ORDER PICKING PROBLEM: ITS VARIATIONS AND INTEGRATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SERHAT SAYLAM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
INDUSTRIAL ENGINEERING

DECEMBER 2022

Approval of the thesis:

ORDER PICKING PROBLEM: ITS VARIATIONS AND INTEGRATION

submitted by **SERHAT SAYLAM** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Esra Karasakal
Head of Department, **Industrial Engineering** ________________

Prof. Dr. Haldun Süral
Supervisor, **Industrial Engineering, METU** ________________

Assoc. Prof. Dr. Melih Çelik
Co-supervisor, **School of Management, University of Bath** ________________

Examining Committee Members:

Prof. Dr. Cem İyigün
Industrial Engineering, METU ________________

Prof. Dr. Haldun Süral
Industrial Engineering, METU ________________

Prof. Dr. Oya Ekin Karaşan
Industrial Engineering, Bilkent University ________________

Prof. Dr. Ferda Can Çetinkaya
Industrial Engineering, Çankaya University ________________

Prof. Dr. Sinan Gürel
Industrial Engineering, METU ________________

Date: 09 / 12 /2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Serhat Saylam

Signature :

ABSTRACT

ORDER PICKING PROBLEM: ITS VARIATIONS AND INTEGRATION

Saylam, Serhat
Ph.D., Department of Industrial Engineering
Supervisor: Prof. Dr. Haldun Süral
Co-Supervisor: Assoc. Prof. Dr. Melih Çelik

December 2022, 147 pages

Order picking is the most costly and labour-intensive warehouse activity. The objective of order picking problem is to collect the items on the pick list in a sequence to ensure a route that minimizes the travel time. In both manual and automated warehouses, a combination of efficient zoning, batching and picker routing plays an important role in improving travel time, congestion, workload balancing and system throughput.

In this thesis, we study single-picker and multi-picker order picking problems on single-block, two-block and multi-block warehouse layouts by also considering synchronised dynamic zone-picking and batch-picking decisions. For this end, we present (1) mathematical models for the optimal solutions of some of these problems, (2) exact dynamic programming approaches to find the optimal solution for some other cases, and (3) simple but effective heuristics for the remaining more complex forms.

Computational experiments on randomly generated instances in line with those in the literature show that the proposed approaches can find optimal and near-optimal solutions in negligible computational times. The comparisons of the resulting objective

function values with the ones in the related literature also show that our approaches perform at least as strongly as the models in the state-of-the-art literature. We also contribute to the literature by introducing the arc routing perspective into the solution methodologies of order picking problems, by also introducing disconnectivity elimination constraints instead of sub-tour elimination constraints and by studying zone-picking and batch-picking decisions as operational level problems integrated with picker routing and workload balancing problems.

Keywords: routing, warehouse management, order picking, zone picking, picker routing, arc routing

ÖZ

SİPARİŞ TOPLAMA PROBLEMİ: VARYASYONLARI VE ENTEGRASYONU

Saylam, Serhat
Doktora, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Haldun Süral
Ortak Tez Yöneticisi: Doç. Dr. Melih Çelik

Aralık 2022 , 147 sayfa

Sipariş toplama, en maliyetli ve emek yoğun depo faaliyetidir. Sipariş toplama probleminin amacı, toplama listesindeki malzemeleri olabilecek en kısa sürede toplayan bir rota oluşturmaktır. Hem manuel hem de otomatikleştirilmiş depolarda; bölgeleme, gruplama ve toplayıcı rotalama kararlarının birlikte ele alınması toplama süresini, tıkanıklığı, iş yükü dengesini ve sistem verimini iyileştirmede önemli bir rol oynamaktadır.

Bu tezde, tek bloklu, iki bloklu ve çok bloklu depo yerleşimleri üzerinde tek toplayıcılı ve çok toplayıcılı sipariş toplama problemlerini, senkronize dinamik bölgesel toplama ve grup toplama kararlarını da dikkate alarak incelemekteyiz. Bu amaçla, (1) bu problemlerin bazılarının optimal çözümleri için matematiksel modeller, (2) diğer bazı durumlar için en uygun çözümü bulmak amacıyla kesin dinamik programlama yaklaşımları ve (3) daha karmaşık problemler için de basit ama etkili sezgisel yöntemler önermekteyiz.

Literatürdekilerle uyumlu olarak rastgele oluşturulmuş örnekler üzerinde yapılan hesaplama deneyleri, önerilen yaklaşımların ihmal edilebilir hesaplama sürelerinde optimal veya optimale yakın çözümler bulabileceğini göstermektedir. Sonuçların amaç fonksiyonu değerlerinin ilgili literatürdekilerle karşılaştırılmaları da yaklaşımlarımızın en az literatürde sunulan en yeni modeller kadar güçlü performans gösterdiğini göstermektedir. Ayrıca, hat rotalama perspektifini sipariş toplama problemlerinin çözüm metodolojilerine dahil ederek, alt tur eleme kısıtlamaları yerine bağlantısızlık eleme kısıtlamalarını getirerek ve bölgesel toplama ve grup toplama kararlarını toplayıcı rotalama ve iş yükü dengeleme problemleri ile tümleşik şekilde ve operasyonel seviye problemleri olarak inceleyerek literatüre katkıda bulunmaktayız.

Anahtar Kelimeler: rotalama, depo yönetimi, sipariş toplama, bölgesel sipariş toplama, toplayıcı rotlama, hat rotalama

To my beloved sons and wife

ACKNOWLEDGMENTS

I wish to express my deepest appreciation to my supervisor Prof. Dr. Haldun Süral for his endless support and encouragement. I wish him all the best in the new chapter of his life.

I especially want to thank my co-supervisor Assoc. Prof. Dr. Melih Çelik for his unprecedented guidance, advice, criticism, encouragements, and insight throughout this research. He was more than an advisor.

I would also like to convey my sincere gratitude to my thesis monitoring committee members, Prof. Dr. Oya Ekin Karaşan and Prof. Dr. Cem İyigün, for providing me extremely useful and enlightening knowledge in the field of operations research.

I would also like to express my special thanks to my friends, Dr. Arsham Atashi Khoei and Mehmet Mert Onağ, for their endless support during this long journey.

Above all I would like to thank my wife, Dr. Rabia Saylam, for her unconditional love and support, for believing in me and not letting me give up. But most of all, thank you for being my best friend. You mean the world to me!

I hope this work inspires my sons, Mustafa Mert and Kemal Yiğit, to do more in the future to make the world a better place. Your creativity, passion, and energy can make a difference in the world. I am so proud of you and I know that you can do great things!

I will always feel privileged to be a graduate of METU.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ALGORITHMS

ALGORITHMS

LIST OF ABBREVIATIONS

ABBREVIATIONS

BIP	Binary integer programming
DP	Dynamic programming
LKH	Lin–Kernighan–Helsgaun
MILP	Mixed integer linear programming
OPP	Order picking problem
RR	Ratliff and Rosenthal
TSP	Travelling salesperson problem
VRP	Vehicle routing problem

CHAPTER 1

INTRODUCTION

In this thesis, we aim to present compact mathematical models and exact/heuristic algorithms for various order picking problems, which cover the most costly and labor-intensive warehouse activities. In order picking operations, items must be collected from the warehouse in order to satisfy the customer demand while aiming a cost and/or service-related objective (De Koster et al. 2007, van Gils et al. 2018b, Masae et al. 2020a). The objective of order picking problem (OPP) is to collect the items on a pick list in a sequence to ensure a route that minimizes the travel time, which can be formulated as a special case of the travelling salesperson problem (TSP) or Steiner TSP (Theys et al. 2010, Scholz et al. 2016, Pansart et al. 2018). Moreover, a growing competition with limited time windows puts extra pressure on order picking operations. Reducing traffic congestion through one-way routing approaches, zone-picking and workload balancing are some ways of dealing with such problems. Domination of online retailing has resulted in many relatively small-size orders with promised time windows. Hence, increasingly tight time windows make it necessary to give effective zone picking, workload balancing and picker routing decisions in short computing times. Orders, online or in-store, arrive continuously. However, decision makers release a large group of orders in a wave in order to take advantage of economies of scale in order picking operations (Çeven & Gue 2017). Besides the economic interpretation of order picking operations, in times of economic crises or pandemics, managers should be able to see a better operational picture under constraints such as limited order pickers, personal space of pickers, etc. Thus, given a number of order pickers, the decision makers should make efficient decisions for operational activities regarding these order pickers. In this study, we present solution methodologies for such enabling problems.

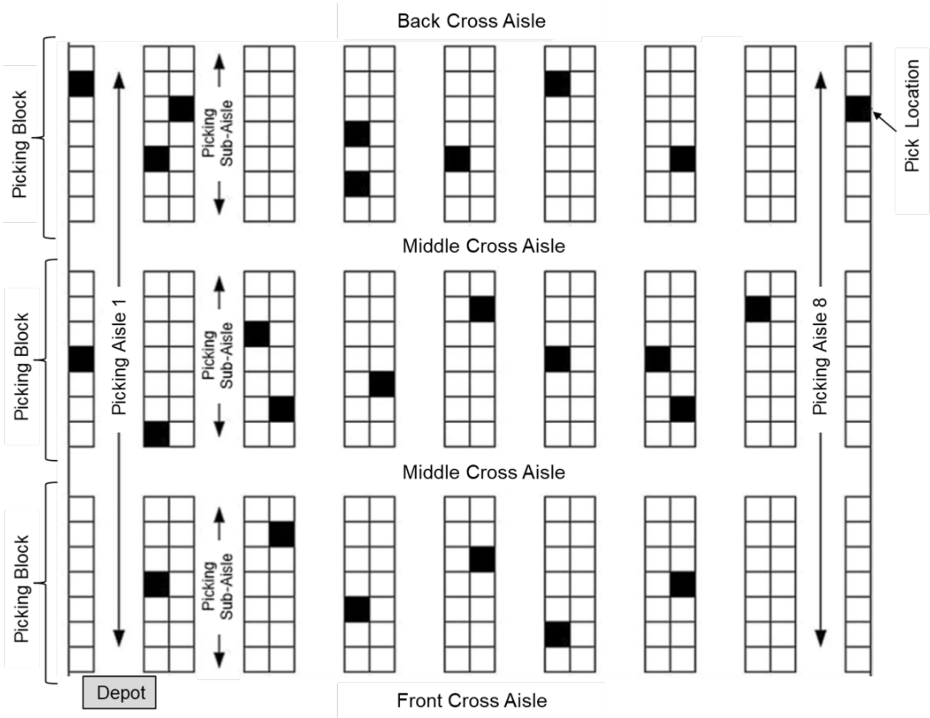


Figure 1.1: A parallel-aisle warehouse layout with 3 blocks, 8 picking aisles and 25 pick locations, Çelik & Süral (2019)

The parallel-aisle warehouses considered throughout the thesis is given in Figure 1.1, which consists of narrow *picking aisles* parallel to one another. It contains *cross-aisles* at the *front* and *back* ends of picking aisles and may also contain *middle cross-aisles*, which perpendicularly divide the picking aisles into equal-length *picking sub-aisles*. We assume narrow picking aisles so that a picker spends no time when making horizontal movements within a picking sub-aisle. A picker starts the tour from the *depot* (also known as the pickup-and-deposit or P&D point), collects all the items in the pick list and returns to the depot. Without loss of generality, the depot is assumed to be located at the left front corner. Figure 1.2 shows the graph representation of the warehouse along with the pick locations using nodes v_q where $q \in N$ while v_0 is the depot. a_j and b_j nodes represent the back and the front intersection points of picking aisle $j \in M$. Node m_{jk} represents the intersection point between picking aisle $j \in M$ and middle cross-aisle $k \in C$. Also we note that a back cross-aisle of a block is the front cross-aisle for the next block.

In general, the OPP is modelled as a variant of standard TSP problem, which is $\mathcal{NP}$-

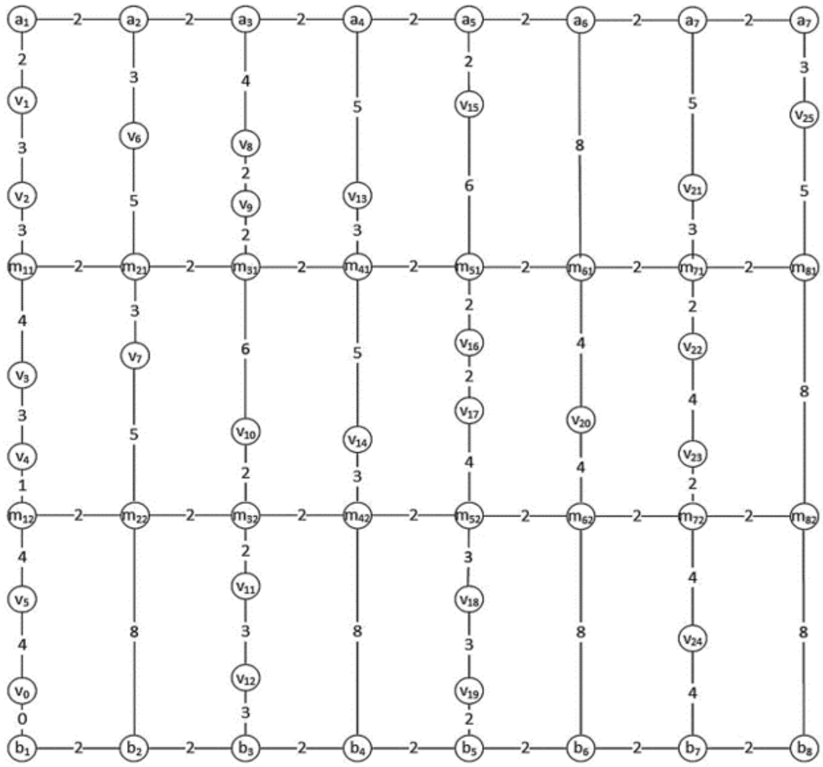


Figure 1.2: The graph representation of Figure 1.1, Çelik & Süral (2019)

hard (Karp 1972), and the data consisting of the pick locations is pre-processed into a standard TSP distance matrix format. In such a case, the complexity of calculating the minimum order picker route exponentially increases with the number of pick locations. The lack of formulations which take into account the special properties of parallel-aisle warehouse layout in the literature motivates us to study the different variants of the OPP by exploiting the properties of a parallel-aisle warehouse layout. Making use of these specific properties and generating aisle-specific distance matrices make it possible to solve the OPP more efficiently. This approach is mostly ignored for the case of mathematical models or such constraints are developed upon a TSP-based formulation. Since travel time is a function of distance, keeping the walking speed fixed and minimizing the walking distance is proposed by many authors as the main factor to optimize the total picking time (e.g., Hall 1993, Vaughan & Petersen 1999, Roodbergen & De Koster 2001a, De Santis et al. 2018, Çelik & Süral 2019). In this regard, we continue with time-units instead of distance-units in the computational experiments throughout the thesis.

1.1 The Outline of the Thesis

In this thesis, we study single-picker and multi-picker OPP on single-block and multi-block warehouse layouts by also considering zone-picking and batch-picking decisions. We present mathematical models, exact dynamic programming approaches, and heuristics. Afterwards, we test the performance of the proposed approaches and explain their strengths and aspects that are open to development. The comparisons of the results with the ones in the related literature also show that our approaches perform at least as strong as the models in the state-of-the-art literature. We also contribute to the literature by introducing the arc routing perspective into the solution methodologies of OPP, by also introducing disconnectivity elimination constraints instead of sub-tour elimination constraints and by studying zone-picking and batch-picking decisions as operational level problems integrated with picker routing and workload balancing problems. Table 1.3 depicts the scope and distribution of this thesis.

In Chapter 3, we study the single-picker OPP in single-block layouts as a variant of the arc routing problem. For this end, we present a binary integer programming (BIP) formulation which exploits the special properties of a parallel-aisle warehouse layout. This formulation only depends on the number of aisles, as opposed to the number of pick locations, which is generally the case of its counterparts in the literature. The OPP can be modelled as an arc routing problem since we search for a strongly connected closed walk of minimum length. The focus of our approach is to clear all picking aisles in the shortest time possible. To do that, we assign the best combination of intra-aisle movements and complementing cross-aisle movements such that it results in the minimum length strongly connected closed walk. Afterwards, we test the performance of the formulation by comparing it with the ones of recent literature. In this formulation, we introduce the *disconnectivity elimination constraints* into the literature instead of TSP sub-tour elimination constraints, where we can ensure a feasible order picking tour with a much smaller number constraints, which significantly increase the computing time performance.

In Chapter 4, we show that the proposed arc routing-based formulation can be straightforwardly extended to the layouts with a middle cross-aisle. In this regard, we focus on two-block layouts and propose a mathematical model which forces a strongly con-

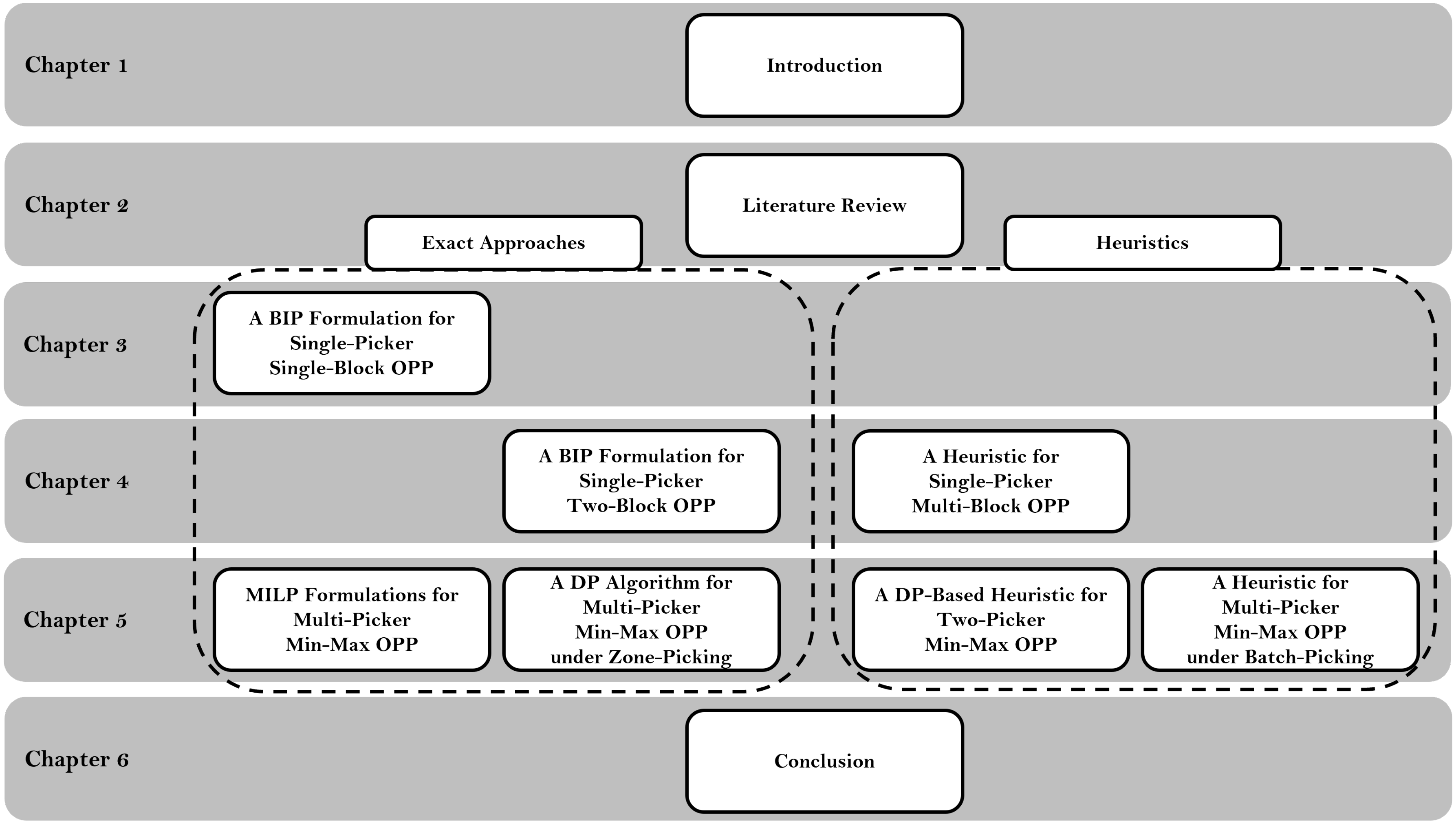


Figure 1.3: The outline of the thesis

nected closed walk using the disconnectivity elimination constraints approach presented in the preceding chapter. In both formulations, we mainly apply three different classes of constraints. Firstly, we assign a movement to each aisle in a feasible sequence, then we ensure that each vertex has in-degree equal to out-degree. Lastly and most importantly, we eliminate the disconnectivities. In warehouse layouts with more than two blocks, the first two classes of constraints still apply similarly. However, one should focus on how to modify the disconnectivity elimination constraints as such occurrences increase exponentially with the number of cross-aisles. Hence, for more than two-block layouts, we have introduced a simple and effective picker routing heuristic and leave the development of a multi-block OPP formulation as a future research.

In Chapter 5, we focus on the multi-picker OPP and solve an integrated zone-picking, picker routing and workforce allocation problem by minimizing the lead time of the pick wave. We focus on zone-picking since there is a clear gap in integrated OPP literature in terms of zone picking operations. Although recent literature regarding multiple order pickers has increased significantly, the advantages of zone picking are ignored as zone picking combinations have not been given any particular attention. Minimizing the lead time of the pick wave forces workload balancing over the zones so that workforce allocation is also achieved. It has the advantages of zone-picking such as assignment of a smaller area for each order picker, reduced traffic congestion, reduced idle times of the pickers, and familiarization with products. It is dynamic in the sense that the assigned zones differ continuously at each order picking activity. Also, it is important to note that waiting times due to congestion can strongly decrease by considering dynamic zone-picking. The study in this chapter is divided into four parts. We first present mixed integer linear programming (MILP) formulations for the multi-picker OPP with a min-max objective including zoning constraints. Following this, we focus on the min-max OPP with two-pickers. We aim to find the shortest travel distances in order to minimize the lead time of two-picker picking process by assigning adjacent aisles to pickers using a dynamic programming algorithm. Then, applying a difference minimization algorithm, we reduce optimality gap at the expense of zone integrity. The study, without applying the difference minimization algorithm, is an example of dynamic zone-picking where each of two pickers is assigned to a specific zone of aisles. Lastly, we focus on the min-max OPP with multiple pickers under synchronized dynamic zone-picking systems. Here, we generalize the dynamic zone-picking approach for multiple order pickers by applying a variant of VRP formulation with introducing the aisle-zone related constraints and a novel dynamic programming approach. Using a dynamic zone-picking approach, we present an exact algorithm extending two-picker OPP to more order pickers by assigning multiple pickers to dynamic zones and minimize the arrival time of the latest picker, i.e., pick wave. Contrary to traditional parallel zone-picking approach, there are no dedicated zones assigned to pickers. Pickers collect items in zones assigned to them at each pick wave. Hence, the zone-picking problem, which is largely studied at tactical level, is reduced to an operational level decision integrated with picker routing and workforce allocation problems. In this way, we solve an integrated zone-picking, picker routing and workforce allocation problem where each zone consists of a certain number of aisles. While the presented mathematical model solves the problem for a given number of pickers, the proposed algorithm solves the problem for each picker and gives a faster picking scheme to the decision makers. As an alternative to zone-picking, in Section 5.6, we consider the min-max OPP under batch-picking policy and present a saving algorithm. Finally, we consider examples to illustrate the algorithms and compare the performance of the solution methodologies in different computational experiments.

1.2 Contributions of this Thesis

As discussed in the literature review section, the OPP literature mainly focuses on polynomially-solvable algorithms since it is assumed that the OPP is closely related to the $\mathcal{NP}$-hard TSP. One of the main contributions of this study arises at this point. Since the picker routing problem in a single-block parallel-aisle warehouse can be solved in polynomial time using dynamic programming approach, then there could be a way to formulate it in a compact way by taking advantage of the movement types introduced by Ratliff & Rosenthal (1983). This is also important in the sense that the mathematical formulations can accommodate side constraints or feasible routes, which can be a part of a combined formulation while these modifications are less likely over the algorithms. The computing time performances of the formulations are significantly short thanks to the one of our main contribution, which is the use of disconnectivity elimination constraints. We also contribute to the literature by introducing the arc routing perspective into the solution methodologies of the OPP. Another contribution of our BIP formulations into the literature is that a feasible order picking tour without disconnectivity elimination constraints is rarely disconnected, hence the use of lazy constraints significantly increases the computing time performance of the formulation. As a result, the OPP turns out to be an appropriate research area for lazy constraints applications. The work in Chapter 4 is especially important in the sense that it paves the way for an arc-routing based OPP formulation for a general number of blocks by induction on the construction of disconnectivity elimination constraints.

The main difference of our work in Chapter 5 from the literature is that while the previous studies consider a static zoning environment, our study focuses on dynamic zoning environment. To the best of our knowledge, there are very few studies considering dynamic zoning strategy (Bartholdi & Eisenstein 1996, Bartholdi et al. 2001, Ho & Liao 2009, Lamballais et al. 2022) where zoning decisions are handled at operational level. The main contributions of this chapter can be stated in four dimensions. First, we define and solve an integrated zone-picking, picker routing and workload balancing problem by minimizing the lead time of the picking wave and applying the policy of synchronized dynamic zone-picking where each picker is assigned to a zone of aisles at each picking wave. For this end, we present an exact polynomial-time DP algorithm, which may be implemented in many real-world warehousing environments and significantly improve performance. Second, for the first time in the literature, the zone-picking approach is studied at operational level under a synchronized dynamic zone-picking policy. Third, this study is one of the few studies which investigate the multi-picker multi-block OPP. Last contribution of our study is that it examines the optimal dynamic zone sizes to increase the system throughput.

We also contribute to the warehouse operations in practice as our solution methodologies help reduce the congestion on the cross-aisles by resulting to one-way movements, avoid congestion within the picking aisles, ensure familiarity of pick locations and spontaneously lead to the balanced partition of the pickers' workload thus reduce the management supervision.

CHAPTER 2

LITERATURE REVIEW

In this section, the most relevant OPP literature is reviewed. There exists extensive literature on warehouse management problems. In the first section, we give brief explanations of existing literature reviews on warehouse operations. Then, we focus on each of these operations which are mostly related with our study. To do that, we aim to highlight the contributions of the studies, focusing especially on picker routing, batching, and zone-picking problems. Finally, we review the literature regarding integrated order picking operations, which focuses on the combinations of different warehouse operation problems.

2.1 Existing Reviews on Warehouse Operations

There is an extensive literature on warehouse problems covering strategic, tactical, operational levels. Reviews classifying warehouse problems include Rouwenhorst et al. (2000), De Koster et al. (2007), Gu et al. (2007), van Gils et al. (2018b), Masae et al. (2020a), and Vanheusden et al. (2022b).

Rouwenhorst et al. (2000) review the existing literature on design and control problems. It is suggested that a joint analysis of various design methodologies is largely required since strategic level problems are highly interrelated. Hence, the literature should focus on clustering the relevant strategic level problems that are possible to solve simultaneously. The authors also claim that the integrated modelling is required at a lesser extent in tactical level problems while the operational level warehouse problems can be handled independently. This is because of the fact that the constraints for the operational level problems are set at strategic or tactical levels and interactions between different processes are typically handled during these levels.

Therefore, the authors suggest that operational level policies would not be required to interact with other same-level policies as long as it is considered at a higher level. As a result, they recommend a more design-oriented research approach.

For another detailed review of warehouse order picking operations, the reader is referred to De Koster et al. (2007). This study gives a literature overview on typical decision problems in design and control of manual order-picking processes focusing on the literature on layout design, storage assignment methods, batching/zone-picking, and routing methods. It is the most comprehensive review regarding order picking objectives and their improvements through various zone-picking, batching and storage assignment approaches. Another literature review focusing on warehouse operations covering functions from receiving to shipping is the study of Gu et al. (2007). Especially batching literature in order-picking activities are explained clearly in this study while it is concluded that zoning accounts for less than 6% in the surveyed literature.

van Gils et al. (2018b) review and classify the most recent literature regarding the integration and interaction of the tactical and operational warehouse problems (e.g., order batching, picker routing, storage assignment, zoning). Authors also note that 75% of the articles on integrated problems have been published in the last decade. The scope is limited with the combinations of storage assignment and batching, storage assignment and routing, batching, and routing, workforce level and batching, workforce level and routing, job assignment and batching, job assignment and routing, job assignment and zone picking, and batching and sorting. The authors mention that zoning and workforce related problems are the ones that have drawn the least attention in the literature. The reader is referred to the appendix of van Gils et al. (2018b), which depicts a brief description for each of the activities related to order picking operations.

Different from the other literature reviews, Masae et al. (2020a) specifically and extensively focus on reviews only regarding the picker routing problems. The solution methodologies are categorized and examined based on the warehouse types (conventional or non-conventional), the number of blocks (single-block or multi-block), and the type of the algorithms (exact, heuristics or meta-heuristics). More recently, Vanheusden et al. (2022b) review the literature to highlight the main practical factors in the order picking operations to fill the gap between the research and practice as the practitioners hardly implement the findings from the researches.

2.2 Picker Routing

Our study is mostly relevant to the routing and zone-picking literature. Picker routing or order picking problem is the mostly studied warehouse operation problem. Due to the importance of order picking on the overall warehouse performance, there exists a vast amount of literature on order picking. van Gils et al. (2018b), in their review, conclude that a significant portion of the warehouse order picking literature addresses picker routing as the main problem or as a subproblem of an integrated problem. The significance of picker routing in order picking operations underlines the importance of formulating an efficient integer programming model for the OPP.

