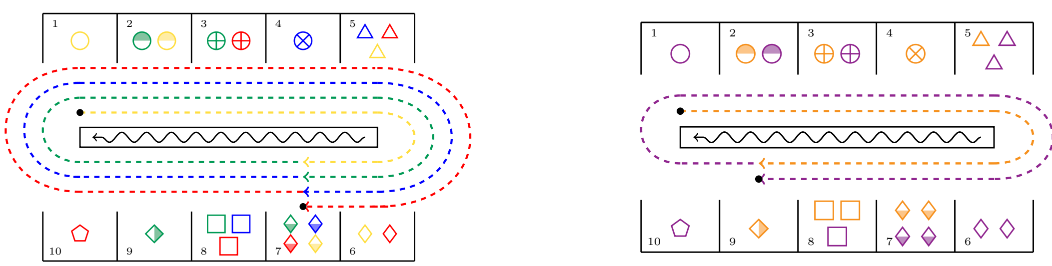
(b) *A schematic representation of a picking line with m locations including a conveyor belt. Source: De Villiers [30].*

FIGURE 6.1: *A schematic representation of the picking line with and without a conveyor belt in the middle. In the picking system, the picking line on the left is on the ground floor and has a conveyor belt. It closely resembles the detailed schematic representation of the picking line on the right. The picking lines next to it do not have conveyor belts in the middle.*

A schematic representation of a unidirectional cyclical picking line with m locations is shown in Figure 6.1(b). A picking line consists of 60 to 76 locations on two pick faces with or without a conveyor belt. Each location has the capacity to store up to five pallets of stock. Sufficient stock can thus be placed on the line before a wave of picking starts. The location of each SKU on the picking line is unique. The orders are deterministic as all store requirements per SKU are known prior to the start of a picking wave. This implies that no replenishment of SKUs are needed during a wave of picking.

The cyclical layout of the picking line system with pickers guided by a voice recognition system (VRS) is similar to a unidirectional carousel system. In a carousel system the operating picker remains in a fixed location during the picking process, while the carousel moves the product to the picker resulting in a parts-to-picker system. Usually small- to medium-sized products are picked by an automated carousel system [73]. In the Retailer's picking system SKUs are stationary and multiple pickers move in a clockwise direction to the next SKU that needs to be picked. All SKUs are at ground level and pickers sequentially process one order at a time while walking around the conveyor belt in a clockwise direction, thus classifying this order picking system as a single layer, single bin, unidirectional carousel according to Hassini [48]. Despite the similarities, the use of wave picking with all orders known and fixed *a priori* is the major difference between this system and carousel systems studied in literature. Furthermore, a unidirectional cyclical picking line can accommodate multiple pickers [78].

There are two fundamental approaches in manual order picking, namely single order picking and batch picking. In batch picking one picker is responsible for the picking of multiple orders simultaneously, while in single order picking the picker only picks one order at a time. In batch picking, the overall picking time can be reduced since multiple orders are picked simultaneously [28]. In Figure 6.2(a) four orders indicated by the colours yellow, red, green and blue are illustrated. Starting with picking the yellow order, followed by green, blue and red a picker would have to pass 37 locations. Randomly forming an orange batch out of the yellow and red order, and a purple batch out of the green and blue order, and starting the picking sequence with the orange batch, leads to only 19 locations that have to be passed as displayed in Figure 6.2(b). This example illustrates that order batching may be an efficient way to minimise walking distance and thus reduce order picking time. In this paper the number of orders that can be batched is restricted to two, as a trolley can only accommodate two orders at a time without having to sort orders after the batch picking process.



(a) *A schematic representation of single order picking.*

(b) *A schematic representation of batch picking.*

FIGURE 6.2: *An example to compare walking distance in single order picking and batch picking.*

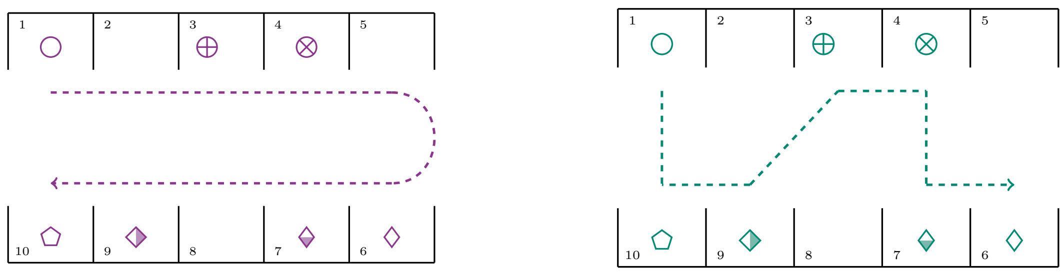
Some picking lines have no conveyor belt in the middle as illustrated in Figure 6.1(a). Therefore, it gives the option to either pick items in a unidirectional configuration (U) with the pickers walking clockwise along the locations, or in a Z-shaped configuration (Z) that allows pickers to cross from locations on the one pick face to the other. The objective of this paper is to determine the configuration – either Z or U – in which a wave should be picked to minimise picking time.

A brief background on the two configurations is given in Section 6.2 characterising each configuration and presenting walking distance minimisation approaches. Some theoretical results are presented in Sections 6.3 and 6.4. The results are confirmed by a simulation of the picking module to compare the U- to the Z-configuration in terms of total picking time in Section 6.5. The numerical experiment in Section 6.6 incorporates the adapted batching metrics of the unidirectional picking line. A potential indicator that helps to select the configuration is investigated.

Finally, an outlook on future work concludes this chapter in Section 6.7.

6.2 Background

The choice between the two configurations (U versus Z) will be the focus of this chapter as illustrated in Figure 6.3. Therefore, their main characteristics are pointed out and literature on the minimisation of walking distance in each configuration is reviewed.



(a) *A picker walking in the U-configuration.*

(b) *A picker walking in the Z-configuration.*

FIGURE 6.3: *A schematic representation of possible walking/picking paths of pickers in the U- and Z-configurations.*

In the U-configuration all pickers move in a clockwise direction along the locations as illustrated in Figure 6.3(a). A trolley is pushed by each picker and the VRS guides the pickers to the closest required SKU. Empty cartons are registered with the VRS before the picking process starts and order packslip codes are attached to each carton. If no conveyor belt is available, full cartons or cartons that contain completed orders are transported on the bottom shelf of the trolley to the next drop off point either at the start or at the end of the picking line [76, 77].

The unidirectional cyclical picking line with a conveyor belt in the middle was first introduced by Matthews and Visagie [78], who adapted methods from bidirectional carousels to sequence orders with the objective to minimise the walking distance per picking wave. They suggested the concept of a maximal cut to determine a solution within one cycle (the distance measurement in the unidirectional cyclical picking line) of a lower bound to the problem. However, the Retailer decided to implement the nearest end heuristic as an easy and fast option to sequence orders. The nearest end heuristic chooses from any given starting position the order with the nearest ending position next [76]. Both these methods can be applied to sequence the picking of order batches, because a batch can simply be viewed as one large order from a modelling perspective.

If there is no conveyor belt in the middle of the picking line, the picking line can also operate under a Z-configuration. In a Z-configuration, a picker alternates between the two pick faces moving in one direction. If a required SKU on the same pick face is closer then the next SKU on the opposite pick face, the picker stays on the same side. Otherwise, the picker crosses over to the opposite pick face, as illustrated in Figure 6.3(b). At the end of the picking line, the picker turns around to move in the opposite direction. The additional picking operations, such as handling cartons, picking, and interacting with the VRS, remain the same in the Z- as in the U-configuration. By taking the conveyor belt out, this new configuration is introduced to the picking system.

Goetschalckx and Ratliff [40] introduced a Z-shaped configuration as a classical order picking problem. In their study, a picker has to cross an aisle to reach the opposite pick face and walk to other aisles to pick all items. For this chapter, the structure of the module (the picking line without the conveyor belt) is such that a picker can remain between the two pick faces of the module and thus stays within one aisle. Picking in a Z-configuration is comparable to the traversal picking policy of Goetschalckx and Ratliff [40] as the picker starts and finishes at two different locations on the picking line. Therefore, this policy is used to determine picker paths in the Z-configuration of this system. The nearest end heuristic suggested by Matthews and Visagie [78] is adapted for the Z-configuration to sequence batches that reduce the overall picking time. Goetschalckx and Ratliff [40] concluded that the density of the order and the width of the aisle determine the walking distance. They showed that for most aisle widths in practice, it is more efficient to pick from both pick faces rather than a single pick face at a time if the order densities are below 0.5 [61]. Pick density might thus be used as an indicator to determine when the Z-configuration outperforms the U-configuration.

Order batching was introduced to the unidirectional cyclical picking line by Hofmann and Visagie [55, 56] as an efficient way to reduce walking distance. In their discrete event simulation (DES) of the unidirectional cyclical picking line with conveyor belt, Hofmann and Visagie [59] showed that the reduction of walking distance through order batching can be translated into a reduction in picking time. The order batching metrics and the DES can be adapted for the Z-configuration that is introduced in this paper.

The management of the Retailer may decide to either run in a Z-configuration, in which case a picker can cross the aisle to different pick faces, or in a U-configuration (similar to the picking line with the conveyor belt). The following question arises: Under which circumstances will a Z-configuration outperform a U-configuration? Similar to the findings of Goetschalckx and Ratliff [40] the pick density may be used as an indicator of when to switch from a Z- to a U-configuration. Gue *et al.* [44] suggests that the pick density has a direct influence on congestion in a warehouse layout with parallel narrow aisles. Furthermore, the pick density is significantly influenced by the batching algorithm according to Hong [60].

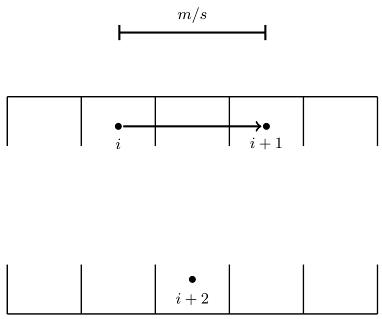
The effect of order batching and the pick density on the U- and Z-configuration are investigated. A worst case, an average case, and a real world simulation are used to determine the best configuration for the Retailer's picking system.

6.3 Worst case behaviour

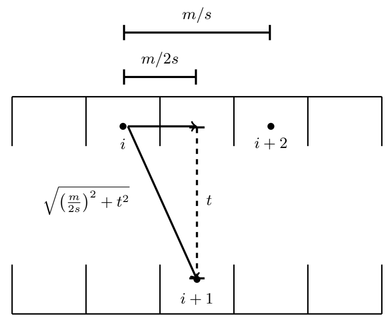
In Figure 6.4 a picking sequence of three items (or SKUs) within an order is illustrated. It is assumed that each order contains at least three or more picks (or stops). The number of locations is indicated by m, the number of stops for this order by s, and the aisle width (the distance between two pick faces) is denoted by t. A gap between two picks within an order are the locations that are passed without stopping to pick. Therefore, a picker walks $\frac{m}{s}$ locations between picks. The pick density of an order is the number of stops for that order divided by the number of locations on the picking line, in other words $\frac{s}{m}$.

The worst case scenario for the U-configuration arises when the largest gap within an order is minimised as illustrated in Figure 6.4(a). This will happen when all the gaps are equal in size – or when the stops are spread out as evenly as possible around the picking line. Such an arrangement will result in the longest possible walking distance to pick all the SKUs in an order. Considering one order at a time, the worst case for the U-configuration is to walk $\frac{m}{s}$ locations $s-1$ times to pick an order.

The worst case scenario for the Z-configuration is when the items of an order are divided in half



(a) *Worst case order location distribution in the U-configuration.*



(b) *Worst case order location distribution in the Z-configuration.*

FIGURE 6.4: *Comparison of worst case location distributions for the U- and Z-configuration.*

and the two halves are spread out as equally as possible along the two pick faces. A picker thus has to cross from one side of the aisle to the other after every pick. An example of this worst case scenario is depicted in Figure 6.4(b). Considering one order at a time, the worst case in the Z-configuration is to walk $\sqrt{\left(\frac{m}{2s}\right)^2 + t^2}$ locations $s-1$ times to visit the s stops to pick an order.

The worst case scenarios for both configurations are depicted in Figure 6.4. This setup results in the U-configuration generating a shorter walking distance if

$$(s-1)(\frac{m}{s}) < \left[\left(\frac{m}{2s}\right)^2 + t^2\right]^{\frac{1}{2}}(s-1) \tag{6.1}$$

$$\frac{m^2}{s^2} < \frac{m^2}{4s^2} + t^2 \tag{6.2}$$

$$\frac{4m^2}{4s^2} - \frac{m^2}{4s^2} - t^2 < 0 \tag{6.3}$$

$$4s^2t^2 > 3m^2 \tag{6.4}$$

$$s^2 > \frac{3m^2}{4t^2} \tag{6.5}$$

$$s > \sqrt{\frac{3}{4}}\frac{m}{t}. \tag{6.6}$$

Note that, because of the assumption that $s \geq 3$, it follows that $s \neq 1$ and thus the division by $s-1$ in equation (6.1) is allowed. Also, because an order can only have a positive number of picks the negative of the square root may be ignored in equation (6.6).

From the result in equation (6.6) it is clear that the number of stops s in an order must be sufficiently large to prefer the U-configuration in terms of walking distance. This confirms the idea that pick density is important in making a decision between U- and Z-configurations.

Table 6.1 depicts a small example with 30 to 120 locations m per picking line and aisle width t ranging from 1 to 5 locations. For example, if the aisle width is two locations and the number of locations on the picking line is 60, then the order has to have more than 26 stops for the U-configuration to generate a shorter distance than the Z-configuration according to equation (6.6). The last column shows at which pick density $\frac{s}{m}$ the configuration on the picking line should change from a Z- to a U-configuration to obtain a shorter walking distance in the worst case scenario.

	Number of locations in the picking line										Critical density
t	30	40	50	60	70	80	90	100	110	120	$\frac{s}{m}$
1	26	35	43	52	61	69	78	87	95	104	0.87
2	13	17	22	26	30	35	39	43	48	52	0.43
3	9	12	14	17	20	23	26	29	32	35	0.29
4	6	9	11	13	15	17	19	22	24	26	0.22
5	5	7	9	10	12	14	16	17	19	21	0.17

TABLE 6.1: *A summary of the worst case scenario for both the U- and Z-configuration. The body of the table contains the number of stops, s, needed to prefer the U-configuration above the Z-configuration when using equation (6.6). These values range from $30 - 120$ locations in the picking line and an aisle width, t, ranging between 1 and 5 locations. The critical density column provides the pick density (expressed as a fraction) at which the U-configuration will result in a shorter walking distance than the Z-configuration for each aisle width.*

In the real world application most picking line set ups hold space for around 70 locations. The aisle width is approximately three locations. In a practical application, the Z-configuration is thus preferred at a pick density $\frac{s}{m}$ that is below 0.29 in the worst case scenario.

In the best case scenario the U-configuration always outperforms the Z-configuration, since there are no empty locations between the stops of an order in this scenario. In other words, for this scenario, crossing the aisle at any point would increase the walking distance since SKUs can simply be picked sequentially along the line.

6.4 Average case behaviour

In an average case, one order is simulated by randomly arranging SKUs (or stops) into locations on the picking line. The shortest walking distance for this random arrangement for both configurations is then calculated and compared. The average case simulation (AC) is described in Algorithm 10 with α simulation replications.

Algorithm 10: `Average case simulation (AC)`

Input: A number of SKUs s, a number of locations m, a aisle width t, α simulation replications
Output: The average difference in walking distance between the configurations for every number of SKUs

1: **for** k in s **do**
2: **for** j in α **do**
3: Randomly arrange k on m
4: Calculate walking distance in U-configuration
5: Calculate walking distance in Z-configuration
6: Subtract walking distance in U-configuration from walking distance in Z-configuration and add it to the total difference
7: **end for**
8: Divide total difference in walking distance by α and add it to the output list
9: **end for**
10: Return a list containing the average difference in walking distance between the U- and Z-configuration for each k

In Table 6.2, the aisle width t ranges from 1 to 5 locations, while the number of locations per picking line m ranges from 30 to 120 locations. If the aisle width is four locations and the number of locations on the picking line is 80 for example, then walking in the Z-configuration is shorter, because it produces a shorter walking distance until up to 13 stops, s. The last column of Table 6.2 shows the pick densities for the average case simulation. These values can be used to determine when one should switch from a Z-configuration to a U-configuration.

	Number of locations in the picking line										**Critical density**
t	30	40	50	60	70	80	90	100	110	120	$\frac{s}{m}$
1	18	25	31	38	44	51	58	64	71	77	0.63
2	9	13	17	21	25	28	32	36	40	44	0.34
3	7	8	11	13	16	19	21	24	27	30	0.22
4	4	5	7	10	12	13	16	18	20	22	0.16
5	3	4	6	7	9	10	12	14	15	17	0.12

TABLE 6.2: *A summary of the average case scenario for both the U- and Z-configurations. The body of the table contains the number of stops, s, needed to prefer the U-configuration above the Z-configuration when using Algorithm 10 with $\alpha = 1\ 000$ simulation replications. These values range from $30 - 120$ locations in the picking line and an aisle width, t, ranging between 1 and 5 locations. The critical density column provides the pick density (expressed as a fraction) at which the U-configuration will result in a shorter walking distance than the Z-configuration for each aisle width.*

In real life most picking lines can accommodate around 70 locations with an aisle width slightly narrower than three locations. The Z-configuration outperforms the U-configuration until a pick density of approximately 0.22 in the average case simulation. This density is marginally lower than in the worst case scenario where only one order is processed. However, a real world application will consist out of more than one order and the locations of the items of an order will be predefined. Therefore, a simulation of the real world application will be used to compare the walking time of both configurations.

6.5 Real world simulation

A DES suggested by Hofmann and Visagie [59] measures the picking time on a unidirectional cyclical picking line with a conveyor belt. This DES is modified to measure picking time in the U-configuration without a conveyor belt and the Z-configuration to compare the total completion times of the two configurations. The module in the Retailer's DC describes the environment in which the simulation takes place and is depicted in Figure 6.5. On each side, unique SKUs are placed at the picking line's locations. Pickers walk according to the configuration along the picking line pushing a trolley with the capacity of two orders. Locations can be either unoccupied or occupied when a picker stops to request SKUs. Each picker carries out the three activities of walking to a SKU, picking a SKU, and handling the carton into which the SKU gets placed. Therefore, each picker's attributes are the speed at which these activities are carried out.

FIGURE 6.5: *Picking line in the module of the Retailer's DC.*

The assumptions that align the simulation with the real life set up of the Retailer's picking module are the same as for the picking line with a conveyor belt described by Hofmann and

Visagie [59]. Firstly, pickers complete an order (or batch) before they start a new one. Secondly, the times for picking and handling are independent of SKU and location. Thirdly, it is assumed that there are no variations between pickers concerning the proportion of time spent per activity.

In Figure 6.6 the logic of the simulation is illustrated in a flow chart. Multiple pickers operate in the picking system. If a new order is started the picker prepares a carton (handling of the box) and walks to the location of the first SKU of the order according to the U- or Z-configuration. The logic of the pick path is the main difference between the two configurations. If a location is occupied, the picker has to wait for the person preceding her to move on to the next location before performing her pick. If the location is unoccupied, the picker picks the SKU required from that location. If the current carton is full, or the requested SKU is the last SKU of the order, the carton is placed on the conveyor belt in the U-configuration (with conveyor belt). In the Z-configuration, the carton gets packed and stored at the bottom of the trolley until it can be dropped off at the start or end of the picking line. Until the picking wave is completed this process is repeated.

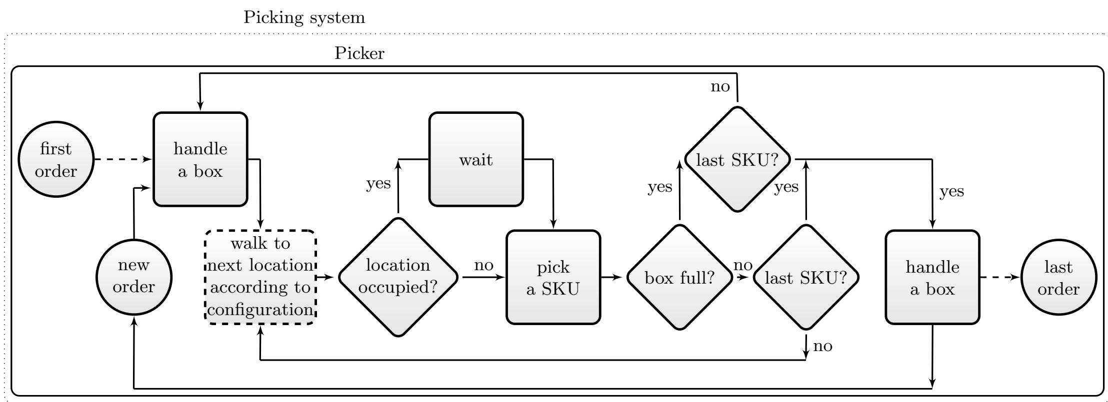


FIGURE 6.6: *A schematic representation of the logical flow chart of the simulation model in the Retailer's module. The pick path logic in the walking activity depends on the configuration.*

For the implementation of the simulation Python 3.6 [94] utilising the C-based libraries Numpy [89] and Pandas [93] was used. The simulation was run on a Dell Optiplex 5050 with a Intel Core i7-7700 CPU at 3.6 GHz, 1x8GB 2400MHz DDR4 RAM, a 2.5" 256GB SSD class 20 drive, and the Microsoft Windows 10 Enterprise 2016 LTSB operating system [84]. The statistical analysis of the input and simulated output data was carried out in R [95].