Regarded as the seminal work in the order picking literature, Ratliff & Rosenthal (1983) introduce an exact dynamic programming (DP) algorithm (further referred to as the RR algorithm) for the single-picker single-block OPP that runs in $\mathcal{O}(|M|+|N|)$ time, where M is the set of picking aisles and N is the set of pick locations. In this algorithm, The depot is assumed to be at the very left front corner and the aisles constitute the stages. Each aisle has two sub-stages denoted by L_j^- and L_j^+. Sub-stage L_j^- contains the nodes a_j and b_j together with all nodes and minimum tour construction edges at the left of the graph. Sub-stage L_j^+ additionally contains all pick locations and minimum tour construction edges in aisle j. The possible connections within and between the picking aisles form the seven different states, called *equivalence classes*. These states are denoted by their (i) degree parity at a_j, (ii) degree parity at b_j, and (iii) the state of connectivity. Possible degree parities are zero (0), even (E) or odd (U), while the state of connectivity can be $0C$, $1C$ or $2C$. Then, any graph in a sub-stage L_j can be represented by an equivalence classes denoted by $(U, U, 1C)$, $(E, 0, 1C)$, $(0, E, 1C)$, $(E, E, 1C)$, $(E, E, 2C)$ and $(0, 0, 1C)$. These states are updated through stages using the possible connections between states called connection types. Ratliff & Rosenthal (1983) show that (i) there are six possible connection types within an aisle (aisle $j-1$) which connect the states at stage L_{j-1}^- with the states at stage L_{j-1}^+ and (ii) there are five possible connection types between two neighboring aisles (aisles $j-1$ and j) which connect the states at stage

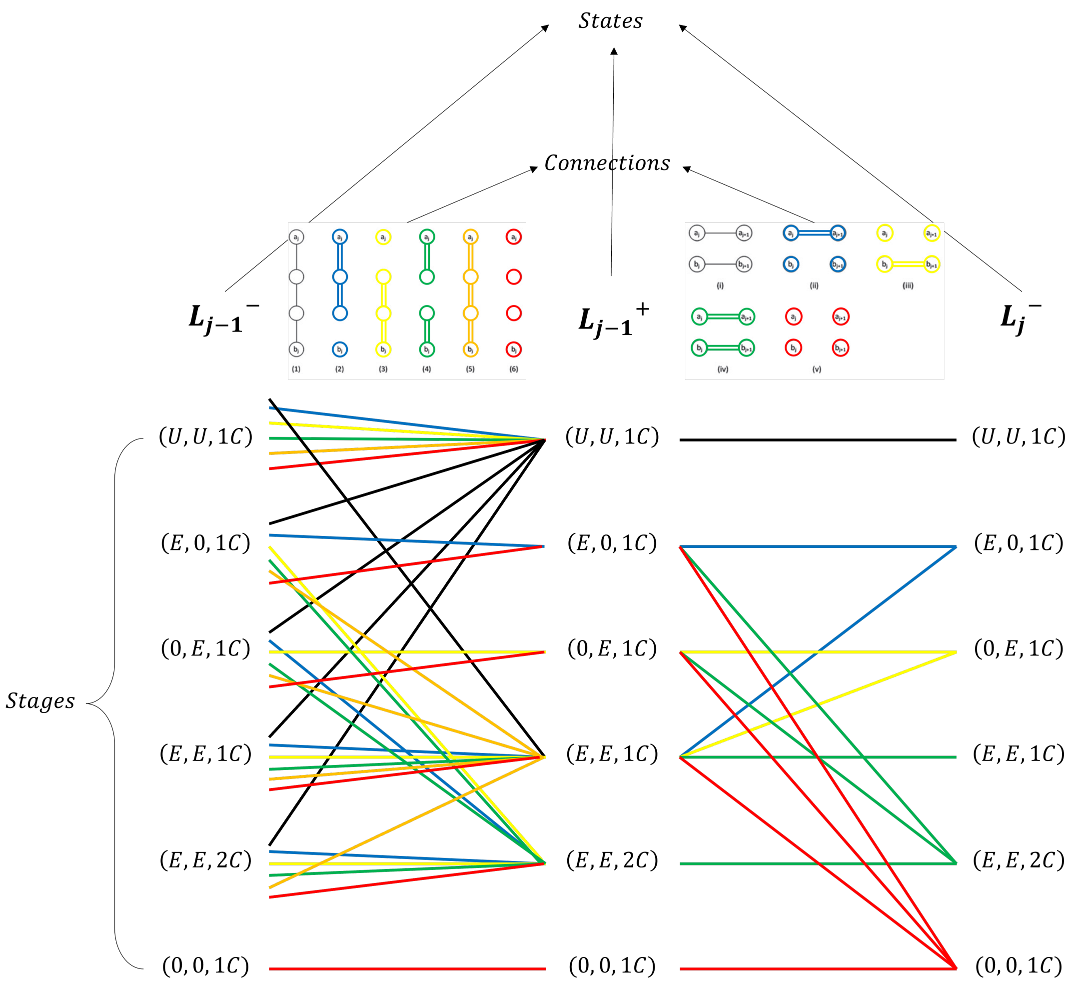


Figure 2.1: The visual representation of the RR algorithm.

L_{j-1}^{+} with the states at stage L_j^{-}. The pick locations and the intersection points between the picking aisles and the cross-aisles determine the length of these possible connections at each stage. The RR algorithm works by forming these states up to the last stage and choosing the shortest travel time solution among the final states satisfying the minimum length completion requirement. The flow of the algorithm in terms of connections among the states at each stage is depicted in Figure 2.1.

Due to the importance of order picking on the overall warehouse performance, the OPP is widely studied. The RR algorithm forms the baseline for the exact solution algorithms of different OPP extensions, including multiple blocks (Roodbergen & De Koster 2001b, Pansart et al. 2018), non-traditional warehouse layouts (Çelik & Süral 2014, Masae et al. 2020c), turn penalties (Çelik & Süral 2016), with multiple pickers using pick waves (Saylam et al. 2022), or with various side constraints (Chabot et al. 2017, Zulj et al. 2018).

Roodbergen & De Koster (2001b) extend the RR algorithm and propose an exact DP approach for the two-block case by increasing the size of the equivalence classes to 25. This substantial increase is due to the fact that the number of possible configurations between two consecutive picking aisles increases from four to fourteen for the layouts where a middle cross-aisle exists. It is also concluded that further extending this algorithm for more than two-block case is difficult. Pansart et al. (2018) propose a DP approach that can exactly solve instances up to five blocks by directly applying the rectilinear TSP algorithm proposed by Cambazard & Catusse (2018) to the case of multi-block OPP. For fish-bone layouts, Çelik & Süral (2014) introduce an exact linear-time algorithm making use of transformation from a graph of a fish-bone layout to a graph of two-block parallel identical aisle warehouse as proposed by Roodbergen & De Koster (2001b). For chevron warehouses, an optimal DP routing algorithm is proposed by Masae et al. (2020c) based on the concept of the graph theory. However, it is also concluded that original two-block picker routing outperforms the chevron warehouse-routing, especially for large number of picking items. As an extension, Çelik & Süral (2016) study the effect of turns on the travel time calculations and show that graph-based heuristics can be modified to take the number of turns into account to travel time minimization including turn penalties. For different layouts, the authors introduce several solution approaches.

The relevant literature also contains several simple heuristics such as the *S-shape, largest gap, return,* and *composite* heuristics due to the complicated nature of the optimal picker routes and the difficulty of their implementation. The S-shape and largest gap heuristics are proposed and analysed by Hall (1993) for the single-block layout. Similarly, return and composite heuristics are put forward by Petersen (1997). Heuristic methods are also proposed for the multi-block layout case (Vaughan & Petersen 1999, Roodbergen & De Koster 2001a, Theys et al. 2010, Çelik & Süral 2019), and the OPP in conjunction with storage replenishment (Çelik et al. 2022) or energy minimization (Atashi Khoei et al. 2022).

Vaughan & Petersen (1999) develop an *aisle-by-aisle* heuristic for multi-block layouts, where the picker visits every aisle exactly once. In this heuristic, the order picker starts from the depot, which is located at the very left bottom corner, and reaches the leftmost picking aisle with a pick location (filled picking aisle). In the DP algorithm, there are $|M|$ stages where M is the set of picking aisles starting from the leftmost filled picking aisle. Each stage has $|B| + 1$ states where B is the set of picking blocks. There are $|B|+1$ different cross-aisles, thus states, to clear pick locations in the leftmost filled picking aisle and to move to the next picking aisle. Each of $|B| + 1$ cross-aisles in the current picking aisle connects to other $|B| + 1$ cross-aisles of the following picking aisle. Using the cross-aisles leading to the minimum travel distance, the order picker clears all the picking aisles and moves to the front cross-aisle of the rightmost picking aisle. In this way, the order picker actually visits every picking aisle exactly once. Finally, the order picker returns to the depot.

Roodbergen & De Koster (2001a) extend the S-shape and largest gap heuristics to multi-block layouts. In S-shape heuristic, the order picker starts from the depot, firstly enters into the leftmost filled picking aisle in order to reach the front cross-aisle of farthest block. Then, the order picker traverses each filled picking sub-aisle of the farthest block with a possibility of a return movement for the last picking sub-aisle if the number of filled picking sub-aisles for the current block is odd. The order picker then clears the next block through S-shape traverses starting from the closest filled picking sub-aisle either the rightmost or the leftmost filled picking sub-aisle (except the leftmost picking aisle). Finally, the order picker returns to the depot after clearing the closest block to the depot. In the largest gap heuristic, the order picker, again, starts from the depot, enters into the leftmost filled picking aisle and reach the front cross-aisle of farthest block. Then, the order picker traverses the first filled picking sub-aisle and reach the back cross-aisle. In this heuristic, using the largest gap policy, all the filled picking sub-aisles are divided into back and front halves. According to the best largest gap movement, the order picker firstly clears all back halves of the farthest block then traverses to the front cross-aisle of the farthest block using the very last filled picking sub-aisle and clears the remaining pick locations through the front halves. Then, the order picker moves to the closest filled picking sub-aisle of the next block, either the rightmost or the leftmost filled picking sub-aisle, and clears this block using the same largest gap strategy. Finally, the order picker returns to the depot after clearing the closest block. We note that, in the largest gap heuristic, the picker travels the middle cross-aisles twice.

In addition to the extensions of S-shape and largest gap heuristics to multi-block layouts, Roodbergen & De Koster (2001a) also introduce the *combined* and *combined+* heuristics, and propose some improvements which are also applicable to S-shape and largest gap heuristics. In these heuristics, the order picker starts from the depot, enters into the leftmost filled picking aisle and reach the front cross-aisle of the farthest block. Then starts the DP algorithm. For the farthest block, the number of stages is equal to the number of picking sub-aisles starting from the leftmost filled picking sub-aisle. For the remaining blocks, the number of stages is equal to the number of picking sub-aisles starting (or ending) after the leftmost filled picking aisle. At each stage, there are two states, the one ending at the back cross-aisle and the one ending at the front cross-aisle. The connection types connecting the stages and forming the states are the same as the ones in RR algorithm except the largest gap type movement. For example, a front cross-aisle ending state is formed either via a traversal movement (cross-aisle shifting movement) connecting the previous back cross-aisle ending state or a return movement (cross-aisle keeping movement) connecting the previous front cross-aisle ending state. At the last picking aisle of each block, the order picker should be at the front cross-aisle which would yield to the minimum tour length for the current block. At the end point of each block, the order picker moves to the closest filled picking sub-aisle of the next block, either the rightmost or the leftmost filled picking sub-aisle, and clears this block using the DP algorithm. Finally, the order picker returns to the depot after clearing the closest block. This combined heuristic is improved with two simple rules to form combined+ heuristic. The first improvement is that the entering picking aisle for the closest block should be the rightmost filled picking sub-aisle. It is to reduce the size of return movement back to the depot. The second one is that the farthest block should not necessarily be reached through the leftmost filled picking aisle. Roodbergen & De Koster (2001a) also note that these improvements could be added to the S-shape and largest gap heuristics as well. In Figure 2.2, we give some heuristic solutions for the instance in Figure 1.1.

Çelik & Süral (2019) propose a *merge-and-reach* heuristic for multi-block layouts by taking advantage of the parallel-aisle property of the rectangular warehouses and

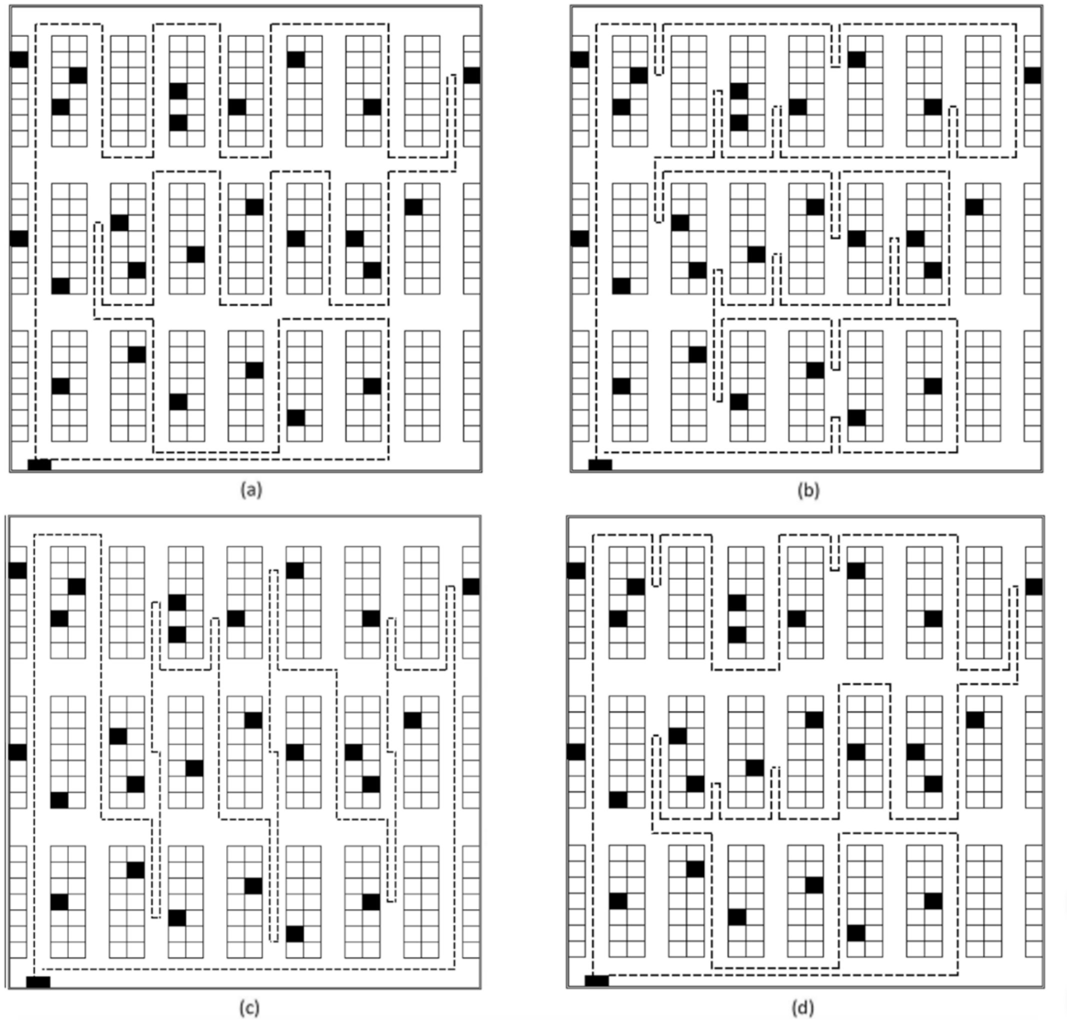


Figure 2.2: Single-picker heuristic solutions for the instance in Figure 1.1. (a) S-shape heuristic solution. (b) Largest gap heuristic solution. (c) Aisle-by-aisle heuristic solution. (d) Combined heuristic solution (adapted from Çelik & Süral (2019))

show that their heuristic outperform the above-mentioned heuristics in terms of solution quality. Different than the heuristics above, which make use of the parallel-aisle properties of a warehouse layout, Theys et al. (2010) suggest the direct use of the Lin–Kernighan–Helsgaun (LKH) TSP heuristic (Helsgaun 2000) to solve the OPP which results in more efficient solutions than previous heuristics. In this algorithm, the authors pre-process the data, omit Steiner points, and then directly apply the LKH heuristic. There are also several metaheuristic approaches that involve the OPP as a direct problem or a subproblem of a combination of multiple order picking planning problems (Tsai et al. 2008, Chen et al. 2013, 2015, Lin et al. 2016, Chen et al. 2016, De Santis et al. 2018, Chen et al. 2019, Ardjmand et al. 2019).

In the OPP, the objective is to visit each of the pick locations on the pick list at least once and in a sequence that minimizes the travel time. Hence, the OPP is indeed a special case of the TSP on a specific graph structure. The OPP is also a special case of the Steiner TSP, where a subset of vertices (*Steiner vertices*) are not necessarily required to be visited in the tour. Such a parallel-aisle warehouse is depicted in Figure 2.3. The Steiner vertices (white vertices) in the case of the OPP correspond to the a_j, b_j and m_{jk} vertices. The vertices required to be visited (black vertices) are the pick locations and the depot. Consequently, the TSP formulations (e.g., Miller et al. 1960, Gavish & Graves 1978, Claus 1984) and the Steiner TSP formulations Letchford et al. (2013) can be directly used to model the OPP, albeit inefficiently.

A number of studies work on modifying the TSP and Steiner TSP formulations by taking advantage of the properties of parallel-aisle warehouse layouts in order to obtain a more efficient formulation. To our best knowledge, the first such work is by Scholz et al. (2016) for the parallel-aisle single-block warehouse. In this work, the warehouse graph is modified by redefining the vertex set, so that the formulation is independent of the number of pick locations. Next, a modified Steiner TSP formulation of Letchford et al. (2013) is applied by using the subtour elimination constraints of Gavish & Graves (1978) as the basis and adding new constraints to reflect the specific properties of the parallel-aisle warehouse layout. The size of the model depends only on the number of picking aisles. The authors show that this formulation leads to faster computing times than the corresponding TSP formulations. The formulation consists of new constraints added onto a TSP-based formulation, which take into account the

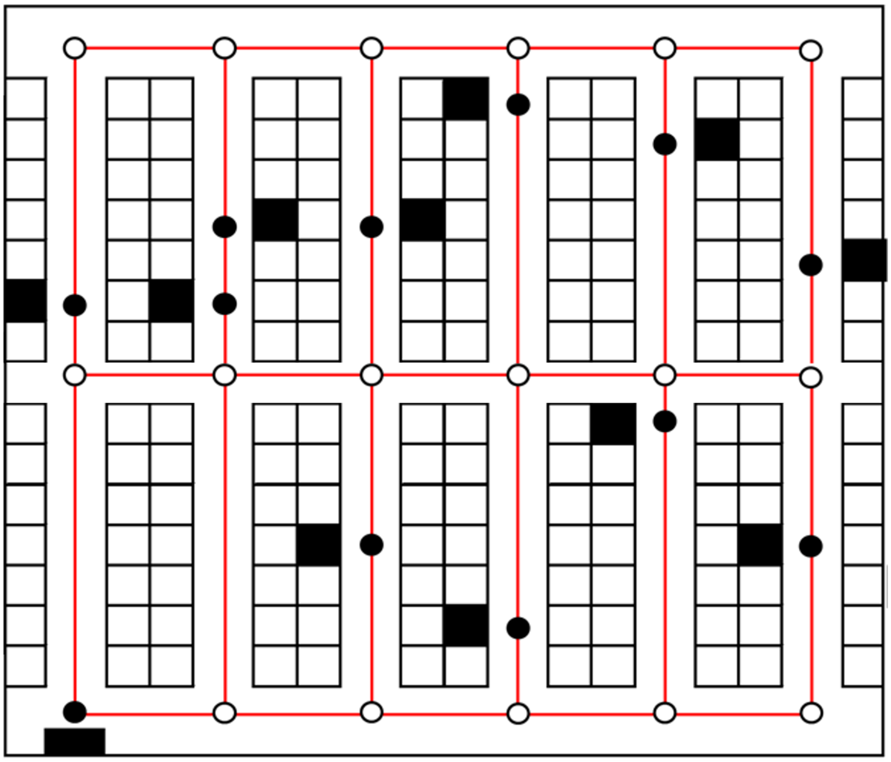


Figure 2.3: Illustration of the OPP as a Steiner TSP

special properties of parallel-aisle warehouse layout.

Similar TSP-based formulations exist for the two and three-block layouts (Ruberg & Scholz 2016, Scholz 2016) or a more general number of blocks (Pansart et al. 2018, Su et al. 2022). Ruberg & Scholz (2016) and Scholz (2016) both extend the single-block formulation introduced in Scholz et al. (2016) to the case of multiple blocks. Pansart et al. (2018) propose a TSP-based MILP formulation for the multi-block OPP where the model is developed by preprocessing the TSP graph in order to reduce the number of vertices and edges and including additional constraints into the single commodity flow formulation of Gavish & Graves (1978), which are resulted from the specific properties of a parallel-aisle warehouse. Su et al. (2022) propose another multi-block formulation built on the single commodity flow formulation proposed by Gavish & Graves (1978) where the authors consider the picking aisles as units. The authors conclude that their approach proves optimality in shorter computing times than the previous studies in the multi-block OPP literature.

To the best of our knowledge, there is only a single study in literature formulating the compact OPP formulation with no use of TSP formulation as the basis. Directly exploiting the properties in RR algorithm, Goeke & Schneider (2021) propose a compact formulation by stipulating that (1) a feasible OPP tour has no more than two connected components, (2) the isolated subtours are prevented as long as a single component is forced at the end of the completion of a tour, and (3) two consecutive picking aisles can only be connected using four possible configurations. The authors show that the proposed approach outperforms its counterparts in the literature and solves large instances within short computing times. Despite its computational efficiency, the formulation by Goeke & Schneider (2021) makes use of the number of possible moves between two consecutive picking aisles, which increases exponentially with the number of blocks. As this would lead to a substantial increase in the number of variables, an extension of this formulation to multiple blocks would not be straightforward or computationally efficient. In other words, the extension of this formulation to more blocks would be difficult if not impossible because of the fact that the number of possible configurations between two consecutive picking aisles, which are defined as decision variables in their study, increases from 4 to 14 for the layouts where a middle cross-aisle exists Roodbergen & De Koster (2001b). This would make it intractable to ensure connectivity through constraints on a progressive OPP tour, hence lead to a more complicated model formulation.

Masae et al. (2022) analyse the impacts of order size, depot location, picker-routing heuristic and storage assignment policies in the order picking time through simulation. Hence, we also note that the performance of the solution approaches for the OPP is largely dependent on tactical level decisions such as storage assignment policies such as random or turnover-based storage decisions. Some of the turnover-based storage policies are shown on Figure 2.4. Since it is beyond the scope of this study, we do not go into details. To the best of our knowledge, there is not an exact solution methodology that solves the OPP with an arbitrary number of blocks.

2.3 Batching

To improve time performance of order picking operations, batching and zone-picking methods are necessary to be considered. If the number of ordered items is relatively small and the number of orders is large, it is inevitable to partition orders into batches

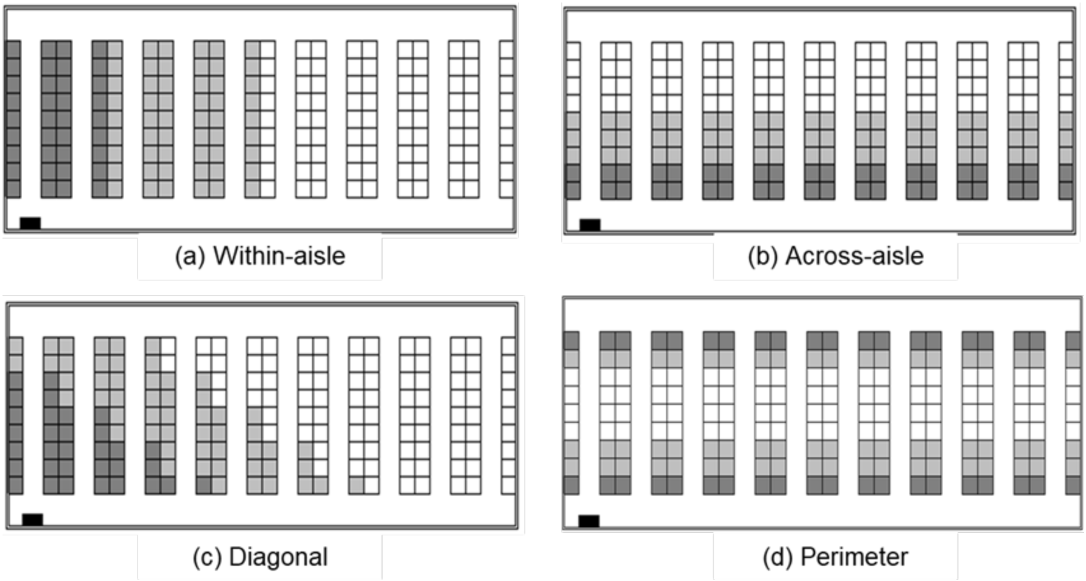


Figure 2.4: Examples of turnover-based storage location assignment policies for a single block warehouse

in picking operations. Similarly, if the number of ordered items is relatively large, it is again inevitable to group items of orders with respect to their zones. Most of the time, orders are not allowed to split into batches while zone-picking requires order split since most of the orders are spread around the picking area. If necessary, batching can also be introduced within zones to more improve the performance, however order integrity would no longer hold, therefore, would not be a constraint. But it is important to note that partitioning a given set of orders into batches more complicates the problem since partition itself is known to be $\mathcal{NP}$-complete (Gademann et al. 2001). Parikh & Meller (2008) study the batch versus zone problem by analyzing their impacts on picking rates, aisle-blocking, workload balancing goals and consolidation/sorting requirements. For this end, the authors propose a cost model, observe the estimated costs on a real-world example and conclude that workload balancing is more ensured by batch picking systems.

The order batching problem is to determine a set of orders to be partitioned into batches such that a specific objective is optimized (Gu et al. 2007, Henn & Wäscher 2012, Scholz et al. 2017). In batching operations, sorting is assumed to be completed on the way using sort-while-pick strategy hence there is no need to sort items afterwards. Batching can be applied using mostly proximity batching where the objective is to minimize the distance travelled or time-window batching where the objective is to maximize the due date performance (Gademann et al. 2001). A batch-picking, on the other hand, is an order picking operation in which the orders are grouped into batches and those batches are picked simultaneously, therefore, forming a wave. The common objective in wave picking operations is to minimize the lead time of a wave, i.e., the maximum travel time of any batch. Partitioning is the main factor of the complexity in batching operations where orders cannot be split into more than one batch. In the literature, several linear-time heuristics, such as rule-based algorithms, seed algorithms, savings algorithms and meta-heuristics are introduced. Koster et al. (1999) compare several batching seed and time-savings batching heuristics combined with largest gap and S-shape routing heuristics in order to come up with fast, robust and user-friendly solutions. Henn & Wäscher (2012) introduce two efficient meta-heuristic approaches, an iterated local search algorithm, and an ant colony optimization algorithm. Gademann et al. (2001) present a branch and bound algorithm where the objective is to minimize the maximum lead time of the batches in a wave picking environment given the number of orders and the number of batches. The model building starts with a basic algorithm, and a preprocessed data. Then a two-opt heuristic is proposed to build a significantly tight upper bound and analyze several lower bounds. The performance of the algorithm significantly decreases with the increase in the number of batches and orders. Gademann & Velde (2005) consider order batching problem using a branch-and-price algorithm with the objective of minimizing the total travel time. To construct the branch-and-price algorithm, the problem is modeled as a generalized set partitioning problem, and then, a column generation algorithm is presented to solve its linear relaxation. Recently, Bayram et al. (2022) consider the order batching problem from a data-centric point of view. They introduce the robust order batching problem, which is defined as the order batching problem subject to uncertainties due to congestion in the picker routes and behavior of the pickers. The authors develop a branch-and-price algorithm integrated with prediction models analyzing the data and predict the batch processing times to improve the overall order processing. For a detailed review of batching operations, the reader is referred to De Koster et al. (2007) and Henn et al. (2012).

2.4 Zone-Picking

In order to reduce travel time, avoid congestion within aisles, and increase the familiarity of pick locations, the picking area could be divided into picking zones. Familiarity of pick locations in a zone has a positive impact if the storage strategy is not random along the order picking area. On the other hand, zone-picking, which is also called picking area zoning, reduces picking time at the expense of order integrity. When zone-picking operation is applied, sorting and consolidation (order assembly) are required at the end of the picking process since the items should be sorted and consolidated according to customer order. This is the main disadvantage of the zone-picking approach. The zone-picking literature can be classified as (i) sequential zone-picking, where the pickers pick the items in a zone and pass to the picker in the next zone, and (ii) synchronized zone-picking, where the pickers pick the items simultaneously in different zones.

The literature regarding sequential zone-picking systems mostly consists of queuing network studies. De Koster (1994) analyses such a system as a Jackson queuing network where the order inter-arrival times and the zone service times are exponentially distributed. Yu & De Koster (2008) generalize the distribution of order inter-arrival and zone service times, and Yu & De Koster (2009) consider the general queuing network to analyse the impact of batching and zone-picking on throughput times. According to this study, since pick density varies across the zones, it leads to imbalance on workload among zones no matter which zone-picking policy is applied. Then, a queuing network model is presented to analyze the impact of batching and zone-picking on throughput times under a sequential zone-picking policy. Melacini et al. (2011) extend the above-mentioned queuing literature by considering the number of zones and the number of pickers for each zone as variables. van Der Gaast et al. (2020) estimate the performance of such systems by also including the buffer capacities at each zone.

In the literature regarding synchronized zone-picking systems, Jane & Laih (2005) present a heuristic algorithm balancing the workloads of pickers at different zones to increase the utilization of a picking wave and reduce the lead time of the wave. First, a similarity measurement approach is presented based on customer order information, then using this measurement, a cluster model with a heuristic solution approach is presented. One of the assumptions is that zone-picking is a strategic level decision, thus zones are known, fixed, and dedicated to pickers before giving operational level decisions. Fixed zone-picking is a common assumption in zone-picking literature. If the items were distributed and stored uniformly among zones in the picking area, then each zone had the same demand data for each wave and workload balance would be perfectly fine. However, items are stored largely according to a turnover-based storage strategy thus fixed zone-picking would not be solely advantageous when workload balancing objective is considered. De Koster et al. (2012) focus on balancing workload over zones and include sorting/consolidation time as a part of wave zone-picking operation. Larger number of zones leads to less picking times but more sorting/consolidation times. It is assumed that all aisles are identical, and all zones consist an equal number of aisles, so that the zone partitioning problem becomes the problem of determining the optimal number of aisles constituting a zone. A model is presented to determine the optimal number of zones which minimizes the system throughput time consisting of picking and sorting/consolidation times. In the model, the number of zones is determined first at the strategic level and then item-to-route assignment problem is solved at the operation level under the constraint of the strategic decision regarding the number of zones. The assumption of fixed zones exists here as well, but it is further noted that the zone-picking problem can also be considered as a short-term problem. Roy et al. (2012) develop a queuing network model and investigate the impact of the number of zones on response time and total travel time. Roy et al. (2019) model a queuing network of movable storage systems and investigate the performance zone-picker assignment strategies.

The above-mentioned zoning studies consider the zoning environment as static, while there are a few studies considering zoning environment as dynamic. To the best of our knowledge, there are only a couple of studies considering dynamic zoning strategy. Bartholdi & Eisenstein (1996) and Bartholdi et al. (2001) introduce the *bucket brigades* concept and model a sequential zone-picking process in a dynamic way where there are no fixed zones since the pickers pick the items in a picking list and pass to the downstream picker, and walk back and fetch the items of the next picking list from the upstream picker. Such systems dynamically balance the workload among pickers without the need of additional operational control. Similarly, Ho & Liao (2009) propose a dynamic zoning strategy where an initial zone partition is designed by distance and flow relationships, then a metaheuristic is developed to help achieving better workload balance and finally a dynamic zone control method is applied for observation. Lamballais et al. (2022) study an order picking process where robots are used instead of conveyor-like fixed systems to reduce pickers' non-value-added travel times. The picker focuses only on picking process while accompanying robot travels between the picker and the depot. The authors compare no-zoning and sequential zoning strategies where, in the sequential zoning case, there exists some fixed zones. In such a collaborative environment, robots are allowed to travel beyond the zones while pickers are not. They develop a Markov Decision Process model together with a closed queuing network and analyse the performance of dynamic switching between zoning and no zoning strategies since there is a trade-off between high robot waiting time in the zoning strategy and high picker travel time in the no zoning strategy. The authors also conclude that it would be significant to examine the optimal dynamic zone sizes to increase the system throughput which is another contribution of our study.

More recently, van der Gaast & Weidinger (2022) introduce a deep learning approach into OPP literature and develop a deep neural networks-based modelling for order picking system selection. The authors apply their model by considering three different static order picking systems; sequential zone-picking, synchronized zone-picking, and bucket brigade picking. The authors conclude that synchronized static zone-picking systems leads to long idle times especially for large order arrivals and long-time pick waves. However, they also conclude that synchronized static zone-picking system is the best when the number of ordered items are small with moderate arrival rates. As the synchronized zone-picking system changes to dynamic, as we propose in Chapter 5, the disadvantages of synchronized static zone-picking systems (e.g., workload-imbalancing, long idle times) also disappear. Similarly, Zhang et al. (2023) analyze the batch and zone-picking integration by comparing the performances of batch-picking and batch-synchronized zone picking, where the batches simultaneously collected in multiple zones. The authors model this problem by considering the pickers' learning effects as this is more possible for the case of zone-picking and propose heuristics for the solution of the models.

2.5 Integrated Order Picking Problems

In the early literature, operational warehouse problems are solved one after another. For example, orders are firstly assigned to batches based on proximity or time-window policy and then routings are calculated. However, in the last decade, literature has started dealing with integrated warehouse operation problems since solving such problems sequentially and independently yields to sub-optimal solutions. Integrated OPP refers to joint analysis of more than one operational warehouse problem (Scholz & Wäscher 2017, van Gils et al. 2018b). Scholz & Wäscher (2017) analyse how to increase the solution quality of the approaches in the literature by jointly studying the exact/heuristic routing algorithms and an iterated local search order batching algorithm. (van Gils et al. 2018b) review and classified the most recent literature regarding the integration and interaction of the tactical and operational warehouse problems and conclude that combining multiple order picking planning problems would results in significant gain on overall order picking operation performance. It is also shown that the most popular integrated problems in literature include picker routing, batching and storage assignment problems.

In the recent literature, there has been a number of studies with solution approaches consisting of mathematical programming problems and meta-heuristics due to complex nature of such problems. Since batching, picker-routing and workload balancing problems are within the operational order picking planning problems, their integration is relatively easy when it is compared to an integration with a tactical level problem such as zoning. For example, Hong et al. (2012) propose an integrated order batching and sequencing model and a simulated annealing algorithm for multiple pickers for single block case with an objective of minimizing the total retrieval time. In this study, congestion caused by order pickers assigned to the same aisle at the same time are also considered. Henn & Schmid (2013) introduce another mathematical optimization model for the order batching and sequencing problem and also presented an iterated local search algorithm. Valle et al. (2017) present a formulation for joint order batching and picker routing problem by introducing several valid inequalities including optimality cuts and symmetry breaking constraints by taking advantage of the graph representation of order picking area and the authors claim that the computational performance is improved significantly. Scholz et al. (2017) propose an integrated solution approach for the order batching and picker routing problem, called joint order batching and picker routing problem. Then, the authors introduce a mathematical model and a variable neighborhood descent algorithm which consider not only batching and routing but also batch assignment and sequencing of batches inbetween. This approach is called joint order batching, assignment and sequencing, and routing problem.

van Gils et al. (2018a) statistically analyze the relations among order picking operations and examine the best simultaneously performing policy combinations among storage, batching, zoning, and routing operations. The results indicate that simultaneous order picking operations with four zones, within-aisle storage, seed batching, and traversal routing would achieve the best combination. Later, van Gils et al. (2019b) investigate the effects of real-life constraints in a warehouse such as safety rules, picker blocking, and high-level storage locations on integrated order picking operations discussed on van Gils et al. (2018a) and conclude that the real-life constraints change the nature of the problem and result in totally different best combination. In this case, the best combination includes a single pick zone, perimeter storage, seed batching, and traversal routing. This reveals the statistically significant impact of real-life constraints on integrated order picking operations especially when the picker density is large. van Gils et al. (2019a) present a mathematical formulation for the integrated batching, routing, and picker scheduling problem with the objective of minimizing the total order picking time while satisfying a customer service level by including order due times as constraints. An efficient iterated local search algorithm is also presented since the model depends on the number of picking items, thus creating exponential number of sub-tour elimination constraints.

Different considerations are also studied in the literature regarding integrated OPP. Vanheusden et al. (2020) present a mathematical model, and an iterated local search algorithm to efficiently solve pickers' workload balancing problem. Vanheusden et al. (2022a) analyse the impact of several workload balancing measures in order picking operations in order to reveal the significant factors ensuring workload balancing. Srinivas & Yu (2022) analyze another integrated OPP by considering human-robot collaboration in order picking systems. The authors assume that humans picks the items and autonomous mobile robots handle the transportation. They propose a formulation as well as a simulated annealing algorithm for the large instances for the joint order batching, batch-assignment/sequencing and routing problem with the objective of minimizing the total tardiness. Guo et al. (2022) takes the COVID-19 pandemic into account and propose a formulation for the zone-wave-batch picking problem under scattered storage policy. Pinto & Nagano (2022) review the literature regarding the joint order batching, assignment and sequencing, and routing problem as it is the mostly studied combination in the integrated OPP literature. D'Haen et al. (2022) extend the joint order batching, assignment and sequencing, and routing problem by also taking into account the online, dynamic order arrivals, which is more realistic in an e-commerce era. To the best of our knowledge, zone-picking operations have received no attention in integrated order picking literature. Finally, we note that scattered storage policy and human-robot collaborations are two trending topics in the order picking literature.

CHAPTER 3

AN ARC ROUTING-BASED COMPACT FORMULATION FOR PICKER ROUTING IN SINGLE-BLOCK PARALLEL-AISLE WAREHOUSES

3.1 Introduction

Warehouses play an increasingly important role in competitive supply chains, and order picking is the core operation in a warehouse, accounting for an estimated 55% of the warehouse operating costs (Tompkins et al. 2010, Bartholdi & Hackman 2019). Among the operations performed within a warehouse, order picking is also the most time-consuming and labor-intensive activity. In order picking operations, items must be collected from the warehouse in order to satisfy the customer demand while a cost and/or service-related objective is to be met (De Koster et al. 2007, van Gils et al. 2018b, Masae et al. 2020a). While the literature has generally modeled the OPP as a special case of the TSP (Burkard et al. 1998, Scholz et al. 2016, Masae et al. 2020a), this study presents an arc routing-based binary integer programming formulation for the OPP in single-block parallel-aisle warehouses, by taking into account the special properties of the graph corresponding to both warehouse layouts. This formulation depends on replacing the subtour elimination constraints with a much smaller number of disconnectivity elimination constraints, which significantly reduces the integrality gap. Our computational experiments show that the proposed formulations are either comparable to or outperform their counterparts in the literature for single-block parallel-aisle warehouses. The efficiency of these formulations implies that not only can they be used to solve the OPP in a timely manner, but they can also be incorporated into integrated models that consider multiple warehouse decision problems at the operational level.

Due to the importance of order picking on the overall warehouse performance, there exists a vast amount of literature on order picking. van Gils et al. (2018b), in their review, conclude that a significant portion of the warehouse order picking literature addresses picker routing as the main problem or as a subproblem of an integrated problem. The significance of picker routing in order picking operations underlines the importance of formulating an efficient integer programming model for the OPP.

Although the OPP is a special case of the $\mathcal{NP}$-hard TSP, it has been shown to be polynomially solvable on single-block warehouse layouts using a dynamic programming algorithm (Ratliff & Rosenthal 1983). It has been also shown that the RR algorithm can be extended to two-block warehouse layouts (Roodbergen & De Koster 2001b). However, the enumerative nature of these algorithms has rendered their extension to more than two blocks impractical. The OPP can also be modeled using the well-known integer programming formulations for the TSP (Miller et al. 1960, Gavish & Graves 1978, Claus 1984) or the Steiner TSP (Letchford et al. 2013). However, these formulations lead to large computing times since they are dependent on the number of pick locations and require computationally expensive subtour elimination constraints. Furthermore, the linear relaxations of these formulations are weak (Pansart et al. 2018). Hence, research focusing on formulating an integer programming model for the OPP has made use of modifying these TSP-based formulations (especially the subtour elimination constraints) by incorporating specific properties of the OPP and its corresponding graph structure (e.g., Theys et al. 2010, Scholz et al. 2016, Pansart et al. 2018).

Existing integer programming approaches for the OPP are relatively inefficient compared to DP-based algorithms, as they depend substantially on the TSP-based formulations, with the exception of Goeke & Schneider (2021), which formulates an integer programming model by taking into account the structural properties of the optimal OPP tours by Ratliff & Rosenthal (1983). While this formulation works efficiently for the single-block case, its reliance on the RR algorithm makes it difficult to extend to multiple blocks in a straightforward way. To bridge this gap, this chapter develops an alternative formulation for the single-block OPP that employs an arc routing-based approach. Based on our computational experiments, the efficiency of the proposed formulation is comparable to that of Goeke & Schneider (2021) and exceeds that of all its remaining counterparts in the literature.

Arc routing problems consist of determining a least cost traversal of some arcs or edges of a graph, subject to side constraints (Eiselt & Laporte 2000) while node routing problems focus on the vertices (nodes). From general routing perspective, the relationship is as follows. Given a connected and undirected graph G with vertex set V and edge set E, a cost c_{ij} for each edge $ij \in E$, a set $V_R \subseteq V$ of *required* vertices and a set $E_R \subseteq E$ of *required* edges, the general routing problem (GRP) is the problem of finding the least cost tour traversing through each $v \in V_R$ and each $ij \in E_R$ at least once. When $E_R = \emptyset$, the GRP reduces to the node routing problem of Steiner TSP. When both $V_R = V$ and $E_R = \emptyset$, the GRP is the node routing problem of TSP. When $V_R = \emptyset$, the GRP reduces to the arc routing problem of Rural Postman Problem. When both $E_R = E$ and $V_R = \emptyset$, the GRP is the arc routing problem of Chinese Postman Problem (Eglese & Letchford 2000).

The OPP consists of determining a least cost traversal of *some* edges of a graph, subject to side constraints. In this regard, the OPP can be defined as an arc routing problem as follows: The picking aisles and the cross-aisles constitute the edges while the intersection points between picking aisles and cross-aisles form the vertices. For a feasible order picking tour, the picker should traverse all non-empty picking aisles (and possibly some of the cross-aisles) in order to maintain a connected closed walk starting and ending at the depot. As we show in the following sections, using an arc routing-based approach for the OPP, as opposed to a TSP-based one, results in two main advantages. First, the formulation does not depend on the number of pick locations. Second, it eliminates subtours by means of a significantly smaller number of more efficient constraints. Our experiments on randomly-generated instances and those from the literature test the performance of the proposed models in terms of computation time and integrality gap of the linear programming relaxation. The results show that the proposed models solve the instances efficiently. For the single-block case, we either outperform or obtain comparable results to the studies in the literature. Furthermore, the linear programming relaxation of our formulation leads to particularly stronger lower bounds when number of aisles and pick locations is relatively large compared to the number of items to be picked.

In this context, taking into account the special properties of parallel-aisle warehouse design, we firstly redefine the parameters based on the movement types introduced by

Ratliff & Rosenthal (1983) and then formulate a compact mathematical model, which is dependent only on the number of aisles. Our integer programming formulation for the single-block OPP includes three main classes of constraints: (i) assignment and sequencing constraints, (ii) degree constraints and (iii) disconnectivity elimination constraints. The first set of constraints ensure that all items are picked, whereas the second and third set of constraints are required for a feasible closed walk on the connected graph corresponding to the warehouse.

The remainder of this chapter is organised as follows. Section 3.2 describes the OPP studied in this chapter. In Section 3.3, we present the formulation for the single-block OPP. Computational experiments and the performance of the models are tested in Section 3.4, and the chapter is concluded in Section 3.5.

3.2 Problem Description

Order picking is the most expensive and labor-intensive warehouse activity. The OPP can be defined as one of sequencing the visits of a picker to item locations on the pick list, so that total travel time is minimized. The OPP is a special case of the TSP arising in warehouse operations. The objective of the OPP is to collect the items on the pick list in a sequence that minimizes the total travel time.

In this chapter, we consider the OPP in a *parallel-aisle single-block* warehouse layout. An example for such a layout is given in Figure 3.1, which consists of narrow *picking aisles* parallel to one another. It also contains *cross-aisles* at the front and back ends of the picking aisles as in Figure 3.1.

Figure 3.2 shows the graph representations of the OPP instances in Figure 3.1, where v_0 refers to the depot, v_i, $i \geq 1$ denote the pick locations and vertices a_j and b_j represent the intersection points between the back/front cross-aisle and the picking aisle j, respectively. In line with the literature, we assume narrow picking aisles so that a picker spends negligible time when making horizontal movements within a picking aisle. A picker starts the picking tour from the depot, collects all the items in the pick list and returns to the depot. Without loss of generality, the depot is assumed to be at the left corner of the front cross-aisle.

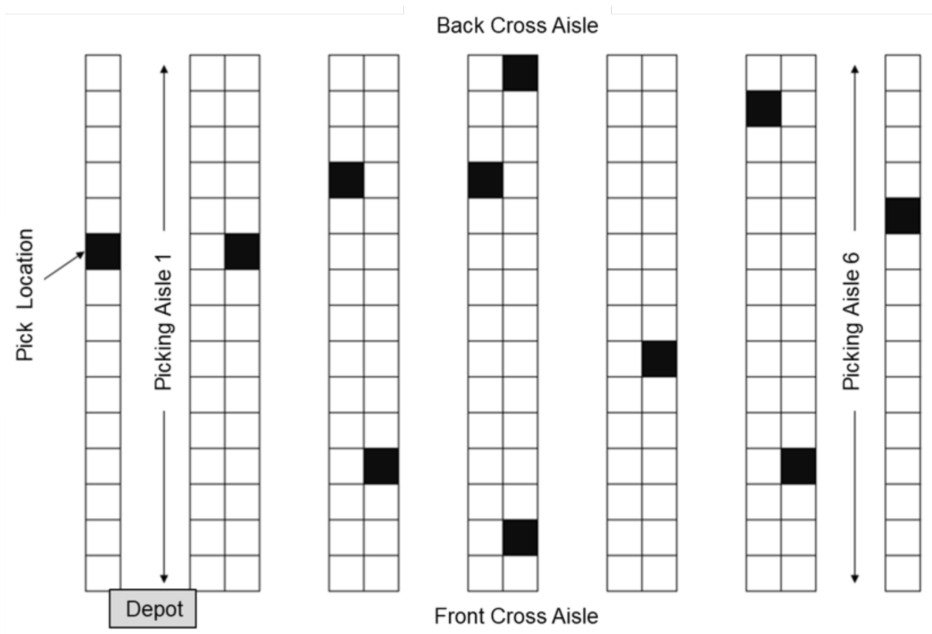


Figure 3.1: A single-block parallel-aisle warehouse 6 picking aisles and 11 items to be picked.

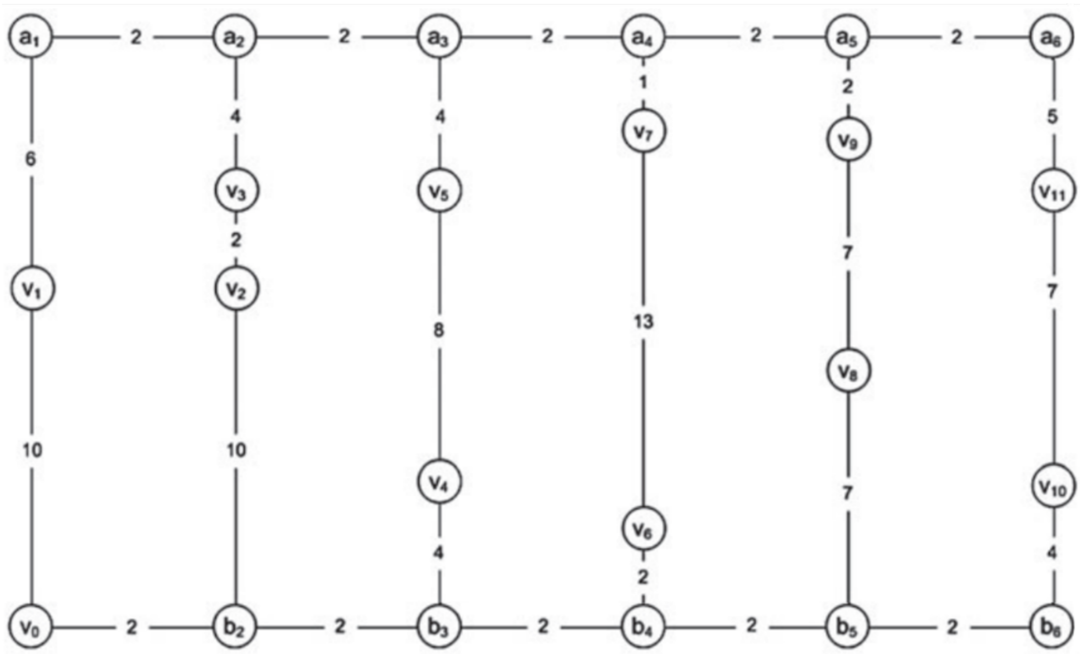


Figure 3.2: The graph representations of Figure 3.1.

Unlike the studies in the relevant literature, this chapter formulates an arc routing-based mathematical model for the OPP that enforces a strongly connected closed walk using *disconnectivity elimination* constraints which can be extended to the case of multi-blocks. These constraints provide a significant contribution to the literature because while enforcing a strongly connected closed walk without the need for sub-tour elimination constraints, these constraints also result in tight linear relaxations for the model, even when the number of pick locations and/or aisles increases significantly. More importantly, these constraints can be extended to incorporate multiple blocks. We leave such an extension for more than two blocks for future research.

To the best of our knowledge, this is the first study in the literature to study the OPP as a variant of the arc routing problem. Our computational results show that our formulation produces results that are either comparable to or better than the existing approaches for the single-block OPP.

3.3 A Binary Integer Programming Formulation for the Single-Block OPP

In this section, we first describe the notation and the useful properties of a parallel-aisle layout for the single-block formulation together with the binary integer programming model.

To obtain an arc routing-based formulation for the OPP, we make use of a number of structural properties of the OPP and redefine the graph so that the set of vertices consists only of the intersection points between the picking aisles and cross-aisles. In this regard, we define M as the set for the picking aisles and the resulting set for the cross-aisles is formed as $C = \{0, 1\}$ referring to front and back cross-aisles, respectively.

Next, we define the six possible intra-aisle movement types for a feasible OPP tour, which are shown in Figure 3.3(a).

(0) The order picker enters the picking aisle using the front intersection point, visits all pick locations and leaves the picking aisle using the back intersection point.

(1) The order picker enters the picking aisle using the back intersection point, visits all pick locations and leaves the picking aisle using the front intersection point.

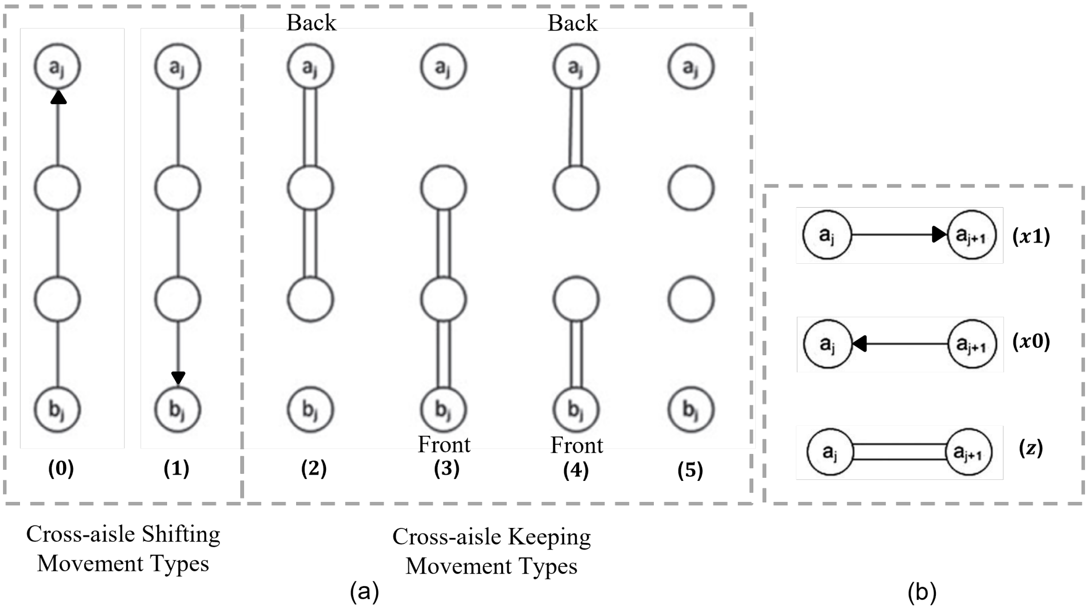


Figure 3.3: Possible movement types. (a) Six possible intra-aisle movement types. (b) Three possible cross-aisle movement types.

(2) The order picker enters the picking aisle using the back intersection point, visits all pick locations, returns after the last pick location, and leaves the picking aisle using the back intersection point.

(3) The order picker enters the picking aisle using the front intersection point, visits all pick locations, returns after the last pick location, and leaves the picking aisle using the front intersection point.

(4) The order picker enters the picking aisle twice, once using the front intersection point and once using the back intersection point. At each time, picker leaves using the same intersection point. The pick locations to make the returns are decided based on the largest gap between any two pick locations or intersection points within the aisle.

(5) The order picker does not enter the aisle when no pick location exists.

We can further classify the intra-aisle movements according to whether they cause a cross-aisle change during the tour. Type (0) and type (1) intra-aisle movements are *cross-aisle shifting movements* since the picker moves from a cross-aisle to the other one with such a move. The remainder of the intra-aisle movements are classified as *cross-aisle keeping movements* since the picker stays at the same cross-aisle after such a move. Moreover, among cross-aisle keeping movements, type (2) is further defined as *back* cross-aisle keeping movement, type (3) is further defined as *front* cross-aisle keeping movement and type (4) is further defined as *both back and front* cross-aisle keeping movement. Finally, type (5) movement is also a cross-aisle keeping movement since the picker continues at the same cross-aisle with such a movement.

We define R as the set of intra-aisle movements in Figure 3.3(a). Furthermore, instead of measuring travel times using those between pick locations (as would be in a TSP-based formulation), we define parameter c_{ij} as the time travelled to clear picking aisle $i \in M$ by making intra-aisle movement $j \in R$. We also note that c_{i5} is set as zero if the picking aisle j does not include a pick location whereas it is set as a very large value otherwise. For the second aisle of Figure 3.1, c_{20} and c_{21} values would be 16 time units since these movements would traverse the whole picking aisle. The travel time parameters for the back and front cross-aisle keeping movements, c_{22} and c_{23}, are 12 and 24 time units, respectively. In the same manner, c_{24}, would be 12 time units. Finally, c_{25} would be a very large value since the aisle is not empty. Defining the parameters in an aisle-dependent way also reduces the size of the data especially when a turnover-based storage policy is applied since the possibility of having a pick location in the further aisles would be relatively small.

The decision variables to represent the intra-aisle movements are defined as follows.

$$y_{ij} = \begin{cases} 1, & \text{if picker clears aisle } i \in M \text{ by making intra-aisle movement } j \in R \\ 0, & \text{otherwise} \end{cases}$$

Similarly, the three possible cross-aisle movement types, also shown in Figure 3.3(b), are as follows:

$(x1)$ The order picker leaves the picking aisle j and enters the picking aisle $j+1$.

$(x0)$ The order picker enters the picking aisle j after leaving the picking aisle $j+1$.

(z) The order picker leaves the picking aisle j and enters the picking aisle $j+1$, and comes back to the picking aisle j using the same cross-aisle after leaving the picking aisle $j+1$.

To define decision variables corresponding to the cross-aisle movements, we firstly introduce set $M' = M \cup \{0\}$, where aisle 0 represents the dummy picking aisle. The parameter h is defined as the travel time between two neighboring aisles. Now we define the binary variables for the cross-aisle movements.

$$
x1_{ik} = \begin{cases} 1, & \text{if picker makes an } x1\text{-type movement from aisle } i \in M' \\ & \text{to } i+1 \in M' \text{ on cross-aisle } k \in C \\ 0, & \text{otherwise} \end{cases}
$$

$$
x0_{ik} = \begin{cases} 1, & \text{if picker makes an } x0\text{-type movement to aisle } i \in M' \\ & \text{from } i+1 \in M' \text{ on cross-aisle } k \in C \\ 0, & \text{otherwise} \end{cases}
$$

$$
z_{ik} = \begin{cases} 1, & \text{if picker makes a } z\text{-type movement from/to aisle } i \in M' \\ & \text{on cross-aisle } k \in C \\ 0, & \text{otherwise} \end{cases}
$$

A feasible picking tour on a parallel-aisle warehouse is a strongly connected closed walk starting from the depot, picking all items on all picking aisles by making intra-aisle movements and returning to the depot. We note that a strongly connected closed walk required here does not necessarily have to be an Eulerian circuit, which is a circuit that uses every edge, but rather only some edges need to be traversed, as in the Rural Postman Problem. Hence, all picking aisles are required to be assigned an intra-aisle movement, whereas some of the cross-aisles can be traversed (at most twice) to ensure a closed walk and that all intra-aisle movements are strongly connected.

For each cross-aisle, the total degree of the vertices due to intra-aisle movement types should be even to complete a closed walk on a parallel aisle warehouse. A return to depot requires equal number of type (0) and (1) intra-aisle movements in total for the single-block picking area. Since other movement types add even degrees on each vertex representing the front and back intersection of a picking aisle, the total degree of vertices due to intra-aisle movement types at each cross-aisle would be even for a strongly connected closed walk.

In this regard, the necessary and sufficient conditions to form a feasible OPP tour, which are also direct implications of being a feasible tour by Ratliff & Rosenthal (1983), are as follows:

(i) One of the intra-aisle movement types in each picking aisle and in a feasible order,

(ii) Even degrees for each of the vertices corresponding to the intersection points between the picking aisles and the cross-aisles, and

(iii) A single connected component for the whole tour.

We constitute the feasible region based on these conditions respectively. The first set of constraints, which we further call the *assignment constraints*, corresponds to the first part of condition (i), where an intra-aisle movement should be assigned to each of the picking aisles to conduct the picking activity.

$$\sum_{j \in R} y_{ij} = 1 \qquad \forall i \in M \tag{3.1}$$

To ensure that the picker starts and ends the tour at the depot (which is on a cross-aisle), there needs to be an equal total number of cross-aisle shifting intra-aisle movements. This implies an equal number of type (0) and (1) intra-aisle movements. Constraint (3.2) guarantees this condition.

$$\sum_{i \in M} y_{i0} = \sum_{i \in M} y_{i1} \tag{3.2}$$

The cross-aisle shifting movements also need to be performed in a specific sequence to guarantee feasibility at any part of the closed walk. Given that the walk starts from the left corner of the front cross-aisle, there are two possible cases. If the picker is at the front intersection point of a picking aisle, the number of type (1) movements to the left of and including that picking aisle should be equal to that of type (0) movements. Similarly, if the picker is at the back intersection point, the number of type (1) movements to the left of and including the picking aisle should be one fewer than that of type (0) movements. In this sense, a type (0) movement occurs before a type (1) movement and do not occur again before the occurrence of a type (1) movement.