6.5.1 Input data capturing

In the DES the picker entity has three different activities, namely (1) walking from one picking location to the next, which might include crossing from one side to the other in the Z-configuration, (2) picking and (3) handling which consists of preparing and packing a carton. The picking process in the module of the DC was analysed to calculate walking velocities, the picking time distributions, and handling time distributions.

Table 6.3(a) shows the proportion of time spent on each task for the U-configuration as it pertains to the picking module. For the U-configuration, picker tasks include: walking, picking, and handling. Table 6.3(b) shows the proportion of time spent on each task for the Z-configuration. For this configuration crossing the aisle is included as a task. Here, the combination of walking and crossing accounts for about 33.98% of the total time spent on tasks. While the time spent on handling is almost the same between the two configurations, the walking time proportion is higher in the Z-configuration as it includes the manoeuvring of the trolley during the crossing activity. Combining the extracted proportions with time stamp data from the Retailer's warehouse management system, the walking, picking and handling times could be calculated.

Walking	28.55%
Picking	51.18%
Handling	20.27%

(a) *Time proportion per activity in the U-configuration.*

Walking	21.86%
Crossing	12.11%
Picking	43.78%
Handling	22.25%

(b) *Time proportion per activity in the Z-configuration.*

TABLE 6.3: *The proportion of time spent by the pickers per activity in the U- and Z-configuration.*

In calculating the average walking velocity, picking, and handling times, a minimum of 10 cycles or orders without excessive time delays (breaks for example) were considered. Over these 10 cycles or orders the number of passed locations, picks, and handled boxes can be determined.

The time stamp data for a picker completing at least 10 cycles in the U-configuration is shown in Table 6.4. If the starting point of the first order (additional 1 location) and the final location of the last order (additional 32 locations) are included and the picking line has 64 locations, this picker has to pass 673 locations. The picker prepared and packed 30 cartons and picked 468 times. The picker spent a total of 3 215 seconds on the picking process in the U-configuration. The average rates per activity can be determined from this. The picker passed 673 locations in 918 seconds. This yields a walking velocity of 1.3639 seconds per location. The picking time and handling time was calculated as 3.5156 seconds per pick, and 21.7266 seconds per carton respectively.

	Proportion	Total time	Events	Average rates
Walking	28.55%	918	673 locations	1.3639 s/location
Picking	51.18%	1 645	468 picks	3.5156 s/pick
Handling	20.27%	652	30 cartons	21.7266 s/carton

TABLE 6.4: *An example of the table to calculate walking velocity, picking and packing times for a specific picker in the U-configuration. Time is measured in (s).*

In Table 6.5 an example of a picker completing 10 orders in the Z-configuration is shown, since no cycles are completed. This picker passes 396 locations and crosses between the locations 97 times to collect the items of the 10 orders. Thereby, the picker picks 379 times and handles 43 cartons. A total of 3 519 seconds is spent on picking the 10 orders. Therefore, the average walking velocity is 1.9427 seconds per location with an average rate of crossing velocity of 4.3949 seconds per crossing. Picking time is on average 3.5156 seconds per pick, and handling a carton can be achieved in 18.2066 seconds per carton in this example.

	Proportion	Total time	Events	Average rates
Walking	21.86%	769	396 locations	1.9427 s/location
Crossing	12.11%	426	97 crossings	4.3949 s/crossing
Picking	43.78%	1 541	379 picks	3.5156 s/pick
Handling	22.25%	783	43 cartons	18.2066 s/carton

TABLE 6.5: *An example of the table to calculate walking velocity, picking and packing times for a specific picker in the Z-configuration. Time is measured in (s).*

The triangular distributions of the average walking (and crossing) velocities, and the average handling and picking times for any specific picker are used as input for the simulation. The lower limit is defined by the smallest value over all pickers and the upper limit is determined by the largest value over all pickers [59].

The pick density measure d_{m} as defined by Hofmann and Visagie [59] shows how often a picker has to stop per distance travelled. It plays an important role in the simulation as it influences the total completion time of a picking wave. Matthews and Visagie [78] proposed the SKU with the greatest number of store requirements as a lower bound for the number of picking cycles that are traversed. This SKU is called the *maximal SKU*. Therefore, the pick density measure d_{m} takes the number of all stops in a picking wave and divides it by an estimate of walking distance defined by the maximal SKU multiplied by the number of locations on the picking line. This density measure ranges from 0 to 1. A value closer to 1 indicates a higher pick density, while a value closer to 0 indicates a lower pick density.

	Lower	Central	Upper
Walking	1.2548	1.5470	1.9626
Picking	3.5022	4.5775	6.0200
Handling	21.2922	24.4368	32.5645

(a) *Triangular distributions with low pick density in the U-configuration.*

	Lower	Central	Upper
Walking	1.1548	1.5701	1.9626
Picking	3.5022	4.4525	5.1556
Handling	18.9453	23.7504	32.5645

(b) *Triangular distributions with high pick density in the U-configuration.*

	Lower	Central	Upper
Walking	1.4559	1.8211	2.3319
Crossing	3.1448	4.7523	7.9214
Picking	3.2627	4.1459	5.6186
Handling	13.9286	21.2867	27.9213

(c) *Triangular distributions with low pick density in the Z-configuration.*

	Lower	Central	Upper
Walking	1.4559	1.9069	2.5069
Crossing	3.1448	4.5023	5.9214
Picking	3.7984	4.6244	5.6186
Handling	13.9286	21.2867	31.9213

(d) *Triangular distributions with high pick density in the Z-configuration.*

Table 6.6: *Triangular distributions for picking lines with different configurations and different pick densities* d_{m}.

There are different distributions for a high and low density picking line, since generally pickers walk faster when the pick density is low since they gain more speed walking towards the next SKU without stopping often. The distributions of each configuration for low pick densities are illustrated in Table 6.6(a) and Table 6.6(c), while the distributions for high pick densities are shown in Table 6.6(b) and Table 6.6(d). Hofmann and Visagie [59] showed that the pick density d_{m} influences picker speed.

An exponential distribution can be applied to both the picking and handling times. The goodness of fit (GOF) of these distributions were measured by the Kolmogorov-Smirnov test [101] and the Anderson-Darling test [26]. The results are shown in Tables 6.7 and 6.8 for both configurations. The test statistics were not rejected at a 99% confidence level.

6.5.2 Verification and validation of output data

The verification and validation (VV) of a simulation ensures that it accurately represents the real world system [100]. Hofmann and Visagie [59] verified and validated the DES model for a unidirectional cyclical picking system with a conveyor belt through conceptual model validation, data validation (statistical testing of input data), white-box, and black-box validation. Whether a conveyor belt is part of the picking line set up or not does not change the main activities of

Exponential distribution: $\lambda = 0.251755$					
Kolmogorov-Smirnov					
Size					2 690
Statistic					0.0162
P-value					0.4798
Alpha	0.2	0.1	0.05	0.02	0.01
C-value	0.021	0.024	0.026	0.029	0.031
Reject?	no	no	no	no	no
Anderson-Darling					
Size					2 690
Statistic					0.5328
P-value					0.7136
Alpha	0.2	0.1	0.05	0.02	0.01
C-value	0.816	1.162	1.321	1.591	1.959
Reject?	no	no	no	no	no

(a) *GOF for the picking distribution.*

Exponential distribution: $\lambda = 0.179220$					
Kolmogorov-Smirnov					
Size					809
Statistic					0.0217
P-value					0.8401
Alpha	0.2	0.1	0.05	0.02	0.01
C-value	0.038	0.043	0.048	0.053	0.057
Reject?	no	no	no	no	no
Anderson-Darling					
Size					809
Statistic					0.3639
P-value					0.8836
Alpha	0.2	0.1	0.05	0.02	0.01
C-value	0.816	1.162	1.321	1.591	1.959
Reject?	no	no	no	no	no

(b) *GOF for the handling distribution.*

TABLE 6.7: *Goodness of fit test for picking and handling distributions in the U-configuration.*

Exponential distribution: $\lambda = 0.243666$					
Kolmogorov-Smirnov					
Size					2 241
Statistic					0.0142
P-value					0.7555
Alpha	0.2	0.1	0.05	0.02	0.01
C-value	0.023	0.026	0.029	0.032	0.034
Reject?	no	no	no	no	no
Anderson-Darling					
Size					2 241
Statistic					0.6826
P-value					0.5736
Alpha	0.2	0.1	0.05	0.02	0.01
C-value	0.816	1.162	1.321	1.591	1.959
Reject?	no	no	no	no	no

(a) *GOF for the picking distribution.*

Exponential distribution: $\lambda = 0.171663$					
Kolmogorov-Smirnov					
Size					773
Statistic					0.0337
P-value					0.3429
Alpha	0.2	0.1	0.05	0.02	0.01
C-value	0.038	0.044	0.049	0.055	0.059
Reject?	no	no	no	no	no
Anderson-Darling					
Size					773
Statistic					0.7004
P-value					0.5585
Alpha	0.2	0.1	0.05	0.02	0.01
C-value	0.816	1.162	1.321	1.591	1.959
Reject?	no	no	no	no	no

(b) *GOF for the handling distribution.*

TABLE 6.8: *Goodness of fit test for picking and handling distributions in the Z-configuration.*

walking, picking, and handling on the picking line. Therefore, the VV carried out by Hofmann and Visagie [59] also confirms the interactions of the DES in the U-configuration without conveyor belt. A black-box validation is carried out for the U- and Z-configuration separately, since it includes the comparison of model output and real life data.

The conceptual model validation for the Z-configuration confirms that the DES is sufficiently detailed since it includes all relevant picker activities (walking, picking, handling, and crossing) [96]. The management of the Retailer confirms that the level of detail of the simulation is adequate. The input data has been statistically tested and thus validated during the capturing in Section 6.5.1. The consistency between the conceptual model and the DES for the Z-configuration is ensured by model verification [96]. The white-box validation of the DES [96] for the Z-configuration has checked whether a picker walks, crosses, picks and handles at the correct times and in the right sequence. A single picker was modelled and the picker's activities were tested to perform this validation. Additionally, the picker behaviour when encountering another picker was assessed for correctness.

The number of simulation replication influences the accuracy of the model in the black box validation of both configurations. Hofmann and Visagie [59] determined that 10 replications are adequate. This is confirmed for both configurations. The Retailer selected 10 sample picking waves of each configuration as representatives. These samples are compared against simulation outputs.

For the U-configuration the comparison between the historical and simulated completion times is illustrated in Table 6.9. The simulation times seem to be accurate since the calculated differences are normally distributed with a mean that is close to zero. The standard deviation indicates the spread of values that is close to the mean. The mean of 10 replications falls between the 95% confidence intervals for each picking wave as illustrated in Table 6.9(a). The Kolmogorov-Smirnov test [101] and Anderson-Darling test [26] are used to measure the GOF of the normal distribution that was fitted to the differences in total completion times. In Table 6.9(b) the example of picking wave 3519 shows that the test statistic was not rejected for a 99% confidence level. All picking wave outputs for the U-configuration have been confirmed in this manner.

Wave ID	**Mean errors**	**Standard deviation of errors**	**Confidence intervals**
3519	12.67	115.19	$[-59, 84]$
3357	46.05	174.33	$[-62, 154]$
3378	71.51	134.23	$[-12, 154]$
3600	87.03	144.54	$[-3, 176]$
3662	−7.35	101.83	$[-71, 56]$
3554	7.28	101.83	$[-73, 88]$
3520	−50.12	144.41	$[-138, 39]$
3317	−30.74	85.90	$[-84, 22]$
3516	−27.44	103.09	$[-91, 36]$
3369	−70.07	144.21	$[-160, 19]$

(a) *Confidence intervals for the time differences. Time in (s).*

Normal distribution: $\mu = -107.33, \sigma = 115.19$					
Kolmogorov-Smirnov					
Size					10
Statistic					0.1994
P-value					0.7520
Alpha	0.2	0.1	0.05	0.02	0.01
C-value	0.323	0.369	0.409	0.457	0.489
Reject?	no	no	no	no	no
Anderson-Darling					
Size					10
Statistic					0.2238
P-value					0.7590
Alpha	0.2	0.1	0.05	0.02	0.01
C-value	0.509	0.631	0.752	0.873	1.035
Reject?	no	no	no	no	no

(b) *GOF for the time differences on picking wave* 3519.

TABLE 6.9: *Differences in total completion times between historical data and simulation data for* 10 *sample picking waves in the U-configuration.*

The comparison of the historical and simulated completion times for the Z-configuration is shown in Table 6.10. Differences are normally distributed with a mean close to zero, indicating that simulation times are accurate. The standard deviation shows that values are spread out close to the mean. Additionally, for each picking wave the mean of 10 replications falls between the 95% confidence intervals as depicted in Table 6.10(a). In Table 6.10(b) the example of picking wave 3660 shows that the test statistic was not rejected for a 99% confidence level in the Kolmogorov-Smirnov [101] and Anderson-Darling [26] GOF test of the normal distribution that was fitted to the differences. All picking wave outputs for the Z-configuration have been confirmed in this manner.

A marginal analysis compares the time lost to congestion to the time gained in overall completion time of a picking wave with an increase in pickers, as introduced to the unidirectional cyclical picking system by Hofmann and Visagie [59]. It shows that 8 or less pickers is a good number for both configurations.

The verification and validation of the simulation shows that the model represents the real world application to a satisfactory degree. To simulate reality, the DES (with the picker number set to 8, and picker attributes as discussed above) will be used to compare the picking wave completion times of the different configurations including order batching. It will be tested if the pick density d_{m} can indicate which configuration leads to faster completion times.

Wave ID	Mean errors	Standard deviation of errors	Confidence intervals
3660	14.00	174.09	$[-94, 122]$
3410	38.20	185.44	$[-77, 153]$
3141	-20.07	154.92	$[-116, 76]$
3569	-54.01	156.83	$[-101, 7]$
3281	61.26	106.27	$[-33, 189]$
3735	27.79	128.80	$[-52, 108]$
3597	-6.43	93.03	$[-64, 51]$
3441	38.41	146.41	$[-14, 91]$
3709	-76.33	149.22	$[-169, 16]$
3348	10.54	46.41	$[-18, 39]$

(a) *Confidence intervals for the time differences. Time in (s).*

Normal distribution: $\mu = 14.004, \sigma = 174.09$					
Kolmogorov-Smirnov					
Size					10
Statistic					0.2348
P-value					0.5632
Alpha	0.2	0.1	0.05	0.02	0.01
C-value	0.323	0.369	0.409	0.457	0.489
Reject?	no	no	no	no	no
Anderson-Darling					
Size					10
Statistic					0.2116
P-value					0.8006
Alpha	0.2	0.1	0.05	0.02	0.01
C-value	0.509	0.631	0.752	0.873	1.035
Reject?	no	no	no	no	no

(b) *GOF for the time differences on picking wave* 3660.

TABLE 6.10: *Differences in total completion times between historical data and simulation data for* 10 *sample picking waves in the Z-configuration.*

6.6 Simulation results

The real world simulation is applied to investigate the influence of the picking system configuration on the total completion time of a picking wave. Order batching has been shown to reduce picking time significantly and will be introduced through adapted batching metrics and tested on both configurations. Furthermore, it will be investigated whether the pick density measure can be used as an indicator to choose one of the configurations.

6.6.1 Input datasets

The total simulated completion time per picking wave was determined for 10 historical picking waves with U-configuration and 10 waves with Z-configuration to compare the two configurations in a simulation with real world input data. Therefore, the average total completion times of 10 replications of each experiment are compared. For each configuration, the Retailer selected sample picking waves that are representative. The Retailer applies the rule of thumb that if a picker has to stop at less than a third of the locations the Z-configuration is used.

The information on the U-configuration waves is depicted in Table 6.11(a), while the waves of the Z-configuration are illustrated in Table 6.11(b). Each picking wave depicts information on the number of orders, the number of locations, the number of picks and the number of cartons as these factors influence the overall picking time [59]. A large dataset contains more than 1 580 orders. This is followed by a medium dataset that contains between 1 551 − 1 580 orders. Finally a small dataset contains less than 1 551 orders.

6.6.2 Simulation output

Different batching metrics were developed by Hofmann and Visagie [55, 56] to reduce picking time on a unidirectional cyclical picking line [59]. They are used here to investigate their effect on the choice of configuration through a minimisation of picking time and the pick density measure d_{m}.

The batching metrics that are applied to these sample picking waves are a first-in-first-out

Wave ID	Orders	Locations	Picks	Cartons
Large data set				
3519	1 783	76	27 177	2 138
3357	1 716	64	15 434	2 017
3378	1 716	76	15 506	2 186
3600	1 672	76	16 082	2 143
3662	1 654	64	18 284	2 275
Medium data set				
3554	1 566	64	31 418	2 030
3520	1 561	76	34 065	1 864
3317	1 558	64	15 152	2 878
Small data set				
3516	1 550	64	10 343	1 706
3369	1 472	76	5 052	1 580

(a) *Sample picking waves in the U-configuration.*

Wave ID	Orders	Locations	Picks	Cartons
Large data set				
3660	1 801	72	22 838	2 709
3410	1 730	60	20 958	1 859
3141	1 725	72	15 806	2 556
3569	1 595	72	16 575	1 899
3281	1 583	72	34 587	1 821
Medium data set				
3735	1 566	60	28 998	1 689
3597	1 563	60	46 164	1 761
3441	1 561	60	24 978	2 915
Small data set				
3709	1 432	72	8 577	1 631
3348	687	60	1 178	690

(b) *Sample picking waves in the Z-configuration.*

TABLE 6.11: *Historical data for 10 sample picking waves in the U- and Z-configuration.*

(FIFO) random approach as a benchmark, the stops ratio batching metric in combination with a greedy random heuristic (R-GR), and the stops non-identical spans metric combined with a greedy smallest entry heuristic (Z-GS). These metrics have shown to reduce the total completion time in this order picking system using a U-configuration [59]. The picking times without order batching are included for comparison.

The stops ratio metric (R-GR) divides the number of stops that are not similar between two orders by the number of stops that the two orders have in common [55]. This metric can be applied to both the U- and Z-configuration. In a unidirectional cyclical picking line the span of an order is the distance that has to be covered for the collection of all items of that order. The *minimum span* is determined by finding the biggest gap between any two SKUs of an order. The end of the biggest gap is the start of the minimum span [78]. Even though there is no unidirectional cycle in the Z-configuration, this logic can be adapted to finding a minimum path in the Z-configuration. Therefore, the stops non-identical spans metric (Z-GS), as suggested by Hofmann and Visagie [56], adds the minimum span or path to the number of stops that are not shared between two orders.

In Table 6.12, which presents picking waves that were run in U-configuration historically, the shortest simulated total completion time is achieved in eight of the 10 picking waves by applying the stops non-identical spans metric to the picking waves. The U-configuration produces shorter total completion times than the Z-configuration in picking waves 3554 and 3520 only. Additionally, picking waves 3554 and 3520 are the only waves that have a pick density measure d_{m} above 0.4. Generally it seems that the higher the pick density the greater the time saving will be when switching to a U-configuration. The random batching of orders (FIFO) could not outperform the other two batching metrics. The average time saving of applying order batching to the 10 sample picking lines of the best performing configuration is 43.60%.

In Table 6.13, which presents picking waves that were run in Z-configuration historically, the stops non-identical spans batching metric also provides the shortest total completion times in eight out of the 10 picking waves. The stops ratio batching metric also seems to work well in combination with the Z-configuration. Even though picking waves 3660, 3410, 3141, 3569, 3709 and 3348 produce the shortest picking time in the Z-configuration, the other four picking waves perform faster in a U-configuration. The pick density measure d_{m} in these cases is above 0.4. The higher the pick density the more often the picker is forced to cross the aisle in the Z-configuration, and this increases the overall picking time. The developed batching metrics all outperform the random batching approach. In comparison to not introducing order batching,

Wave ID	No batch	Random (FIFO)	Stops ratio (R-GR)			Stops non-identical spans (Z-GS)		
			Density d_m	U	Switch to Z	Density d_m	U	Switch to Z
3519	41 716	29 263	0.330	29 021	$\boxed{25\,760}$	0.327	28 967	25 964
3357	30 264	20 079	0.208	19 526	17 096	0.218	19 471	$\boxed{17\,093}$
3378	33 482	21 899	0.183	21 337	16 567	0.188	21 286	$\boxed{16\,560}$
3600	32 162	21 199	0.380	20 837	$\boxed{16\,094}$	0.366	20 591	16 143
3662	31 893	20 907	0.232	20 859	19 871	0.240	20 824	$\boxed{19\,802}$
3554	37 633	28 264	0.427	28 226	29 164	0.466	$\boxed{28\,222}$	29 214
3520	40 868	29 898	0.466	28 458	29 942	0.478	$\boxed{28\,430}$	29 973
3317	28 721	18 906	0.253	18 704	16 664	0.263	18 606	$\boxed{16\,639}$
3516	22 012	14 925	0.298	14 013	11 228	0.302	13 767	$\boxed{11\,098}$
3369	19 916	13 047	0.100	10 778	6 471	0.102	10 680	$\boxed{6\,238}$

TABLE 6.12: *Comparison of different batching metrics in the U-configuration. Picking time per batching metric is measured in (s). Density is the number of stops per distance measured in the number of locations.*

on average 25.23% saving in total completion time can be achieved by applying the batching metrics in the best performing configuration.