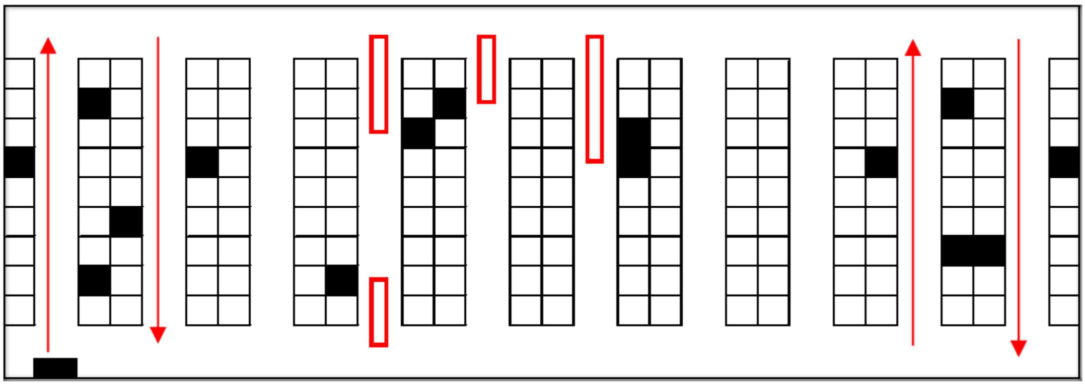


Figure 3.4: A set of intra-aisle movements satisfying Constraints (3.1)-(3.4)

Constraints (3.3) and (3.4) guarantee this logical sequencing as follows:

$$\sum_{p=1}^{i} y_{p0} \geq \sum_{p=1}^{i} y_{p1} \qquad \forall i \in M \qquad (3.3)$$

$$\sum_{p=1}^{i} y_{p0} \leq 1 + \sum_{p=1}^{i} y_{p1} \qquad \forall i \in M \qquad (3.4)$$

Altogether, we refer to Constraints (3.2)-(3.4) as the *sequencing constraints*. Applying only the assignment and sequencing constraints may result in the degree parities to be non-even for the vertices where the corresponding aisles are assigned by a cross-aisle shifting type of movement. Furthermore, each cross-aisle keeping movement may lead to a one-aisle disconnected closed walk. An example with both of these issues is depicted in Figure 3.4.

Before proceeding further, we further classify type (0) and type (1) intra-aisle movements as well as $x1$ and $x0$-type cross-aisle movements as the *odd-degree movements* as they increase the degree of their corresponding vertices by an odd number. In this regard, due to Constraints (3.3) and (3.4), an $x1$-type cross-aisle movement only occurs at the back cross-aisle since type (0) intra-aisle movement occurs before type (1) intra-aisle movement. Similarly, all $x0$-type cross-aisle movements only occur at the front cross-aisle. This is because if an order picker walks to the back cross-aisle using a type (0) intra-aisle movement, there are two possibilities: The picker may (1) continue to the next aisle by making an $x1$-type cross-aisle movement or (2) continue to the previous cross-aisle and come back, thus making a z-type cross-aisle movement. The remaining possibilities would violate the sequencing Constraints (3.3) and

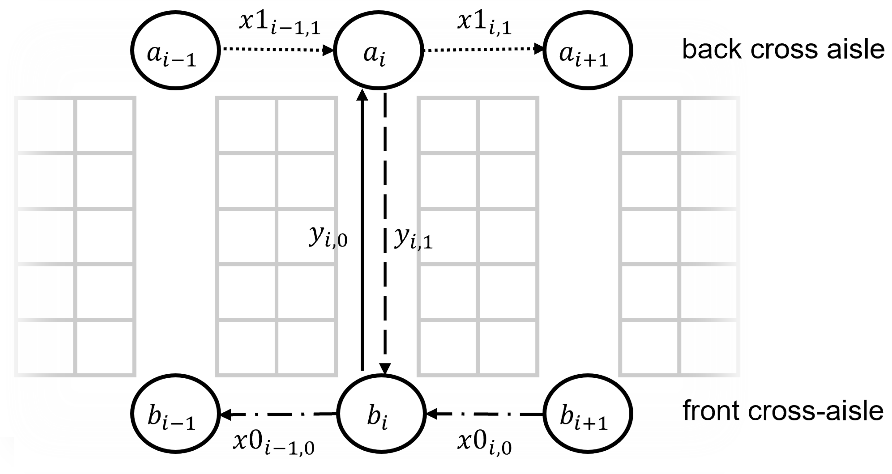


Figure 3.5: Visual representation of all possible odd-degree movements involving aisle i in a single-block layout

(3.4). A similar reasoning applies at the front cross-aisle for an $x0$-type cross-aisle movement. Hence, using odd-degree movements while establishing a sequential relationship between type (0) and type (1) movements breaks the symmetry and halves the number of feasible order picking tours. Moreover enforcing intra-aisle movements in this way is also advantageous in practice as it reduces the congestion on the front and back cross-aisles by leading to one-way movements. All possible odd-degree movements in single-block layouts resulting from above observations are depicted in Figure 3.5.

Addition of Constraints (3.5) and (3.6), defined as *degree constraints*, prevents the occurrence of odd-degree vertices on the front and back intersection points, respectively. Here, each constraint forces the corresponding vertices to be even degree by ensuring equal number of incoming and outgoing odd-degree movements as in Figure 3.5. To formulate such even degree vertex constraints, there is no need to force the right-hand side of the constraints to be even by using dummy integer variables as modelled in Goeke & Schneider (2021). It is possible by equalizing the number of incoming and outgoing odd-degree movements. In other words, there is no need to include any even-degree movements into degree constraints as they don't change the even-degree status of a vertex.

$$x0_{(i-1)0} + y_{i0} = x0_{i0} + y_{i1} \qquad \forall i \in M \tag{3.5}$$

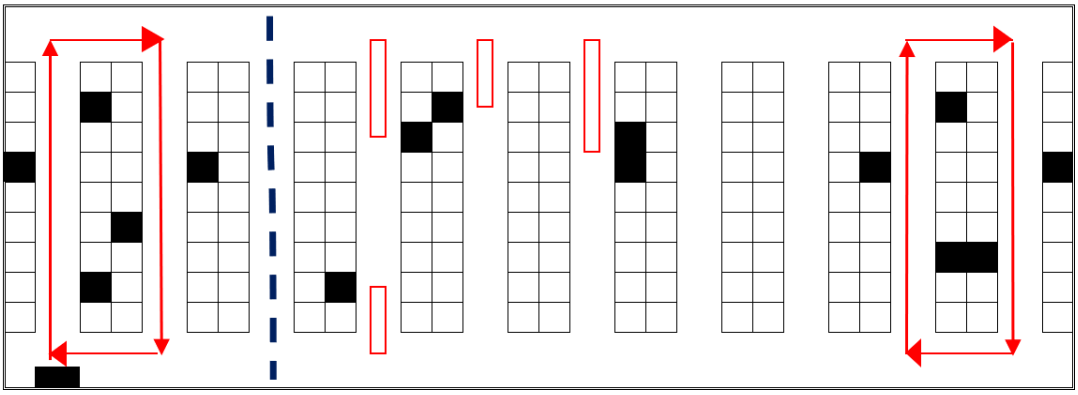


Figure 3.6: A set of intra-aisle and cross-aisle movements satisfying the assignment, sequencing and degree constraints

$$x1_{(i-1)1} + y_{i0} = x1_{i1} + y_{i1} \qquad \forall i \in M \tag{3.6}$$

With the assignment, sequencing and degree constraints, a disconnected closed walk could still exist, as exemplified in Figure 3.6. Here, each aisle is assigned to a movement with the sequential relations satisfied, and each vertex is of even degree. However, each cross-aisle keeping movement generates disconnected closed walks (subtours) by itself. Moreover, an even number of cross-aisle shifting movements connect with each other, hence they also form disconnected closed walks. To explain in which condition such disconnectivities occur, we use the following definition.

Definition 1. When the picker is on the front cross-aisle, the state of the tour is called *state 0*. Otherwise, the state is called *state 1*. In other words, when the total number of type (0) movement equals to the total number of type (1) movement, the picker is on the front cross-aisle and it is called state 0. On the contrary, when the picker is on the back cross-aisle, i.e. when the total number of type (0) movement is one greater than the total number of type (1) movement, it is called *state 1*.

We note that the assignment, sequencing and degree constraints are sufficient to prevent disconnectivity on an ongoing OPP tour as long as the state is 1. This is because when the state is 1, the picker is on the back cross-aisle, hence at least one of the vertices would be an odd-degree vertex. At this point, degree constraints would force each of the vertices to be of even degree, thus enforce connectivity, until the state of the tour turns to state 0.

We also note that, for a feasible order picking tour, only a z-type cross-aisle movement can occur when the state of the tour is 0 (and $x1$ and $x0$-type cross-aisle movements can occur when the state of the tour is 1) to maintain the degrees of vertices to be even on an ongoing OPP tour until the state of the tour changes to 1.

In Figure 3.6, the state becomes 0 at the second aisle. At this point, all vertices are even degree but disconnected closed walks exist. Hence, as long as the state is 0, the assignment, sequencing and degree constraints will not be sufficient to prevent disconnectivity. Before we formulate the constraints to break such disconnectivities, we define the possible cases of disconnectivities that might occur in the single-block layouts.

Definition 2. For a single-block layout, three possible types of disconnectivities can occur.

(i) *Vertical disconnectivity* occurs when there is neither front nor back cross-aisle movement between two adjacent aisles. In other words, one can draw a vertical line which does not cut any cross-aisle movements. Such a vertical line is shown on the third picking aisle in Figure 3.6 above.

(ii) *Horizontal disconnectivity* occurs when either adjacent front or adjacent back cross-aisle keeping movements horizontally connect among themselves to satisfy the degree constraints but remain disconnected from the remainder of the tour. Such a disconnected closed walk can be seen in Figure 3.7.

(iii) *Diagonal disconnectivity* occurs when both a front and a back cross-aisle movement are missing, and thus one can draw a diagonal line which does not cut any intra-aisle or cross-aisle movements. An example of such a disconnectivity is given in Figure 3.8.

As we define *when* and *how* the disconnectivities occur, we now begin formulation of disconnectivity elimination constraints starting with Constraint (3.7), which prevents the vertical disconnectivity by stipulating that each picking aisle must be connected to the next aisle through the back and/or front cross-aisle no matter what the state of

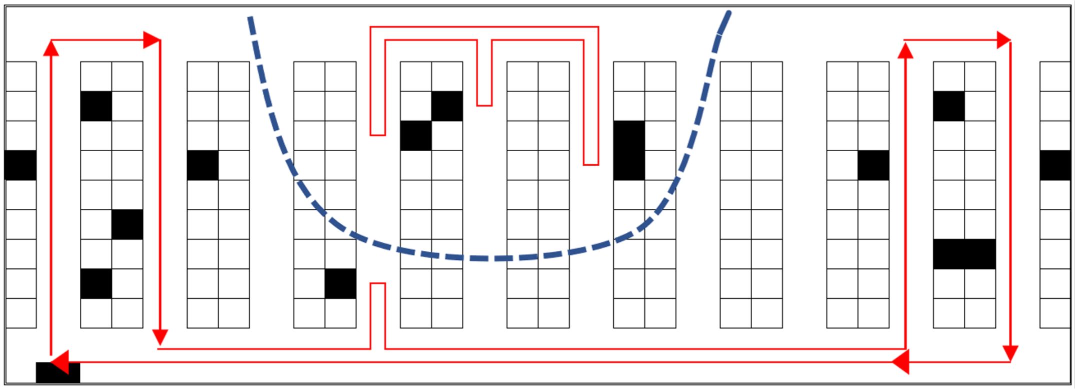


Figure 3.7: Disconnected closed walks resulting due to horizontal disconnectivities

the tour is.

$$x1_{i1} + x0_{i0} + 2z_{i1} + 2z_{i0} \geq 2 \qquad \forall i \in M \setminus \{|M|\} \tag{3.7}$$

Since disconnectivities occur when the state of the tour is 0 and only z-type cross-aisle movements can occur at this state, the remaining disconnectivity constraints are formulated using only z-type decision variables. Moreover, the last part of the right hand side of the disconnectivity constraints ensures that the following constraints are binding as long as the state of the picker is 0. In this regard, Constraints (3.8) to (3.9) eliminate possible horizontal disconnectivities on the back cross-aisle.

$$z_{(i-1)1} + z_{i1} \geq y_{i2} + y_{i4} - |M| \sum_{p=1}^{i} (y_{p0} - y_{p1}) \qquad \forall i \in M \tag{3.8}$$

$$u\Big(z_{(i-1)1} + z_{(i+u)1}\Big) \geq \sum_{p=0}^{u-1} z_{(i+p)1} - |M| \sum_{q=0}^{u+1} \Big(\sum_{p=1}^{i-1+q} (y_{p0} - y_{p1}) \Big) \tag{3.9}$$

$$u \in \{1, \dots, |M| - 1\}, i \in \{1, \dots, |M| - u\}$$

Constraint (3.8) eliminates the one-aisle disconnected closed walks formed by a back cross-aisle keeping movement (e.g., aisles 4, 5 and 6 in Figure 3.6). Constraint (3.9) eliminates the multi-aisle disconnected closed walks formed by z-type cross-aisle movements. Such disconnectivities contain at least one and at most $|M| - 1$ adjacent z-type cross-aisle movements as exemplified in Figure 3.7. The number of adjacent

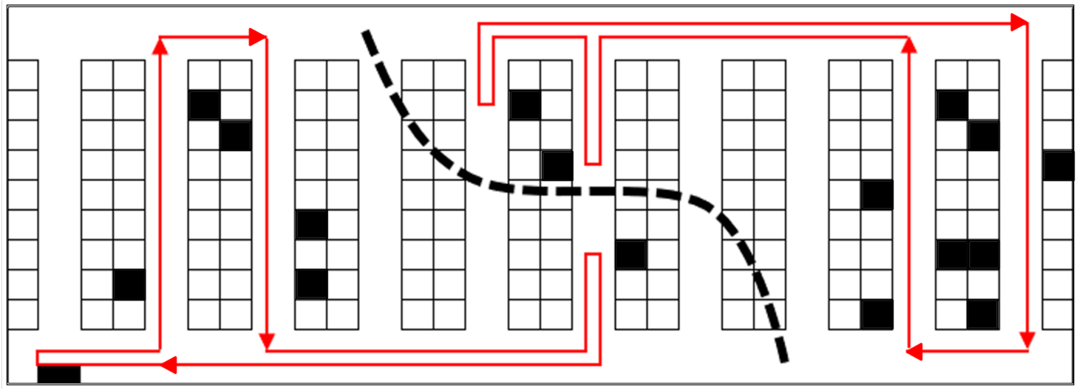


Figure 3.8: Disconnected closed walks resulting due to diagonal disconnectivities

z-type cross-aisles in a disconnected closed walk is represented by the parameter u. Here, the constraint firstly checks if the state is 0 throughout from aisle i to $i+u$. If so, it means that u adjacent z-type cross-aisles may form a disconnected closed walk. To prevent this, Constraint (3.9) forces either a back z-type cross-aisle movement before aisle i and/or a back z-type cross-aisle movement after aisle $i + u$. For example, Constraint (3.9) with $i = 4$ and $u = 2$ eliminates the disconnectivity depicted in Figure 3.7. Here, the constraint firstly checks if the state is 0 throughout from aisle 4 to 6. Since the state is 0 and there are 2 adjacent back z-type cross-aisle movements, it forces either z_{31} and/or z_{61} to exist.

In the same manner, Constraints (3.10) and (3.11) eliminate the horizontal disconnectivities possible on the front cross-aisle.

$$z_{(i-1)0} + z_{i0} \geq y_{i3} + y_{i4} - |M| \sum_{p=1}^{i} (y_{p0} - y_{p1}) \qquad \forall i \in M \tag{3.10}$$

$$u\Big(z_{(i-1)0} + z_{(i+u)0}\Big) \geq \sum_{p=0}^{u-1} z_{(i+p)0} - |M| \sum_{q=0}^{u+1} \Big(\sum_{p=1}^{i-1+q} (y_{p0} - y_{p1}) \Big) \tag{3.11}$$

$$u \in \{1, \ldots, |M| - 1\}, i \in \{1, \ldots, |M| - u\}$$

Diagonal disconnectivities are prevented in a similar fashion by applying Constraints

(3.12) and (3.13).

$$z_{(i-1)1} + z_{(i+u)0} \geq 1 - |M| \sum_{q=0}^{u+1} \Big(\sum_{p=1}^{i-1+q} (y_{p0} - y_{p1}) \Big) \qquad u \in \{0, \dots, |M|-5\}, \; i \in \{3, \dots, |M|-u-2\} \tag{3.12}$$

$$z_{(i-1)0} + z_{(i+u)1} \geq 1 - |M| \sum_{q=0}^{u+1} \Big(\sum_{p=1}^{i-1+q} (y_{p0} - y_{p1}) \Big) \qquad u \in \{0, \dots, |M|-5\}, \; i \in \{3, \dots, |M|-u-2\} \tag{3.13}$$

Constraints (3.12) and (3.13) firstly check if the state is 0 throughout from aisle i to $i + u$ using the right hand side of the equations. If so, Constraint (3.12) forces either a back z-type cross-aisle movement before aisle i and/or a front z-type cross-aisle movement after aisle $i + u$. At the same time, Constraint (3.13) forces either a front z-type cross-aisle movement before aisle i and/or a back z-type cross-aisle movement after aisle $i + u$. For example, in Figure 3.8, Constraint (3.12) with $i = 5$ and $u = 1$ is violated since neither z_{41} nor z_{60} exists in the solution.

We also observe that, in the first and last picking aisles, diagonal disconnectivity cannot occur because it requires at least one couple of type (0) and type (1) intra-aisle movements, otherwise it would be a horizontal disconnectivity. Hence, the above constraints do not include the first and last picking-aisles.

Since they increase the problem size significantly, but are only required when the state is 0, we implement Constraints (3.9), (3.11), (3.12) and (3.13) as lazy constraints.

Lazy constraints are the constraints that solvers do not initially put into the problem being solved. Only the ones that are violated are included into the problem. Lazy constraints are considered to be significantly useful when (1) the lazy constraints are out of the problem, the most of the instances would still be solved to optimality and also even when they are violated only a small portion of instances are violated, (2) the lazy constraints are out of the problem the computing time would significantly improved since there are too many lazy constraints. For a detailed review of lazy constraints, the reader is referred to Pearce (2019).

Lazy constraint approach is especially important for the computing time performances of our models. This is because of the fact that disconnectivities occur when the state is 0 and the disconnectivity elimination constraints are the most time consuming constraints. For this end, we can formulate these constraints as lazy constraints since, most of the time these constraints will not be required for the optimal OPP tour. In the computational experiment section we will show how significantly the performance of our models are increased with the use of disconnectivity elimination constraints as lazy constraints.

Constraint (3.14) includes the depot into the tour, which is at the very left front corner, without loss of generality. This can be updated in the same manner for different depot locations.

$$z_{10} + x0_{10} = 1 \tag{3.14}$$

The remaining constraints define the domains of the decision variables.

$$x1_{0k}, x1_{|M|k}, x0_{0k}, x0_{|M|k}, z_{0k}, z_{|M|k} = 0 \qquad \forall k \in C \tag{3.15}$$

$$y_{ij} \in \{0,1\} \qquad \forall i \in M, j \in R, \tag{3.16}$$

$$x1_{ik}, x0_{ik}, z_{ik} \in \{0,1\} \qquad \forall i \in M', k \in C \tag{3.17}$$

Finally, we present the objective function which minimizes the total time travelled to complete the order picking tour. The first part gives the total travel time of intra-aisle movements, and the remaining part gives the total travel time of cross-aisle movements.

$$\min \quad \sum_{j \in R} \sum_{i \in M} c_{ij} y_{ij} + h \sum_{k \in C} \sum_{i \in M} (x1_{ik} + x0_{ik} + 2z_{ik}) \tag{3.18}$$

Objective function (3.18), subject to Constraints (3.1)- (3.17), defines a complete binary integer programming formulation for the single-block OPP. To increase the efficiency of the formulation, we propose a set of valid inequalities in Section 3.3.1.

3.3.1 Valid Inequalities for the Single-Block OPP in Parallel-Aisle Warehouse Layouts

Although the proposed formulation is sufficient to constitute a complete binary integer programming model for the OPP, one can significantly increase the efficiency of computing time performance through valid inequalities by taking advantage of (i) the special properties of the parallel-aisle warehouse layout, (ii) the sequential relation between the odd-degree movements, and (iii) the starting and ending aisles of the states of the blocks. Such valid inequalities are presented as follows.

$$x1_{i0}, x0_{i1} = 0 \qquad \forall i \in M' \tag{3.19}$$

$$y_{i5} \geq 1 - \sum_{j=2}^{5} c_{ij} \qquad \forall i \in M \tag{3.20}$$

$$z_{i0} + y_{i0} \leq 1 \qquad \forall i \in M \setminus \{|M|\} \tag{3.21}$$

$$z_{i1} + y_{i0} \leq 1 \qquad \forall i \in M \setminus \{|M|\} \tag{3.22}$$

$$x0_{i0} + y_{i1} \leq 1 \qquad \forall i \in M \setminus \{|M|\} \tag{3.23}$$

$$x1_{i1} + y_{i1} \leq 1 \qquad \forall i \in M \setminus \{|M|\} \tag{3.24}$$

$$z_{(i-1)0} + y_{i1} \leq 1 \qquad \forall i \in M \setminus \{1\} \tag{3.25}$$

$$z_{(i-1)1} + y_{i1} \leq 1 \qquad \forall i \in M \setminus \{1\} \tag{3.26}$$

$$x0_{(i-1)0} + y_{i0} \leq 1 \qquad \forall i \in M \setminus \{1\} \tag{3.27}$$

$$x1_{(i-1)1} + y_{i0} \leq 1 \qquad \forall i \in M \setminus \{1\} \tag{3.28}$$

$$x0_{i0} + z_{i1} \leq 1 \qquad \forall i \in M \setminus \{|M|\} \tag{3.29}$$

$$x1_{i1} + z_{i0} \leq 1 \qquad \forall i \in M \setminus \{|M|\} \tag{3.30}$$

$$x0_{i0} = x1_{i1} \qquad \forall i \in M \setminus \{|M|\} \tag{3.31}$$

$$(|M| - 2) \sum_{i \in M} y_{i0} \geq \sum_{i \in M} (y_{i2} + y_{i4}) \tag{3.32}$$

Constraint (3.19) forces one-way cross-aisle movements, since $x1$-type cross-aisle movement only occurs at the back and $x0$-type cross-aisle movement only occurs at the front cross-aisles. Constraint (3.20) ensures that no intra-aisle movement other than a type (5) movement can occur in an empty aisle.

Constraints (3.21) through (3.28) regulate the occurrence of cross-aisle movements when they are adjacent to cross-aisle shifting intra-aisle movements. In this regard, Constraints (3.21) and (3.22) prevent the simultaneous occurrences of both a type (0) intra-aisle movement and a z-type cross-aisle movement for an aisle as the state would change to 1 with a type (0) intra-aisle movement. In the same manner, Constraints (3.23) and (3.24) prevent the simultaneous occurrences of both a type (1) intra-aisle movement and an x-type cross-aisle movement for an aisle as the state would change to 0 with a type (1) intra-aisle movement. Moreover, Constraints (3.25) and (3.26) prevent the simultaneous occurrences of both a type (1) intra-aisle movement for aisle i and a z-type cross-aisle movement for aisle $i-1$ as both cannot occur at the same time. Similarly, Constraints (3.27) and (3.28) prevent the simultaneous occurrences of both a type (0) intra-aisle movement for aisle i and an x-type cross-aisle movement for aisle $i-1$, which is not possible in an order picking tour.

Constraints (3.29) and (3.30) limit occurrences of the number of cross-aisle movements corresponding to a specific aisle as an x-type cross-aisle movement cannot occur with a z-type cross-aisle movement at the same time. Constraint (3.31) ensures the simultaneous occurrences of two x-type cross-aisle movements as one occurs, the other naturally occurs. Finally, Constraint (3.32) ensures that the back cross-aisle movements can only occur if a type (0) intra-aisle movement is occurred, i.e., the picker should visit the back cross-aisle at least once.

3.4 Computational Experiments

In this section, we present the computational experiments conducted on various instance sets which are generated in line with those in the literature (Scholz et al. 2016). All instances assume uniform demand, in that the pick locations are assumed to be distributed independently and uniformly over the order picking area. We firstly analyse the computing time performance of the single-block model and compare it with the mathematical models studied in the single-block OPP literature on the instance set generated in line with Scholz et al. (2016). This analysis also assesses the number of constraints/variables as well as our LP relaxation integrality gaps. Note that, henceforth, we refer to our basic single-block model as SCS and the model where lazy constraints are applied as SCS+. We implement the models using CPLEX 20.1.0.0 in AMPL modelling language on a personal computer with AMD Ryzen 7 4.2 GHz processor and 8 GB dedicated RAM.

First, we compare the computing time performance of our single-block formulation with the state-of-the-art formulations in the literature. The instance set for single-block OPP is generated in line with Scholz et al. (2016) where the number of picking aisles is set as $M \in \{5, 10, 15, 20, 25, 30\}$, the aisle-length is set to 46 time units,

Table 3.1: Comparison of computing times (in seconds) on Scholz et al. (2016)'s instances

Aisles	Items	LNT	SHSW	PCC	GS	SCS	SCS+
5	30	2.84	0.09	0.03	0.02	0.03	0.02
5	45	8.71	0.09	0.05	0.02	0.03	0.02
5	60	25.66	0.09	0.08	0.02	0.04	0.03
5	75	63.22	0.09	0.08	0.02	0.03	0.03
5	90	146.31	0.10	0.08	0.03	0.03	0.03
10	30	4.57	1.60	0.05	0.03	0.04	0.03
10	45	14.66	1.03	0.09	0.03	0.04	0.03
10	60	37.09	1.42	0.13	0.02	0.04	0.03
10	75	156.22	1.36	0.09	0.02	0.04	0.03
10	90	303.68	0.62	0.09	0.02	0.04	0.03
15	30	7.45	2.29	0.06	0.04	0.08	0.04
15	45	24.85	5.28	0.11	0.04	0.08	0.05
15	60	90.30	10.64	0.12	0.05	0.08	0.05
15	75	357.27	15.10	0.13	0.04	0.08	0.04
15	90	811.61	19.41	0.40	0.05	0.08	0.04
20	30	9.47	10.57	0.09	0.06	0.12	0.06
20	45	41.30	27.32	0.09	0.07	0.14	0.07
20	60	147.52	114.33	0.26	0.06	0.15	0.07
20	75	614.11	216.63	0.24	0.07	0.16	0.07
20	90	1627.68	485.71	0.82	0.09	0.16	0.07
25	30	15.07	54.46	0.10	0.05	0.21	0.07
25	45	41.55	85.46	0.22	0.06	0.26	0.09
25	60	173.87	258.92	0.37	0.08	0.30	0.10
25	75	858.44	527.39	0.58	0.09	0.31	0.11
25	90	1764.21	646.59	1.10	0.09	0.32	0.11
30	30	14.00	204.18	0.08	0.06	0.33	0.08
30	45	43.01	406.19	0.19	0.08	0.45	0.11
30	60	293.87	508.80	0.54	0.10	0.54	0.13
30	75	1102.47	638.89	0.72	0.11	0.61	0.14
30	90	1800.00	786.29	1.63	0.15	0.62	0.15
Average		353.37	167.70	0.29	**0.06**	0.18	**0.06**
Variable Size		$\mathcal{O}(\lvert N^2 \rvert M\rvert)$	$\mathcal{O}(\lvert M\rvert)$	$\mathcal{O}(\lvert M\rvert)$	$\mathcal{O}(\lvert N\rvert\lvert M\rvert)$	$\mathcal{O}(\lvert M\rvert)$	$\mathcal{O}(\lvert M\rvert)$
Constraint Size		$\mathcal{O}(\lvert N^2 \rvert M\rvert)$	$\mathcal{O}(\lvert M\rvert)$	$\mathcal{O}(\lvert M\rvert)$	$\mathcal{O}(\lvert N\rvert\lvert M\rvert)$	$\mathcal{O}(\lvert M^2\rvert)$	$\mathcal{O}(\lvert M^2\rvert)$

and the number of pick locations is set as $N \in \{30, 45, 60, 75, 90\}$. Table 3.1 depicts the average computing times and sizes of the Steiner TSP formulation given by LNT (Letchford et al. 2013), the TSP-based formulations, SHSW (Scholz et al. 2016) and PCC (Pansart et al. 2018), the formulation of GS (Goeke & Schneider 2021) and our formulations, SCS ad SCS+ for the single-block OPP. As the instances for these studies are not publicly available, we generate 2000 instances for each of the 30 classes of settings for SCS and SCS+, and compare it with the results of LNT, SHSW, PCC and GS which are the results of 30, 30, 10 and 10 instances per setting, respectively.

From Table 3.1 we firstly observe that both SCS and SCS+ significantly outperform the Steiner TSP formulation proposed by Letchford et al. (2013), and the formulation of Scholz et al. (2016). The proposed formulation also outperforms that of Pansart et al. (2018) especially when the ratio of the number of pick locations to the number of aisles gets larger. Furthermore, significant contribution of the use of lazy constraints can be observed more clearly as there is an increasing performance of SCS+ compared to SCS especially with a larger number of items and aisles. More importantly, the performance of SCS+ is comparable to that of Goeke & Schneider (2021) as both approaches have an average computing time of 0.06 seconds.

The single-block OPP comparison and the significant contribution of the use of lazy constraints can be observed more clearly in Figure 3.9. Here, one can also observe that computing times required for the solution of SCS increase with an increase in the number of pick locations, even though the size of the model is not dependent on the number of pick locations. This discrepancy is more significant for the results of Scholz et al. (2016) and Pansart et al. (2018). As Scholz et al. (2016) also point out, this seems largely due to the fact that a large number of pick locations results in many good solutions, hence it becomes more time-consuming to prove optimality.