Wave ID	No batch	Random (FIFO)	Stops ratio (R-GR)			Stops non-identical spans (Z-GS)		
			Density d_m	Z	Switch to U	Density d_m	Z	Switch to U
3660	28 804	23 971	0.236	22 031	26 918	0.241	$\boxed{22\,029}$	26 818
3410	29 386	23 889	0.336	22 005	22 820	0.342	$\boxed{21\,910}$	22 752
3141	22 016	17 744	0.187	$\boxed{16\,144}$	21 614	0.169	16 173	21 616
3569	23 703	18 907	0.198	$\boxed{17\,236}$	21 934	0.193	17 276	21 942
3281	44 896	38 243	0.389	34 697	32 541	0.403	35 183	$\boxed{32\,454}$
3735	28 993	24 688	0.611	25 141	25 060	0.639	22 981	$\boxed{22\,977}$
3597	41 635	35 808	0.672	33 872	33 728	0.676	33 450	$\boxed{33\,389}$
3441	32 102	26 749	0.503	24 657	24 364	0.509	24 580	$\boxed{24\,294}$
3709	13 969	12 005	0.257	10 440	14 228	0.276	$\boxed{10\,271}$	13 990
3348	2 117	2 108	0.123	1 485	2 145	0.130	$\boxed{1\,470}$	2 132

TABLE 6.13: *Comparison of different batching metrics in the Z-configuration. Picking time per batching metric is measured in (s). Density is the number of stops per distance measured in the number of locations.*

Additionally, introducing order batching to the picking system increases the pick density measure d_m as distance decreases. Table 6.14 compares 10 sample picking lines. Single orders are picked first, then batches of two are formed through the stops ratio metric and then batches of four are generated using the same approach. Without batching, a pick density measure d_m that exceeds approximately 0.4 produces the shortest picking time in the U-configuration. Forming batches of two orders generates shorter completion times in the U-configuration with a d_m above 0.43. Moreover, a d_m exceeding 0.45 produces the shortest picking times in the U-configuration for batches containing four orders. These results indicate that order batching increases pick density and thereby favours the U-configuration.

The configuration seems to influence the completion time of the picking waves. The pick density may help to determine which configuration to choose. This will be further investigated in the following statistical analysis.

6.6.3 Pick density and configuration selection

A regression analysis was performed to evaluate the total completion time difference between the U- and Z-configuration on the dataset in which both batching metrics were switched to

Wave ID	Orders	No batch			Batch of 2			Batch of 4		
		Density d_m	Time Z	Time U	Density d_m	Time Z	Time U	Density d_m	Time Z	Time U
4759	1 807	0.160	$\boxed{32\ 694}$	38 591	0.209	$\boxed{21\ 147}$	25 274	0.288	$\boxed{14\ 972}$	15 148
4772	1 573	0.275	$\boxed{31\ 485}$	37 011	0.337	$\boxed{22\ 562}$	26 083	0.372	$\boxed{15\ 889}$	16 325
4798	1 815	0.174	$\boxed{30\ 476}$	35 022	0.214	$\boxed{21\ 197}$	24 166	0.278	$\boxed{13\ 328}$	14 066
4905	1 647	0.273	$\boxed{32\ 725}$	37 257	0.388	$\boxed{23\ 312}$	25 694	0.448	16 959	$\boxed{16\ 632}$
4784	1 792	0.391	$\boxed{35\ 908}$	38 205	0.527	28 194	$\boxed{28\ 187}$	0.620	20 215	$\boxed{19\ 062}$
4850	1 573	0.423	36 763	$\boxed{35\ 710}$	0.496	27 750	$\boxed{26\ 046}$	0.505	18 878	$\boxed{16\ 535}$
4774	1 576	0.396	$\boxed{29\ 186}$	32 326	0.427	22 755	$\boxed{22\ 506}$	0.446	13 819	$\boxed{13\ 120}$
4825	1 575	0.485	60 265	$\boxed{49\ 656}$	0.526	45 240	$\boxed{39\ 555}$	0.581	24 772	$\boxed{21\ 114}$
4902	1 655	0.416	35 167	$\boxed{34\ 729}$	0.474	25 603	$\boxed{24\ 828}$	0.460	17 902	$\boxed{15\ 452}$
5010	1 573	0.410	47 774	$\boxed{43\ 415}$	0.510	34 346	$\boxed{31\ 848}$	0.460	21 588	$\boxed{18\ 724}$

TABLE 6.14: *Comparison between no order batching, batches of two and batches of four in both configurations. Time is measured in (s).*

the other configuration. The p-value of the dummy variable for the Z-configuration is significant, suggesting that there is a statistical evidence of a difference in average completion time between configurations. This is displayed in Table 6.15. The average completion time in the U-configuration is estimated to be 28 306 seconds, whereas the average completion time in the Z-configuration is estimated to be 15 596 seconds. An adjusted R^2 of 0.49 indicates that 49% of the total completion time difference can be explained by the configuration. Therefore, if the introduced Z-configuration generates the lowest total completion time, it is on average 12 710 seconds less.

Coefficients	Estimate	Std. error	t-value	p-value
Intercept	28 306	2 420	11.697	7.61×10^{-10}**
Configuration	−12 710	2 892	−4.394	0.00035**
Residual standard error:	5 928 on 18 degrees of freedom			
Adjusted R^2:	0.49			
F-statistic:	19.31			

Note: Two asterisks indicate significance at the 5% level or below.

TABLE 6.15: *Regression model summary on the difference in average total completion time between the configurations.*

To investigate if the pick density measure d_m can be used as an indicator to choose a configuration, the correlation between the pick density measure d_m and the configuration that generates the smallest picking time throughout all order batching metrics is evaluated. It results in a Pearson correlation coefficient [101] of 0.77 showing a strong correlation between d_m and completion time, thus indicating that d_m can be used as an indicator to support the configuration selection.

Figure 6.7 contains the total simulated completion times (for batched orders) plotted against the pick density measure d_m for both configurations. Between a density of 0 and 0.22 the Z-configuration seems to generate the shortest total completion times. Below a density of 0.22 the Z-configuration also outperforms the U-configuration in the average scenario presented in Section 6.4. The average completion time in the U-configuration is 16 145 seconds, while the average completion time in the Z-configuration is 11 736 seconds for the picking lines with a density less than 0.22. There is no clear indication whether to choose the Z- or U-configuration for densities between 0.22 and 0.38. For the U-configuration the average completion time is 19 551 seconds and for the Z-configuration it is 19 110 seconds. At a density of 0.38 the two fitted regression lines cross as depicted in Figure 6.7. Therefore, at a density above 0.38 the U-

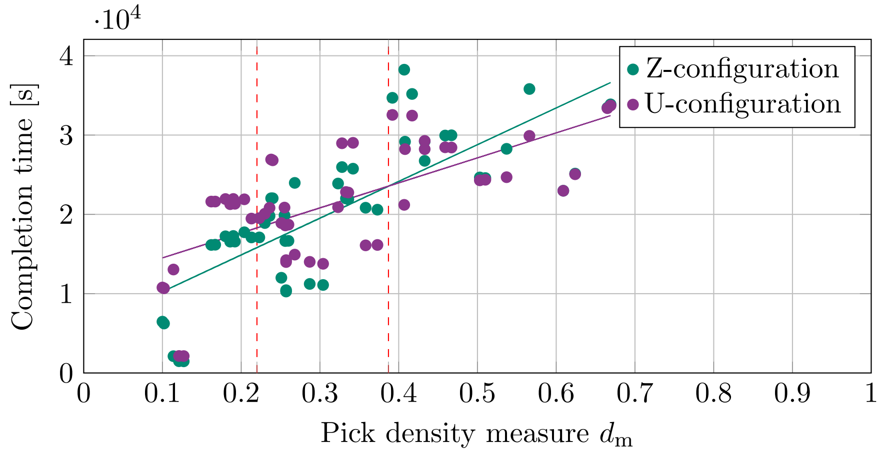


FIGURE 6.7: *The completion times plotted against an increasing pick density measure together with the fitted regression lines for both configurations.*

configuration seems to consistently outperform the Z-configuration. In this interval, the average completion time in the U-configuration is 27 949 seconds, while the average completion time in the Z-configuration is 30 120 seconds.

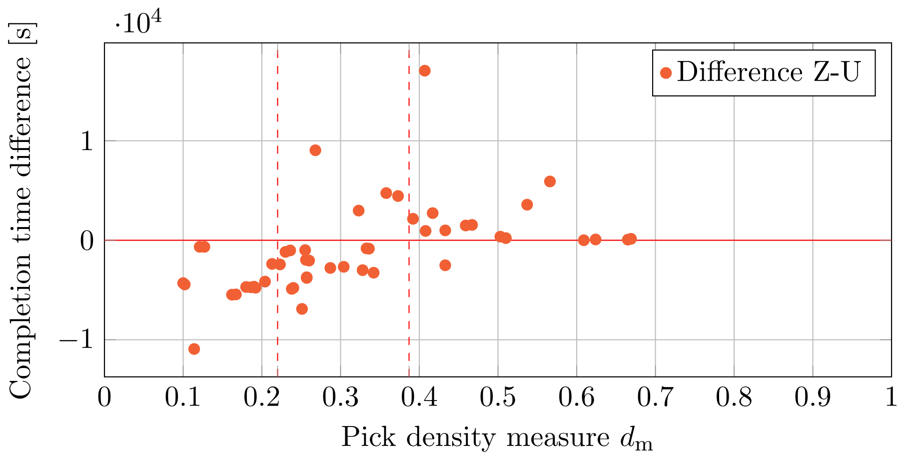


FIGURE 6.8: *The completion time difference between the Z- and U-configuration plotted against an increasing pick density measure. As pick density increases the Z- becomes the U-configuration, thus the difference tends to zero as density increases.*

These observations are confirmed by the difference in completion time depicted in Figure 6.8 in which the total completion time for the U-configuration is subtracted from the Z-configuration. The higher the pick density, the more positive value points are generated as the U-configuration outperforms the Z-configuration.

Regression analyses were performed between the completion time for both the U-configuration and Z-configuration including batching versus the pick density measure d_m. A summary of the results from the regression models is depicted in Tables 6.16 and 6.17. According to the adjusted R^2, 43.81% of the total completion times in the U-configuration can be explained by d_m, while 59.94% of the total completion times in the Z-configuration can be explained by the pick density measure.

Coefficients	Estimate	Std. error	t-value	p-value
Intercept	11 358	1 789	6.347	7.37×10^{-8}**
Pick density d_{m}	31 530	5 036	6.261	1×10^{-7}**
Residual standard error:	5 332 on 48 degrees of freedom			
Adjusted R^2:	0.4381			
F-statistic:	39.2			

Note: Two asterisks indicate significance at the 5% level or below.

TABLE 6.16: *Regression model summary on the density d_{m} and the U-configuration.*

Coefficients	Estimate	Std. error	t-value	p-value
Intercept	5 618	1 910	2.942	0.00501**
Pick density d_{m}	46 338	5 375	8.621	2.58×10^{-11}**
Residual standard error:	5 691 on 48 degrees of freedom			
Adjusted R^2:	0.5994			
F-statistic:	74.33			

Note: Two asterisks indicate significance at the 5% level or below.

TABLE 6.17: *Regression model summary on the density d_{m} and the Z-configuration.*

6.7 Conclusion

All three approaches, namely the worst case, the average case, and the DES, predicted the same ball park in which to switch from a Z- to a U-configuration. In the DES each configuration was combined with adapted batching metrics that have been developed for the unidirectional cyclical picking line to reduce walking distance and thus save picking time. The pick density measure suggested to choose the Z-configuration if d_{m} is lower than 0.22 and to choose the U-configuration if d_{m} is higher than 0.38. In between these densities, the choice of configuration is not clear.

No conveyor belt can lead to more congestion since more workers are crossing the aisle. In the Z-configuration the cartons are dropped off at different points in the layout and collection by external workers may cause additional obstacles to the process flow. It seems that the more the pick density measure increases, the more beneficial the U-configuration becomes. The question then becomes: should both systems be used, or would it be more beneficial to optimise assignments and arrangements on the picking line to favour a particular configuration?

The importance of the pick density measure was emphasised in this chapter. Future work should focus on increasing pick density. The question of which SKUs to assign to which picking lines and how to arrange these SKUs on the picking line to increase pick density should be answered. This might help to choose a configuration before orders (or batches) are sequenced. Order batching seems to increase pick density and thereby shorten the total picking time to complete a picking wave.

6.8 Chapter summary

In this chapter, the structural setup of the picking module in the Retailer's picking system was investigated. Pickers can either walk in the existing U-configuration, that is clockwise around the conveyor belt, or in a Z-configuration, that allows for crossing between the two sides of the picking line, since there is no conveyor belt between the two pick faces.

A worst case model which explores the worst distribution of items for each configuration, an average case simulation with the items of an order randomly spread out in each configuration, and a DES for the existing U- and introduced Z-configuration were investigated. The DES, incorporating real world picking wave samples, showed that the pick density measure d_{m} can be used as an indicator to decide when the Z-configuration will outperform the U-configuration.

The next chapter attempts to optimise the Retailer's order picking system holistically including order batching as an additional layer in the decision making process.

CHAPTER 7

Batching orders in a unidirectional cyclical picking system

Contents

The picking system with unidirectional cyclical picking lines gives rise to four optimisation problems or decision tiers that must be solved. All four decision tiers together constitute a picking wave. Therefore, the implementation of order batching to all four tiers is investigated. When to introduce order batching to the four decision tiers has to be determined, since orders are not finalised yet at the beginning of the optimisation process. The effect of increasing pick density to minimise walking distance and thereby reduce picking time is analysed.

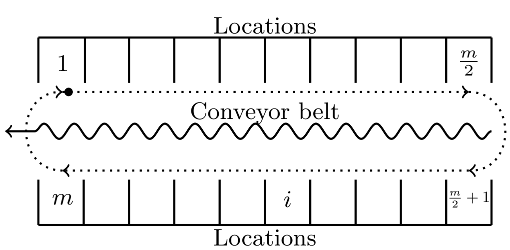
7.1 Problem background

The prominent South African retailer, referred to as the Retailer, that is considered here serves approximately 2 000 stores in Southern Africa on a daily basis. These stores mainly sell apparel to the lower income part of the South African population. Serving such a large number of stores has led to the evolution of a unique picking system. An order picking system transforms bulk supply into customer orders and is thus the main activity in a company's distribution centre (DC). This activity accounts for 50% to 65% of the operational cost of a DC [29, 113].

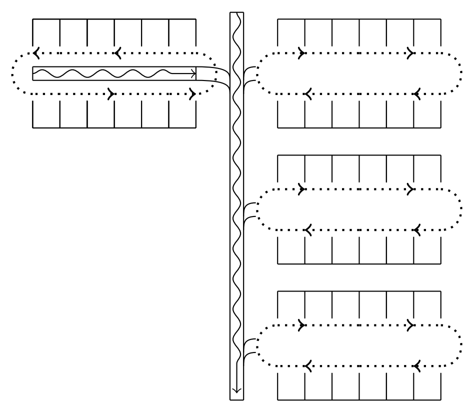
The Retailer's 2 000 stores generate a large number of non-uniform orders to reach a market with varying customer profiles and locations all across the country. The Retailer has introduced a central inventory planning approach to keep operational cost low, while managing this large number of stores. Instead of store managers individually placing orders, planners at the central planning department allocate stock keeping units (SKUs) to stores. The order picking system evolved from this centralised approach.

A central planner releases all the store requirements for a subset of SKUs to be picked in a specific time window (typically a week). The picking process operates in waves. A *picking wave* consists of bringing stock to a picking line, arranging the stock on the picking line, picking the stock on the picking line, and cleaning the picking line of leftover stock (if there is any). A wave of picking thus fulfils all stores requirements for the SKUs present in that wave. Picking instructions are released in distributions (DBNs). DBNs are groups of SKUs of the same product (for example white T-shirts). SKUs within a DBN thus only differ in the sizes of this product (for example small, medium and large). Once the DBNs with their pick instructions for all the stores are released to the DC, all the SKUs in the DBNs are scheduled to be picked in a picking wave [78].

The Retailer's DC uses unidirectional cyclical picking lines that operate independently and in parallel to pick the SKUs in the DBNs during a picking wave. The picking line set up is illustrated in Figure 7.1.



(a) *A schematic representation of a picking line with m locations in a U-configuration. Source: De Villiers [30].*



(b) *A schematic representation of the four picking lines in the Cape Town DC.*

FIGURE 7.1: *A schematic representation of the order picking system in the Retailer's DC. On the left, the detailed schematic representation of a picking line has a conveyor belt in the middle. On the right, the picking system has a picking line with a conveyor belt and three picking lines without conveyor belts.*

Figure 7.1(a) schematically illustrates a unidirectional cyclical picking line with m locations. A picking line normally consists of 64 to 76 locations and has the option of a conveyor belt in the middle. Each SKU in the DBN gets assigned to a unique location on the picking line. Up to five pallets of SKUs can be stored at each location. Therefore, no replenishment of stock is necessary during a wave of picking. Pickers are guided by a voice recognition system (VRS). If the U-configuration is used, the pickers walk in a clockwise direction around the conveyor belt to fulfil the orders. In the Z-configuration, the pickers may also cross over to a picking location at the opposite side of the picking line [58].

If the pickers are viewed as static relative to moving SKUs, the cyclical layout of the unidirectional picking line shows similarities to a unidirectional carousel system from literature. In carousel systems operating pickers remain at a fixed location during the picking process. A typical carousel system is classified as a parts-to-picker system [73]. The unidirectional cyclical picking line, on the other hand, would be classified as a picker-to-parts, single layer, single bin, unidirectional carousel [48].

Besides the structural similarities of a unidirectional cyclical picking line and a unidirectional carousel, the presence of wave picking with all orders known and deterministic before the start of the picking process, is a major difference between the two picking systems. Additionally, a unidirectional cyclical picking line can accommodate multiple pickers, while a unidirectional carousel can only be operated by one picker at a time [78].

Figure 7.1(b) schematically illustrates an order picking system with unidirectional cyclical picking lines. In the example of the Retailer's DC in Cape Town, one picking line with a conveyor belt is placed on the floor, while three additional picking lines run in a picking module. The picking lines that run in the module do not have a conveyor belt in the middle which allows for different configurations of the picking system. A conveyor connects the three floors of the module. This conveyor transports the cartons to other processing areas.

The Retailer's picking system with the unidirectional cyclical picking line layout could be described as a synchronised zone picking system as each picking line runs independently and in parallel to the other [29]. The independently run picking lines allow pickers to join any other picking wave once the current picking wave is completed. Thereby the challenge of work balance in zone picking systems is circumvented [80]. Nevertheless, the rate at which picking lines are populated and picked has to be balanced. This can be achieved by increasing the number of pickers while still keeping congestion to a minimum. Hofmann and Visagie [59] suggested that the pick density determines a good number of pickers in this picking system. In this example, a good number of pickers is determined to be around eight pickers.

This chapter focuses on the Cape Town DC, but findings can be equally applied to the Retailer's DC in Durban and Johannesburg, South Africa. Even though there are structural differences between the DCs, the fundamental layout and logic of the order picking system stays the same.

7.2 Problem description

Four sequential and dependent decision tiers are generated by the process of managing picking waves. During Tier 1 the DBNs (collection of SKUs) are assigned to available picking lines. SKUs are arranged into locations during Tier 2. A picking configuration is selected during Tier 3, while sequencing orders is the focus of the final decision tier, Tier 4. Minimising the total travel distance of pickers to reduce the total completion time per picking wave is the main objective for all four decision tiers [82]. The four decision tiers are presented in more detail next.

Tier 1 in which DBNs are assigned to available picking lines give rise to the SKU to picking line assignment problem (SPLAP). All the SKUs in a DBN should be processed during the same wave of picking. This decision tier must be solved on a daily basis. The number of waves that start each day is dependent on the number of available picking lines on that day. Once the customer requirements for each SKU are released, the associated DBNs can be assigned to a picking wave [82]. Currently, picking line managers assign DBNs based on in-house rules such as minimisation of the number of different store readiness groups (departments per product) and lead time considerations. With the progression of the week the importance of the deadline based rationale increases, making the assignment of older DBNs more urgent.

In Tier 2 SKUs are assigned to locations, in other words, it is arranged on an available picking line. This decision results in the SKU arrangement problem (SAP). The interaction between Tier 1 and 2 is depicted in Figure 7.2, where SKUs are first assigned to picking lines and then arranged on the picking lines to start a new wave of picking [82]. Picking line managers currently arrange SKUs according to personal guidelines such as evenly spreading out high volume SKUs to avoid congestion.

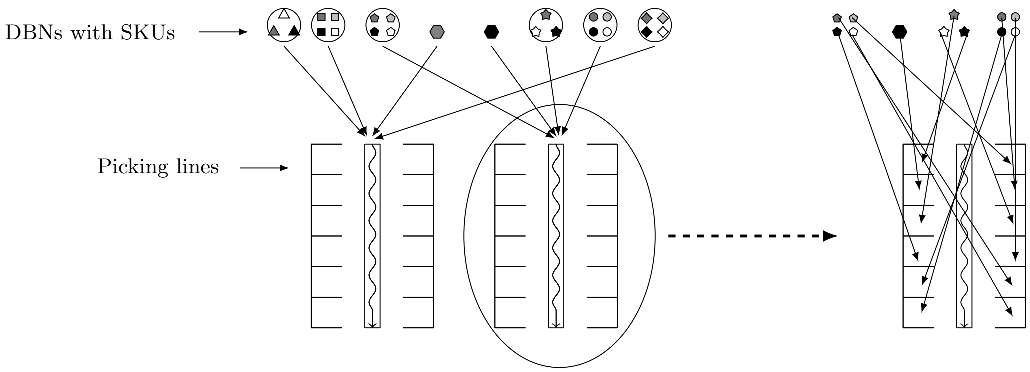
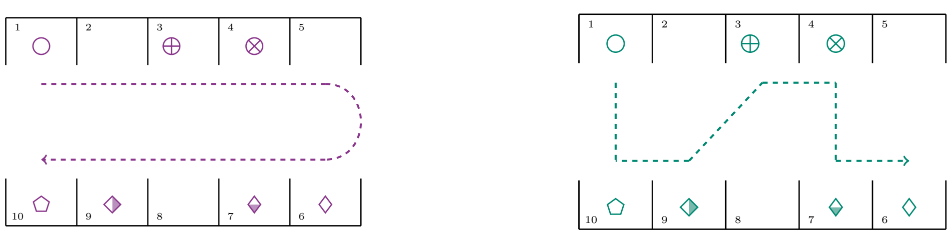


FIGURE 7.2: *A schematic representation of Tiers 1 (left) and 2 (right). A DBN is represented by a shape. Each shaded shape represents a SKU. Different shades are SKUs of the same DBN. Source: Matthews [77].*

Tier 3 entails the selection of the best configuration in which the picking line should operate. This decision tier only exists in picking lines that have no conveyor belt in the middle as shown in Figure 7.3. Tier 3 generates the system configuration problem (SCP), which determines the logic of how pickers walk. In the U-configuration, all pickers move in a clockwise direction around the conveyor belt, as illustrated in Figure 7.3(a). In the Z-configuration, the picker moves in one direction alternating between the two pick faces. The picker only turns around at the end of the picking line to move in the opposite direction. If a required SKU on the same pick face is closer then the next SKU on the opposite pick face, the picker stays on this side. Otherwise, the picker crosses over to the opposite pick face. This scenario is depicted in Figure 7.3(b) [58]. Currently, picking line managers use a rule of thumb to switch from a Z- to a U-configuration: if the picker has to stop at more than a third of the locations the U-configuration is used.

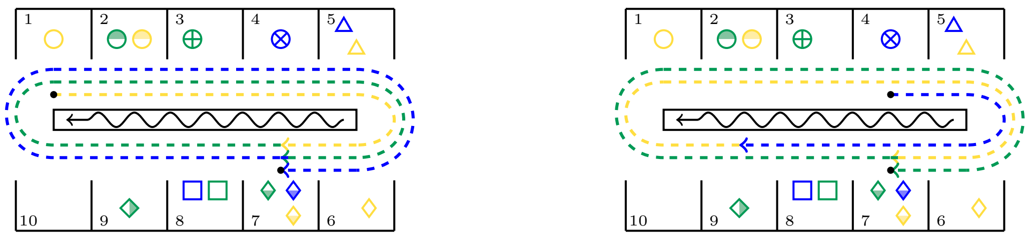


(a) *A schematic representation of a U-configuration in the module.*

(b) *A schematic representation of a Z-configuration in the module.*

FIGURE 7.3: *Comparison of picker walking (dashed lines) in a U- and Z-configuration. The different shapes depict the SKUs that must be picked for an order. Thus a picker must visit all those locations to pick an order.*

Finally, before the wave of picking commences, Tier 4 sequences the orders of that wave. This forms the order sequencing problem (OSP). Orders do not necessarily require all SKUs present in a picking wave. Orders may be assigned to any picker at any location that is available after completing a previous order [82]. The effect of alternative sequences for orders is illustrated in an example in Figure 7.4. In the U-configuration of a unidirectional cyclical picking line the pick path of three orders are depicted by the dashed lines in the colours yellow, green and blue. The different symbols indicate different SKUs. In Figure 7.4(a), the yellow order is picked first, followed by green and blue, resulting in the picker passing 27 locations. In Figure 7.4(b), the picking starts with the blue order, followed by yellow and green. This order sequencing leads to a pick path length of only 24 locations. In this small example, the choice of order sequence results in a difference in walking distance of three locations.



(a) *Order sequence option* 1. (b) *Order sequence option* 2.

FIGURE 7.4: *A schematic representation of Tier* 4*. Depending on the sequencing of orders a picker either has to pass* 27 *locations in Option* 1 *or* 24 *locations in Option* 2*.*

Between all four decision tiers there is a flow of information. This information flow is shown in Figure 7.5. It is critical to analyse the reciprocal paths of information flow between all decision tiers simultaneously because all tiers are interconnected. For example, the DBNs that are assigned to a picking wave define the SKUs that have to be arranged on the picking line in Tier 2. The SKU locations determine the configuration in the third tier and the sequence of orders in the fourth. It is essential to understand the interactions between the four decision tiers to optimise the picking system holistically. Therefore, each decision tier must be investigated carefully [82].

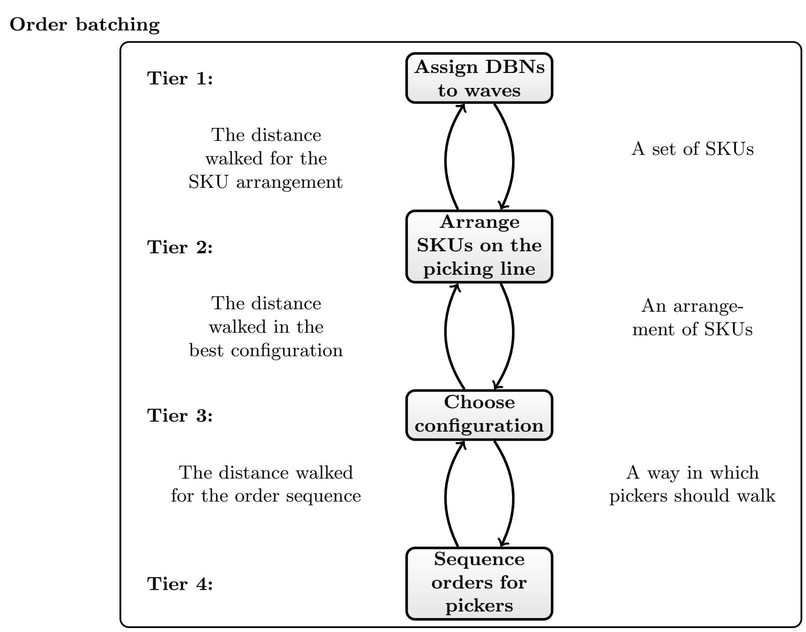
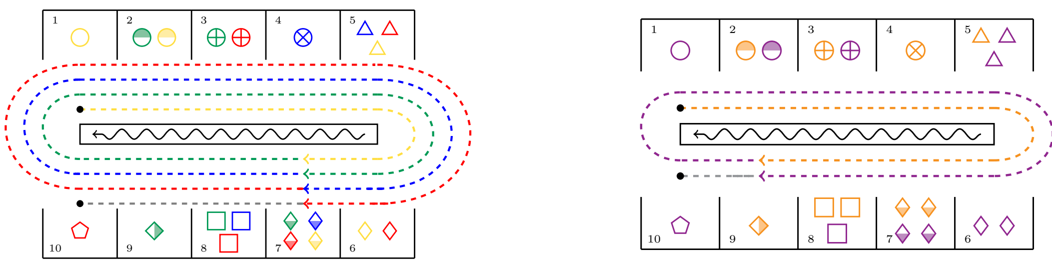


FIGURE 7.5: *Schematic representation of the information flow between the four decision tiers.*

The introduction of order batching to the DC's picking system influences the walking distance of a picker. In manual order picking the two main concepts are single order picking or batch picking. While in single order picking one order is picked by one picker, in batch picking multiple orders are picked simultaneously by one picker. Thereby order batching can effectively reduce walking distance. The reduction in walking distance may result in overall picking time savings, which can be translated into a reduction of labour cost [28].

In Figure 7.6 the impact of order batching on walking distance in the unidirectional cyclical picking line is illustrated in a small example. Matthews and Visagie [78] introduced the measurement of *cycles traversed* to measure the walking distance in the unidirectional cyclical picking line. Therefore, one cycle traversed is added to the count every time a picker completes a full cycle during the picking process. In single order picking, as illustrated in Figure 7.6(a), the four orders indicated by the colours yellow, red, green, and blue, are picked starting with yellow, then green, then blue and finally red. If the number of cycles traversed are counted, it results in four cycles. In batch order picking, if the orders yellow and red form the orange batch, and green and blue the purple batch as depicted in Figure 7.6(b), a picker only has to traverse two cycles. This example illustrates how batching two orders can reduce walking distance by as much as 50%.



(a) *A schematic representation of single order picking.*

(b) *A schematic representation of batch picking.*

FIGURE 7.6: *Comparison of the walking distance in single order picking and batch picking.*

Hofmann and Visagie [55, 56] introduced order batching to the layout of a unidirectional cyclical picking line by developing distance approximation metrics and combination heuristics. Furthermore, Hofmann and Visagie [59] showed that the reduction in walking distance influences picker congestion, but still leads to an overall reduction in picking time. Therefore, order batching may be included in the optimisation of the four decision tiers in the order picking system of the Retailer.

The main objective of this chapter is to reduce walking distance to shorten the total completion time of a picking wave. Order batching will be included in the optimisation of the picking system, since it has shown to efficiently reduce walking distance and picking time [55, 56, 59]. It can be seen as another layer to the optimisation problem as depicted in Figure 7.5, since it potentially may interact with all four decision tiers. The decision tiers have a different impact on order batching, because orders are not formed before Tier 1 and locations are not assigned before Tier 2. Tier 3 is a novel inclusion in this optimisation process.

This chapter aims to achieve two objectives. The first objective aims at identifying solution approaches in each decision tier that support the introduction of order batching. The second objective is to determine when to introduce order batching to the decision making process to minimise the total walking distance of all pickers to pick all orders in a picking wave.

After the brief background of the real world application and an introduction to the decision problem, a literature review on order picking optimisation in the unidirectional cyclical picking line is presented in Section 7.3. The four decision tiers are then described and optimisation approaches are presented in Section 7.4. In Section 7.5 different solution approaches are tested on real life historical data of the Retailer's DC, spanning an entire month. The chapter is concluded in Section 7.6 with a summary and an outlook on future research.

7.3 Literature

For the order picking system with a unidirectional cyclical picking line, three decision tiers have been studied [78, 80, 81, 82]. The picking line module introduced a fourth decision tier, namely deciding on a configuration of the picking system [58]. Therefore, a brief overview on the published research within each of the four decision tiers (in this layout as compared to other layouts) will be presented next, starting with Tier 1 and ending with Tier 4.

The SKU to picking line assignment problem (SPALP) considers the assignment of DBNs, a collection of SKUs, to picking waves. The picking waves may be viewed as zones as each picking line functions independently, but non-zoned picking system approaches may also be used and modified [82].

For the assignment of SKUs to zones, Matthews and Visagie [80] developed a decision framework based on the method of Kim and Smith [65] that assigns SKUs of the same order to locations that are near to one another. Therefore, they introduced a greedy insertion procedure that assigns DBNs to picking waves in a greedy manner and thus reduces walking distance by minimising the sum of the sizes of the maximal SKU for each picking line. The *maximal SKU* is the SKU with the largest number of stores requiring it. It serves as a lower bound for the number of cycles that has to be traversed during the picking process [78]. However, these zones consisted of a single aisle occupied by a single picker, and thus did not take into account a cyclical layout or the presence of multiple pickers [80].

For the assignment of SKUs in non-zoned picking systems, Manzini [75] introduced a correlation measure that is calculated by the number of orders requiring two particular SKUs to minimise total picking time. Bindi *et al.* [11] developed a similarity measure based on SKU correlations, a stock turn coefficient, and a Jaccard statistic which improved DC throughput. Accorsi *et al.* [2] considered storage allocation (inventory level per SKU) and storage assignment (location of SKU) simultaneously. They proposed a top-down hierarchical procedure for overall picking optimisation. Therefore, different approaches such as allocation rules, correlation measures and clustering algorithms were combined and tested. Based on these studies, Matthews and Visagie [81] developed four desirability scores to combine with the greedy insertion procedure to not only address the problem of minimising walking distance, but also to reduce the number of small cartons and to improve the volume distribution. Their desirability score, which considers the number of stores required by the candidate DBN including at least one DBN that was already assigned to a picking line, was recommended.

In addition to the zone and non-zoned assignment approaches, store requirements may also be used to assign DBNs to picking waves. Chen and Wu [21] adopted association rules to identify order connections that reflect the demand relationships of customers to form batches. According to Chen *et al.* [20] the association rules apply the apriori algorithm to extract large ordersets. These groups of orders, that are frequently requested together, are indicated by these demand patterns. No approximation metrics for travel distance have to be developed as demand patterns can be directly extracted from the order database. Their test instances use a rectangular warehouse with parallel picking aisles applying a S-shape picking strategy, but as no distance approximations are necessary this does not influence the batching algorithm. Chiang *et al.* [22] applied a data mining approach to generate similarity measures between SKUs to use in a storage assignment model. They developed an association index between each available storage location and new SKU and assigned these SKUs. They then assigned SKUs to their locations using a general assignment model that minimises the number of aisles.

In the SKU arrangement problem (SAP) Matthews and Visagie [82] identified solution approaches to minimise walking distance within a picking wave by rearranging SKUs. Their exact mixed integer linear programming formulation was not able to solve real life data instances, thus they modified heuristic approaches from literature to meet the requirements of this layout. In a bidirectional carousel the organ pipe arrangement proposed by Vickson and Fujimoto [119] minimises the long run average travel time that is spent on picking a sequence of single independent and identically distributed orders. In a unidirectional carousel it is optimal to assign SKUs to locations in a greedy sequential manner [118]. Both cases were applied to carousels with an infinite set of stochastic orders and adapted to process a set of deterministic orders [82]. Litvak and Vlasiou [73] proposed a SKU allocation approach developed by Stern [105] which places SKUs close to each other if there is a high probability of them appearing in the same order. This heuristic was adapted by considering the number of orders requiring two SKUs as the number of adjacencies between them according to a maximal adjacency principle [82]. Analysing the effect of SKU arrangements on congestion Hagspihl and Visagie [46] introduced a classroom discipline heuristic. This heuristic uses a set that is ordered based on pick frequency inserting SKUs to minimise congestion. The effects of the classroom discipline approach on minimising walking distance was tested [82].

Matthews and Visagie [82] showed that the order sequencing (Tier 4) outweighs SKU assignment (Tier 2). In other words, walking distance can be reduced much more by proper order sequencing than by arranging SKUs in a particular way. By implementing the last decision tier, walking distance can be reduced by as much as 15% as opposed to a 1% improvement for Tier 2. None of the modified algorithms produced an optimal solution in the unidirectional cyclical layout with a set of deterministic orders. Therefore, they recommended using any of the easily implementable heuristic solution approaches.

Kress *et al.* [67] investigated the storage assignment in carousel racks as a SKU partitioning problem to minimise the number of groups that have to be accessed when retrieving an order. They were able to minimise the average picker idle times (when the carousel rotates) with their branch-and-bound algorithm which performed better when compared to a random and an organ pipe arrangement. However, they did not include the order sequence in the assumptions of their model, which according to Matthews and Visagie [82] outweighs the gain of the SKU arrangement optimisation.

The choice of configuration (SCP) of a unidirectional cyclical picking line was investigated by Hofmann and Visagie [58]. They suggest the *pick density measure* d_{m} that divides all stops of a picker by a distance approximation (number of locations $\times$ maximal SKU) as an indicator of when to switch from a Z- to a U-configuration.

Matthews and Visagie [78] investigated the order sequencing problem (OSP) for a single picking wave with a fixed SKU arrangement. They assumed that the time to pick stock and time to handle boxes does not influence the sequence of orders and thus minimised walking distance to reduce overall completion time. Even though they introduced a maximal cut approach, they recommended an easily implementable nearest end heuristic that yields comparable walking distances. Based on the nearest order and nearest item heuristic by Bartholdi and Platzman [8], it selects the order with the nearest ending location, given a starting point, to be sequenced next.

A batch can be viewed as a single but bigger order. Order batching as an additional layer to the decision process can be handled by adopting the solution approaches of Matthews and Visagie [78, 80, 81, 82], Chen *et al.* [20], Chen and Wu [21], and Hofmann and Visagie [58]. All four decision tiers may provide support for order batching and the picking system could be optimised holistically.

7.4 Incorporating order batching in the four decision tiers

The four decision tiers of the picking system are analysed in the following section. Ways to incorporate order batching in each decision tier (even if no orders are defined yet, since they are defined by decisions made in the first tier) are suggested.

7.4.1 DBN assignment to picking waves

Tier 1 assigns the already released DBNs into specific picking waves for a given day while minimising the total walking distance [80]. The problem can be described as a generalised assignment problem (GAP) in which DBNs are assigned to picking waves. A wave of picking (on an available picking line) may also be viewed as a knapsack which has to be filled with a set of DBNs. Each DBN has to be assigned to a picking line with a limited capacity (or locations). The DBN's location requirements can be interpreted as an item's weight. With the introduction of the maximal SKU as a lower bound for the number of cycles traversed in the picking line, the objective of this decision tier can be restated as minimising the maximal SKU for a picking line [71].

No orders are yet formed and thus no order based batching can be introduced at this stage of the order picking optimisation process. However, selecting DBNs for a high pick density reduces walking distance and thereby benefits order batching [58, 59]. Tier 1 aims at increasing pick density, which indirectly supports better order batching.

According to Matthews and Visagie [80] an integer programming formulation of the GAP was not solvable within 10 minutes, even when they introduced a relaxation of the formulation that adjusted the size of all maximal SKUs. Toth and Martello [111] developed an insertion heuristic for the GAP. This approach ranks all unassigned items with regards to its best and second best possible knapsack assignment in decreasing order. The greedy insertion approach (GI) as described in Algorithm 11 was designed on this basis, ensuring that all DBNs get assigned [80]. If DBNs with a large number of stores requiring it are added to a picking line, the pick density of this picking line may increase.

Algorithm 11: `Greedy insertion heuristic (GI)`

Input: A set of picking lines $\mathcal{L}$ ordered by the number of floor and rack locations available for the picking line l. A set of DBNs $\mathcal{D}$.
Output: An assignment of DBNs to picking lines.

1: **while** Unassigned DBN which fits into the remaining l **do**
2: Select the DBN with the largest maximal SKU which fits into picking line l.
3: Assign DBN to the picking line l.
4: **end while**

Correlation approaches between adjacent SKUs can also be used to solve Tier 1 [77]. This approach improves the speed of the order picking operation and balances the carton flow. The desirability measure (DM) considers how many stores require a particular DBN. The candidate DBN d must form part of a set $\mathcal{D}_l$ and this set must already be assigned to a picking line l. This approach showed the best results in terms of minimising walking distance. All assigned DBNs are merged and considered as a single DBN and the intersection (or correlation) of this new DBN and the set of stores requiring the candidate DBN is calculated. According to Matthews [77] the desirability score $\mathcal{A}(\mathcal{D}_l, d)$, with $\mathcal{G}$ representing the set of stores, can be calculated by

$$\mathcal{A}(\mathcal{D}_l, d) = |\mathcal{G}_l \cap \mathcal{G}_d|. \tag{7.1}$$

The desirability score is inserted into the greedy insertion algorithm as displayed in Algorithm 12 as the greedy insertion heuristic using a desirability measure (GIDM). The pick density may increase, if the same set of stores require a DBN that is already assigned to the picking line.

Algorithm 12: `Greedy insertion heuristic using a desirability measure (GIDM)`

Input: A set of picking lines $\mathcal{L}$ ordered by the number of floor and rack locations available for the picking line l. A set of DBNs $\mathcal{D}$ and a set of pre-assigned $\mathcal{D}_l$ associated with each picking line.
Output: An assignment of DBNs to picking lines.

while Unassigned DBN which fits into the remaining l **do**
  Select DBN with largest desirability score $\mathcal{A}(D_l, d)$ which fits into picking line l.
  Assign DBN to picking line l.
end while

A pattern mining approach to form batches out of orders was first introduced by Chen and Wu [21] and further extended by Chen *et al.* [20]. However, there are no orders at this stage of the decision process as DBNs are not assigned to picking waves yet. Nevertheless, SKUs are already linked with store requirements. If SKUs, that are required by the same stores, are assigned to the same picking line, the pick density may increase. The apriori algorithm as developed by Agrawal *et al.* [3] is modified to uncover relationships between data and also used to form picking waves out of similar DBNs. In the first stage, associations between DBNs in terms of *support*, *confidence* and *lift* are recognised. In this case, support shows how often DBN 1 and 2 are requested together by all stores. Confidence describes the likelihood that if a store requests DBN 1 it will also request DBN 2. Lift is a simple correlation measure that defines the degree to which the request of DBN 1 increases the request of DBN 2 [47]. In the second stage, a clustering procedure that maximises the sum of the support values is employed [20].

The pattern mining approach (PMA) assigns DBNs to picking waves as depicted in Algorithm 13. The support value is lowered dynamically until no significant association rules can be generated [47]. For cases where DBNs are still unassigned, the algorithm fills up available locations in a greedy manner.

Algorithm 13: `Pattern mining approach (PMA)`

Input: A set of picking lines $\mathcal{L}$. A set of DBNs $\mathcal{D}$ with store requirements per SKU.
Output: An assignment of DBNs to picking lines.

while Unassigned DBN which fits into the remaining l **do**
  while Association rules generated by apriori algorithm not empty **do**
    Select DBNs with highest support values which fit into picking line l.
    Assign DBNs to picking line l.
    if Unassigned DBNs **then**
      Lower support value and update association rules.
    end if
  end while
  Assign DBNs to remaining l in a greedy manner.
end while

Besides the GI, the GIDM, and the PMA that aim at increasing pick density, a greedy random assignment approach (GRA) will be used as a benchmark. This comparison tries to determine whether an increase in pick density in Tier 1 supports order batching. The four algorithms will be compared to historical assignments (HA).