Although the need for disconnectivity elimination constraint increases polynomially with the increase in the number of picking aisles these constraints are applied as lazy constraints and also there is a trade off such that the number of constraints increase while the lower bound obtained by solving the LP relaxation gets stronger. Figure 3.10 depicts the size of the formulations in terms of the number of variables and the number of constraints. The Y-axis represents the relative sizes (in terms of

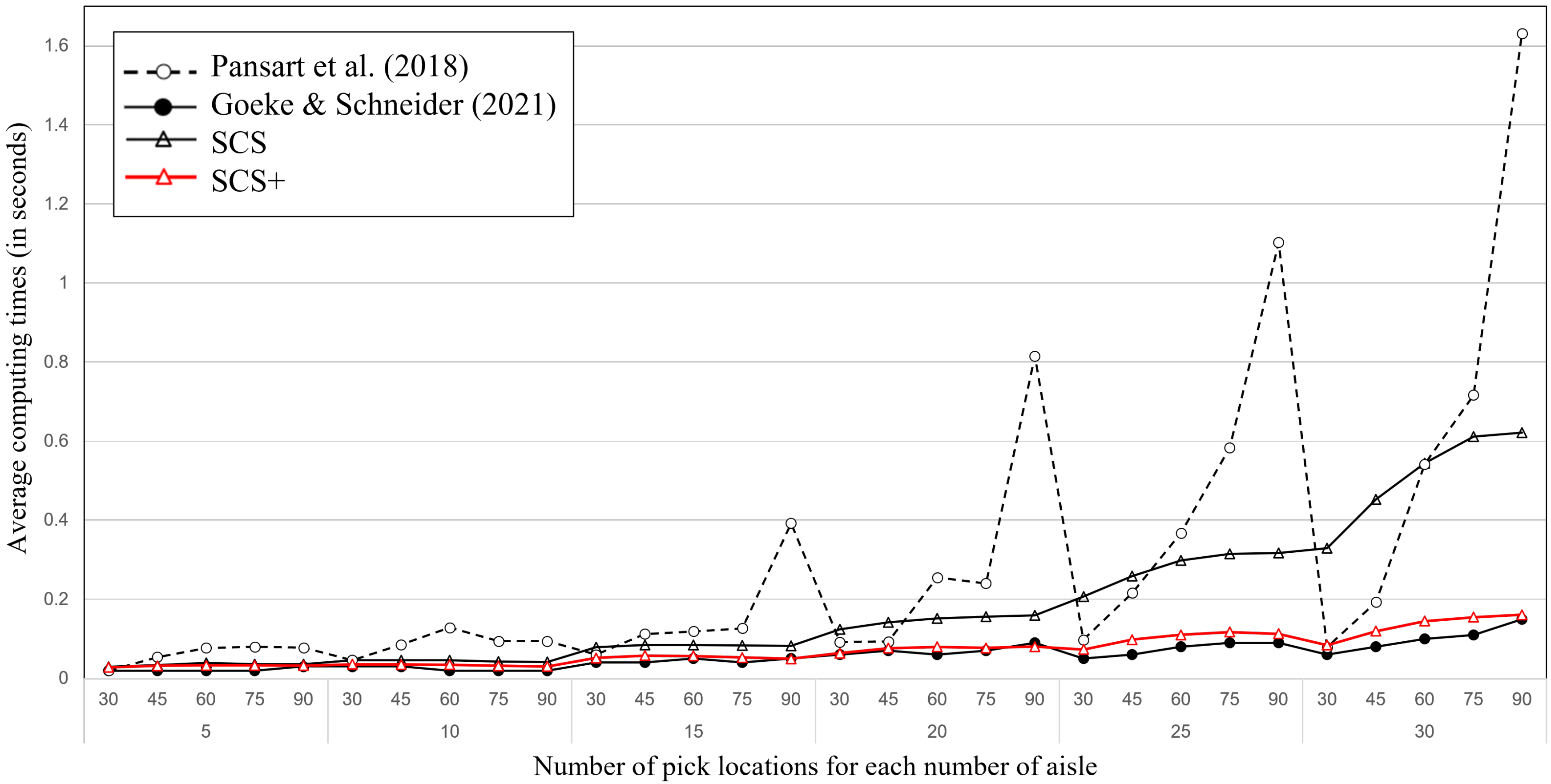


Figure 3.9: Graphical representation of Table 3.1

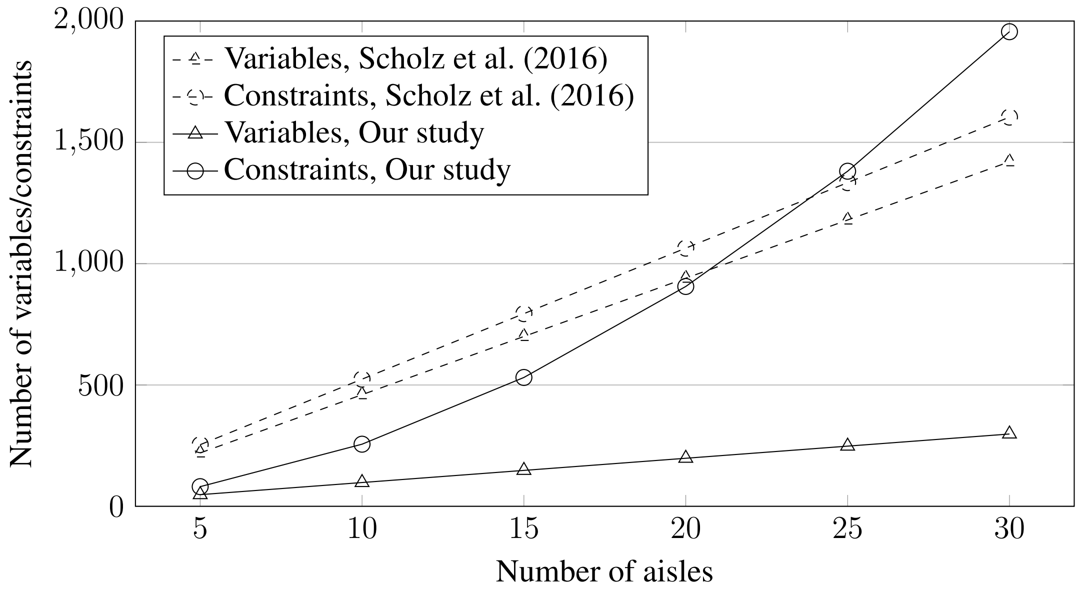


Figure 3.10: Comparison of model size with Scholz et al. (2016)

variables and constraints) of Scholz et al. (2016) and our formulations. In Figure 3.10, we observe that the formulation of Scholz et al. (2016) has $O(|M|)$ variables and $O(|M|)$ constraints while our formulation has $O(|M|)$ variables and $O(|M|^2)$ constraints. Moreover in Table 3.1 we observe that the SCS and SCS+ formulations both involve $\mathcal{O}(|M|)$ decision variables, as do SHSW and PCC. The formulation by GS, on the other hand, includes $\mathcal{O}(|N||M|)$ variables and hence also depends on the number of items. This dependency on the number of pick locations is due to the fact that GS keeps a more general formulation than the assumptions made by Ratliff & Rosenthal (1983). SCS and SCS+ require $\mathcal{O}(|M|^2)$ constraints, which is higher than that of SHSW, PCC and GS when $|M| > |N|$. However, the use of lazy constraints implies that the actual number of constraints is much fewer in our formulation than the worst-case.

On the other hand, relatively good performance of SCS is not only correlated with its compact size but also with its tight lower bounds obtained from its linear relaxation solutions. For this end, we evaluate the performance of the proposed model in terms of percentage integrality gap, which is the difference between the lower bound given by the LP relaxation solution and the value of the true binary integer optimum,

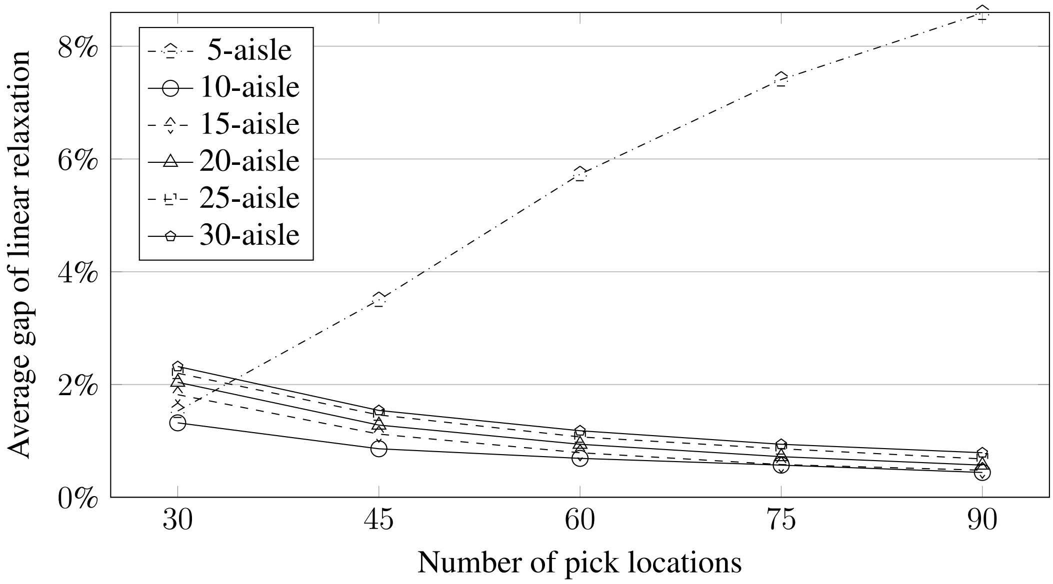


Figure 3.11: Percentage integrality gaps for SCS formulation

expressed as a percentage of the latter:

$$\frac{z_{opt} - z_{LP}}{z_{opt}},$$

where z_{LP} denotes the LP relaxation solution and z_{opt} refers to the optimal OPP time. Figure 3.11 depicts the percentage integrality gaps of our formulation for Scholz et al. (2016) instances. This reveals how effective the disconnectivity elimination constraints are in reducing the integrality gap. We first observe that when the number of aisles is 5, the integrality gap is relatively large and increases significantly with the number of pick locations. This is due to the fact that there is only a single disconnectivity elimination constraint in effect (for $i = 3$ and $u = 0$). However, for larger numbers of aisles (and therefore more realistic warehouse layouts), disconnectivity elimination constraints, which polynomially increase with the number of aisles, reduce this gap considerably. Moreover we also observe a reduction below 1% in the gap with the increase in the number of pick locations as more disconnectivity elimination constraints are in effect.

3.5 Concluding Remarks

This chapter presents a compact arc routing-based formulation for the single-block OPP in parallel-aisle warehouses, taking into account the graph structure of the warehouse and the properties of a feasible order picking tour. Our approach is important in the sense that it is an arc routing-based formulation making use of specifics of the graph structure corresponding to the warehouse layout. Since it is also a compact formulation, it can be a base picker routing model for more complex integrated operational warehouse problems or OPPs with multiple blocks or multiple pickers.

Our computational experiments show that the performance of the proposed formulation for single-block layout outperforms all TSP-based formulations while it is comparable to that of Goeke & Schneider (2021). Other noteworthy findings obtained from our computational experiments include: (i) although the number of constraints is in quadratic order of the number of aisles, applying the multi-aisle disconnectivity elimination constraints as lazy constraints keeps the constraint size linear for the most of the instances and also significantly decreases the actual number of constraints and the computing times, (ii) the integrality gap of the LP relaxation is particularly lower as the size of the instance increases, due to an increase in the number of disconnectivity elimination constraints, and (iii) the computing times are significantly shorter as the ratio of the number of pick locations to the number of aisles is larger.

CHAPTER 4

EXTENSION OF THE SINGLE-BLOCK PICKER ROUTING FORMULATION TO THE CASE OF TWO-BLOCK LAYOUTS

4.1 Introduction

It is noteworthy that our proposed mathematical model for the single-picker OPP can be extended for different variants of the OPP, where each of these variations could yield a significant contribution to the OPP literature. In general, the OPP is defined as the problem of collecting the items on a given pick list in the minimum-time picking tour. From an arc routing perspective, the OPP is the problem of clearing all picking aisles in the minimum-time while assuring a strongly connected closed order picking walk. In this chapter, we aim to extend the arc routing-based single-picker formulation to two-block warehouse layouts. Following the model building phase, we analyze the performance of the model in terms of computing time. Our computational experiments show that our formulation performs better than any alternatives in the literature for the case of two block parallel-aisle warehouses. Finally, for parallel-aisle warehouses with more than two blocks, we propose a simple and effective heuristic, which has an increasing performance with more blocks and larger size of aisle lengths.

Two-block OPP is of concern since having middle cross-aisles further shortens the unnecessary travel time in order picking. In the previous study we have focused on a warehouse with parallel picking aisles that are perpendicular to the two cross-aisles at both ends of the picking aisles. We, again, consider an OPP in a parallel-aisle warehouse layout in this chapter, but it also includes a *middle cross-aisle*, which perpendicularly divides the warehouse into *blocks*, and thereby divide the aisles into *picking sub-aisles*. Order pickers can change aisles at the ends of every picking aisle or at the middle cross-aisle halfway along the picking aisles. An example for such a

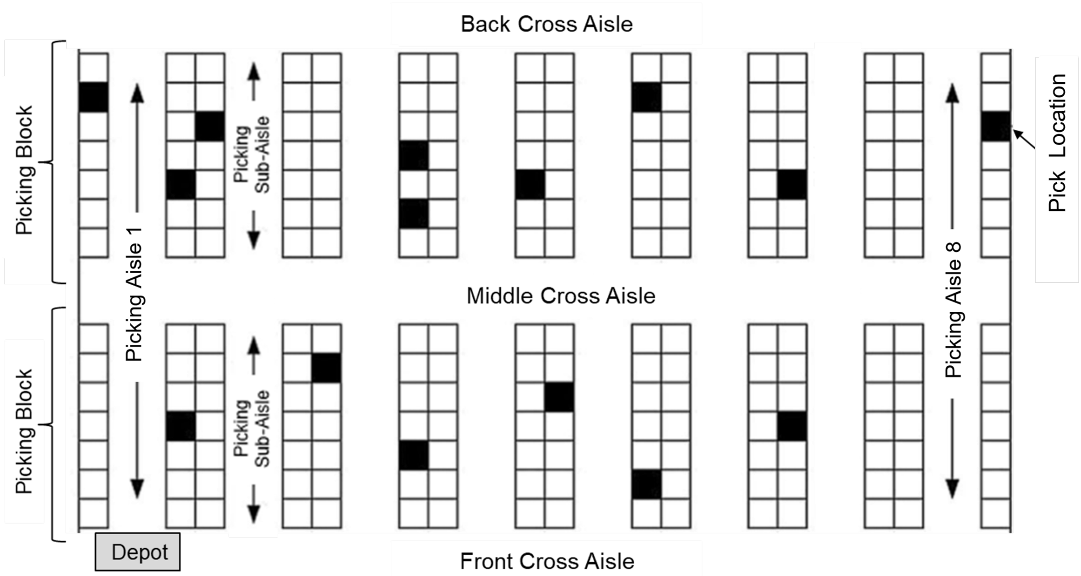


Figure 4.1: A two-block warehouse with 8 picking aisles and 15 items to be picked.

layout is given in Figure 4.1, which consists of narrow *picking aisles* parallel to one another.

The remainder of this chapter is organised as follows. Section 4.2 describes the OPP in two-block layouts. In Section 4.3, we discuss the properties and arc routing-based observations regarding two-block parallel-aisle warehouse layout and present an extended formulation for the OPP when a middle cross-aisle exists. In the Section 4.4, we present a heuristic for layouts with multiple blocks. Computational experiments and the performance of the approaches are tested in Section 4.5, and the chapter is concluded in Section 4.6.

4.2 Problem Description

Figure 4.2 shows the graph representations of the OPP instances in Figure 4.1, where v_0 refers to the depot, v_i, $i \geq 1$ denote the pick locations and vertices a_j and b_j represent the intersection points between the back/front cross-aisle and the picking sub-aisle j, respectively. Vertex m_{j1} represents the intersection point of picking-aisle j and the middle cross-aisle. In line with the literature, we assume narrow picking sub-aisles so that a picker spends negligible time when making horizontal movements within a picking sub-aisle. The middle cross-aisle is the back cross-aisle of the first

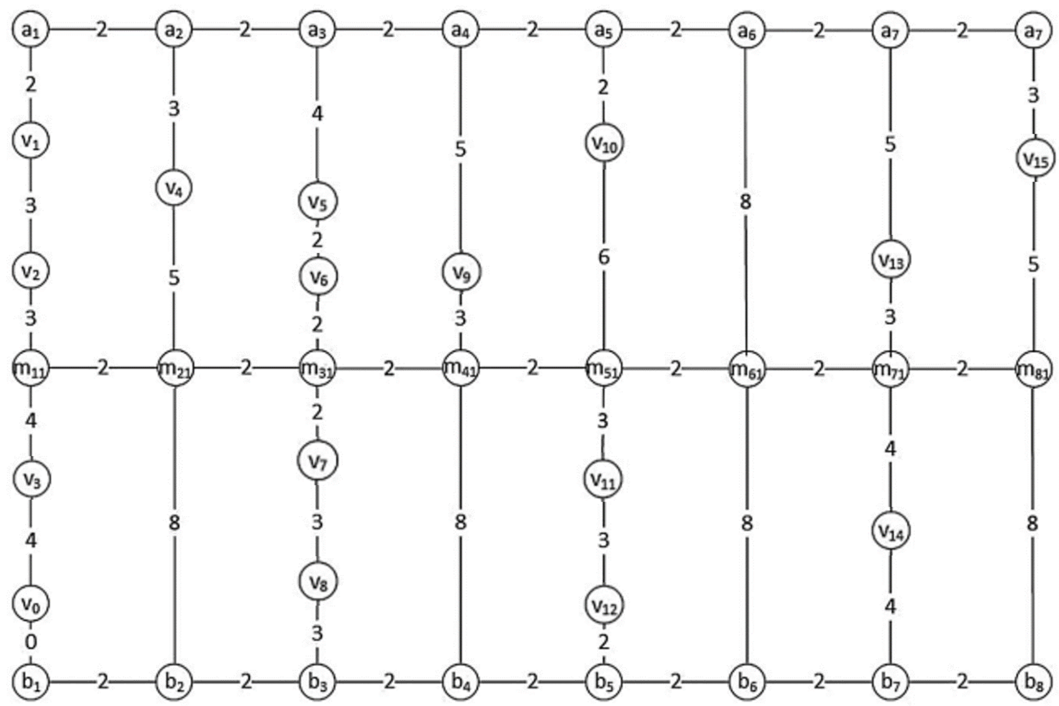


Figure 4.2: The graph representations of Figure 4.1.

block and the front cross-aisle for the second block. In general, a picker starts the picking tour from the depot, collects all the items in the pick list and returns to the depot. Without loss of generality, the depot is assumed to be at the left corner of the front cross-aisle.

Single-picker OPP with two blocks have been studied in the literature since it has been acknowledged as a steppingstone for the multi-block layout studies. The solution approaches in the related literature includes algorithms (Roodbergen & De Koster 2001b, Jang & Sun 2012, Masae et al. 2020b) or TSP-based formulations (Ruberg & Scholz 2016, Scholz 2016, Pansart et al. 2018, Su et al. 2022). Although very efficient for single-block layouts, the formulation proposed by Goeke & Schneider (2021), which is different than the TSP-based formulations and directly exploiting the properties in Ratliff & Rosenthal (1983), is not practically extensible to multiple blocks.

Unlike the studies in the relevant literature, this chapter extends the mathematical model for the single-block OPP especially by focusing on the *disconnectivity elimination* constraints which extended to the case of two-blocks in a relatively straightforward manner. We consider how such an extension is possible for two blocks in this chapter. This work is also important in the sense that it paves the way for an arc-routing based multi-block OPP formulation by induction on the construction of disconnectivity elimination constraints. Our computational results show that our formulation produces results that outperforms the best-known approaches for the two-block OPP to date (e.g., Pansart et al. 2018, Su et al. 2022).

4.3 A Binary Integer Programming Formulation for the Two-Block OPP

This section extends the binary integer programming formulation in the preceding chapter to the two-block OPP. We first discuss how the index sets and parameters are modified to incorporate the existence of multiple blocks in the formulation. Following this, we present the additional changes in cross-aisle movements in the existence of multiple blocks, and provide the modified assignment, sequencing and degree constraints, which are straightforward extensions of their counterparts in the single-block model. The main difference of the two-block model arises from the disconnectivity elimination constraints. We provide a detailed discussion of how such disconnectivities occur and present the constraints to eliminate them.

In this regard, we define the new index set $B = \{0, 1\}$ as the set for the blocks, thus the resulting set for the cross-aisles is updated as $C = \{0, 1, 2\}$ referring to front, middle and back cross-aisles, respectively. We define M as the set for the picking sub-aisles for each block as in Chapter 3. Due to multiple blocks, we present c_{ijb} as the unit time travelled to clear picking sub-aisle $i \in M$ at block $b \in B$ by making intra-aisle movement $j \in R$.

The six possible intra-aisle movement types in Figure 3.3(a) still occur in the same way for the two-block case. As in the single-block case, type (0) and type (1) intra-aisle movements are classified as cross-aisle shifting movements, type (2) is further classified as back cross-aisle keeping movement, type (3) is further classified front cross-aisle keeping movement, type (4) and type (5) movements are further defined as both back and front cross-aisle keeping movements. By also considering the block index, we update the respective binary variable corresponding to these movements in each block as follows.

$$y_{ijb} = \begin{cases} 1, & \text{if picker clears sub-aisle } i \in M \text{ by making intra-aisle movement } j \in R \\ & \text{at block } b \in B \\ 0, & \text{otherwise} \end{cases}$$

Additionally, we use the same cross-aisle movements as in Figure 3.3(b) and their corresponding decision variables as in the single-block case as follows.

$$x1_{ik} = \begin{cases} 1, & \text{if picker makes an } x1\text{-type movement from aisle } i \in M' \\ & \text{to } i+1 \in M' \text{ on cross-aisle } k \in C \\ 0, & \text{otherwise} \end{cases}$$

$$x0_{ik} = \begin{cases} 1, & \text{if picker makes an } x0\text{-type movement to aisle } i \in M' \\ & \text{from } i+1 \in M' \text{ on cross-aisle } k \in C \\ 0, & \text{otherwise} \end{cases}$$

$$z_{ik} = \begin{cases} 1, & \text{if picker makes a } z\text{-type movement from/to aisle } i \in M' \\ & \text{on cross-aisle } k \in C \\ 0, & \text{otherwise} \end{cases}$$

Next, we introduce Constraints (4.1)-(4.4) as straightforward extensions of the assignment and sequencing constraints in the single-block case.

$$\sum_{j \in R} y_{ijb} = 1 \qquad \forall i \in M, b \in B \tag{4.1}$$

$$\sum_{i \in M} y_{i0b} = \sum_{i \in M} y_{i1b} \qquad \forall b \in B \tag{4.2}$$

$$\sum_{p=1}^{i} y_{p0b} \geq \sum_{p=1}^{i} y_{p1b} \qquad \forall i \in M, b \in B \tag{4.3}$$

$$\sum_{p=1}^{i} y_{p0b} \leq 1 + \sum_{p=1}^{i} y_{p1b} \qquad \forall i \in M, b \in B \tag{4.4}$$

Constraint 4.1 ensures each sub-aisle is cleared by a movement type. Constraint 4.2 implies an equal number of type (0) and (1) intra-aisle movements at each block to guarantee a return to the depot. Constraints 4.3 and 4.4 guarantee that type (0)

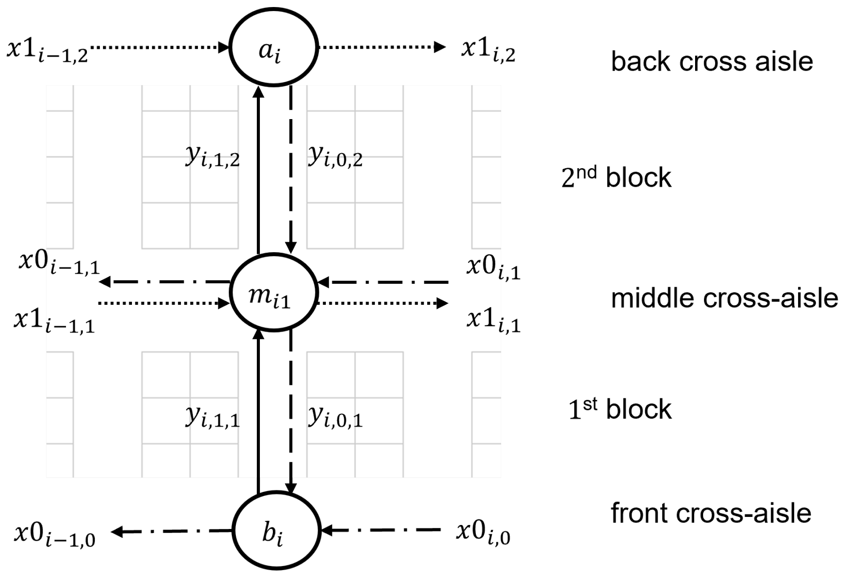


Figure 4.3: Visual representation of possible odd-degree movements involving sub-aisle i in a two-block layout

movement occurs before type (1) movement and cannot occur again before type (1) movement for each block.

This group of constraints ensure the assignment of a movement to each sub-aisle and the sequential relation between the movements. Applying only these sets of constraints would lead to non-even degree parity for the vertices connected with cross-aisle shifting movements and sub-tours for the vertices connected with cross-aisle keeping movements.

The definition *odd-degree movements* also applies in two-block case for type (0) and type (1) intra-aisle movements as well as $x1$ and $x0$-type cross-aisle movements. For the two-block layouts, it is still valid that $x1$-type cross-aisle movement only occurs at the back cross-aisle and $x0$-type cross-aisle movement only occurs at the front cross-aisle. Additionally, at the middle cross-aisles, the occurrence of one of these two odd-degree cross-aisle movements is possible. In this respect, the possible odd-degree movements involving an aisle in two-block layouts would be as in Figure 4.3.

Addition of degree constraints (4.5)-(4.7) prevents the occurrence of odd-degree vertices on the front, middle and back cross-aisles, respectively, by ensuring equal number of incoming and outgoing odd-degree movements shown in Figure 4.3.

$$x0_{(i-1)0} + y_{i01} = x0_{i0} + y_{i11} \qquad \forall i \in M, \quad (4.5)$$

$$x1_{(i-1)1} + x0_{i1} + y_{i01} + y_{i12} = x0_{(i-1)1} + x1_{i1} + y_{i11} + y_{i02} \qquad \forall i \in M \quad (4.6)$$

$$x1_{(i-1)2} + y_{i02} = x1_{i2} + y_{i12} \qquad \forall i \in M \quad (4.7)$$

Addition of degree constraints prevents the odd-degree vertices however subtours could still occur due to disconnected closed walks.

We also note that definitions of horizontal, vertical and diagonal disconnectivities hold for the case of two-blocks as well. Constraint (4.8) prevents the vertical disconnectivity in the same manner as in the single-block case by ensuring that each picking aisle must be connected to the next aisle through back, middle and/or front cross-aisle.

$$\sum_{k \in C} (x1_{ik} + x0_{ik} + 2z_{ik}) \geq 2 \qquad \forall i \in M \setminus \{|M|\} \qquad (4.8)$$

For two-block layouts, we define the state of the tour for each block separately, further referred to as *state of the block*. When the picker is on the front cross-aisle, the state of both first and second blocks are 0 as the number of cross-aisle shifting movements are equal. When the picker is on the middle cross-aisle, the state of the tour is 1 for the first block but it is still 0 for the second block since the middle cross-aisle is the front cross-aisle of the second block. In the same manner, when the picker is on the back cross-aisle, the state of tour is 1 for both first and second blocks.

From our observations for the single-block layouts, horizontal and diagonal disconnectivities may only occur when the state is 0. This also applies for the two-block case. In this sense, no horizontal and diagonal disconnectivity constraints would be required if the states of both blocks are 1. For example, when the picker is on the middle cross-aisle, such a disconnectivity is only possible at the second block since only its state is 0. This is also still valid for two-block layouts that only z-type cross-aisle movement can occur when the state of the block is 0 to maintain the degrees of vertices to be even on an ongoing OPP tour until the state of the corresponding block changes to 1.

In this regard, constraints (4.9) to (4.12) are the straightforward extensions of single-block horizontal disconnectivty elimination constraints (3.8)-(3.11).

$$z_{(i-1)2} + z_{i2} \geq y_{i22} + y_{i42} - |M| \sum_{p=1}^{i} (y_{p02} - y_{p12}) \qquad \forall i \in M \tag{4.9}$$

$$\begin{aligned} u\Big(z_{(i-1)2} + z_{(i+u)2}\Big) \geq \sum_{p=0}^{u-1} z_{(i+p)2} - |M| \sum_{q=0}^{1} \Big(\sum_{p=1}^{i-1+q} (y_{p02} - y_{p12})\Big) \\ u \in \{1, \dots, |M|-1\}, \\ i \in \{1, \dots, |M|-u\} \end{aligned} \tag{4.10}$$

$$z_{(i-1)0} + z_{i0} \geq y_{i31} + y_{i41} - |M| \sum_{p=1}^{i} (y_{p01} - y_{p11}) \qquad \forall i \in M \tag{4.11}$$

$$\begin{aligned} u\Big(z_{(i-1)0} + z_{(i+u)0}\Big) \geq \sum_{p=0}^{u-1} z_{(i+p)0} - |M| \sum_{q=0}^{u+1} \Big(\sum_{p=1}^{i-1+q} (y_{p01} - y_{p11})\Big) \\ u \in \{1, \dots, |M|-1\}, i \in \{1, \dots, |M|-u\} \end{aligned} \tag{4.12}$$

Constraint (4.9) eliminates the one-aisle disconnected closed walks formed by a back cross-aisle keeping movement. Constraint (4.10) eliminates the multi-aisle disconnected closed walks formed by z-type cross-aisle movements. In the same manner, Constraints (4.11) and (4.12) eliminate the horizontal one-aisle and multi-aisle disconnectivities possible on the front cross-aisle exemplified in Figure 3.7.