7.4.2 SKU arrangement on the picking line

Tier 2 considers the arrangement of a set of SKUs in a single picking line while minimising the walking distance of pickers in the process. Tier 2 has a small impact on the overall solution, and the first and last decision tier outweighs this impact according to Matthews and Visagie [82]. Therefore, they recommended quick solution approaches to make implementation times of the global problem feasible. Orders do not have any location information yet. Therefore, order-to-route closeness metrics cannot be applied to introduce order batching. The objective at this stage thus still remains to increase pick density.

A greedy random and a greedy sequential heuristic are introduced to analyse the effect of SKU arrangement on batch forming. In the greedy random approach the SKUs are arranged randomly on the available locations along the picking line. In the greedy sequential arrangement approach the set of SKUs is ordered according to the maximal SKU (pick frequency) and is assigned to available locations as illustrated in Algorithm 14.

Algorithm 14: `Greedy sequential arrangement (GSL)`

Input: A set of SKUs $\mathcal{U}$ in descending order by maximal SKU. A number of floor and rack locations m.
Output: An assignment of SKUs to locations on the picking line.

1: **while** Unassigned SKU which fits into the remaining locations m **do**
2: Assign SKUs to available locations m according to the sequence in $\mathcal{U}$.
3: **end while**

The greedy random arrangement (GRL) serves as a benchmark. The greedy sequential arrangement (GSL) increases pick density by assigning favoured SKUs (or SKUs on high demand) close to each other. The latter may support order batching.

7.4.3 Configuration selection

The objective during Tier 3 is to select a configuration for the picking wave that was determined by the first two decision tiers. Orders can only be formed after Tier 1 is complete and locations are arranged during Tier 2. Picking location based order batching metrics are not influenced by the configuration choice. However, the configuration does influence the order batching metric that takes the picker path into consideration. A configuration should be chosen before order batching is introduced to allow for different order-to-route closeness metrics.

Hofmann and Visagie [58] suggest the pick density measure d_{m} as an indicator for when to switch from a Z- to a U- configuration. The density measure includes the total number of stops s of a picker to pick all orders, the number of locations m available in the system, and the maximal SKU v as an approximation of the number of cycles needed to pick all orders [59]. The pick density measure can only take on values between 0 and 1 and is calculated as

$$d_{\mathrm{m}} = \frac{s}{m \cdot v}. \tag{7.2}$$

According to Hofmann and Visagie [58] a high pick density measure d_{m} generates shorter overall picking times in a U-configuration. Therefore, if pick densities have been increased in Tier 1 and 2 the implementation of a U-configuration becomes more likely. The choice of configuration then influences the order batching metrics directly.

7.4.4 Order sequencing including order batching

The orders (or in this case batches of orders) are sequenced in Tier 4. Before the sequencing, batches are formed according to the batching metrics developed by Hofmann and Visagie [55, 56], since all information necessary to include order batching is now available.

The picking location metric with the highest reduction in walking distance suggested by Hofmann and Visagie [55] is the stops ratio metric combined with a greedy random heuristic (RGR). The metric divides the number of stops n_{ij} that are not shared between orders i and j by the number of stops t_{ij} that are similar between orders i and j. The matrix $\mathbf{R}$ with elements r_{ij} can be calculated as

$$r_{ij} = \frac{n_{ij}}{t_{ij}}, \text{ for all orders } i \text{ and } j, \text{ with } i \neq j. \tag{7.3}$$

The greedy random heuristic (GR) searches the rows of any matrix $\mathbf{M}$ with elements m_{ij} in a random sequence. This is displayed in Algorithm 15. The algorithm randomly selects a row (say k) and then finds the minimum element (say m_{kq}) in row k and adds it to set $\mathcal{B}$ to indicate that orders k and q are batched. Both rows and columns corresponding to k and q are then removed from $\mathbf{M}$. These steps are repeated until a list of batched orders is generated.

Algorithm 15: `Greedy random heuristic (GR)`

Input: A picking location metric consisting of a $n \times n$ matrix $\mathbf{M}$ with entries m_{ij}, an empty solution set $\mathcal{B}$.
Output: The solution set $\mathcal{B}$ as a list of batched orders.

$\mathcal{B} \leftarrow \emptyset$
while $\mid \mathcal{B} \mid < n/2$ **do**
  $k \leftarrow$ random row from $\mathbf{M}$
  $m_{kq} = \min_j [m_{kj}]$
  $\mathcal{B} \leftarrow \mathcal{B} \cup (k, q)$
  Remove both rows and columns k and q from $\mathbf{M}$
end while
Return $\mathcal{B}$

The *span* (or path in the Z-configuration) of an order is the distance that has to be covered to collect all items of an order, given a starting location [78]. The stops non-identical spans metric that is combined with a greedy smallest entry heuristic (ZGS) includes this layout specific measurement. The ZGS route overlap metric is discussed in greater detail by Hofmann and Visagie [56]. The sets $\mathcal{S}_i$ and $\mathcal{S}_j$ contain all stops for orders i and j respectively. The notation $\widetilde{P}_i^{\min}$ and $\widetilde{P}_j^{\min}$ represent the biggest gap between the minimum spans of order i and j, and $|P_i^{\min} \Cap P_j^{\min}|$ indicates the length of the overlap in number of locations between $P_i^{\min}$ and $P_j^{\min}$. The stops non-identical spans metric can be calculated by

$$z_{ij} = (|\mathcal{S}_i| + |\mathcal{S}_j| - |\mathcal{S}_i \cap \mathcal{S}_j|) + (|\widetilde{P}_i^{\min}| + |\widetilde{P}_j^{\min}| - |\widetilde{P}_i^{\min} \Cap \widetilde{P}_j^{\min}|), \text{ with } i \neq j, \tag{7.4}$$

to form matrix $\mathbf{Z}$. The greedy smallest entry heuristic (GS) searches globally for the smallest entry through all elements m_{ij} of any matrix $\mathbf{M}$. This is shown in Algorithm 16. The entry that has to be batched next is the global smallest entry. The global smallest entry is the next smallest entry with the biggest difference to the next second smallest entry in matrix $\mathbf{M}$.

After batches are formed according to the these batching metrics, they have to be sequenced to complete the fourth or final tier. Matthews and Visagie [78] developed a nearest end heuristic

Algorithm 16: `Greedy smallest entry heuristic (GS)`

Input: A batching metric comprising a $n \times n$ matrix $\mathbf{M}$ with entries m_{ij}, an empty solution set $\mathcal{B}$.
Output: The solution set $\mathcal{B}$ as a list of batched orders.

```
1: B ← ∅
2: while | B | < n/2 do
3:     m_kq = min_{i,j}[m_ij]
4:     B ← B ∪ (k, q)
5:     Remove the rows and columns corresponding to k and q from M
6: end while
7: Return B
```

(NE) that is easy to implement. As displayed in Algorithm 17 the batch with the nearest ending location, given the current position, is added to the sequence. This NE algorithm will sequence batches in U- and Z-configurations.

Algorithm 17: `Nearest end heuristic (NE)`

Input: A set of batches and SKU locations.
Output: A sequence of batches.

```
1: Set the current position to the first location
2: while The set of batches is not empty do
3:     Search for the batch with the closest ending location, if picking starts from the current position
4:     Add this batch to the sequence
5:     Delete from the set of batches
6:     Update the current position to the ending positing of the recently added batch
7: end while
8: Return the sequence of batches
```

7.5 Results

In Tier 1, order batching has an indirect impact on the stops and minimum spans. Depending on which DBNs are assigned to which picking line, orders are formed that are later combined into batches. The rearrangement of SKUs in Tier 2 has an effect on the minimum span or path, but it does not affect the number of stops of an order [82]. The minimum span is dependent on the arrangement of the locations of SKUs. Additionally, the configuration chosen in Tier 3 influences the stops non-identical spans metric, but not the stops ratio metric as all stops remain the same. At Tier 4, orders have been finalised and SKUs have been arranged. This then allows the approximation metrics to measure distances and combine orders into batches. After forming the batches, they get sequenced for picking. A global performance of the picking system can thus only be evaluated by first assigning DBNs to picking waves (Tier 1), then arranging the SKUs on the picking line (Tier 2), then choosing the configuration of the picking wave (Tier 3), then batching the newly composed orders, and finally sequencing the batches (Tier 4).

7.5.1 Data and scenarios

Historical data from the Retailer's DC in Cape Town was used to test the solution approaches. The data consist of 45 picking waves over 27 work days with the number of picking lines working in parallel ranging from two to four. There are 1 206 unique DBNs containing 2 212 unique SKUs. The master data for the numerical experiment was reorganised into four scenarios due to the different numbers of scheduled lines per day. Thereby algorithms can be compared more comprehensively and picking waves that are not completed during a single working day can be investigated. The data set is cleaned of outliers such as picking waves that contained less than 15 SKUs for example. Each scenario contains a uniform set of instances. In other words, the data for each scenario were taken from similar picking lines, workdays, and DBNs. The properties of these scenarios are illustrated in Table 7.1.

Scenario	Number of lines	Number of days	Number of DBNs	Number of waves
1	2	27	767	30
2	3	21	708	24
3	4	9	330	12

TABLE 7.1: *The composition of three test scenarios from historical data of the Cape Town DC.*

All numerical experiments were implemented in Python 3.6 [94] utilising the C-based libraries Numpy [89] and Pandas [93] and performed on a Dell Optiplex 5050 with a Intel Core i7-7700 CPU at 3.6 GHz, 1x8GB 2400MHz DDR4 RAM, a 2.5" 256GB SSD class 20 drive, and the Microsoft Windows 10 Enterprise 2016 LTSB operating system [84]. The statistical analysis of the experimental results was carried out in R [95].

In all three scenarios, the main objective is to minimise the total completion time of a picking wave. The influence of including order batching is analysed in every decision tier of the optimisation process. All combinations of solution approaches for each decision tier are illustrated in Figure 7.7.

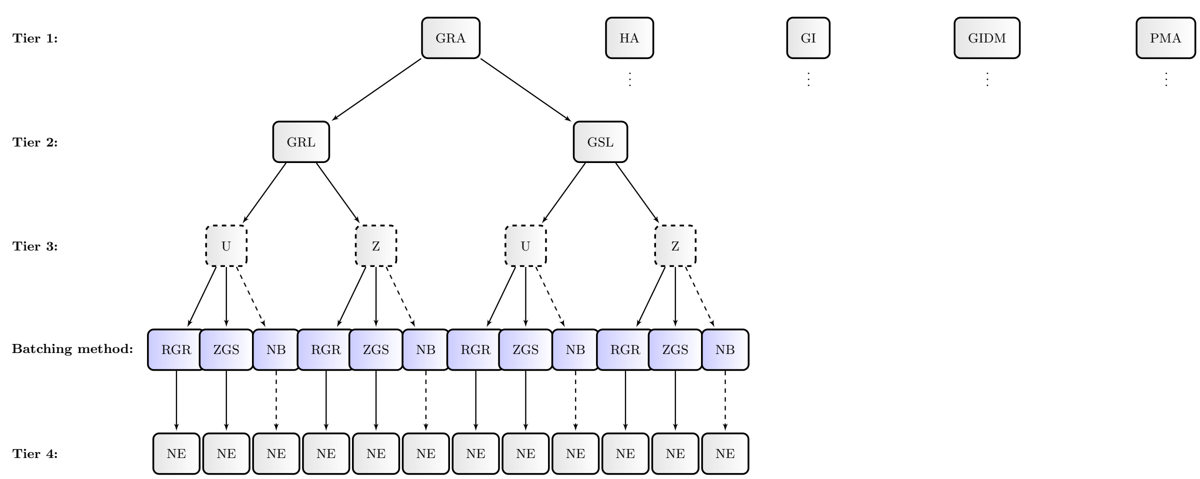


FIGURE 7.7: *Decision tree for different optimisation combinations including order batching. The newly introduced Tier 3 is pointed out by a dashed line, while the additional layer of order batching is depicted in blue.*

In Tier 1 DBNs are assigned to SKUs according to a greedy random assignment approach (GRA), the historical assignment (HA), the greedy insertion heuristic (GI), the greedy insertion heuristic using a desirability measure (GIDM), and a pattern mining approach (PMA). In Tier 2 all SKUs contained in the DBNs that have been assigned to a picking wave are arranged on a picking line. The greedy random heuristic (GRL) and the greedy sequential heuristic (GSL) are applied to arrange SKUs on the locations. The historical arrangement of SKUs on picking lines is not included as it cannot be applied to DBN assignments that differ from the historical data. In Tier 3 orders are formed and information about the SKU locations is available. Depending on the pick density, a configuration can be chosen. Batches can then be formed applying the stops ratio metric with the greedy random heuristic (RGR) and the stops non-identical spans metric with the greedy smallest entry heuristic (ZGS). Additionally, no batching is included (NB) for comparison. These order combinations are sequenced with the nearest end heuristic (NE) in Tier 4. No further sequencing approaches are tested, since order batching happens before the sequencing. A discrete event simulation (DES), as developed by Hofmann and Visagie [58, 59], determines the total completion time per wave. The completion times can then be compared for the different solution approaches. The optimisation approach of using GRA for the SPALP, GRL for the SAP, a U-configuration of the picking line, and RGR for the batching of orders would for example results in the abbreviation GRA-GRL-U-RGR-NE. Thereby, any path that can be followed through the layers of Figure 7.7 defines a solution approach and can be described through an abbreviation as explained.

Figure 7.8 contains the sum of the completion times measured in seconds for the 30 sample picking waves in Scenario 1, if every solution approach (meaning every possible algorithm combination) is tested. The configuration choice (Tier 3) has the biggest influence on the total completion time (on average a difference of about 1 hour per picking wave). The ZGS batching metric produces the shortest time in the U-configuration. In the Z-configuration the RGR seems to generate slightly lower completion times. The arrangement of SKUs (Tier 2) does not have a significant impact since it influences the total completion time by less than 1%. This is in line with the findings of Matthews and Visagie [82]. The assignment using the GI seems to generate the lowest total completion time. Nevertheless, GIDM and PMA also produce completion times that are lower than GRA and HA and even outperform GRL on some lines.

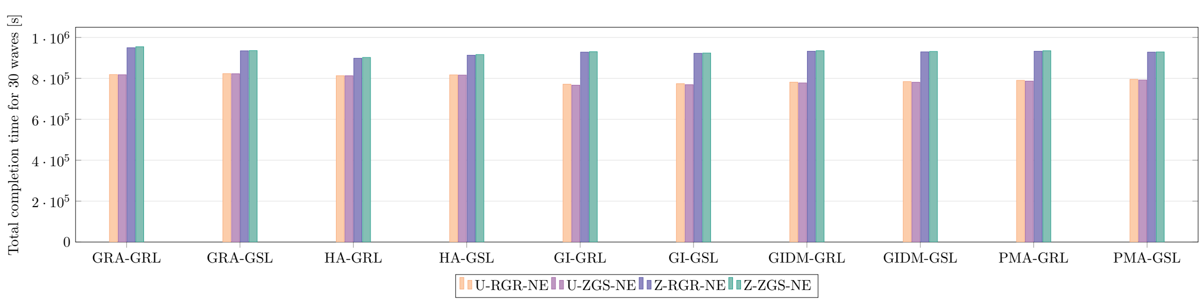
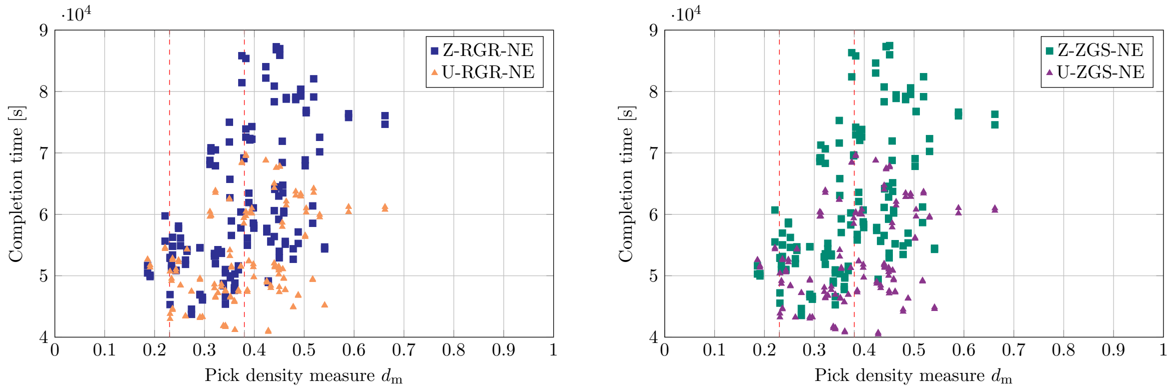


FIGURE 7.8: *The total completion time for Scenario* 1 *with all potential optimisation approaches. The x-axis label describes the first two choices (Tier 1 and 2), while the key describes the last three choices (Tier 3, batching method, and Tier 4). Together they form one experimental set of algorithms. The y-axis label indicates the time it takes to complete all* 30 *waves for each combination.*

The average pick density can already be calculated after Tier 1, since the specific location of SKUs does not have an influence. However, it is only calculated after Tier 3, since its influence on order batching is investigated. In Figure 7.9 the corresponding pick density measure d_m to the total completion time of each picking wave is illustrated. In the experimental set up of Hofmann and Visagie [58] the Z-configuration outperformed the U-configuration consistently at a d_m below 0.22. The U-configuration consistently returned the shortest total completion times for a d_m larger than 0.38. Therefore, the pick density measure 0.22 and 0.38 are depicted in Figure 7.9 by red dashed lines. The GRA and HA assignment approaches do not focus on increasing the pick density to support order batching. Therefore, only a few picking lines that applied the GRA and HA assignment approach generated a pick density measure that is above 0.22. The average pick densities for GI, GIDM and PMA are all above a pick density measure of 0.38. The findings of Hofmann and Visagie [58] are supported by the results of this chapter, since all assignments that have a pick density measure above 0.38 generate lower total completion times in the U-configuration for the 30 sample picking lines of Scenario 1. This is illustrated in Figure 7.9(a) for batching metric RGR and in Figure 7.9(b) for batching metric ZGS.



(a) *Time and density with RGR for the Z- and U-configuration.* (b) *Time and density with ZGS for the Z- and U-configuration.*

FIGURE 7.9: *The completion times plotted against an increasing pick density measure for both configurations in Scenario* 1 *with batching metrics RGR and ZGS.*

In the second scenario, the sum of the completion times for all solution approaches in the 24 sample picking waves is illustrated in Figure 7.10. Tier 3 seems to have an even bigger influence in the difference between picking waves – about 1.5 hours. The lowest total completion time for all samples is generated with the ZGS metric in the U-configuration. The impact of Tier 2 remains lower than 1%. GI followed by GIDM generates the lowest completion times per picking wave. PMA produces slightly longer completion times because it only assigns scheduled SKUs. The central planning department would have to schedule more similar SKUs to increase the efficiency of the PMA.

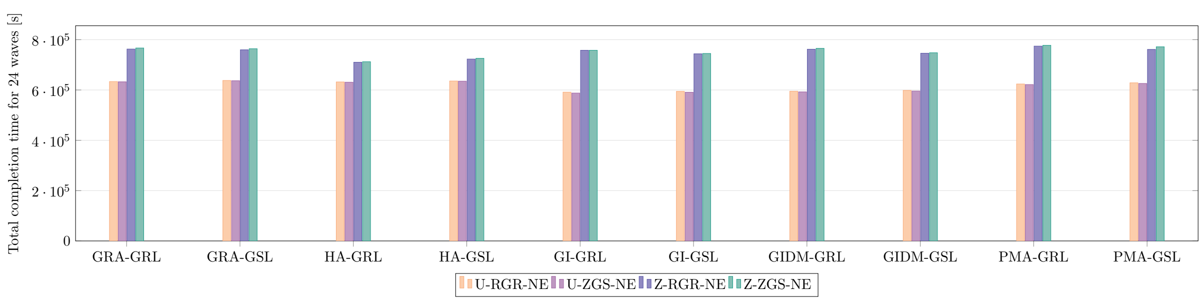
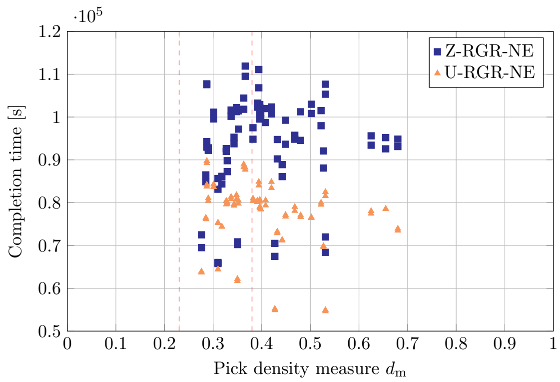


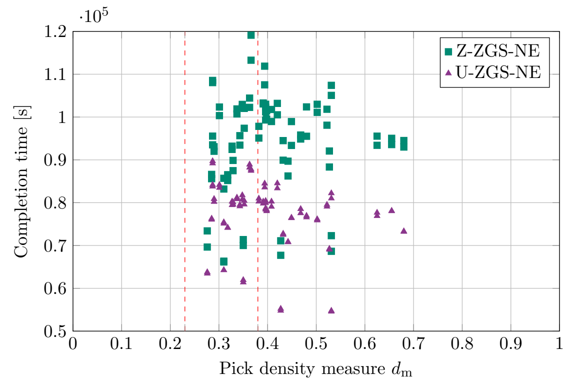
FIGURE 7.10: *The total completion time for Scenario* 2 *with all potential optimisation approaches. The x-axis label describes the first two choices (Tier 1 and 2), while the key describes the last three choices (Tier 3, batching method, and Tier 4). Together they form one experimental set of algorithms. The y-axis label indicates the time it takes to complete all* 24 *waves for each combination.*

The corresponding pick density measure d_{m} to the total completion time of each picking wave is depicted in Figure 7.11. No picking wave has a lower average pick density measure than 0.276 for all different combinations. Therefore, the U-configuration results in the lowest completion times. For the RGR batching metrics this is depicted in Figure 7.11(a), while for the ZGS batching metric it is shown in Figure 7.11(b). The GI assignment approach results in the highest d_{m} of 0.680 thus supporting order batching.

In Figure 7.12 the sum of the completion times for all potential decision tier combinations of the 12 sample picking waves in Scenario 3 is illustrated. Here the choice of configuration influences the total completion time the most. The difference in the third scenario can amount to almost



(a) *Time and density with RGR for the Z- and U-configuration.*



(b) *Time and density with ZGS for the Z- and U-configuration.*

FIGURE 7.11: *The completion times plotted against an increasing pick density measure for both configurations in Scenario* 2 *with batching metrics RGR and ZGS.*

2 hours per picking wave. The GI assignment followed by GIDM results in the lowest completion times. Tier 2 again influences the overall picking time by less than 1%. Using ZGS for batching the orders and walking in a U-configuration generates the lowest total completion time.

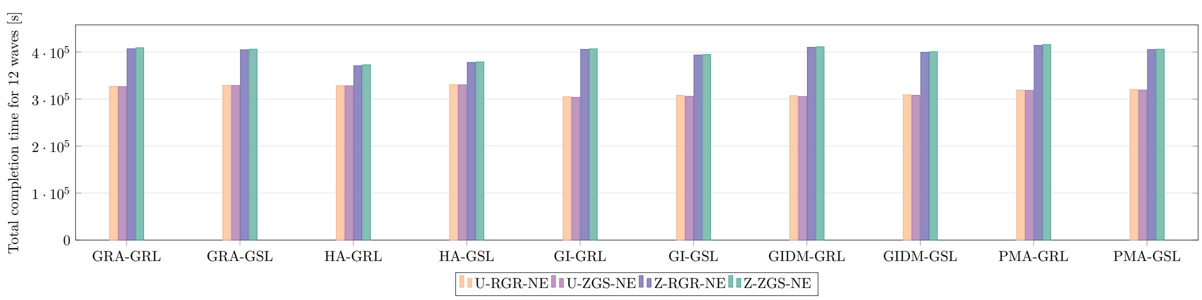
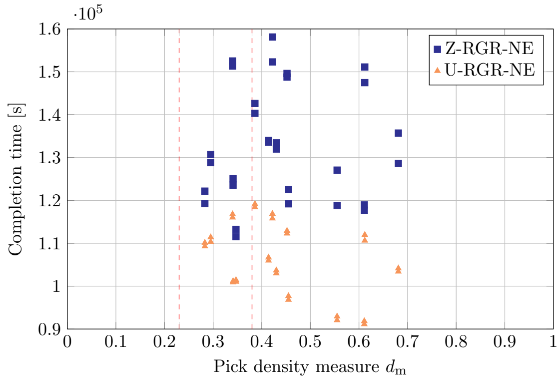


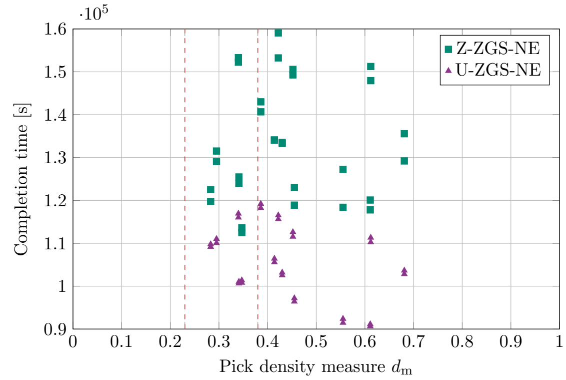
FIGURE 7.12: *The total completion time for Scenario* 3 *with all potential optimisation approaches. The x-axis label describes the first two choices (Tier 1 and 2), while the key describes the last three choices (Tier 3, batching method, and Tier 4). Together they form one experimental set of algorithms. The y-axis label indicates the time it takes to complete all* 12 *waves for each combination.*

In Figure 7.13 the pick density measure d_m with the total completion time of each picking wave in the third scenario is illustrated. The lowest d_m value is 0.283. Therefore, the U-configuration always yields lower total completion times. The highest pick density measure is 0.681 and is again generated by the GI assignment in support of order batching. For the RGR batching metric this is illustrated in Figure 7.13(a) and for the ZGS metric it is depicted in Figure 7.13(b).

The preferred solution method for each decision tier may be determined by analysing the three scenarios. Tier 1 should make use of the GI assignment as it consistently increases pick density the most. The influence of Tier 2 is very limited. Therefore, an easily implementable random arrangement approach such as the GRL arrangement should be deployed. Implementing the GI increases the pick density. Therefore, the pick density measure d_m would favour the U-configuration in Tier 3. The order batching metric ZGS generates the lowest completion times, which can be determined after the orders are formed, locations of their SKUs are determined, and a configuration of the picking system is chosen. In Tier 4 the easily implementable NE heuristic sequences the batches for the pickers. Therefore, using the GI-GRL-U-ZGS-NE combination consistently results in the lowest overall picking times. This holistic solution approach runs well within the time frame that is deemed acceptable by the Retailer.



(a) *Time and density with RGR for the Z- and U-configuration.*



(b) *Time and density with ZGS for the Z- and U-configuration.*

FIGURE 7.13: *The completion times plotted against an increasing pick density measure for both configurations in Scenario 3 with batching metrics RGR and ZGS.*

The GI-GRL-U-ZGS-NE solution approach is tested against a benchmark solution approach including historical assignment (HA), followed by a random arrangement of SKUs (GRL). With a high enough pick density, the U-configuration is chosen but no batching (NB) is implemented and the NE heuristic is applied to sequence the orders. This choice of algorithms generates the benchmark solution approach as HA-GRL-U-NB-NE. The completion times for GI-GRL-U-ZGS-NE and HA-GRL-U-NB-NE are plotted in Figure 7.14 for the three scenarios. In the first scenario 28.4% of picking time can be saved, in the second scenario 27.9% and in the third scenario 27.2% can be saved by applying the solution approach GI-GRL-U-ZGS-NE that includes order batching.

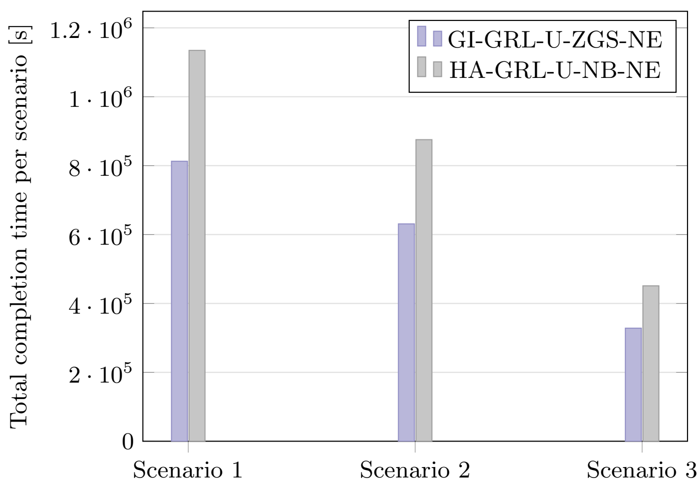


FIGURE 7.14: *Comparison between the holistically optimised picking system including order batching and a benchmark without order batching.*

7.5.2 Statistical analysis

The influence of the four independent variables, namely Tier 1, Tier 2, Tier 3 and batching (additional layer) on the total completion time per picking wave is investigated applying inferential statistics. Tier 4 is not included since batching is introduced before this tier and batches can simply be viewed as larger single orders. To see whether there is statistical difference between variables a one-way ANOVA per independent variable with a Tukey's HSD *post-hoc* test and a fractional ANOVA that includes all independent variables are applied to all scenarios.

In Table 7.2 the ANOVA for Tier 1 and Tier 2 are not statistically significant. However, the ANOVA on batching is statistically significant ($F(2, 1\ 557) = 117.2, p = 0.000$) pointing out the influence of order batching on the total completion time. The Tukey's HSD *post-hoc* test shows a significant difference between RGR and NB (difference between means = −21 449.4, $p = 0.000$) and between ZGS and NB (difference between means = −21 418.8, $p = 0.000$). There is a statistically significant difference between the configurations Z and U in Tier 3 (difference between means = 12 148.67, $p = 0.000$) [106]. Nevertheless, the factorial ANOVA that includes all four independent variables shows no significant influence through the interaction of Tier 1, Tier 2, Tier 3 and batching ($F(8, 1\ 500) = 0.002, p = 1.000$) as illustrated in Table 7.2 [107].

	Sum of squares	*df*	Mean square	*F*	*p*
One-way ANOVA					
Tier 1	$3.85E+09$	4	$9.62E+08$	1.233	0.213
Error	$1.21E+12$	1 555	$7.8E+08$		
Tier 2	$2.08E+06$	1	2 081 219	0.003	0.955
Error	$1.22E+12$	1 558	$7.81E+08$		
Tier 3	$5.76E+10$	1	$5.76E+10$	77.32	0.000**
Error	$1.16E+12$	1 558	$7.45E+08$		
Batching	$1.59E+11$	2	$7.96E+10$	117.2	0.000**
Error	$1.06E+12$	1 557	$6.80E+08$		
Four-way ANOVA					
Tier 1 × Tier 2	$5.48E+07$	4	$1.37E+07$	0.021	0.999
Tier 1 × Tier 3	$2.58E+09$	4	$6.44E+08$	0.976	0.419
Tier 1 × Batching	$2.19E+09$	8	$2.74E+08$	0.414	0.913
Tier 2 × Tier 3	$9.15E+06$	1	$9.15E+06$	0.014	0.906
Tier 2 × Batching	$2.26E+07$	2	$1.13E+07$	0.017	0.983
Batching × Tier 3	$1.36E+09$	2	$6.80E+08$	1.031	0.357
Tier 1 × Tier 2 × Tier 3	$5.09E+07$	4	$1.27E+07$	0.019	0.999
Tier 1 × Tier 2 × Batching	$9.38E+06$	8	$1.17E+06$	0.002	1.000
Tier 1 × Batching × Tier 3	$2.46E+08$	8	$3.08E+07$	0.047	1.000
Tier 2 × Batching × Tier 3	$7.91E+07$	2	$3.96E+07$	0.060	0.942
Tier 1 × Tier 2 × Batching × Tier 3	$1.17E+07$	8	$1.47E+06$	0.002	1.000
Error	$9.90E+11$	1 500	$6.60E+08$		

Note: Two asterisks indicate significance at the 5% level or below.

TABLE 7.2: *One-way ANOVA and four-way ANOVA on Tier 1, Tier 2, batching and Tier 3.*

The results suggest that order batching and the choice of configuration have the biggest influence on the total completion time. Therefore, an integrated approach of optimising the picking system should have a strong focus on Tier 3 and the additional layer of order batching.

7.6 Conclusion

Order batching reduces walking distance significantly and thus contributes to the main objective of minimising the total completion time for picking waves. It is therefore an important additional layer in the decision process. In the three scenarios tested in this chapter, an average of 27.8% of picking time can be saved when the GI-GRL-U-ZGS-NE solution approach (which includes order batching) is tested against the benchmark of HA-GRL-U-NB-NE.

The U-configuration, given a high enough pick density on the picking line, outperforms the Z-configuration. The question then becomes: should both systems be used, or would it be more beneficial to focus optimisation efforts exclusively on the U-configuration? If only one configuration choice is implemented, the optimisation process could be reduced to three decision tiers. This suggestion should be discussed with the Retailer's management.

Alternatively, the competitiveness of the Z-configuration could be improved by including a bidirectional option. In this bidirectional option, pickers would be able to pick orders that are located in the opposite direction of their current picking path. Thereby, pickers would be allowed to change directions, if the order that lies in the opposite direction is closer to the current location. Future studies could thus aim at introducing a bidirectional option in the Z-configuration and compare its efficiency to this of the current U-configuration.

7.7 Chapter summary

In this chapter, the picking system of a South African retailer was solved by looking at four decision tiers within the system. Order batching was included as an additional layer in the optimisation process. Tier 1 aims to answer which DBN should be assigned to which picking line. The greedy insertion heuristic (GI), introduced by Matthews and Visagie [80], generated the lowest completion time by increasing pick density. This increase in pick density indirectly supports order batching. Secondly, the locations of the subset of SKUs that belongs to the DBN are arranged. According to Matthews and Visagie [82] this decision does not have a big influence on the overall picking time. This was confirmed by the results of this chapter. Therefore, an easily implementable greedy random arrangement (GRL) solution approach is recommended. Depending on the level of pick density measure either a U- or Z-configuration was chosen in Tier 3. This choice determines how pickers have to walk along (U-configuration) or between (Z-configuration) the locations of the picking line. If the pick density measure is above 0.38, Hofmann and Visagie [58] recommended the application of a U-configuration. This recommendation is supported by the results of this chapter. Therefore, an increase of pick density through Tier 1 may result in a preference for the U-configuration.

Order batching was directly included between Tier 3 and 4. For the batching, a stops non-identical spans batching metric with a greedy smallest entry heuristic (ZGS) is recommended. Tier 4 views batches as bigger orders. Therefore, the nearest end heuristic, as introduced by Matthews and Visagie [78], was the chosen solution approach for sequencing. This chapter shows that all decision tiers can either indirectly support order batching through increasing the pick density (Tier 1, 2 and 3) or directly include batches formed by batching metrics and combination algorithms (Tier 4).

The introduction of order batching to the unidirectional cyclical picking line is summarised in the next chapter. Recommendations for the practical application will be provided and some ideas for future studies are highlighted.

CHAPTER 8

Conclusion

Contents

In conclusion of this dissertation, a summary of the dissertation is presented in the following chapter. A summary of the main conclusions are provided and recommendations are suggested to practitioners working in a picking system with a unidirectional cyclical picking line. In Chapter 1 objectives were set for this dissertation. Section 8.3 explains how and where these objectives were met throughout the dissertation. The major contributions of this dissertation are also pointed out. To wrap up this chapter and dissertation, ideas for future research on this topic are presented.

8.1 Dissertation summary

In Chapter 1, supply chain management was discussed and why it is important for supply chains to run efficiently. The logistics network of a company forms part of its supply chain. Distribution centres form nodes within the logistics network. DCs transform bulk stock from suppliers into customer orders through the process of order picking. The process of order picking is the most labour-intense and thus most costly process in the operation of a DC. In batch picking, multiple customer orders are picked simultaneously thereby reducing walking distance during the picking process. The logistics network of a prominent South African retailer, who provides the test instances for the dissertation, was described. The focus was on the order picking system with the specific picking line structure of a unidirectional cyclical picking line. This layout can be compared to the unidirectional carousel discussed in literature.

In Chapter 2 a literature review on order batching was carried out. The standard order batching problem is formulated in a single-block warehouse layout with parallel aisles that assigns orders to batches to optimise the objective function. Larger problem instances cannot be solved to optimality. Therefore, heuristics and metaheuristics have been developed. Construction heuristics include priority rule-based algorithms such as a first-in-first-out approach, seed algorithms that initiate batches and allocate orders to these batches, and saving algorithms that compare the time saving of a pair of orders to the individual order. Data mining approaches search for similarities between SKUs before they are grouped into orders. Metaheuristics that are used to solve the OBP are single solution-based algorithms such as iterated local search, variable neighbourhood search, tabu search, simulated annealing, great deluge and hybrid methods combining different metaheuristics. In literature, most OBPs are either solved in the standard single-block parallel aisle warehouse layout or adapted to an automated storage and retrieval system. Applying the OBP to a unidirectional cyclical picking line structure has not been considered in literature.

In Chapter 3 order batching was introduced to a picking system with the layout of a unidirectional cyclical picking line. The objective was to minimise the walking distance of pickers in the picking line. Three order-to-route closeness metrics were introduced to approximate walking distance, since the orders will be batched before the pickers are routed. All metrics were based on the picking location that determines when a picker has to stop at a location to collect the items for an order. These metrics comprise a number of stops, a number of non-identical stops and a stops ratio measurement. A regression analysis supported the idea that the introduced metrics can be used to approximate walking distance. Besides exact solution approaches, four greedy heuristics as well as six metaheuristics were applied to combine similar orders in batches. The capacity of the picking device is restricted, thus only two orders per batch are allowed. All metrics were tested using real life data of 50 sample picking waves from the DC of the Retailer in Durban. The combination of stops ratio metric and the greedy random heuristic generated the best results in terms of minimum number of total cycles traversed. It also seemed to be the most computationally efficient.

In Chapter 4 order batching metrics were extended to include information about the specific layout of a unidirectional cyclical picking line. Route overlap metrics were developed that add information about the route similarity to approximate walking distance before picker routing and thus identify compatible orders for batching. The stops non-identical spans, the non-identical stops-spans, and the stops-spans ratio metric were developed. The best combination of route overlap metric and solution approach was determined in a numerical experiment using real life data of 50 sample picking waves from the Retailer's DC in Durban. The usability of route overlap metrics as approximations for walking distance before picker routing was confirmed by the results. The combination of stops non-identical spans metric and the smallest entry heuristic generated a lower number of total cycles traversed compared to picking location metrics.

An investigation into the effect of order batching on the overall picking time in a unidirectional cyclical picking line was the main focus of Chapter 5. The three main actions of pickers in this system are to walk (in a clockwise direction) to the next pick location, perform the picking operation and to prepare the packaging carton. The total amount of work for the last two actions remain constant during a given wave of picking, rendering the minimisation of walking distance as the main objective when minimising the total picking effort. Order batching increases the pick density, which might lead to an increase in congestion. Congestion can also be influenced by the number of pickers in the picking wave. The question then became, what is the total time saving, and what is the best number of pickers before and after batching? Therefore, this system was modelled with a discrete event simulation to determine the influence of batching on the total completion time. Historical data were compared to the output of the simulation for the verification and validation of the simulation. It showed that the simulation represents the real world picking system to a satisfactory degree. The simulation runs also showed that the order batching heuristics that aim at minimising walking distance reduce the overall picking time by up to 21% and a good number of pickers are dependent on the pick density. Furthermore, pick density can be used as a predictor for picking time reduction.

Chapter 6 introduced a new configuration to a unidirectional cyclical picking line. In the previous chapters, a picking line with a conveyor belt was considered. If this conveyor belt is removed, the picking line can operate in either a U- or a Z-configuration. In the U-configuration pickers move in a clockwise direction to pick items from a single pick face at a time. To adapt this to a Z-configuration, pickers may cross the aisle to pick from either pick face. Through the introduction of a Z-configuration, the question arises which configuration will minimise walking distance and thereby overall picking time. Three scenarios, namely a worst case, an average case, and a real world simulation were evaluated to determine whether pick density can indicate which configuration to choose. The configurations were then combined with adaptations of the order batching metrics for the unidirectional cyclical picking line, since they have shown to reduce walking distance. All three approaches showed that the Z-configuration only outperforms the U-configuration when pickers on average stop at every fourth or fifth location. For real life instances the U-configuration consistently outperformed the Z-configuration.

In Chapter 7 the total picking system was considered. The picking system gives rise to four optimisation problems or decision tiers that must be solved. Together these four tiers constitute a wave of picking. The effect of introducing order batching on all four decision tiers in this picking system was investigated. Tier 1 assigns stock to a picking wave. Tier 2 arranges the stock around the picking line for each wave of picking. This is followed by Tier 3 that decides on the configuration and thereby determines the logic of how pickers should walk. Pickers can either walk along the picking line (U-configuration) or cross the picking aisle (Z-configuration). Tier 4 sequences the orders to be picked during a picking wave. Order batching cannot be introduced in the first two tiers of the decision process because orders are only formed once picking waves have been finalised and location information known. In the absence of this information the best approach was to solve the first two tiers with the objective to increase pick density. An increase in pick density supported the use of a U-configuration over the Z-configuration in Tier 3. After the first three decision tiers, order batching could be introduced through specific order batching metrics and combination algorithms. A nearest end heuristic then sequenced the batches in the last decision tier to minimise walking distance. This four decision tier optimisation approach yielded about 28% saving in total completion time, when compared to benchmark scenarios.

8.2 Recommendations

Reducing walking distance increased congestion, but with a good number of pickers it decreased the overall picking time. This was emphasised by the simulation of the picking process in a unidirectional cyclical picking line developed in Chapter 5 and extended to include both configurations of the picking line in Chapter 6. Optimising a picking system should focus on reducing walking distance, as the time for picking and handling of products remains constant. Therefore, minimising walking distance reduces the total completion time in a unidirectional cyclical picking line.

Order batching showed to reduce walking distance effectively. Introducing order batching to a unidirectional cyclical picking line was best achieved by applying the stops non-identical spans batching metric combined with the greedy smallest entry heuristic. Even though solely implementing picking locations based batching metrics shorten the walking distance considerably, as shown in Chapter 3, adding information about the specific layout of the picking line, as provided by Chapter 4, resulted in the shortest walking distance and thus reduced picking time for all test instances provided by the Retailer. The additional time saving generated by this route overlap metric constantly increases and with the Retailer running up to 12 picking waves in parallel each day, this reduction contributes significantly to the saving in operational cost. The introduction of order batching as an extra layer should take place after the configuration of the picking line has been determined in Tier 3 as this choice influences the span or path that a picker has to walk during the order picking process.