Additionally, a horizontal disconnectivity may occur not only by cutting through a picking sub-aisle as in the single-block case, but it can also cover a whole picking sub-aisle for the two-block case as exemplified in Figure 4.4. Constraints (4.13) and (4.14) eliminate such disconnectivities covering the picking sub-aisles entirely.

$$\begin{aligned} 2u \sum_{k=1}^{2} (x1_{(i-1)k} + x1_{(i+u)k} + x0_{(i-1)k} + x0_{(i+u)k} + z_{(i-1)k} + z_{(i+u)k}) \\ \geq y_{i02} + y_{(i+u)12} - |M| \sum_{p=0}^{u-1} (1 - z_{(i+p)0}) \\ u \in \{1, \dots, |M|-1\}, i \in \{1, \dots, |M|-u\} \end{aligned} \tag{4.13}$$

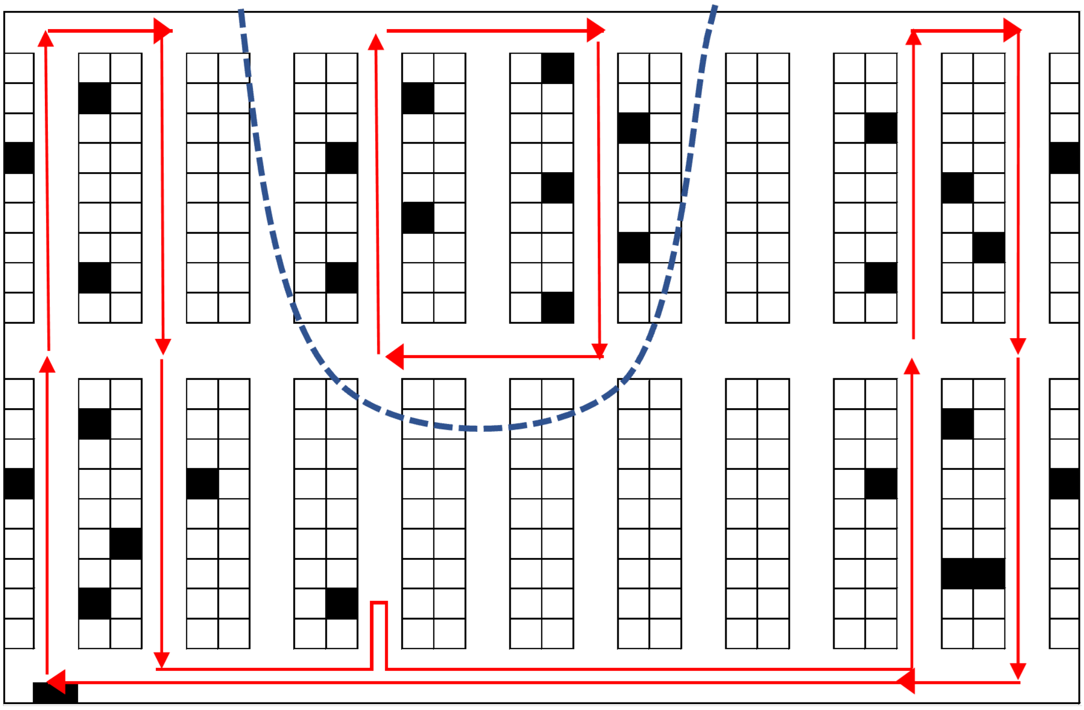


Figure 4.4: A horizontal disconnectivity covering a whole picking sub-aisle

$$\begin{aligned} 2u\sum_{k=0}^{1}(x1_{(i-1)k}+x1_{(i+u)k}+x0_{(i-1)k}+x0_{(i+u)k}+z_{(i-1)k}+z_{(i+u)k}) \\ \geq y_{i01}+y_{(i+u)11}-|M|\sum_{p=0}^{u-1}(1-z_{(i+p)2}) \\ u\in\{1,\ldots,|M|-1\}, i\in\{1,\ldots,|M|-u\} \end{aligned} \tag{4.14}$$

With Constraints (4.13) and (4.14), the connection to the remainder of the tour is ensured not only through a z-type cross-aisle movement both also using both $x1$ and $x0$-type cross-aisle movements. This is because of the fact that such a disconnected closed walk covering the entire block can be connected to the remainder of the tour not only through front and/or back z-type movements but also through front and back $x0$ and $x1$-type movements related to the corresponding block.

In single block layouts, diagonal disconnectivities occur when there is an absence on both front and back cross-aisle movements while the state of the tour is 0. In the two-block layout case, the middle cross-aisle should also be absent to observe a diagonal disconnecitivity. As in the single-block case, diagonal disconnectivities do not occur in the first and last picking sub-aisles. As a result, we can infer that the simultaneous absence of the front, middle and back cross-aisle movements can take place in $(|M|-3)^{|C|}$ different ways where $|C| = 3$ in a two-block layout. As an example, Figure 4.5 shows all possible diagonal disconnectivity occurrences for a six-aisle two-block warehouse. Fortunately, they occur only when the state of the blocks are 0, and thus can be modeled as lazy constraints, so they do not result in substantial computational challenges.

The diagonal disconnectivities may only occur when the states of both blocks are 0. Constraints (4.15) through (4.22) eliminate diagonal disconnectivities. To better explain these diagonal disconnectivity elimination constraints, we decompose the constraints according to the starting and ending blocks of these states.

Assume that the state of the second block is 0 and the state of the first block is turned to 0 at aisle i.

(i) While the state of the first block is 0, the state of the second block cannot turn to 1 at aisle $i+u$ unless there is a z-type middle cross-aisle connection, which includes u adjacent z-type cross-aisle movements between these aisles. Constraint (4.15) ensures such connectivity.

$$\begin{aligned} 1+(\frac{1}{u})\sum_{p=0}^{u-1}(z_{(i+p)1}) &\geq y_{i11}+y_{(i+u)02} \\ -|M|\sum_{q=0}^{u-1}\Big(\sum_{p=1}^{i+q}(y_{p01}+y_{p02}-y_{p11}-y_{p12})\Big) & \quad \begin{array}{l} u \in \{1,\ldots,|M|-1\}, \\ i \in \{1,\ldots,|M|-u\} \end{array} \end{aligned} \tag{4.15}$$

(ii) While the state of the second block is 0, the state of the first block cannot turn to 1 at aisle $i+u$ unless there is a z-type front and/or middle cross-aisle connection. This is ensured by Constraint (4.16).

$$\begin{aligned} 1+(\frac{1}{u})\sum_{p=0}^{u-1}(z_{(i+p)0}+z_{(i+p)1}) &\geq y_{i11}+y_{(i+u)01} \\ -|M|\sum_{q=0}^{u-1}\Big(\sum_{p=1}^{i+q}(y_{p01}+y_{p02}-y_{p11}-y_{p12})\Big) & \quad \begin{array}{l} u \in \{1,\ldots,|M|-1\}, \\ i \in \{1,\ldots,|M|-u\} \end{array} \end{aligned} \tag{4.16}$$

Assume that the state of the first block is 0 and the state of the second block is turned to 0 at aisle i.

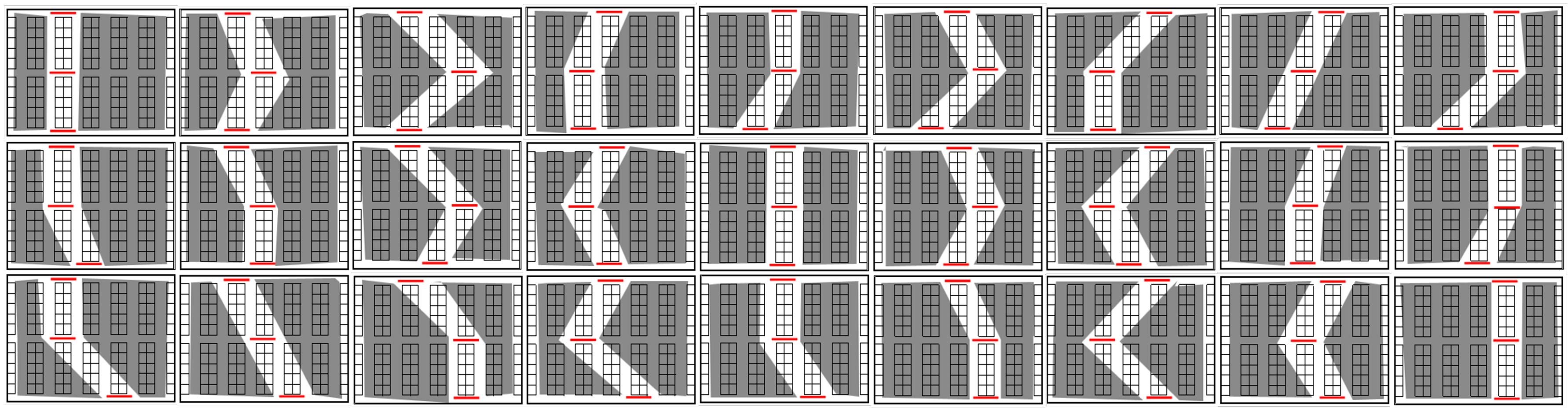


Figure 4.5: Possible diagonal disconnectivity patterns for a six-aisle two-block warehouse layout

(i) While the state of the first block is 0, the state of the second block cannot turn to 1 at aisle $i+u$ unless there is a z-type middle and/or back cross-aisle connection, which includes u adjacent z-type cross-aisle movements between these aisles. This is stipulated by Constraint (4.17).

$$\begin{aligned} 1 + (\frac{1}{u}) \sum_{p=0}^{u-1} (z_{(i+p)2} + z_{(i+p)1}) \geq y_{i12} + y_{(i+u)02} \qquad & u \in \{1, \ldots, |M|-1\}, \\ -|M| \sum_{q=0}^{u-1} \Big(\sum_{p=1}^{i+q} (y_{p01} + y_{p02} - y_{p11} - y_{p12}) \Big) \qquad & i \in \{1, \ldots, |M|-u\} \end{aligned} \tag{4.17}$$

(ii) While the state of the second block is 0, the state of the first block cannot turn to 1 at aisle $i+u$ unless there is a z-type middle cross-aisle connection. Constraint (4.18) ensures such connectivity,

$$\begin{aligned} 1 + (\frac{1}{u}) \sum_{p=0}^{u-1} (z_{(i+p)1}) \geq y_{i12} + y_{(i+u)01} \qquad & u \in \{1, \ldots, |M|-1\}, \\ -|M| \sum_{q=0}^{u-1} \Big(\sum_{p=1}^{i+q} (y_{p01} + y_{p02} - y_{p11} - y_{p12}) \Big) \qquad & i \in \{1, \ldots, |M|-u\} \end{aligned} \tag{4.18}$$

Finally, the uninterrupted continuity of u adjacent z-type cross-aisle movements, as long as the state of the blocks are 0, is ensured by Constraint (4.19) for the front cross-aisle, by Constraint (4.20) and (4.21) for the middle cross-aisle and by Constraint (4.22) for the back cross-aisle.

$$\begin{aligned} 2 + z_{(i+u-1)0} \geq z_{(i)0} + y_{i11} + y_{(i+u)01} - |M| \sum_{q=0}^{u-1} \Big(\sum_{p=1}^{i+q} (y_{p01} + y_{p02} - y_{p11} - y_{p12}) \Big) \\ u \in \{1, \ldots, |M|-1\}, i \in \{1, \ldots, |M|-u\} \end{aligned} \tag{4.19}$$

$$\begin{aligned} 2 + z_{(i+u-1)1} \geq z_{(i)1} + y_{i11} + y_{(i+u)01} - |M| \sum_{q=0}^{u-1} \Big(\sum_{p=1}^{i+q} (y_{p01} + y_{p02} - y_{p11} - y_{p12}) \Big) \\ u \in \{1, \ldots, |M|-1\}, i \in \{1, \ldots, |M|-u\} \end{aligned} \tag{4.20}$$

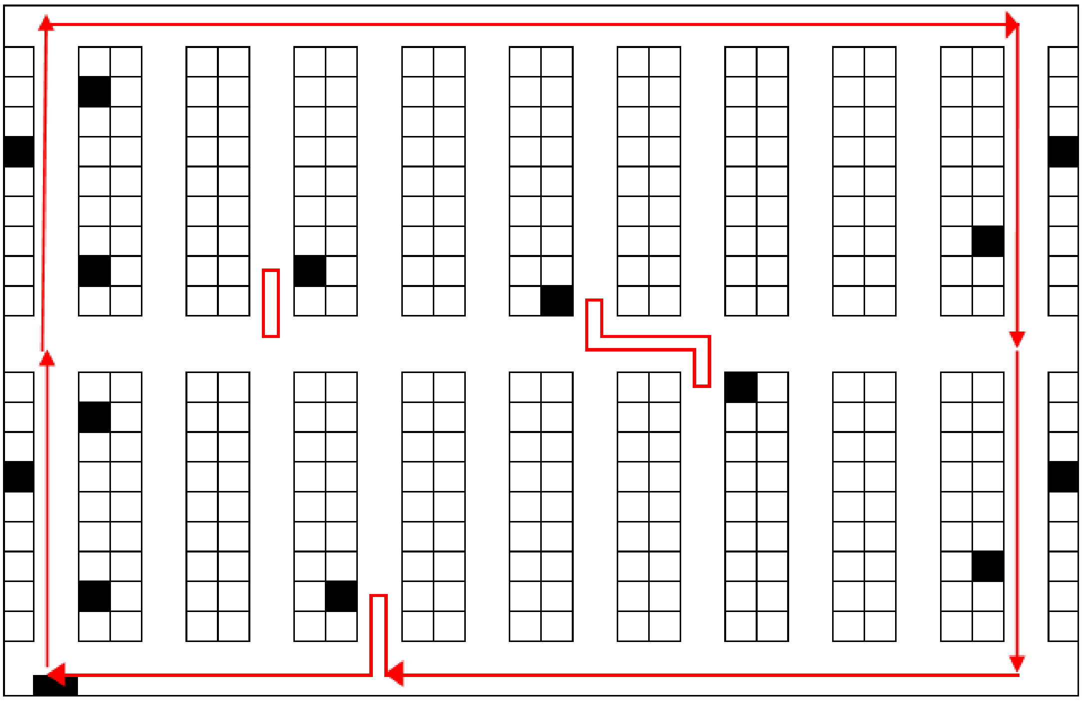


Figure 4.6: One-aisle and multi-aisle inner disconnectivities

$$2 + z_{(i+u-1)1} \geq z_{(i)1} + y_{i12} + y_{(i+u)02} - |M| \sum_{q=0}^{u-1} \Big(\sum_{p=1}^{i+q} (y_{p01} + y_{p02} - y_{p11} - y_{p12}) \Big)$$
$$u \in \{1, \dots, |M| - 1\}, i \in \{1, \dots, |M| - u\} \tag{4.21}$$

$$2 + z_{(i+u-1)2} \geq z_{(i)2} + y_{i12} + y_{(i+u)02} - |M| \sum_{q=0}^{u-1} \Big(\sum_{p=1}^{i+q} (y_{p01} + y_{p02} - y_{p11} - y_{p12}) \Big)$$
$$u \in \{1, \dots, |M| - 1\}, i \in \{1, \dots, |M| - u\} \tag{4.22}$$

Apart from the disconnectivities defined above, there exists an additional type of disconnectivity, referred to as *the inner disconnectivity* where a closed walk can occur on the intersection points between the middle cross-aisle and the picking aisle $i \in M \setminus \{1, |M|\}$ isolated from the remainder of the tour as shown in Figure 4.6.

Unlike other disconnectivity types, inner disconnectivity occurs only when both blocks are at the same state, i.e., the state of both blocks are 0 or 1. For example, Figure 4.6 shows the instances of inner disconnectivities while the states of both blocks are 1. Constraints (4.23) through (4.26) prevent this occurence by forcing a middle z-type cross-aisle movement before aisle i and/or after aisle $i+u$.

Constraints (4.23) and (4.24) eliminate the one-aisle inner disconnectivities formed by cross-aisle keeping movements as exemplified in Figure 4.6 aisle 3. Constraint (4.23) is binding when the state of both blocks are 0 while Constraint (4.24) is binding (e.g., Figure 4.6) when the state of both blocks are 1.

$$\begin{aligned} 2(z_{(i-1)1} + z_{i1}) &\geq y_{i21} + y_{i41} + y_{i32} + y_{i42} \\ -|M| \sum_{q=0}^{1} \Big(& \sum_{p=1}^{i-1+q} (y_{p01} + y_{p02} - y_{p11} - y_{p12}) \Big) \quad \forall i \in M \end{aligned} \tag{4.23}$$

$$\begin{aligned} 2(z_{(i-1)1} + z_{i1}) &\geq y_{i21} + y_{i41} + y_{i32} + y_{i42} \\ -|M| \sum_{q=0}^{1} \Big(2 - & \sum_{p=1}^{i-1+q} (y_{p01} + y_{p02} - y_{p11} - y_{p12}) \Big) \quad \forall i \in M \end{aligned} \tag{4.24}$$

Similarly, Constraints (4.25) and (4.26), binding when the state of both blocks are 0 or 1 respectively, eliminate the multi-aisle inner disconnectivities formed by z-type cross-aisle movements as exemplified in aisles 6 and 7 in Figure 4.6.

$$\begin{aligned} u(z_{(i-1)1} + z_{(i+u)1}) \geq \sum_{p=0}^{u-1} z_{(i+p)1} - |M| \sum_{q=0}^{u+1} \Big(\sum_{p=1}^{i-1+q} (y_{p01} + y_{p02} - y_{p11} - y_{p12}) \Big) \\ u \in \{1, \ldots, |M|-1\}, i \in \{1, \ldots, |M|-u\} \end{aligned} \tag{4.25}$$

$$\begin{aligned} u(z_{(i-1)1} + z_{(i+u)1}) \geq \sum_{p=0}^{u-1} z_{(i+p)1} - |M| \sum_{q=0}^{u+1} \Big(2 - \sum_{p=1}^{i-1+q} (y_{p01} + y_{p02} - y_{p11} - y_{p12}) \Big) \\ u \in \{1, \ldots, |M|-1\}, i \in \{1, \ldots, |M|-u\} \end{aligned} \tag{4.26}$$

Constraint (4.27) includes the depot into the tour, while the remaining constraints define the domains of the decision variables.

$$z_{10} + x0_{10} = 1 \tag{4.27}$$

$$x1_{0k}, x1_{|M|k}, x0_{0k}, x0_{|M|k}, z_{0k}, z_{|M|k} = 0 \qquad \forall k \in C \tag{4.28}$$

$$y_{ijb} \in \{0, 1\} \qquad \forall i \in M, j \in R, b \in B \tag{4.29}$$

$$x1_{ik}, x0_{ik}, z_{ik} \in \{0, 1\} \qquad \forall i \in M', k \in C \tag{4.30}$$

The function (4.31), along with constraints (4.1)- (4.30), constitutes a complete binary integer programming model for the two-block OPP, which minimizes the total time travelled to complete the order picking tour.

$$\min \quad \sum_{b\in B}\sum_{j\in R}\sum_{i\in M} c_{ijb}y_{ijb} + h\sum_{k\in C}\sum_{i\in M'}(x1_{ik} + x0_{ik} + 2z_{ik}) \tag{4.31}$$

The set of valid inequalities in Section 3.3.1 are also applicable to the two-block layouts and significantly increase the efficiency of the formulation by taking advantage of the special properties of parallel-aisle warehouse layout. Next, we present these valid inequalities, which are enriched by some additional constraints and inclusion of the block index. We note that these constraints can be implemented for single-block warehouses by ignoring the block index.

4.3.1 Valid Inequalities for the Two-Block OPP in Parallel-Aisle Warehouse Layouts

Although the SCS and SCS+ formulations are sufficient to constitute a complete binary integer programming model for the OPP, one can significantly increase the efficiency of computing time performance through valid inequalities by taking advantage of (i) the special properties of the parallel-aisle warehouse layout, (ii) the sequential relation between the odd-degree movements, and (iii) the starting and ending aisles of the states of the blocks. Such valid inequalities are presented as follows.

$$x1_{i0}, x0_{i2} = 0 \qquad \forall i \in M' \tag{4.32}$$

$$y_{i52} \geq 1 - \sum_{j=2}^{5} c_{ij2} \qquad \forall i \in M \tag{4.33}$$

$$z_{i0} \leq 1 - \sum_{p=1}^{i}(y_{p01} - y_{p11}) \qquad \forall i \in M \setminus \{|M|\} \tag{4.34}$$

$$z_{i2} \leq 1 - \sum_{p=1}^{i}(y_{p02} - y_{p12}) \qquad \forall i \in M \setminus \{|M|\} \tag{4.35}$$

$$x0_{i0} \leq \sum_{p=1}^{i}(y_{p01} - y_{p11}) \qquad \forall i \in M \setminus \{|M|\} \tag{4.36}$$

$$x1_{i2} \leq \sum_{p=1}^{i}(y_{p02} - y_{p12}) \qquad \forall i \in M \setminus \{|M|\} \tag{4.37}$$

$$z_{i0} + y_{i01} \leq 1 \qquad \forall i \in M \setminus \{|M|\} \tag{4.38}$$

$$z_{i2} + y_{i02} \leq 1 \qquad \forall i \in M \setminus \{|M|\} \tag{4.39}$$

$$x0_{i0} + y_{i11} \leq 1 \qquad \forall i \in M \setminus \{|M|\} \tag{4.40}$$

$$x1_{i2} + y_{i12} \leq 1 \qquad \forall i \in M \setminus \{|M|\} \tag{4.41}$$

$$z_{(i-1)0} + y_{i11} \leq 1 \qquad \forall i \in M \setminus \{1\} \tag{4.42}$$

$$z_{(i-1)2} + y_{i12} \leq 1 \qquad \forall i \in M \setminus \{1\} \tag{4.43}$$

$$x0_{(i-1)0} + y_{i01} \leq 1 \qquad \forall i \in M \setminus \{1\} \tag{4.44}$$

$$x1_{(i-1)2} + y_{i02} \leq 1 \qquad \forall i \in M \setminus \{1\} \tag{4.45}$$

$$x0_{i1} + z_{i2} \leq 1 \qquad \forall i \in M \setminus \{|M|\} \tag{4.46}$$

$$x1_{i1} + z_{i0} \leq 1 \qquad \forall i \in M \setminus \{|M|\} \tag{4.47}$$

$$\sum_{k=0}^{1} x0_{ik} \leq 1 \qquad \forall i \in M \setminus \{|M|\} \tag{4.48}$$

$$\sum_{k=1}^{2} x1_{ik} \leq 1 \qquad \forall i \in M \setminus \{|M|\} \tag{4.49}$$

$$\sum_{k=0}^{1} x0_{ik} = \sum_{k=1}^{2} x1_{ik} \qquad \forall i \in M \setminus \{|M|\} \tag{4.50}$$

$$\sum_{k=0}^{1} x0_{ik} + \sum_{k=0}^{2} z_{ik} \leq 2 \qquad \forall i \in M \setminus \{|M|\} \tag{4.51}$$

$$\sum_{k=1}^{2} x1_{ik} + \sum_{k=0}^{2} z_{ik} \leq 2 \qquad \forall i \in M \setminus \{|M|\} \tag{4.52}$$

$$x1_{ik} + x0_{ik} + z_{ik} \leq 1 \qquad \forall i \in M \setminus \{|M|\}, k \in C \tag{4.53}$$

$$(|M| - 2) \sum_{i \in M} y_{i0b} \geq \sum_{i \in M} (y_{i2b} + y_{i4b}) \qquad \forall b \in B \tag{4.54}$$

Constraint (4.32) forces one-way cross-aisle movements, since $x1$-type cross-aisle movement only occurs at the back and $x0$-type cross-aisle movement only occurs at the front cross-aisles. Constraint (4.33) ensures that a type (5) intra-aisle movement should occur on the back-most block if no pick location exists in the corresponding aisle. This is only applicable for the back-most block as the picker has a possibility to make a cross-aisle shifting travel on an empty picking aisle in the previous blocks in order to reach to the back-most block. Constraints (4.34)-(4.37) arrange the cross-aisles movement occurrences by checking the state of the blocks. For example, a front cross-aisle z-type movement cannot occur during the state of the first block is 0, as ensured by Constraint (4.34).

Constraints (4.38)-(4.45) regulate the occurrence of cross-aisle movements when they are followed by cross-aisle shifting intra-aisle movements. In this regard, Constraints (4.38) and (4.39) prevent the simultaneous occurrences of both a type (0) intra-aisle movement and a z-type cross-aisle movement at an aisle as the state would change to 1 with a type (0) intra-aisle movement. In the same manner, Constraints (4.40) and (4.41) prevent the simultaneous occurrences of both a type (1) intra-aisle movement and an x-type cross-aisle movement at an aisle as the state would change to 0 with a type (1) intra-aisle movement. Moreover, Constraints (4.42) and (4.43) prevent the simultaneous occurrences of both a type (1) intra-aisle movement at aisle i and a z-type cross-aisle movement at aisle $i-1$ as both cannot occur at the same time. Similarly, Constraints (4.44) and (4.45) prevent the simultaneous occurrences of both a type (0) intra-aisle movement at aisle i and an x-type cross-aisle movement at aisle $i-1$, which is not possible in an order picking tour.

Constraints (4.46) and (4.53) limit occurrences of the number of cross-aisle movements corresponding to a specific aisle. Constraint (4.46) ensures that both a middle $x0$-type cross-aisle movement and a back z-type cross-aisle movement cannot occur at the same time since a front $x1$-type cross-aisle movement is not possible. In the same manner, Constraint (4.47) prevents the simultaneous occurrences of both a middle $x1$-type cross-aisle movement and a front z-type cross-aisle movement since a back $x0$-type cross-aisle movement is not possible. Constraints (4.48) ensures that at most one $x0$-type cross-aisle movement can occur at an aisle. Similarly, Constraints (4.49) ensures that at most one $x1$-type cross-aisle movement can occur at an aisle. Constraint (4.50) ensures the simultaneous occurrences of two x-type cross-aisle movements as one occurs, the other naturally occurs. Constraints (4.51) and (4.52) limit the simultaneous occurrences of x and z types movements to at most two. Constraints (4.53) states that only one of the cross-aisle movements can appear on a cross-aisle. Finally, Constraint (4.54) ensures that the back cross-aisle movements of a block can only occur if type (0) intra-aisle movement is occurred, i.e., the picker should visit the back cross-aisle at least once.

4.4 Z-Shape Heuristic: A Modified Largest Gap Picker Routing Algorithm for the Multi-Block OPP

In this section we present a simple, easy to remember and effective picker routing algorithm for routing heuristic for the OPP in parallel-aisle warehouses with multiple block layouts, called the Z-shape heuristic, which is a modified version of the largest gap picker routing heuristic.

The main motivation in this section is to present a heuristic by (i) decreasing the unnecessary cross-aisle movements developed by the largest gap heuristics and (ii) reducing the travel time in picking aisles with large aisle lengths where large gaps are highly likely. This is not taken into account by the Combined+ heuristics proposed by Roodbergen & De Koster (2001a). In largest gap heuristics, the picker travels each cross-aisle almost twice. The contribution of this study is to avoid this duplication in the cross-aisle movements while keep the picker routing still easy to remember. The parallel aisle multi-block warehouse considered in this study is given in Figure 1.1. It contains cross-aisles at front and back of the picking-aisles and contains middle cross-aisles, which perpendicularly divide the warehouse into blocks, and thereby divide the aisles into sub-aisles.

It a user-friendly cross-aisle-oriented algorithm. At each middle cross-aisle, the picker follows the cross-aisle and collects the items in the upper and lower sub-aisles according to largest gap routing policy by making the shortest *cross-aisle keeping movement* from Figure 3.3(a). Only exceptions are the left-most and right-most filled sub-aisles as the picker reaches to another block with a *cross-aisle shifting movement*. What we aim to develop with our algorithm is exemplified in Figure 4.7(a). In this regard we explain the steps of the algorithm as follows.

As in line with the literature, the depot is assumed to be at the left bottom corner. The picker route starts by going all the way up to the back cross-aisle from the leftmost filled picking sub-aisles by making the type (0) cross-aisle shifting movement. Afterwards, the back cross-aisle is travelled while the pick locations close to the back cross-aisle is collected. The collection decision is given by comparing and choosing the minimum of cross-aisle keeping movements. In this regard, at the beginning of the algorithm, all the necessary intra-aisle movements are determined firstly by determin-

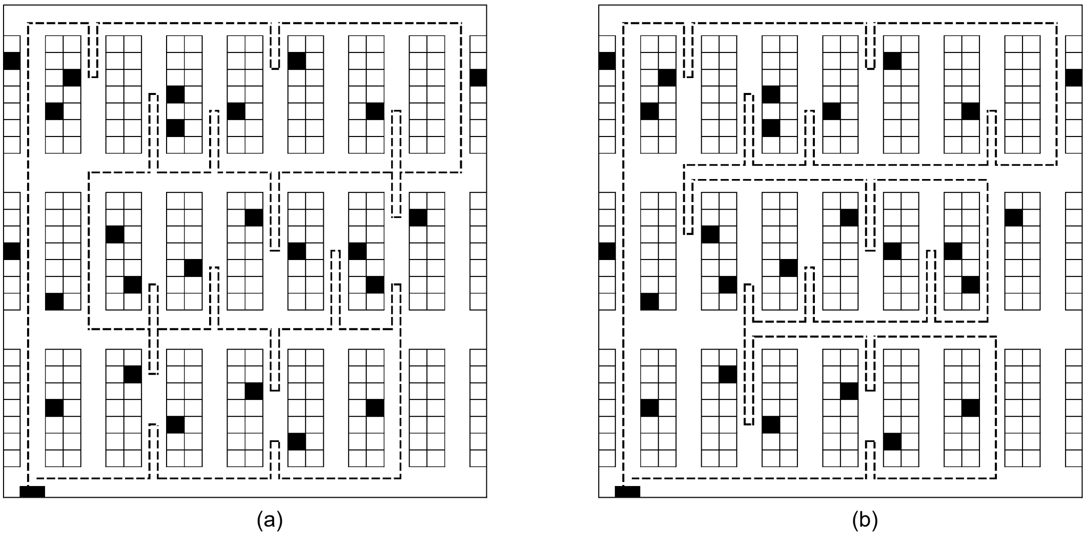


Figure 4.7: (a) Z-shape heuristic solution and (b) the solution of largest gap heuristic proposed by Roodbergen & De Koster (2001a)

ing the cross-aisle shifting movements to change blocks and secondly by determining the minimum of cross-aisle keeping movements for the remaining picking sub-aisles to conduct the picking activity.

After travelling the back cross-aisle and visiting the necessary pick locations, the picker travels to the next cross-aisle by making type (0) cross-aisle shifting movement. On this cross-aisle, the picker visits the necessary pick locations by visiting both adjacent blocks to the corresponding cross-aisle by making cross-aisle keeping movements. The picker keeps going at the same cross-aisle until the last filled picking sub-aisle and makes type (0) cross-aisle shifting movement to enter into the next cross-aisle. In this manner, the picker reaches to the front cross-aisle and returns to the depot by clearing the pick locations close to the front cross-aisle. A disadvantage of this heuristic is that the picker makes a double-cross-aisle movement if the number of blocks is even. It is to say that the picker travels the front or back cross-aisle twice to continue with the Z-shape heuristics route.

By introducing this algorithm, we hypothesize that, for large size of aisle lengths and large number of blocks, Z-Shape heuristics performs better than all simple heuristics for all cases including the Combined+ heuristics, which is the most sophisticated one among the simple-heuristic literature. This is firstly because of the fact that Z-shape heuristic cross-aisle movements are single per cross-aisle when compared to the largest gap heuristic, which has double cross-aisle movements per cross-aisle (see Figure 4.7 (b). Moreover we know from the literature that the largest gap heuristic performs better than the remaining simple heuristics except the Combined+ heuristic. Secondly, Combined+ heuristic never makes largest gap movements. Hence, for the cases with large gaps between pick locations in an aisle, the performance of Combined+ heuristic is expected to lag behind even largest gap heuristics. We especially expect good performance for the layouts with odd number of blocks as Z-Shape heuristic makes single cross-aisle movements on each cross-aisle. So, we expect that Z-Shape heuristic to outperform all its counterparts for the layouts with large number of blocks and large size of aisle lengths. We recommend practitioners to apply this algorithm for picker routing in parallel-warehouse layouts with large size aisles and large number of blocks.

4.5 Computational Experiments

In this section, we present the computational experiments conducted on various instance sets which are generated in line with those in the literature (Scholz 2016, Theys et al. 2010, Roodbergen & De Koster 2001a). All instances assume uniform demand, in that the pick locations are assumed to be distributed independently and uniformly over the order picking area. Firstly, we present the computing time performance of the single- and two-block formulations and compare them with the TSP formulation by Miller et al. (1960) on randomly generated instance sets. Secondly, we compare the computing time performance of the two-block formulation with the formulations of Scholz et al. (2016) and Scholz (2016) on instance set generated in line with Scholz (2016). Thirdly, we test the performance of the two-block formulation with the formulations of Pansart et al. (2018) and Su et al. (2022) on the instance set generated by Theys et al. (2010). Lastly, we compare the performance of our Z-Shape algorithm with the ones in the literature as presented in Figure 2.2. We refer to our basic single- and two-block models as SCS and the models where lazy constraints are applied as SCS+. We implement the models using CPLEX 20.1.0.0 in AMPL modelling language on a personal computer with AMD Ryzen 7 4.2 GHz processor and 8 GB dedicated RAM.

Table 4.1: Computing times (in seconds) of the solved instances within 60 seconds (and the number of solved instances out of 50 instances) on newly generated instances

Aisles	Items	Single-Block			Two-Block		
		TSP-MTZ	SCS	SCS+	TSP-MTZ	SCS	SCS+
10	10	0.33 (50/50)	0.07	**0.02**	0.27 (50/50)	0.13	**0.06**
10	20	10.87 (49/50)	0.07	**0.03**	11.16 (45/50)	0.15	**0.10**
10	30	43.03 (25/50)	0.07	**0.03**	23.47 (35/50)	0.19	**0.12**
20	10	0.53 (50/50)	0.10	**0.03**	0.56 (50/50)	0.58	**0.15**
20	20	22.74 (34/50)	0.10	**0.04**	18.87 (42/50)	1.10	**0.23**
20	30	53.65 (9/50)	0.12	**0.07**	36.71 (26/50)	1.72	**0.30**
30	10	0.48 (50/50)	0.14	**0.03**	0.41 (50/50)	1.96	**0.18**
30	20	41.80 (24/50)	0.23	**0.05**	29.84 (29/50)	4.55	**0.45**
30	30	59.08 (2/50)	0.32	**0.08**	43.23 (21/50)	10.20	**0.81**

In our first set of experiments, we compare the computing time performances of the single- and two-block versions of SCS and SCS+ to each other, and as well as to those of the TSP-MTZ. Here we generate instances with 10, 20 or 30 aisles and 10, 20 or 30 pick items for both single- and two-block warehouses. We set a time limit of 60 seconds for the computing time comparison, as the OPP needs to be solved repetitively in short intervals for each pick list. Table 4.1 shows the average computing times and the number of solved instances within 60 seconds. We leave out the latter for SCS and SCS+ as they solve all instances within the time limit. The results show that the SCS+ formulation solves all instances for the single- and two-block warehouses to optimality in less than a second on average.

Comparing our formulations to TSP-MTZ, the proposed single- and two-block formulations outperform the corresponding TSP-MTZ formulations at all instances. TSP-MTZ is unable to solve part of the instances when the pick list size exceeds 20. Comparing single- to two-block formulation, we note that there is an increasing difference between computing times with the increase in the number of aisles large due to the need of more disconnectivity elimination constraints for the two-block layouts. Nevertheless, the computing time of SCS+ for two-block layouts is still under 0.80

seconds on average even for the largest set of instances. Comparing SCS+ to SCS, we continue to note the significant contribution of the use of lazy constraints when applied for the two-block layouts.

With our second set of experiments, we compare the computing time performances of SCS+ for the two-block layout with the performance of the TSP-based formulations, SHSW (Scholz et al. 2016) and S (Scholz 2016). For this end, we generate two-block OPP instance sets in line with Scholz (2016) where we set the number of picking aisles as 5, 10, 15, 20, 25 or 30, aisles, the picking sub-aisle-length as 26 time units, and the number of pick items as 30, 45, 60, 75 or 90. This experiment is the two-block version of the experiment in Chapter 3. In line with Scholz et al. (2016) and Scholz (2016), we generate 30 instances for each setting. Table 4.2 presents the average computing times and the number of solved instances of SHSW, S and SCS+ within a time limit of 30 minutes. We again leave out the number of solved instances for SCS+ as they solve all instances within the time limit. SHSW is able to solve all instances to optimality within the time limit for the setting with 5 and 10 aisles, however the number of solved instances decreases quickly for the settings with 15 aisles and more. Formulation S performs relatively better and solves all instances to optimality except part of those with 30 aisles. The results show that the performance of SCS+ significantly outperforms TSP-based formulations, Scholz et al. (2016) and Scholz (2016) as it solves even the largest set of instances to optimality in 2.21 seconds on average. We also note that unlike the previous experiments, the computing times only grows moderately as the number of pick locations increases. Moreover, the computing times start decreasing as the ratio of the number of pick items to that of aisles increases, particularly for the settings with 90 pick locations and less than 25 aisles.

Our third experiments compare the two-block formulation with those of Pansart et al. (2018) and Su et al. (2022), denoted as PCC+ and SZ+ respectively, on the two-block random instance settings in Theys et al. (2010). As in Pansart et al. (2018) and Su et al. (2022), we generate instance sets with 5, 15 or 60 aisles; 15, 60 or 240 pick locations, 11 time units for the aisle length, and 10 instances in each set. Although this settings are far from applicability, it can better compare the computing time performance of the state-of-the-art formulations Table 4.3 depicts the average

Table 4.2: Comparison of computing times (in seconds) and the proportions of solved instances on Scholz (2016)'s two-block instances with 30-minute time limit

Aisles	Items	SHSW	S	SCS+
5	30	0.78 (30/30)	0.44 (30/30)	**0.05**
5	45	0.71 (30/30)	0.47 (30/30)	**0.05**
5	60	0.67 (30/30)	0.46 (30/30)	**0.04**
5	75	0.74 (30/30)	0.52 (30/30)	**0.04**
5	90	0.78 (30/30)	0.55 (30/30)	**0.03**
10	30	14.29 (30/30)	1.03 (30/30)	**0.10**
10	45	12.17 (30/30)	1.42 (30/30)	**0.13**
10	60	13.69 (30/30)	1.38 (30/30)	**0.14**
10	75	10.85 (30/30)	1.58 (30/30)	**0.10**
10	90	10.03 (30/30)	1.42 (30/30)	**0.09**
15	30	428.07 (27/30)	6.08 (30/30)	**0.23**
15	45	351.52 (29/30)	6.54 (30/30)	**0.25**
15	60	355.98 (29/30)	19.50 (30/30)	**0.25**
15	75	271.85 (30/30)	13.10 (30/30)	**0.24**
15	90	478.34 (27/30)	47.94 (30/30)	**0.22**
20	30	1000.09 (17/30)	6.91 (30/30)	**0.34**
20	45	1010.82 (16/30)	16.44 (30/30)	**0.39**
20	60	1010.61 (15/30)	53.12 (30/30)	**0.55**
20	75	1111.41 (17/30)	113.47 (30/30)	**0.52**
20	90	1015.24 (17/30)	168.85 (30/30)	**0.49**
25	30	1695.72 (2/30)	20.02 (30/30)	**0.65**
25	45	1727.68 (2/30)	44.54 (30/30)	**0.72**
25	60	1517.52 (7/30)	162.40 (30/30)	**1.27**
25	75	1654.45 (5/30)	270.24 (30/30)	**1.12**
25	90	1529.55 (8/30)	317.64 (28/30)	**1.24**
30	30	1800.00 (0/30)	59.00 (30/30)	**0.90**
30	45	1752.28 (1/30)	184.08 (29/30)	**1.77**
30	60	1800.00 (0/30)	330.98 (27/30)	**2.11**
30	75	1800.00 (0/30)	581.39 (25/30)	**2.13**
30	90	1800.00 (0/30)	904.34 (20/30)	**2.21**

Table 4.3: Computing times (in seconds) of the solved instances within 30 minutes on Theys et al. (2010) two-block instances

Aisles	Items	PCC+	SZ+	SCS	SCS+
5	15	0.15	0.53	0.05	**0.03**
5	60	1.46	1.32	0.05	**0.03**
5	240	32.15	2.91	0.05	**0.04**
15	15	0.20	0.67	0.34	**0.13**
15	60	55.92	9.59	0.95	**0.22**
15	240	1800.00	976.00	0.48	**0.09**
60	15	**0.81**	1.07	38.249	1.38
60	60	1800.00	86.59	934.60	**12.01**
60	240	1800.00	1800.00	1553.03	**18.44**

computing times for PCC+, SZ+, SCS and SCS+ with a time limit of 30 minutes. PCC+ solves all intances to optimality in 6 out of 9 settings in less than a minute on average, however it is not able to solve any of the instances within the 30 minute-time limit for the settings with 15 aisles, 240 pick locations, and 60 aisles, 60 and 240 pick locations. SZ+ performs relatively better as it solves all the instances within the limits except the case of 60 aisles and 240 pick locations. Of our proposed formulations, SCS+ solves all instances in all settings within the time limit (under 20 seconds on average for the largest set of instances) while SCS also solves all instances except 5 out of 10 instances with 60 aisles and 240 pick items. More importantly, SCS+ significantly outperforms its counterparts in all except one set of instances, where its average difference from the best result is half a second. Moreover, the average of 0.09 seconds for the setting of 15 aisles and 240 pick locations shows that the computing times are particularly shorter as the ratio of the number of pick locations to the number of aisles is larger.

As the last experiment, we analyze the performance of the Z-shape heuristic on multi-block layouts by comparing it with four well-known simple heuristics explained in the literature section on a set of instances with up to 5 blocks where there are 7 or 15 aisles, each with a length of 10 or 30 time units, and 10 or 15 pick items in the picking list, as in line with those in the literature (Roodbergen & De Koster 2001a). During the analysis, we only focus on the easy-to-memorize heuristics hence the heuristics which results in relatively more complex routes are left out of consideration (e.g., Theys et al. 2010, Çelik & Süral 2019).

Following Roodbergen & De Koster (2001a), we use a set of 2,000 instances for each combination. In all settings, the horizontal length between two adjacent aisles set to 2 time units. The end of a sub-aisle (length) of a block also refers to the center of the back cross-aisle of that block and the front cross-aisle of the next block. Table 4.4 gives the average travel times of the heuristics for the multi-block OPP instances. For each combination, the best heuristic is indicated in bold-underlined.

Table 4.4 shows that the Z-Shape heuristic outperforms all other heuristics for large size of aisle lengths and large number of blocks. It performs worse only when there is a two-block layout or when the picking aisle is short. As the picking aisles are deeper in practice, we conclude that Z-Shape algorithm performs the best in real-world scenarios. It performs poorly for two-block layouts because of the unnecessary double cross-aisle movement on the front cross-aisle to reach to the depot. The negative effect of these unnecessary movements disappears quickly when the number of blocks is more than two. We also observe that the combined heuristic outperforms other heuristics when the picking aisles are short. In numbers, the combined heuristic has the best performance in 24 of the 40 settings, the largest gap strategy has the best performance in 1 setting, and Z-Shape algorithm has the best performance in 15 of the 40 settings.

4.6 Concluding Remarks

This chapter extends the arc routing-based formulation for the single-block layouts to two-block OPP in parallel-aisle warehouses. Based on similar observations on the states of the blocks, the occurrences of disconnectivities and the possible movements we have modified the index sets and the parameters to incorporate the existence of a middle aisle in the formulation. Then, we present the assignment, sequencing and degree constraints, which are straightforward extensions of their counterparts in the single-block model. More importantly, we have analyzed change in the ways and

Table 4.4: Average travel time for picker routing heuristics (in seconds) for multi-block layout settings

Method	Aisles	Items	Length	Number of Blocks				
				1	2	3	4	5
Z-Shape	7	10	10	74.7	102.0	114.3	132.7	150.8
	7	10	30	**163.0**	220.9	**258.6**	**304.4**	**353.4**
	15	10	10	114.2	156.0	165.4	188.0	207.1
	15	10	30	**219.0**	286.2	**318.1**	**367.0**	**415.3**
	7	15	10	87.4	125.7	140.8	163.0	178.8
	7	15	30	196.9	275.6	**318.7**	**367.2**	**413.0**
	15	15	10	136.6	192.1	202.7	235.9	251.5
	15	15	30	**273.9**	362.6	**398.2**	**454.4**	**495.8**
S-Shape	7	10	10	79.2	100.4	122.6	141.0	159.5
	7	10	30	191.9	255.9	309.1	354.4	403.4
	15	10	10	127.3	143.5	174.2	194.1	217.2
	15	10	30	278.0	325.0	381.0	425.3	476.8
	7	15	10	88.5	121.5	154.3	174.3	195.3
	7	15	30	218.5	318.8	393.3	445.7	499.5
	15	15	10	151.3	178.0	223.5	243.0	267.7
	15	15	30	345.4	425.9	505.3	550.4	603.2
Largest Gap	7	10	10	76.0	95.8	114.2	132.8	152.0
	7	10	30	164.1	218.8	268.3	317.4	369.1
	15	10	10	123.7	146.0	168.7	188.9	211.7
	15	10	30	228.2	280.8	332.6	382.0	436.7
	7	15	10	88.5	116.3	140.9	161.2	181.0
	7	15	30	197.5	**269.0**	329.2	382.5	435.6
	15	15	10	146.8	179.3	212.9	237.1	260.3
	15	15	30	284.0	353.3	419.9	474.1	527.7
Aisle-by-Aisle	7	10	10	70.8	97.1	126.5	157.4	188.2
	7	10	30	163.5	244.6	333.7	424.9	517.9
	15	10	10	111.2	137.8	170.1	204.1	238.4
	15	10	30	226.6	307.3	403.0	505.9	610.0
	7	15	10	80.7	116.7	154.0	192.9	231.5
	7	15	30	193.0	301.6	415.3	528.1	650.8
	15	15	10	132.3	168.9	211.3	256.8	303.1
	15	15	30	283.8	394.6	525.3	658.8	795.4
Combined	7	10	10	**70.8**	**88.8**	**111.3**	**130.6**	**150.6**
	7	10	30	163.5	**218.8**	272.1	321.3	373.3
	15	10	10	**111.2**	**126.9**	**158.7**	**180.7**	**205.5**
	15	10	30	226.6	**272.1**	332.4	381.9	438.0
	7	15	10	**80.7**	**107.1**	**137.4**	**157.6**	**178.5**
	7	15	30	**193.0**	271.4	338.9	391.7	445.1
	15	15	10	**132.3**	**154.2**	**198.8**	**219.8**	**246.1**
	15	15	30	283.8	**348.1**	425.4	475.5	530.4

types of occurences of the disconnectivities for two-block case, and formulated the disconnectivity elimination constraints accordingly. The main difference of the two-block model arises from these disconnectivity elimination constraints. Finally, we propose a simple and effective heuristic for multiple block layouts, which increases its performance with more blocks and deeper aisles.

Our findings in computational experiments section show that refraining from implementing TSP subtour elimination constraints increases the efficiency of the OPP formulations significantly as the performance of the proposed formulation for the two-block layout outperforms the best-known approaches for the two-block OPP to date. Other noteworthy findings obtained from our computational experiments continues to support (i) the significant contribution of the use of lazy constraints, (ii) the shorter computing times when the number of items to number of aisles ratio is large. Moreover, (iii) we observe that the difference between computing times of single- and two-block formulations increases with a larger number of aisles as the need of more disconnectivity elimination constraints also increases. Finally we observe that (iv) our proposed multi-block heuristic outperforms all its simple counterparts for the layouts with large number of blocks and large size of aisle lengths.

CHAPTER 5

THE MIN-MAX ORDER PICKING PROBLEM IN SYNCHRONIZED DYNAMIC ZONE-PICKING SYSTEMS

5.1 Introduction

In order picking operations, items must be collected from the warehouse to satisfy the customer demand while aiming to optimize a cost or/and service-related objective (e.g., De Koster et al. 2007, Scholz et al. 2016, van Gils et al. 2018b, Çelik & Süral 2019, Masae et al. 2020a). In the last decades, a growing competition with limited time windows has put extra pressure on order picking operations. Furthermore, the increasing demands from online retailing have resulted in many relatively small-size orders with promised time windows, which make effective order picking and workload balancing necessary (Ardjmand et al. 2018, Vanheusden et al. 2020). Consequently, warehouse managers are under increased pressure to make use of their order picking resources in an efficient manner. The literature clearly reveals the gap in integrated operational level warehouse problems considering zone picking operations. Although recent literature regarding multiple order pickers has increased significantly, the advantages of zone picking are ignored, thus, zone picking combinations have not been given any particular attention.

One way of ensuring a more efficient order picking process in terms of response time is the use of zone-picking. Zone-picking is a method of order picking where the picking area is divided into a number of zones so that the picking activity is being conducted at each zone by a different picker, sequentially or synchronously. Zone-picking also has other advantages including reduction of the travel time, achieving better workload balance, avoiding congestion within aisles, and ensuring familiarity of pick locations.

Although zone-picking and batch-picking are the two main factors influencing the performance of order picking processes at tactical and operational levels, zone-picking has received less attention in the literature among all order picking operations (Yu & De Koster 2009). *Synchronized zone-picking* refers to the zone-picking process where all pickers work on the same order simultaneously, thus creating a picking wave, whereas *dynamic zone-picking* refers to the zone-picking decisions made at the operational level, since it is possible to arrange the zone sizes and re-assign the pickers to zones at each picking wave. A dynamic zoning environment is especially important when synchronized zone-picking is considered, as fixed zones would lead to significant idle times where the pickers wait until all the pickers complete the picking activity. Hence, as opposed to static zoning, dynamic zoning is considered as an operational level decision problem and can be simultaneously addressed with other operational level problems, including picker routing and workload balancing.

In most picker routing problems, the main objective is to minimize the total travel time of the order pickers, as it constitutes more than half of the order picking time (Bartholdi & Hackman 2019). On the other hand, in synchronized zone-picking, the lead time of an order picking is determined by the longest time taken by any of the order pickers in each zone. Therefore, a min-max approach where the objective is to minimize the latest travel time of any of the pickers is not only a good way of synchronizing zone-picking operations, but it also helps ensure fairness among pickers. This is also a common objective for other wave-picking operations such as batch-picking (Ardjmand et al. 2018). For this end, in this chapter we make use of a min-max objective for the OPP in a parallel-aisle warehouse with a synchronized dynamic zone-picking system. Such zone-picking systems with a makespan minimization objective dynamically balance the workload among pickers without the need of additional operational control. To the best of our knowledge, this is the first study that defines the OPP in a synchronized zone-picking system with a min-max objective, while also simultaneously considering the picker routing, zone assignment, and workload balancing decisions. The findings of this research should be helpful in warehouses where multiple pickers are employed, and lead time requirements are stringent.

Throughout this chapter, we first present the min-max OPP without zoning and give the VRP min-max formulation. Afterwards, we discuss the OPP with dynamic zoning under the assumption that each zone of adjacent aisles is assigned to a single picker and present the relevant zoning constraints. For the min-max VRP without zoning, we then propose a heuristic approach for the two-picker case based on the exact picker routing algorithm and the knapsack problem. This approach also constitutes a baseline for our exact DP approach for the min-max VRP with zoning, which is presented afterwards. As the last approach, we present a batch-picking algorithm with a min-max objective where we similarly minimize the make-span of a wave-picking operation. Finally, we work on randomly-generated order picking instances where the pick locations in a picking list are independently and uniformly distributed over the order picking area to illustrate the algorithms and compare the performance of the solution methodologies in various computational experiments.

The remainder of this chapter is organised as follows. Section 5.2 describes the warehouse corresponding to the OPP. In Section 5.3, we develop the min-max VRP formulations for multi-picker OPP with and without zoning constraints. In Section 5.4, we present the heuristic approach for the two-picker OPP without zoning. In Section 5.5, we develop an exact approach for the min-max OPP under synchronized dynamic zone-picking policy. Finally, in Section 5.6 we present a modified Clarke & Wright saving heuristic for min-max OPP under batch-picking policy. Computational experiments are presented in Section 5.7, and the chapter is concluded in Section 5.8.

5.2 Problem Description

In both manual and automated warehouses, a combination of efficient zoning and picker routing plays an important role in improving travel time, congestion, and system throughput. This chapter considers the order picker routing problem in a dynamic and synchronized zoning environment, where the items corresponding to each customer order are picked simultaneously in multiple zones, and zones may change between different orders. The objective is to minimize the maximum time of completing the picking activities in any zone. Using a min-max type of objective not only minimizes the makespan of an order picking wave, but it also helps balance the workload of the order pickers more effectively. We present a mathematical model for the optimal solution of this problem, as well as a DP approach to find the optimal solution for the case where a zone is a set of adjacent aisles. Computational experiments on randomly generated instances show that the DP approach is able to find optimal solutions in negligible computational times.

The parallel-aisle warehouse considered in this chapter is given in Figure 5.1. It contains *cross-aisles* at *front* and *back* of the picking-aisles as in Figure 5.1(a) and may contain *middle* cross-aisles, which perpendicularly divide the warehouse into *blocks*, and thereby divide the aisles into sub-aisles, as shown on Figure 5.1 (b).

Figure 5.2 shows the graph representations of the warehouses in Figure 5.1(a) and (b), where each pick location is represented by node v_p, $p \geq 1$ while v_0 represents the depot, and the edge (p, q) represents the path providing direct access between node v_p and v_q. The nodes a_i and b_i represent the back and front intersection points of picking-aisle i, respectively. Nodes m_{ij} represent the intersection point of picking-aisles i and middle cross-aisle j. A back cross-aisle of a block is the front cross-aisle of the next block. A picker starts a feasible picking route from the depot, collects all the items in the picking list and returns to the depot. The OPP is to find such a feasible route that can be completed in the minimum time.

5.3 Min-Max OPP with Dynamic and Synchronized Zone-Picking

In this section, we propose a min-max VRP formulation to minimize the lead time of the multi-picker wave-picking process, after which we introduce zoning constraints into the formulation. The following assumptions are made to model the problem:

- A zone is a set of contiguous identical aisles. An aisle cannot belong to more than one zone. The aisles are narrow enough so that order picking can be performed simultaneously from both sides of an aisle in negligible time.
- The warehouse performs wave-picking with a min-max type objective, and we focus on one wave at a time by applying synchronized dynamic zone-picking policy to operationally control the zone sizes and therefore picker workloads.
- Each order picking tour starts and ends at the same point (the depot), and all the items in the wave are picked in one picking tour (of each picker).