In a holistic optimisation approach of the picking system four decision tiers, namely SPALP (Tier 1), SAP (Tier 2), SCP (Tier 3) and OSP (Tier 4), have to be addressed. Tier 1 cannot implement order batching as no orders are formed yet. However, increasing the pick density measure d_{m}, that was introduced in Chapter 5, while assigning orders to picking waves supports batch forming which in turn reduces walking distance. Therefore, a greedy heuristic that increases pick density is recommended. This was shown in the results of Chapter 7. A simple greedy random arrangement heuristic can assign SKUs to their locations as proposed in Chapter 7, because Tier 2 is far outweighed by Tier 4. The nearest end heuristic is an easily implementable option to sequence orders for pickers, while reducing walking distance in the last decision tier. This option is thus recommended and was implemented throughout Chapters 3, 4 and 7. The optimisation approach should focus on order batching and Tier 3.

If the pick density measure d_{m} is above a critical threshold in Tier 3, the U-configuration will always be preferred to the Z-configuration as presented in Chapter 6. The question thus arises whether two configurations of the picking lines should be accessible, since increasing the pick density measure d_{m} supports order batching in the first two decision tiers and a high pick density indicates a shorter picking time as pointed out in Chapter 5. Increasing the pick density measure d_{m} and only keeping the U-configuration alive would simplify the holistic optimisation of the order picking system of the Retailer.

8.3 Achievements of objectives

The following objectives were identified in Section 1.5.

OBJECTIVE I: Investigate the order picking system on a unidirectional picking line:

a Describe the layout and operations of the Retailer's DCs to comprehend the broader context of the problem;

b Describe the order picking system in detail to emphasise the characteristics of a unidirectional cyclical picking line;

c Describe the different configurations of the order picking system.

OBJECTIVE II: Perform a literature study:

a Describe the optimisation approaches on a unidirectional cyclical picking line;

b Describe the standard order batching problem and its solution approaches;

c Identify the differences (in layout) to a unidirectional cyclical picking line.

OBJECTIVE III: Apply order batching to a unidirectional cyclical picking line:

a Model order batching on a unidirectional cyclical picking line;

b Emphasise the specific layout in the order batching approach.

Objective IV: Build a simulation of the order picking system to measure picking time:

a Build a tool that can measure picking time by simulating the U-configuration of the unidirectional cyclical picking line;

b Build a tool that can measure picking time by simulating the Z-configuration of the unidirectional cyclical picking line;

c Identify a predictor for picking time;

c Determine an indicator for the selection of configuration.

Objective V: Apply order batching as an extra layer to all four decision tiers:

a Apply order batching to Tier 1;

b Apply order batching to Tier 2;

c Apply order batching and define Tier 3;

d Apply order batching to Tier 4.

Objective VI: Propose directions for future research:

a Summarise the contributions of this dissertation;

b Give recommendations to practitioners;

c Propose further optimisation ideas for a unidirectional cyclical picking line.

The research question in Section 1.4 is answered by addressing each objective. The achievement of these objectives is presented.

Objective I was achieved in Chapter 1. The Retailer's operation and its specific approach to central planning of inventory and wave picking were pointed out. The set up of the unidirectional cyclical picking line was compared to the unidirectional carousel in detail. Chapters 3 to 7 included information about the unidirectional cyclical picking line. Especially Chapter 6 emphasised the characteristics of the two configurations of the picking system.

Objective II was achieved in Chapter 2. Optimisation approaches on a unidirectional cyclical picking line have been presented. The standard order batching problem in a single-block parallel aisles warehouse layout was described. Several solution approaches such as construction heuristics and metaheuristics were introduced. The differences in the structural set up of the unidirectional cyclical picking line layout were pointed out.

Objective III was achieved in Chapters 3 and Chapter 4. Order batching was introduced to the unidirectional picking line by developing picking location based distance approximation metrics in Chapter 3. Additionally, layout specific characteristics were taken into account by introducing the route overlap addition metrics in Chapter 4. These distance approximations seemed to produce batches that reduce walking distance the most.

Objective IV was achieved in Chapter 5 and extended to include both configurations of the picking line in Chapter 6. The simulation that was developed can measure picking time in the U-configuration and the Z-configuration. The pick density measure d_{m} was identified as a predictor for picking time together with the explanatory variables of orders, locations, and picks. Furthermore, d_{m} can be used to indicate when to switch from a Z- to a U-configuration to generate the shortest total completion time for a picking wave.

OBJECTIVE V was achieved in Chapter 7. Order batching was applied indirectly through an increase in pick density measure to the first and second decision tier. In Tier 3, the pick density measure helped to choose the configuration that generates the shortest picking time. The U- and Z-configurations were defined as they influence the introduction of order batching by determining a picker's path. Order batching was then applied before Tier 4. A batch of orders can be viewed as one larger order to be sequenced in this last decision tier.

OBJECTIVE VI will be achieved in this chapter. Firstly, the dissertation was summarised. Recommendations to practitioners were provided. Further optimisation ideas for a unidirectional cyclical picking line will be proposed after the contribution of the dissertation.

The three subproblems identified in Section 1.6, namely the introduction of order batching, the simulation of a picking wave, and the optimisation of all decision tiers in the picking system were addressed in five articles. The five articles resulted in Chapters 3 to 7 of this dissertation.

8.4 Contribution

The contribution of this dissertation lies in applying new and known heuristics to a new application through adaptation, change, and improvement. The research question focused on the effects of introducing current practices in an attempt to optimise an order picking system on a unidirectional cyclical picking line in different configurations. By performing the research, the identified gap in research was closed. Additionally, the application of this dissertation has practical significance.

There was a gap in research in optimising the order picking system on a unidirectional cyclical picking line. This problem was addressed through the established picking planning problem of order batching. Order batching aims at reducing walking distance. To generate batches that fulfil this aim successfully, information about the picker routes was needed. However, picker routing before order batching is computationally too expensive. Therefore, distance approximation measures were developed that take the specific structural set up of the unidirectional cyclical picking line into account.

It was shown that walking distance can be translated into picking time. While order batching increases congestion in the picking system, a good number of pickers can reduce the overall picking time. This was shown in discrete event simulations of the order picking system in a U- and a Z-configuration.

Additionally, in a holistic optimisation of the order picking system on a unidirectional cyclical picking line, the general distance pick density measure d_{m} has been defined. It can be used as a picking time predictor and as an indicator of when to switch from a Z- to a U-configuration. Increasing the pick density measure supports order batching through the first two decision tiers and favours the U-configuration in Tier 3. This valuable measure can be helpful in making decisions about managing the picking system. The effect of introducing the methodology of order batching and structural changes through different configurations on this specific environment was investigated.

The practical advantage of the optimisation lies in the efficient operation of the DC. Order picking is the highest cost of operation. Therefore, it should be carried out in an efficient way. DC managers can use the results of the simulation model as a decision support tool to improve the current order picking system.

This dissertation focuses on the order picking system. Other operations in the DC are not investigated in detail. Additionally, the layout of the entire DC is fixed and thus not further investigated. The methodologies chosen for the scope of the dissertation are a representation of possible methodologies that are not exhaustive.

8.5 Future work

The importance of the pick density measure in reducing overall picking time has been determined by the simulation. The batching metrics in the Chapters 3 and 4 increase pick density indirectly (especially by including the characteristics of the unidirectional picking line in the stops non-identical spans metric). While these metrics minimise the differences between orders and thereby batch similar orders, a batching metric that maximises d_{m} and thus increases pick density directly could be developed to deepen the research.

The discrete event simulation developed in this chapter is used as a tool to measure picking time instead of walking distance. Thereby the effect of order batching on congestion was analysed. The picking system of the Retailer could be modelled as an agent-based simulation to simulate the picking system even more exhaustive and thus include picker specific behaviour for example.

Beside the pick density measure, Chapter 5 showed that the explanatory variables, namely orders, locations, and picks are also statistically significant in predicting the overall picking time. Therefore, including these explanatory variables in the optimisation algorithms of Tier 1 may result in an even shorter overall picking time in Chapter 7.

Even though the Z-configuration does not generate shorter walking distances if the pick density reaches a threshold, the Z-configuration could be improved. It might be extended to include a bidirectional option that allows pickers to change directions if the next item is closer to the current location. This may result in pickers increasing the pick density through operating in the Z-configuration. Therefore, Z-picking could be a viable option in Tier 3. If no improvements can be made to the Z-configuration the option should be taken out to simplify the decision making process.

To broaden the research further, other methodologies such as self-organising systems could be introduced. An example of a self-organising system in warehousing is the organisation of pickers in a bucket brigade (BB). A BB is a variant of a traditional assembly line. Workers are required to sequentially pass work from one to another. Workers are flexible, since there is no fixed assignment of a worker to a station. A simple local rule to determine what to do next is followed by each individual worker. The rule is to carry work from station to station until it is finished or handed over to the next worker and then go back and start a new task. This enables workers to share their workload. After the last worker completes the task, the worker walks back upstream to take over work from the predecessor, who walks back to take over work from her predecessor. This process is carried out until the first worker begins a new task at the starting point of the assembly line. As a final requirement, workers in a BB have to be sequenced from slowest to fasted along the direction of the material flow. However, if a worker catches up to her successor she has to remain inactive until the station becomes available again. In a DC, bucket brigades are most successfully implemented in the activity of order picking. Bucket brigades change the set up of the picking system and the routing of the pickers [6, 7].

The implications of order batching on other operational areas of the Retailer could also be investigated. The instructions from the central planning department dictate the operations at the DC. The composition of these instructions could be optimised to improve data mining approaches for example. Including the planning department in the optimisation process would present a company-wide global optimisation approach.

Glossary

Cycles traversed counts the number of cycles that have to be traversed on a unidirectional cyclical picking line to pick all items that are requested during a picking wave. It is used as a distance measurement [78].

Maximal SKU is needed by the maximum number of orders and thus provides a minimum number of cycles that has to be traversed to pick all orders of a picking wave [78].

Order is the set of store requirements for a single store for all the SKUs that are selected to be processed in a picking wave [78].

Order picking retrieves products from storage or buffer areas and turns them into specific orders to respond to a costumer request [29].

Order batching problem assigns orders to batches simultaneously and calculates a picking tour for each batch to optimise an objective function [36].

Order sequencing problem sequences the orders to be picked by pickers during a picking wave (Tier 4) [78].

Pick density is a measurement for how often a picker stops to pick at a location in relation to the distance walked to collect the items [46, 59].

Picking location metrics are distance approximations that are based on each picking location where a picker has to stop to pick items for an order [55].

Picking wave involves populating a picking line with SKUs to unique locations, the actual order picking process and removing of excess stock from the picking line [78].

Route overlap metrics are distance approximations based on the span of each order [56].

SKU arrangement problem arranges SKUs on locations around the picking line (Tier 2) [82].

SKU assignment problem assigns DBNs containing SKUs to available picking lines (Tier 1) [80].

Span is the distance that has to be covered by a picker to collect all products of an order, given a starting location [78].

System configuration problem determines either a U- or a Z-configuration for the picking wave (Tier 3) [57].

Unidirectional carousel is a picking system that consists of a number of shelves that are linked together and rotate in one direction in a closed loop. A picker that has a fixed position in front of the carousel operates it [73].

Unidirectional cyclical picking line is a picking system implemented by a prominent South African retailer. SKUs are placed in locations around a conveyor belt in the middle of the set up. Pickers operate it by moving clockwise around the conveyor belt collecting all requested items [73].

U-configuration moves all pickers in a clockwise direction around a conveyor belt [78].

Z-configuration makes pickers cross the aisles if the next required SKU on the other side is closer than the SKU on the same side [58].

Bibliography

[1] Aarts E and Korst J, 1988, *Simulated annealing and Boltzmann machines*, John Wiley and Sons Inc., New York, United States of America.

[2] Accorsi R, Manzini R and 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] Agrawal R, Imieliński T and Swami A, 1993, *Mining association rules between sets of items in large databases*, Proceedings of the Proceedings of the 1993 ACM SIGMOD international conference on Management of data, pp. 207–216.

[4] Albareda-Sambola M, Alonso-Ayuso A, Molina E and De Blas CS, 2009, *Variable neighborhood search for order batching in a warehouse*, Asia-Pacific Journal of Operational Research, **26(05)**, pp. 655–683.

[5] Banks J, 1998, *Handbook of simulation: Principles, methodology, advances, applications, and practice*, John Wiley & Sons, Hoboken, New Jersey, United States of America.

[6] Bartholdi JJ, Eisenstein DD and Lim YF, 2010, *Self-organizing logistics systems*, Annual Reviews in Control, **34(1)**, pp. 111–117.

[7] Bartholdi JJI and Hackman ST, 2016, *Warehouse & Distribution Science: Release 0.97*, Supply Chain and Logistics Institute, Available from `https://www.warehouse-science.com/book/index.html`.

[8] Bartholdi JJI and Platzman LK, 1986, *Retrieval strategies for a carousel conveyor*, IIE transactions, **18(2)**, pp. 166–173.

[9] Bartholdi III JJ and Platzman LK, 1988, *Heuristics based on spacefilling curves for combinatorial problems in euclidean space*, Management Science, **34(3)**, pp. 291–305.

[10] Beamon BM, 1998, *Supply chain design and analysis: Models and methods*, International journal of production economics, **55(3)**, pp. 281–294.

[11] Bindi F, Manzini R, Pareschi A and 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.

[12] Blum C, Puchinger J, Raidl GR and Roli A, 2011, *Hybrid metaheuristics in combinatorial optimization: A survey*, Applied Soft Computing, **11(6)**, pp. 4135–4151.

[13] Blum C and Roli A, 2003, *Metaheuristics in combinatorial optimization: Overview and conceptual comparison*, ACM computing surveys (CSUR), **35(3)**, pp. 268–308.

[14] Boussaïd I, Lepagnot J and Siarry P, 2013, *A survey on optimization metaheuristics*, Information Sciences, **237(1)**, pp. 82–117.

[15] Bozer YA and Kile JW, 2008, *Order batching in walk-and-pick order picking systems*, International Journal of Production Research, **46(7)**, pp. 1887–1909.

[16] Burghout W, 2004, *A note on the number of replication runs in stochastic traffic simulation models*, [Online], [Cited August 27th, 2019], Available from `https://pdfs.semanticscholar.org/ea3b/70673c2e27ead5c9b932fb3e2acdccb959ab.pdf`.

[17] Cano JA, Correa-Espinal AA and Gómez-Montoya RA, 2018, *A review of research trends in order batching, sequencing and picker routing problems*, Revista Espacios, **39(04)**, pp. 3–15.

[18] Cergibozan Ç and Tasan AS, 2019, *Order batching operations: an overview of classification, solution techniques, and future research*, Journal of Intelligent Manufacturing, **30(1)**, pp. 335–349.

[19] Černỳ V, 1985, *Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm*, Journal of optimization theory and applications, **45(1)**, pp. 41–51.

[20] Chen MC, Huang CL, Chen KY and Wu HP, 2005, *Aggregation of orders in distribution centers using data mining*, Expert Systems with Applications, **28(3)**, pp. 453–460.

[21] Chen MC and Wu HP, 2005, *An association-based clustering approach to order batching considering customer demand patterns*, Omega, **33(4)**, pp. 333–343.

[22] Chiang DMH, Lin CP and 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.

[23] Chiarandini M, Paquete L, Preuss M and Ridge E, 2007, *Experiments on metaheuristics: Methodological overview and open issues*, Technical Report DMF-2007-03-003, Available from `https://pdfs.semanticscholar.org/9fcd/f48d4ae1770a6c653e33caee25aeb109b44d.pdf`.

[24] Clarke G and Wright JW, 1964, *Scheduling of vehicles from a central depot to a number of delivery points*, Operations research, **12(4)**, pp. 568–581.

[25] Coyle JJ, Bardi EJ and Langley Jr JC, 2003, *The management of business logistics: A supply chain perspective*, volume 7, Thomson Learning, Boston, United States of America.

[26] D'Agostino RB, 1986, *Goodness-of-fit-techniques*, volume 68 of *Statistics: A Series of Textbooks and Monographs*, CRC press, Florida, United States of America.

[27] Dallari F, Marchet G and Melacini M, 2009, *Design of order picking system*, The International Journal of Advanced Manufacturing Technology, **42(1)**, pp. 1–12.

[28] De Koster M, Van der Poort ES and Wolters M, 1999, *Efficient order batching methods in warehouses*, International Journal of Production Research, **37(7)**, pp. 1479–1504.

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

[30] De Villiers AP, 2012, *Minimising the total travel distance to pick orders on a unidirectional picking line*, Masters Thesis, Stellenbosch University.

[31] Dréo J, Pétrowski A, Siarry P and Taillard E, 2006, *Metaheuristics for hard optimization: Methods and case studies*, Springer Science & Business Media, Berlin.

[32] Dueck G, 1993, *New optimization heuristics*, Journal of Computational physics, **104(1)**, pp. 86–92.

[33] Elsayed EA and Stern RG, 1983, *Computerized algorithms for order processing in automated warehousing systems*, The International Journal of Production Research, **21(4)**, pp. 579–586.

[34] Elsayed EA and Unal O, 1989, *Order batching algorithms and travel-time estimation for automated storage/retrieval systems*, The International Journal of Production Research, **27(7)**, pp. 1097–1114.

[35] Frazelle E, 2002, *World-class warehousing and material handling*, volume 1, McGraw-Hill, New York, United States of America.

[36] Gademann N and Velde S, 2005, *Order batching to minimize total travel time in a parallel-aisle warehouse*, Institute of Industrial Engineers Transactions, **37(1)**, pp. 63–75.

[37] Gibson DR and Sharp GP, 1992, *Order batching procedures*, European Journal of Operational Research, **58(1)**, pp. 57–67.

[38] Glover F, 1986, *Future paths for integer programming and links to artificial intelligence*, Computers & Operations Research, **13(5)**, pp. 533–549.

[39] Glover F and Laguna M, 1998, *Tabu search*, pp. 2093–2229 in *Handbook of combinatorial optimization*, pp. 2093–2229. Springer, New York, United States of America.

[40] Goetschalckx M and Ratliff D, 1988, *Order picking in an aisle*, IIE transactions, **20(1)**, pp. 53–62.

[41] Gong Y, 2009, *Stochastic modelling and analysis of warehouse operations*, Doctoral Dissertation, Erasmus University, Rotterdam.

[42] Gu J, Goetschalckx M and McGinnis LF, 2007, *Research on warehouse operation: A comprehensive review*, European Journal of Operational Research, **177(1)**, pp. 1–21.

[43] Gu J, Goetschalckx M and McGinnis LF, 2010, *Research on warehouse design and performance evaluation: A comprehensive review*, European Journal of Operational Research, **203(3)**, pp. 539–549.

[44] Gue KR, Meller RD and Skufca JD, 2006, *The effects of pick density on order picking areas with narrow aisles*, Independent Institute of Education Transactions, **38(10)**, pp. 859–868.

[45] Hagspihl R, 2012, *The number of pickers and sku arrangement on a unidirectional picking line*, [Online], [Cited January 8th, 2020], Available from `http://www.orssa.org.za/wiki/uploads/Awards/Hagspihl_Project.pdf`.

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

[47] Han J, Pei J and Kamber M, 2011, *Data mining: concepts and techniques*, Elsevier.

[48] Hassini E, 2009, *One-dimensional carousel storage problems: Applications, review and generalizations*, INFOR: Information Systems and Operational Research, **47(2)**, pp. 81–92.

[49] Henn S, Koch S, Doerner KF, Strauss C and Wäscher G, 2010, *Metaheuristics for the order batching problem in manual order picking systems*, Business Research, **3(1)**, pp. 82–105.

[50] Henn S, Koch S and Wäscher G, 2012, *Order batching in order picking warehouses: a survey of solution approaches*, pp. 105–137 in *Warehousing in the Global Supply Chain*, pp. 105–137. Springer, London.

[51] Henn S and Wäscher G, 2012, *Tabu search heuristics for the order batching problem in manual order picking systems*, European Journal of Operational Research, **222(3)**, pp. 484–494.

[52] Ho YC, Su TS and Shi ZB, 2008, *Order-batching methods for an order-picking warehouse with two cross aisles*, Computers & Industrial Engineering, **55(2)**, pp. 321–347.

[53] Ho YC and Tseng YY, 2006, *A study on order-batching methods of order-picking in a distribution centre with two cross-aisles*, International Journal of Production Research, **44(17)**, pp. 3391–3417.

[54] Hofmann F, 2020, *PHD*, [Online], [Cited September 30th, 2020], Available from `https://github.com/floratheeexplorer/PHD`.

[55] Hofmann F and Visagie S, 2019, *Picking location metrics for order batching on a unidirectional cyclical picking line*, ORiON, **35(2)**, pp. 161–186.

[56] Hofmann F and Visagie S, 2019, *Route overlap metrics for order batching on a unidirectional cyclical picking line*, submitted.

[57] Hofmann F and Visagie S, 2020, *Batching orders in a unidirectional cyclical picking system*, to be submitted.

[58] Hofmann F and Visagie S, 2020, *Configuration selection on a unidirectional cyclical picking line*, submitted.

[59] Hofmann F and Visagie S, 2020, *The effect of order batching on a unidirectional picking line's completion time*, accepted, to appear in International Journal of Logistics Systems and Management.

[60] Hong S, 2015, *Order batch formations for less picker blocking in a narrow-aisle picking system*, Industrial Engineering & Management Systems, **14(3)**, pp. 289–298.

[61] Hsieh Lf and Tsai L, 2006, *The optimum design of a warehouse system on order picking efficiency*, The International Journal of Advanced Manufacturing Technology, **28(5-6)**, pp. 626–637.

[62] Hwang H and Lee MK, 1988, *Order batching algorithms for a man-on-board automated storage and retrieval system*, Engineering Costs and Production Economics, **13(4)**, pp. 285–294.

[63] IBM, 2018, *Spss version 25*, [Online], [Cited October 8th, 2018], Available from `https://www.ibm.com/analytics/spss-statistics-software`.

[64] Jarvis JM and McDowell ED, 1991, *Optimal product layout in an order picking warehouse*, IIE transactions, **23(1)**, pp. 93–102.

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

[66] Kirkpatrick S, Gelatt CD and Vecchi MP, 1983, *Optimization by simulated annealing*, Science, **220(4598)**, pp. 671–680.

[67] Kress D, Boysen N and Pesch E, 2017, *Which items should be stored together? A basic partition problem to assign storage space in group-based storage systems*, IISE Transactions, **49(1)**, pp. 13–30.

[68] Laerd Statistics, 2018, *One-way anova in spss statistics (cont)*, [Online], [Cited October 5th, 2018], Available from `https://statistics.laerd.com/spss-tutorials/one-way-anova-using-spss-statistics-2.php`.

[69] Laerd Statistics, 2018, *Two-way anova in spss statistics (cont)*, [Online], [Cited October 5th, 2018], Available from `https://statistics.laerd.com/spss-tutorials/two-way-anova-using-spss-statistics-2.php`.

[70] Law AM, 2015, *Simulation modeling and analysis*, volume 5, McGraw-Hill Education, New York, United States of America.

[71] Le Roux GJ and Visagie SE, 2017, *A multi-objective approach to the assignment of stock keeping units to unidirectional picking lines*, South African Journal of Industrial Engineering, **28(1)**, pp. 190–209.

[72] Lindo Systems, 2018, *Lingo 11*, [Online], [Cited October 8th, 2018], Available from `https://www.lindo.com/`.

[73] Litvak N and Vlasiou M, 2010, *A survey on performance analysis of warehouse carousel systems*, Statistica Neerlandica, **64(4)**, pp. 401–447.

[74] Lourenço HR, Martin OC and Stützle T, 2003, *Iterated local search*, pp. 320–353 in *Handbook of metaheuristics*, pp. 320–353. Springer Science & Business Media, Berlin.

[75] Manzini R, 2006, *Correlated storage assignment in an order picking system*, International Journal of Industrial Engineering-Theory Applications and Practice, **13(4)**, pp. 384–394.

[76] Matthews J, 2012, *Order sequencing and SKU arrangement on a unidirectional picking line*, Masters Thesis, Stellenbosch: Stellenbosch University.

[77] Matthews J, 2015, *SKU assignment in a multiple picking line order picking system.*, Doctoral Dissertation, Stellenbosch: Stellenbosch University.

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

[79] Matthews J and Visagie SE, 2014, *Picking line data set a*, [Online], [Cited October 5th, 2018], Available from http://hdl.handle.net/10019.1/86143.

[80] Matthews J and Visagie SE, 2015, *Assignment of stock keeping units to parallel unidirectional picking*, South African Journal of Industrial Engineering, **26(1)**, pp. 235–251.

[81] Matthews J and Visagie SE, 2015, *SKU assignment to unidirectional picking lines using correlations*, ORiON, **31(2)**, pp. 61–70.

[82] Matthews J and Visagie SE, 2019, *SKU arrangement on a unidirectional picking line*, International Transactions in Operational Research, **26(1)**, pp. 100–130.

[83] Matusiak M, de Koster R, Kroon L and Saarinen J, 2014, *A fast simulated annealing method for batching precedence-constrained customer orders in a warehouse*, European Journal of Operational Research, **236(3)**, pp. 968–977.

[84] Microsoft, 2018, [Cited October 8th, 2018], Available from https://www.microsoft.com/.

[85] Mladenović N and Hansen P, 1997, *Variable neighborhood search*, Computers & Operations Research, **24(11)**, pp. 1097–1100.

[86] Muter I and Öncan T, 2015, *An exact solution approach for the order batching problem*, IIE Transactions, **47(7)**, pp. 728–738.

[87] Nicolas L, Yannick F and Ramzi H, 2017, *Optimization of order batching in a picking system with carousels*, IFAC-PapersOnLine, **50(1)**, pp. 1106–1113.

[88] Nicolas L, Yannick F and Ramzi H, 2018, *Order batching in an automated warehouse with several vertical lift modules: Optimization and experiments with real data*, European Journal of Operational Research, **267(3)**, pp. 958–976.

[89] NumPy Steering Council, 2020, *Numpy v1.19.0*, [Online], [Cited September 29th, 2020], Available from https://numpy.org//.

[90] Öncan T, 2015, *MILP formulations and an iterated local search algorithm with tabu thresholding for the order batching problem*, European Journal of Operational Research, **243(1)**, pp. 142–155.

[91] Orlin JB and Lee Y, 1993, *Quickmatch - a very fast algorithm for the assignment problem*, [Online],].

[92] Pan CH and Liu S, 1995, *A comparative study of order batching algorithms*, Omega, **23(6)**, pp. 691–700.

[93] Pandas, 2020, *Pandas 1.1.2*, [Online], [Cited September 29th, 2020], Available from https://pandas.pydata.org//.

[94] Python Software Foundation, 2018, *Python 3.6*, [Online], [Cited October 8th, 2018], Available from https://www.python.org/.

[95] R Core, 2019, *The r project for statistical computing*, [Online], [Cited August 27th, 2019], Available from https://www.r-project.org/.

[96] Robinson S, 1997, *Simulation model verification and validation: increasing the users' confidence*, Proceedings of the Winter Simulation Conference, Atlanta, Georgia, United States of America, pp. 53–59.

[97] Rosenwein M, 1996, *A comparison of heuristics for the problem of batching orders for warehouse selection*, International Journal of Production Research, **34(3)**, pp. 657–664.

[98] Rouwenhorst B, Reuter B, Stockrahm V, van Houtum GJ, Mantel R and Zijm W, 2000, *Warehouse design and control: Framework and literature review*, European Journal of Operational Research, **122(3)**, pp. 515–533.

[99] Ruben RA and Jacobs FR, 1999, *Batch construction heuristics and storage assignment strategies for walk/ride and pick systems*, Management Science, **45(4)**, pp. 575–596.

[100] Sargent RG, 2010, *Verification and validation of simulation models*, Proceedings of the Proceedings of the 2010 Winter Simulation Conference, Baltimore, Maryland, United States, pp. 166–183.

[101] Sheskin DJ, 2000, *Handbook of parametric and nonparametric statistical procedures*, CRC Press, Boca Raton, Florida, United States of America.

[102] SPSS Tutorials, 2018, *Spss one-way anova tutorial*, [Online], [Cited October 5th, 2018], Available from `https://www.spss-tutorials.com/spss-one-way-anova/`.

[103] SPSS Tutorials, 2018, *Spss one-way anova tutorial*, [Online], [Cited October 5th, 2018], Available from `https://www.spss-tutorials.com/spss-one-way-anova-with-post-hoc-tests-example/`.

[104] SPSS Tutorials, 2018, *Spss two way anova – basics tutorial*, [Online], [Cited October 5th, 2018], Available from `https://www.spss-tutorials.com/spss-two-way-anova-basics-tutorial/`.

[105] Stern H, 1986, *Parts location and optimal picking rules for a carousel conveyor automatic storage and retrieval system*, Proceedings of the Proceedings of the 7th International Conference on Automation in Warehousing, pp. 185–193.

[106] STHDA, 2020, *One-way anova test in r*, [Online], [Cited February 8th, 2020], Available from `http://www.sthda.com/english/wiki/one-way-anova-test-in-r#relaxing-the-homogeneity-of-variance-assumption`.

[107] STHDA, 2020, *Two-way anova test in r*, [Online], [Cited February 8th, 2020], Available from `http://www.sthda.com/english/wiki/two-way-anova-test-in-r`.

[108] Stützle T, 1998, *Local search algorithms for combinatorial problems*, Doctoral Dissertation, Darmstadt University of Technology.

[109] Talbi EG, 2009, *Metaheuristics: From design to implementation*, John Wiley & Sons, New Jersey, United States of America.

[110] Tompkins JA, White JA, Bozer YA and Tanchoco JMA, 2010, *Facilities planning*, John Wiley & Sons, Hoboken, New Jersey, United States of America.

[111] Toth P and Martello S, 1990, *Knapsack problems: Algorithms and computer implementations*, Wiley, Hoboken, New Jersey, United States of America.

[112] Valle CA and Beasley JE, 2020, *Order batching using an approximation for the distance travelled by pickers*, European Journal of Operational Research, **284(2)**, pp. 460–484.

[113] Van den Berg JP and Zijm WH, 1999, *Models for warehouse management: Classification and examples*, International Journal of Production Economics, **59(1-3)**, pp. 519–528.

[114] Van Gils T, Caris A and Ramaekers K, 2018, *Reducing picker blocking in a high-level narrow-aisle order picking system*, Proceedings of the 2018 Winter Simulation Conference, Gothenburg, Sweden, pp. 2953–2965.

[115] Van Gils T, Ramaekers K, Braekers K, Depaire B and Caris A, 2018, *Increasing order picking efficiency by integrating storage, batching, zone picking, and routing policy decisions*, International Journal of Production Economics, **197(1)**, pp. 243–261.

[116] Van Gils T, Ramaekers K, Caris A and de Koster RB, 2018, *Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review*, European Journal of Operational Research, **267(1)**, pp. 1–15.

[117] Van Laarhoven PJ and Aarts EH, 1987, *Simulated annealing*, pp. 7–15 in *Simulated annealing: Theory and applications*, pp. 7–15. Springer, Berlin.

[118] Vickson R and Lu X, 1998, *Optimal product and server locations in one-dimensional storage racks*, European journal of operational research, **105(1)**, pp. 18–28.

[119] Vickson RG and Fujimoto A, 1996, *Optimal storage locations in a carousel storage and retrieval system*, Location Science, **4(4)**, pp. 237–245.

[120] Wäscher G, 2004, *Order picking: a survey of planning problems and methods*, pp. 323–347 in *Supply chain management and reverse logistics*, pp. 323–347. Springer, Heidelberg.

[121] Winston WL and Goldberg JB, 2004, *Operations research: Applications and algorithms*, volume 3, Thomson Brooks Cole, Belmont, California, United States of America.

[122] Žulj I, Kramer S and Schneider M, 2018, *A hybrid of adaptive large neighborhood search and tabu search for the order-batching problem*, European Journal of Operational Research, **264(2)**, pp. 653–664.

APPENDIX A

A.1 Comparison of different batching metrics on sample picking waves in Durban

The Retailer's DC in Durban is made up of 12 picking lines. Each picking line consists of 56 locations with a conveyor belt in the middle. The 50 sample picking lines presented in Chapter 3 and 4 were recorded in Durban and are available online [79]. The discrete event simulation presented in Chapter 5 has been tested on these sample picking waves. Table A.1 illustrates the comparison between no batching and different batching metrics (random, stops ratio, spans ratio, and stops non-identical spans) in terms of number of cycles, total picking time and pick density measure as described in Section 5.3.

Wave ID	No batch			Random (RAND)			Stops ratio (R-GR)						Spans ratio (A-GS)			Stops non-identical spans (Z-GS)					
	Cycles	Time	d_g	Cycles	Time	d_g	Cycles	Time	d_g	d_m	d_r	d_b	Cycles	Time	d_g	Cycles	Time	d_g	d_m	d_r	d_b
3150	1 156	26 321	18.60	647	20 024	17.32	615	19 837	16.19	0.29	0.25	13.94	635	19 907	16.68	622	19 882	17.27	0.29	0.24	13.29
1065	1 110	27 002	24.19	617	20 237	23.30	586	20 054	23.05	0.34	0.25	14.01	602	20 053	22.49	580	20 031	21.85	0.36	0.26	14.61
2964	1 236	34 241	18.04	651	28 112	16.54	641	28 073	17.28	0.46	0.44	24.55	649	28 064	16.95	640	28 056	16.42	0.47	0.45	25.10
2137	860	21 152	26.92	542	15 533	24.59	462	15 193	25.50	0.33	0.14	7.87	465	15 332	23.77	456	15 166	23.73	0.35	0.15	8.50
2147	848	21 536	16.30	519	16 107	16.85	452	15 875	16.29	0.41	0.20	11.14	476	16 020	16.48	447	15 765	14.77	0.41	0.21	11.50
2968	1 235	33 099	30.78	663	25 765	28.85	647	25 748	28.80	0.44	0.40	22.17	662	25 743	29.05	648	25 619	28.65	0.45	0.42	22.88
1756	1 206	27 021	20.87	662	20 704	20.20	629	20 491	19.33	0.30	0.26	14.41	653	20 498	19.67	638	20 523	19.98	0.29	0.25	13.84
521	1 081	25 705	16.05	620	18 977	16.01	579	18 864	16.15	0.34	0.24	13.14	593	18 852	15.87	571	18 844	15.24	0.34	0.25	13.78
852	1 188	33 338	31.46	645	26 529	30.24	620	26 551	30.00	0.54	0.46	25.21	628	26 478	30.04	620	26 444	29.92	0.55	0.47	26.03
1814	1 274	41 327	12.25	652	34 953	10.77	651	34 835	10.76	0.53	0.53	29.07	648	34 917	10.62	646	34 786	10.01	0.54	0.54	29.61
3142	1 256	32 987	27.66	675	26 223	26.94	648	25 780	25.90	0.45	0.42	22.69	644	25 896	26.37	651	25 817	26.42	0.44	0.41	21.90
2459	1 043	25 738	22.54	605	19 262	21.97	563	19 143	21.90	0.38	0.25	13.24	575	19 099	22.28	556	19 052	20.99	0.39	0.26	13.94
1810	997	25 062	19.39	581	18 857	17.69	538	18 785	17.98	0.40	0.25	13.42	540	18 797	17.48	523	18 742	17.08	0.42	0.26	14.17
1865	1 033	24 913	20.57	605	17 881	20.87	556	17 769	20.01	0.29	0.24	13.03	569	17 672	19.25	558	17 640	18.86	0.30	0.26	13.79
2913	1 327	36 681	28.77	675	30 413	27.21	671	30 384	27.02	0.28	0.29	15.30	670	30 335	26.98	669	30 331	26.92	0.30	0.31	16.02
1813	1 336	45 682	13.18	670	38 111	11.60	666	38 192	12.02	0.44	0.46	24.00	668	38 708	11.17	666	38 085	11.17	0.46	0.48	24.94
1966	1 316	45 199	18.47	658	38 330	16.94	658	38 094	16.45	0.48	0.51	26.64	658	38 265	16.74	658	37 960	16.15	0.49	0.53	27.34
1503	1 219	28 085	24.49	667	21 505	23.00	644	21 314	22.39	0.32	0.31	16.02	658	21 446	22.97	651	21 389	22.97	0.31	0.30	15.50
505	1 314	30 630	28.56	658	24 047	27.00	658	23 795	27.03	0.45	0.34	17.32	658	23 811	26.88	658	23 606	26.84	0.47	0.35	17.88
1868	966	30 077	24.24	575	23 827	22.91	520	23 820	22.48	0.27	0.30	15.30	522	23 825	22.14	513	23 813	21.88	0.28	0.31	15.94
926	1 167	23 791	20.98	639	17 362	20.12	609	17 236	17.33	0.34	0.20	10.02	615	17 207	18.67	605	17 016	19.41	0.35	0.21	10.51
2503	1 054	42 654	13.24	559	36 402	12.65	548	36 146	12.78	0.46	0.47	23.35	549	36 179	12.42	536	36 135	11.43	0.49	0.48	23.93
2569	1 303	37 650	24.73	680	31 591	18.54	671	31 336	18.55	0.50	0.34	17.17	675	31 518	18.44	672	31 269	18.36	0.52	0.35	17.71
1569	1 195	32 463	24.51	652	25 916	19.26	628	25 638	19.19	0.37	0.40	19.79	652	25 656	19.15	627	25 551	18.90	0.38	0.41	20.26
2817	1 363	31 282	20.12	683	24 235	19.03	657	24 162	19.21	0.29	0.31	15.11	682	24 178	18.72	656	24 126	17.65	0.31	0.33	16.03
1460	1 238	39 650	12.59	672	33 470	12.58	650	33 482	13.08	0.50	0.56	27.30	630	33 454	12.37	648	33 476	12.94	0.49	0.54	26.69
779	1 360	31 362	20.17	670	25 004	19.07	682	24 805	20.29	0.39	0.36	17.07	673	24 762	19.61	681	24 963	20.00	0.38	0.35	16.49
1501	1 249	24 538	23.93	681	17 424	22.94	648	17 305	21.76	0.30	0.26	12.05	666	17 456	22.18	657	17 469	22.58	0.29	0.25	11.53
1079	1 110	34 111	16.96	634	27 751	15.92	583	27 751	12.43	0.33	0.39	17.86	594	27 799	14.67	589	27 769	15.43	0.32	0.38	17.25
1080	1 311	26 885	15.41	658	20 219	13.74	657	19 870	13.32	0.21	0.22	10.11	658	19 893	13.40	658	20 003	13.70	0.20	0.21	9.81
2502	419	47 024	8.79	251	40 784	7.13	200	40 615	7.10	0.54	0.64	29.47	238	40 699	7.01	200	40 601	6.68	0.55	0.65	29.98
2550	224	34 858	23.28	167	28 400	22.45	132	28 293	21.95	0.29	0.34	15.36	153	28 261	21.78	122	28 222	18.99	0.31	0.36	16.37
2640	224	39 225	26.36	163	33 016	25.21	131	32 918	25.09	0.32	0.37	16.84	150	32 912	24.92	123	32 909	24.06	0.33	0.39	17.38
2037	179	32 560	22.21	146	26 292	20.04	104	26 239	20.12	0.28	0.33	14.70	137	26 269	20.21	102	26 186	19.99	0.29	0.35	15.20
1605	284	43 860	23.03	162	37 733	19.77	153	37 648	19.86	0.43	0.51	22.53	157	37 583	19.92	148	37 550	19.13	0.45	0.52	23.09
1715	206	34 727	25.25	132	28 023	22.08	93	27 722	20.86	0.34	0.35	15.20	114	28 031	22.10	108	28 110	21.76	0.34	0.34	14.79
704	1 146	30 994	19.77	629	24 582	18.28	602	24 428	18.18	0.34	0.33	14.15	602	24 540	17.88	595	24 264	17.42	0.35	0.34	14.68
1942	1 301	40 092	23.08	661	33 904	23.15	658	33 665	23.20	0.35	0.45	19.40	661	33 674	22.79	660	33 525	21.43	0.36	0.46	19.89
2022	1 175	7 802	15.99	643	6 412	17.81	608	5 860	11.22	0.13	0.06	3.63	616	6 166	16.57	607	5 705	10.82	0.13	0.07	3.68
701	1 030	6 460	27.09	609	5 442	27.51	553	4 879	25.60	0.40	0.06	3.36	583	5 343	27.12	558	4 849	22.85	0.44	0.06	3.53
591	1 316	6 481	13.86	658	5 355	13.14	658	4 869	13.48	0.59	0.08	4.62	658	5 202	13.21	658	4 811	11.85	0.58	0.09	4.83
790	1 293	5 237	29.54	677	4 146	26.36	664	3 871	26.19	0.30	0.07	3.76	661	4 143	26.13	656	3 801	26.05	0.31	0.07	3.93
592	1 326	6 033	16.60	682	4 386	16.27	668	4 300	16.00	0.40	0.08	4.58	677	4 441	14.89	666	4 025	12.67	0.42	0.08	4.64
910	1 145	4 862	22.72	635	3 903	21.81	592	3 612	21.17	0.31	0.07	3.63	604	3 894	22.84	589	3 562	20.98	0.33	0.07	3.81
2710	18	525	24.47	19	443	23.96	10	426	23.24	0.45	0.09	4.71	15	433	23.46	11	424	20.66	0.49	0.10	5.08
669	23	417	13.00	15	378	11.48	14	337	11.80	0.33	0.09	3.84	12	360	12.02	14	334	11.38	0.34	0.09	3.95
517	14	751	21.65	21	685	19.73	12	611	20.19	0.36	0.05	2.20	19	652	16.84	14	596	16.38	0.36	0.05	2.22
867	12	738	19.89	8	631	18.67	7	630	15.66	0.50	0.43	16.00	6	633	18.66	6	628	15.32	0.51	0.50	18.63
515	6	627	16.95	3	546	16.17	3	491	15.43	0.25	0.07	2.36	3	524	13.74	3	475	13.68	0.27	0.07	2.52
3160	19	6 480	1.90	15	5 931	1.54	10	5 948	1.46	0.70	0.87	45.00	14	5 938	1.54	9	5 916	1.46	0.70	0.87	45.00
	46 711	**1 289 933**		**25 451**	**1 035 762**		**24 209**	**1 027 690**					**24 647**	**1 031 517**		**24 149**	**1 025 823**				

TABLE A.1: *Comparison of different batching metrics for 50 sample picking waves in Durban. Walking distance is measured in cycles traversed on the picking line. Time is measured in (s). Density is the average number of locations between picks.*