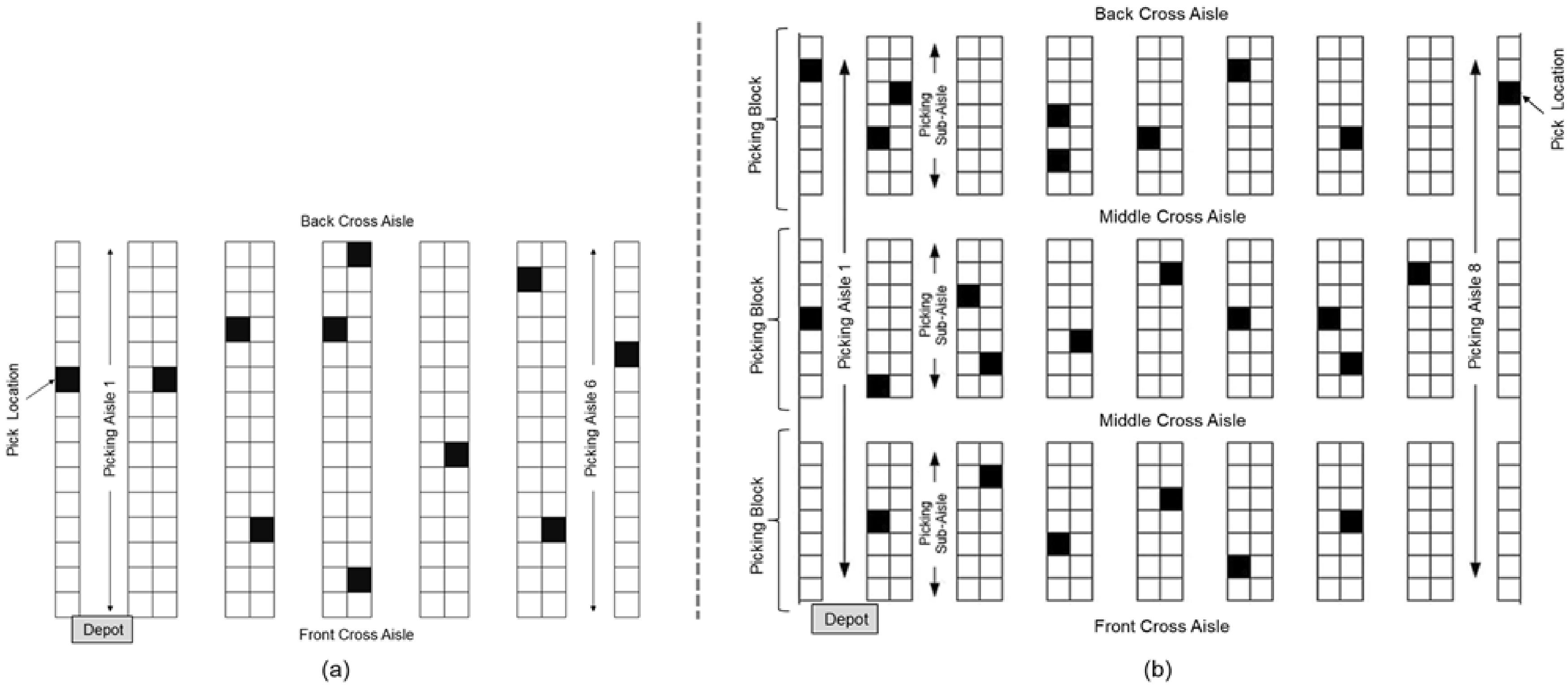
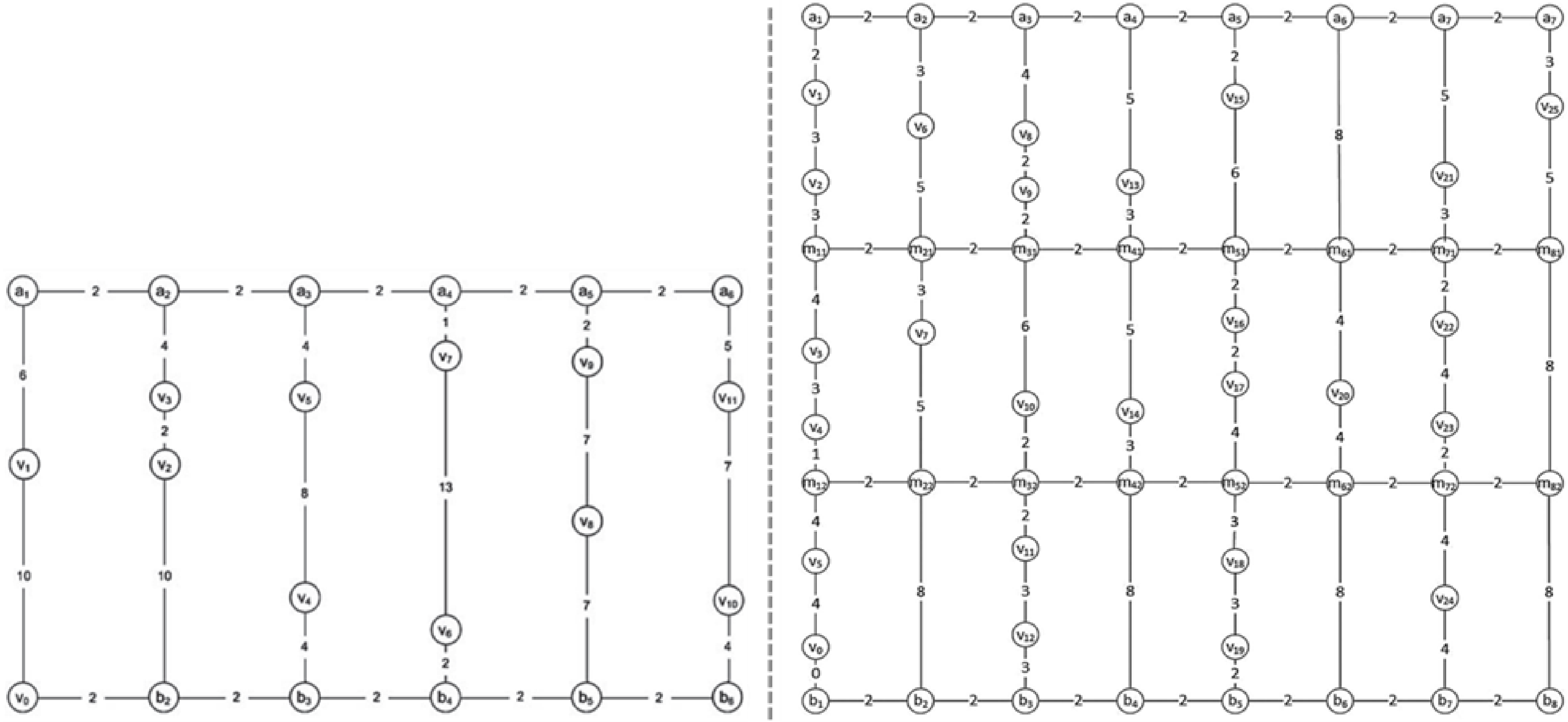


Figure 5.1: Two OPP instances. (a) A single-block warehouse with 6 aisles each with 16 time-unit length and 11 pick locations. (b) A three-block warehouse with 8 aisles each with 16 time-unit length and 25 pick locations, (Çelik & Süral 2019)



(a) (b)

Figure 5.2: The graph representations of Figure 5.1(a) and (b), (Çelik & Süral 2019)

5.3.1 VRP Formulation for the Min-Max OPP

Minimum total time travelled in a multi-picker OPP can be obtained by applying a two-index VRP variant of commodity flow formulation by Gavish & Graves (1978) (VRP-MINSUM). However, in our case, using a two-index formulation would not help to balance the workload of pickers, since these indices refer only to the pick locations (vertex and edge sets), not to the pickers. To consider balancing, we require a three-index formulation, where the additional index k is introduced for pickers to clarify which picker traverses the edge (p, q):

$$x_{pqk} = \begin{cases} 1, & \text{if } k^{\text{th}} \text{ picker traverses edge } (p,q) \\ 0, & \text{otherwise} \end{cases}$$

Let us define N as the set of all pick locations including the depot v_0, and K as the set of all pickers. Also let c_{pq} be the distance of edge (p, q) and g_{pqk} be the number of units of the commodity passed onto pick location $q \in N$ directly from pick location $p \in N$ by picker $k \in K$. The following single commodity flow formulation aims to minimize the maximum time travelled by any one of the pickers (VRP-MINMAX).

$$\min \quad L \tag{5.1}$$

$$s.t. \quad \sum_{k \in K} \sum_{\substack{p \in N \\ p \neq q}} x_{pqk} = 1 \qquad q \in N \setminus \{0\} \tag{5.2}$$

$$\sum_{k \in K} \sum_{\substack{q \in N \\ q \neq p}} x_{pqk} = 1 \qquad p \in N \setminus \{0\} \tag{5.3}$$

$$\sum_{q \in N \setminus \{0\}} x_{0qk} = 1 \qquad k \in K \tag{5.4}$$

$$\sum_{p \in N \setminus \{0\}} x_{p0k} = 1 \qquad k \in K \tag{5.5}$$

$$\sum_{\substack{p \in N \\ p \neq q}} x_{pqk} = \sum_{\substack{p \in N \\ p \neq q}} x_{qpk} \qquad q \in N \setminus \{0\}, k \in K \tag{5.6}$$

$$\sum_{k \in K} \sum_{\substack{p \in N \\ p \neq q}} g_{pqk} - \sum_{k \in K} \sum_{\substack{p \in N \setminus \{0\} \\ p \neq q}} g_{pqk} = 1 \qquad q \in N \setminus \{0\} \tag{5.7}$$

$$g_{pqk} \leq (|N| - |K|) x_{pqk} \qquad p \in N, q \in N \setminus \{0\}, p \neq q, k \in K \tag{5.8}$$

$$\sum_{p \in N} \sum_{q \in N} c_{pq} x_{pqk} \leq L \qquad k \in K \tag{5.9}$$

$$x_{pqk} \in \{0, 1\} \qquad p, q \in N, p \neq q, k \in K \tag{5.10}$$

$$g_{pqk} \geq 0 \qquad p, q \in N, p \neq q, k \in K \tag{5.11}$$

In this formulation, the objective function 5.1 minimizes the time travelled by the latest picker. Constraints 5.2 and 5.3 state that each pick location should be visited exactly once by a picker. Constraints 5.4 and 5.5 state that exactly $|K|$ pickers should leave and return to the depot. Constraint 5.6 guarantees a tour for each picker. Constraint 5.7 ensures that exactly one unit of commodity is left to each pick location and constraint 5.8 states that no commodity is passed through the arcs that are not included in the picker tour. The main constraints leading to weakness of the formulation is the big-M constraint 5.8. $(|N| - |K|)$ is used as the big-M, since there are $|N|$ pick locations including the depot, and there are $|K|$ pickers, each of whom requires visiting at least one pick location. So, each picker can deliver at most $(|N| - |K|)$ units of the commodity. Constraint 5.9 ensures that the time travelled by each picker cannot be more than the longest tour, L, which is to be minimized in the objective function. Finally Constraints 5.10 and 5.11 define the domains of the decision variables.

5.3.2 VRP-MINMAX Formulation with Zone-Picking Constraints

To incorporate zone-picking constraints into the formulation in the preceding section, we assume the warehouse follows a synchronized zone-picking policy where all pickers start simultaneously to better control throughput times. When pickers complete the picking tour, they wait for the completion of the overall picking wave at the depot. A zone is defined as a set of adjacent aisles and zones do not necessarily have the same number of aisles. A picker can be assigned at most one zone and an aisle cannot belong to more than one zone. We focus on minimizing the lead time of a picking wave; hence, workload balancing is also achieved.

In such a case, another decision variable is required to introduce aisles and zones into the model. Let M be the set of all aisles and parameter e_p be the aisle on which the item $p \in N$ is located. Then define:

$$y_{ik} = \begin{cases} 1, & \text{if } k^{\text{th}} \text{ picker is assigned to aisle } i \in M \\ 0, & \text{otherwise} \end{cases}$$

Now we can use the min-max formulation presented in the previous section as a basis to come up with the VRP formulation with zone-picking constraints. The following formulation (referred to as VRP-Z), together with Constraints 5.2 - 5.11, aims to minimize the maximum travel time subject to zoning constraints.

$$\min \quad L \tag{5.12}$$

$$s.t. \quad y_{11} + y_{|M|,|K|} = 2 \tag{5.13}$$

$$\sum_{k \in K} y_{ik} = 1 \qquad i \in M \tag{5.14}$$

$$\sum_{i \in M} y_{ik} \geq 1 \qquad k \in K \tag{5.15}$$

$$2x_{pqk} \leq y_{(e_p)(k)} + y_{(e_q)(k)} \qquad p, q \in N \setminus \{0\}, p \neq q, k \in K \tag{5.16}$$

$$y_{ik} \leq y_{(i+1)k} + y_{(i+1)(k+1)} \qquad i \in M \setminus \{|M|\}, k \in K \setminus \{|K|\} \tag{5.17}$$

$$y_{ik} + y_{(i+1)(k-1)} \leq 1 \qquad i \in M \setminus \{|M|\}, k \in K \setminus \{1\} \tag{5.18}$$

$$y_{ik} \in \{0, 1\} \qquad i \in M, k \in K \tag{5.19}$$

Constraint 5.13 guarantees that the first picker is assigned to the first aisle and the last picker is assigned to the last aisle. Constraint 5.14 ensures that each aisle is assigned to exactly one picker. Constraint 5.15 guarantees that each picker is assigned to at least one aisle. Constraint 5.16 is the linking constraint and guarantees that if k^{th} picker is traversing the edge (p, q), then the aisles, in which items p and q are located, are assigned to picker k. Constraints 5.17 and 5.18 ensure that a zone is a set of adjacent aisles such that if picker k is assigned to aisle i, then the next aisle is assigned either to picker k or picker $k + 1$ and no further aisle can be assigned to the previous pickers.

5.4 A Dynamic Programming-Based Heuristic for the Min-Max OPP with Two Pickers

The integer programming formulations in Section 5.3 are computationally challenging to solve for larger instances, as will become apparent in Section 5.7. We first con-

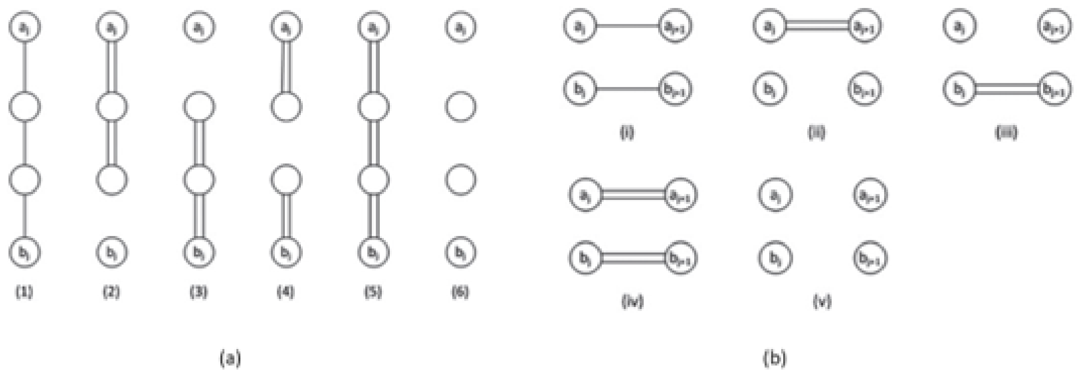


Figure 5.3: (a) Possible intra-aisle connection types, (b) Possible inter-aisle connection types, (Ratliff & Rosenthal 1983)

sider the case with two pickers and propose a heuristic approach for VRP-MINMAX, which also serves as the baseline for the exact approach for VRP-Z with a general number of pickers. The heuristic introduces a simple but effective DP algorithm for the min-max OPP without zoning. Although the heuristic tackles the problem without zoning, it uses the idea of assigning pickers to aisles, which may be considered as constituting temporary zones. This is followed by an improvement scheme (*the travel time balancing algorithm*) derived from the knapsack problem, to reduce the optimality gap at the expense of aisle-zone integrity.

For the single-picker OPP, Ratliff & Rosenthal (1983) provide an exact algorithm, the RR algorithm, consisting of $|M|$ stages where $|M|$ is the number of aisles. In this algorithm, there are two sub-stages for each aisle. Each stage has a number of states called equivalence classes represented by: (i) degree parity at the back of an aisle, (ii) degree parity at the front of an aisle, and (iii) number of connected components. Degree parities can be zero (0), even (E) or odd (U), while connectivity can be $0C$, $1C$ or $2C$. A partial tour can be represented by one of the six equivalence classes. These states are updated along the stages using two classes of connection types: intra-aisle and inter-aisle, as shown in Figure 5.3. At each stage, minimum tour lengths for each state are found by adding the related possible connection types to the minimum tour lengths of previous sub-stage. At the last aisle, the partial tour with the minimum length sum is determined, which yields the optimal solution for the single-picker OPP.

Before developing the proposed algorithms at this and following sections, we primar-

Table 5.1: Notation used in the proposed algorithms

M	Set for the aisles; $M = \{1, 2, \ldots, \{\|M\|\}$
$RR(i, j)$	Minimum tour length solution of the RR algorithm starting from depot, entering the zone starting with aisle i and exiting to return to the depot from aisle j
$minL_i^1$	Minimum tour length for picker 1 when aisle i is set as the stopping aisle i.e., when first i aisles are assigned to picker 1.
$minL_i^2$	Minimum tour length for picker 2 when aisle i is set as the stopping aisle i.e., when last $\|M\| - i$ aisles are assigned to picker 2.
$F(i, k)$	Cost-to-go function, which returns the lead time of the picking wave, i.e., the travel time of the latest picker, in Section 5.5, for optimally assigning $k \in K$ pickers to $i \in M$ aisles

ily summarize the notation in Table 5.1.

5.4.1 Construction of Temporary Zones

In the first phase of the heuristic, we apply $R(i, j)$ repeatedly for each picker after assigning their starting and ending aisles. Initially, the first aisle is assigned to picker 1 and the last $|M| - 1$ aisles are assigned to picker 2, thus aisle 1 is set as the stopping aisle. Then, minimum tour lengths are calculated for $RR(1, 1)$ and $RR(2, |M|)$, and called $minL_1^1$ and $minL_1^2$ respectively. Subsequently, aisle 2 is set as the stopping aisle and minimum tour lengths, $minL_2^1$ and $minL_2^2$, are calculated again. The stopping aisle i is increased in this manner until the minimum tour length for picker 1 exceeds the one for picker 2, i.e., $minL_i^1 \geq minL_i^2$. At this final stage, the first i aisles are assigned to picker 1 and the remaining $|M| - i$ aisles are assigned to picker 2 where picker 1 travels at least as many time units as picker 2. Note that, at $(i-1)^{th}$ stopping aisle stage, $minL_{i-1}^1 \leq minL_{i-1}^2$. At this previous stage, the first $i - 1$ aisles are assigned to picker 1 and the remaining $|M| - i + 1$ aisles are assigned to picker 2 where picker 1 travels at most as many time units as picker 2.

The aim of the two-picker algorithm is to return the smaller wave time, i.e., smaller maximum time travelled, at the final and the previous stages. Before concluding the two-picker algorithm, we introduce an improvement algorithm in the following subsection as the second phase, which further reduces the wave time by minimizing the absolute travel time difference between the two pickers by focusing on the final and the previous stages.

5.4.2 Travel Time Balancing Algorithm

In the first phase of the heuristic, there is an aisle-zone integration and also one of the pickers will travel more than the other picker. The difference, if any, could be further reduced to more balance the time travelled by each of two pickers at the expense of aisle-zone integration. Reassignment of moves from one picker to the other is likely to improve the workload balance at the expense of zone integrity. We observe that, at the final stage, picker 1 travels at least as many time units as picker 2 where some possible front movements made by picker 1 can be assigned to picker 2. There is also a similar case for the previous stage. Some movements made by picker 2 can be assigned to picker 1. For this end, we develop an exact DP algorithm called travel time balancing algorithm.

Reassignment of moves from one picker to the other is described as follows. Since picker 2 starts from depot and uses the front cross-aisle to reach the starting aisle point, i.e., $b_{(i+1)}$, picker 2 is only allowed to make additional connection type 3 intra-aisle movements in aisles assigned to picker 1 where $j = \{1, 2, \ldots, i\}$. Moreover, picker 2's connection type 3 movement in aisle j is only possible as long as the optimum movement made by picker 1 in aisle j at the final stage is a connection type 3 or connection type 4 movement, because otherwise the connectivity of picker 1's tour will be lost. Finally we note that if the optimum movement of picker 1 is a connection type 4 movement, the front part of the movement should be considered as a candidate connection type 3 movement taken from picker 1 and given to picker 2.

The problem stated here can be described as a variant of the knapsack problem that fills the travel time difference between two pickers as much as possible. Suppose we solve a 0-1 knapsack problem having capacity W with i items each with weight w_j and value v_j with the additional notation in Table 5.2.

Then, the objective for the 0-1 knapsack problem is to select a subset of items to

Table 5.2: Additional notation used in travel time balancing algorithm

W	Twice the size of the difference between pickers' travel times, $2(minL_i^1 - minL_i^2)$
$w_j = v_j$	Connection type 3 movement length for aisle $j \in \{1, 2, \ldots, i\}$
w	Travel time difference index between the two pickers; $w \in \{1, 2, \ldots, W\}$
$K_1(j, w)$	The maximal movement obtainable when filling a knapsack of capacity w time units using reassignable movements among aisles from 1 to j.
$K_2(j, w)$	The minimum difference between pickers obtainable when filling a knapsack of capacity w time units using reassignable movements among aisles from 1 to j

maximize the total value while satisfying the capacity constraint. K_1 value function stated as the following recursive formula refers to the knapsack DP algorithm. If it were just a knapsack problem, the optimal value would be stored at $K_1(i, W)$ where i refers to the last aisle for picker 1:

$$K_1(j, w) = \begin{cases} K_1(j-1, w), & w_{(j-1)} \geq w \\ max\Big(K_1(j-1, w), K_1(j-1, w - 2w_{j-1}) + w_{j-1}\Big), & \text{otherwise} \end{cases} \tag{5.20}$$

where the base cases are $K_1(0, w) = 0$ for $w \in \{1, 2, \ldots, W\}$ and $K_1(j, 0) = 0$ for $j \in \{1, 2, \ldots, i\}$. However, our objective is not to maximize the value. We would like to reach as close as possible to half of the knapsack capacity, $(minL_i^1 - minL_i^2)$, either above or below. Hence, we aim to choose a subset of connection type 3 intra-aisle movements from first i aisles, take it from picker 1 and give it to picker 2 to minimize the difference between two pickers. To solve this problem, we use another value function storing the actual difference. Such values can be stored at K_2 value function as:

$$K_2(j, w) = \Big|minL_i^1 - minL_i^2 - 2K_1(j, w)\Big| \tag{5.21}$$

The optimal subset of movements, $K_1(j, W)$, is the one which minimizes the difference $K_2(i, W)$. Thus, this subset can be taken from picker 1 and given to picker

1. We note that $K_2(j, w)$ is found by subtracting the twice of the optimal subset of movements from the difference between pickers. This can be explained by the fact that $K_1(j, w)$ value is taken from picker 1 and given to picker 2, thus multiplied by 2. A pseudocode of the travel time balancing algorithm is depicted in Algorithm 1.

Algorithm 1: Travel Time Balancing Algorithm

```
for aisle j = 1 to i do
    for difference w = 0 to W do
        if j = 1 or w = 0 then
            K_1(j, w) = 0
        else if 2w_{j-1} ≤ w then
            K_1(j, w) = max( K_1(j − 1, w), K_1(j − 1, w − 2w_{j-1}) + w_{j-1} )
        else
            K_1(j, w) = K_1(j − 1, w)
        end
        K_2(j, w) = |minL_i^1 − minL_i^2 − 2K_1(j, w)|
    end
end
for aisle j = 1 to i do
    for difference w = 0 to W do
        if temp ≥ K_2(j, w) then
            temp ← K_2(j, w)
            Δ_i = K_1(j, w)
        end
    end
end
minL_i^1 = minL_i^1 − Δ_i
minL_i^2 = minL_i^2 − Δ_i
```

By applying Algorithm 1, we have further strengthen the results for the final stage. A similar method can be applied to the previous stage to reduce the difference by assigning possible movements made by picker 2 to picker 1. Finally, the maximum time travelled at the final stage and the previous stage are compared and the smaller of these two maximums is found.

$$minmax_h = min\Big(max(minL_i^1, minL_i^2), max(minL_{i-1}^1, minL_{i-1}^2)\Big) \tag{5.22}$$

In consequence, the heuristic returns $minmax_h$, the best split between two pickers that has the smallest wave time. Here, we find a near optimal min-max solution for the two-picker min-max OPP as we also balance the workload of the pickers and use the minimum-time routes. The pseudocode of the two-picker min-max OPP algorithm is depicted in Algorithm 2.

Algorithm 2: The Two-Picker Min-Max OPP Algorithm

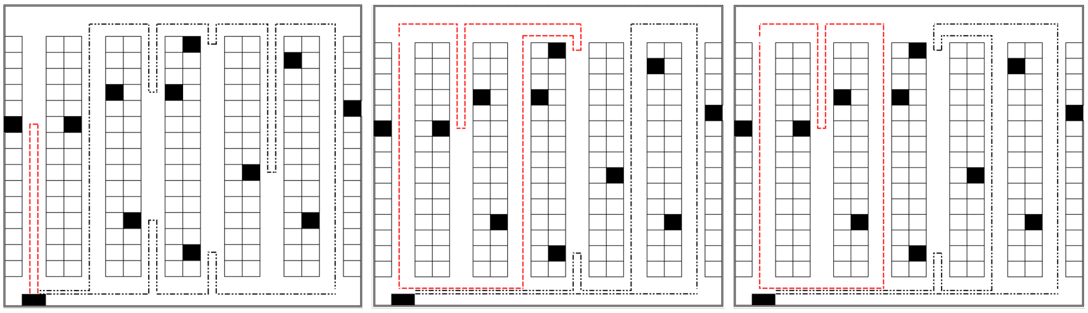
for *aisle* $i = 1$ **to** $|M|$ **do**
- $minL_i^1 \leftarrow RR(1, i)$
- $minL_i^2 \leftarrow RR(i+1, |M|)$
- **if** $minL_i^1 \geq minL_i^2$ **then**
  - Apply Algorithm 1 for stage i
  - Apply a simpler reassignment for stage $(i-1)$
  - $minmax_h \leftarrow$ $min\Big(max(minL_i^1, minL_i^2), max(minL_{i-1}^1, minL_{i-1}^2)\Big)$
  - **break**
- **end**

end

5.4.3 Numerical Example

For the example given in Figure 5.1(a), we demonstrate the solutions of MINSUM, MINMAX and $minmax_h$, and the first phase of the two-picker OPP algorithm in Figures 5.4(a), (b) and (c), respectively.

The VRP-MINSUM model minimizes the total time travelled at 112 time units, with a maximum travel time of 92 units. The VRP-MINMAX model minimizes the maximum time travelled at 58 units. Moreover, the difference between pickers is only 2 units. The first phase of the proposed heuristic yields an approximate solution for the VRP-MINMAX problem with 62 units. The difference between the travel times of the pickers is 10 units. Finally, this difference is further reduced by the balancing heuristic at the expense of aisle-zone integrity using the travel time balancing



(a) (b) (c)

Figure 5.4: The minimum time solutions for the example in Figure 1(a). (a) VRP-MINSUM, (b) VRP-MINMAX and the two-picker OPP algorithm, (c) initial phase of the two-picker OPP algorithm

Table 5.3: The solution approach of the two-picker min-max OPP algorithm

Picker 1		Picker 2			
Aisles	Lead Time	Aisles	Lead Time	Abs.Dif.	Remarks
{}	0	{1,2,3,4,5,6}	**104**	104	1P-OPP
{1}	20	{2,3,4,5,6}	92	72	
{1,2}	36	{3,4,5,6}	76	40	
{1,2,3}	52	{4,5,6}	**62**	10	Phase-1
{1,2,3,4}	66	{5,6}	52	14	
{1,2,3,4}	**58**	{4,5,6}	56	2	Phase-2

algorithm, which yields the same solution of the VRP-MINMAX formulation. The solution approach of the complete two-picker algorithm is shown on Table 5.3.

In the following section, we extend the temporary zoning approach of this section to develop an exact approach for the min-max OPP under dynamic zoning and a general number of pickers. This would be tedious by extending the algorithm suggested for two-picker case. Thus, we propose an exact and more efficient DP algorithm based on the graph representation of Pascal's Triangle.

5.5 An Exact Algorithm for the Min-Max OPP under Synchronized Dynamic Zone-Picking

In this section, we propose an exact and efficient DP approach for the min-max OPP in synchronized dynamic zone-picking systems. For this end, we use Pascal's Triangle and develop an exact algorithm which runs in polynomial time.

Contrary to tactical level zone-picking policy, there are no dedicated zones assigned to pickers at the operational level. Zones are assigned to pickers at each picking wave. Thus, the zone-picking problem is reduced to an operational level decision integrated with routing and workload balancing problems. In this way, we solve an integrated zone-picking, picker routing and workload balancing problem where each zone consists of a certain number of aisles.

In the proposed algorithm for the min-max OPP in synchronized dynamic zone-picking systems, we first determine all minimum tour lengths for each possible zone configuration. Based on this information, we recursively look at the optimal combination of zone assignments that minimizes the longest-time tour length for a given number of pickers.

Proposition 1 states the required number of minimum tour length information prior to zone assignment.

Proposition 1. *RR algorithm can be calculated for $\frac{(|M|)(|M|+1)}{2}$ different zone configurations for an $|M|$-aisle picking area in polynomial time.*

Proof. This arises from the assumption that a zone is a set of adjacent aisles. There is only a single possibility for a zone consisting of $|M|$ aisles. RR algorithm runs in $\mathcal{O}(|M|)$. There are two possibilities to construct a zone consisting of $(|M|-1)$ aisles with $\mathcal{O}(|M|-1)$ running time for each. There are $(|M|-1)$ different ways to form zones consisting of 2 aisles with $\mathcal{O}(2)$ running time for each. Finally, there are $|M|$ different ways to form a 1-aisle zone configuration with $\mathcal{O}(1)$ running time for each. Then, for the total number of routing calculation, the proof is the same as the one of sum of finite arithmetic series formula by induction, thus $\frac{(|M|)(|M|+1)}{2}$. When the running times are also included in the summation, the overall running time ends up with:

$$
\begin{aligned}
&= (1)(|M|) + (2)(|M|-1) + \ldots + (|M|-1)(2) + (|M|)(1) \\
&= \sum_{r \in M} r(|M| + 1 - r) \\
&= (|M|+1) \sum_{r \in M} r - \sum_{r \in M} r^2 \\
&= (|M|+1) \frac{(|M|)(|M|+1)}{2} - \frac{(|M|)(|M|+1)(2|M|+1)}{6} \\
&= \frac{(|M|)(|M|+1)(|M|+2)}{6}
\end{aligned}
$$

□

Proposition 2 states the number of possible zones for picker k and implies that zone-assignment problem can be studied using combinational calculations.

Proposition 2. *There are $\binom{|M|-1}{k-1}$ different zone configurations to assign k pickers to $|M|$ aisles in an $|M|$-aisle picking area.*

Proof. This can be shown by the "Stars and Bars" technique by Feller (2008). Suppose $|M|$ aisles (stars) are fixed and $(|M| - 1)$ gaps between aisles, in each of which there may or may not be a “bar”. A zone configuration is obtained by ‘bar’ing $(k - 1)$ of these $(|M| - 1)$ gaps. □

As an example, a visual representation on which k pickers are assigned to 6 aisles where $k \in \{1, 2, \ldots, \}$ is shown on Figure 5.5.

Next, we present two corollaries to better define the later DP recursion.

Corollary 1. *The $\binom{|M|-1}{k-1}$ possible zone configurations can be represented in a Binomial Expansion where each row represents the number of aisles and each “top-right to bottom-left diagonal line” represents the number of pickers as shown on Figure 5.6. This can be coded down where each row represents the number of aisles and each column represents the number of pickers.*

The coefficients in the Binomial Expansion also correspond to the entries of Pascal’s Triangle such that the $\binom{i}{k}^{\text{th}}$ coefficient in the Binomial expansion is equal to the entry at aisle i, picker k in the Pascal’s Triangle.

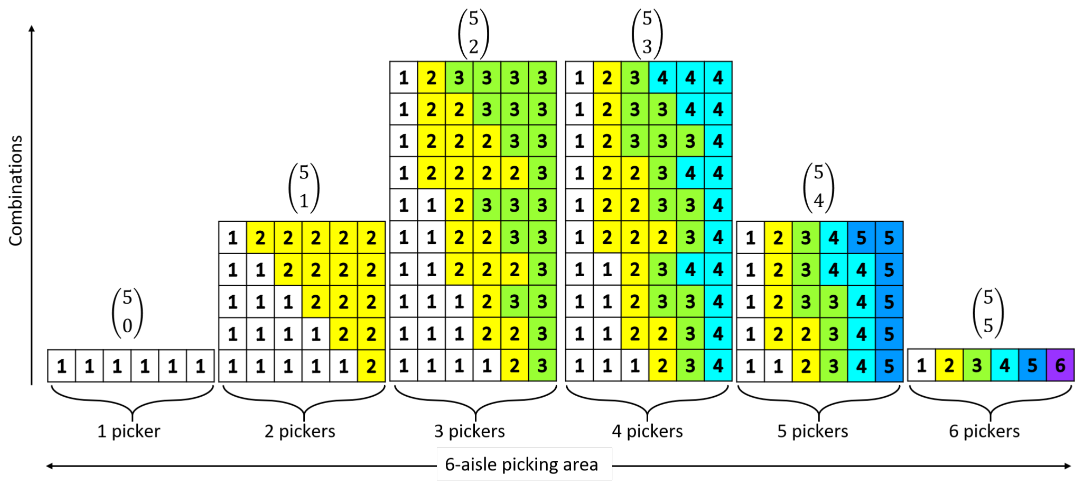


Figure 5.5: Visual representation of 6-aisle picking area zone-assignment

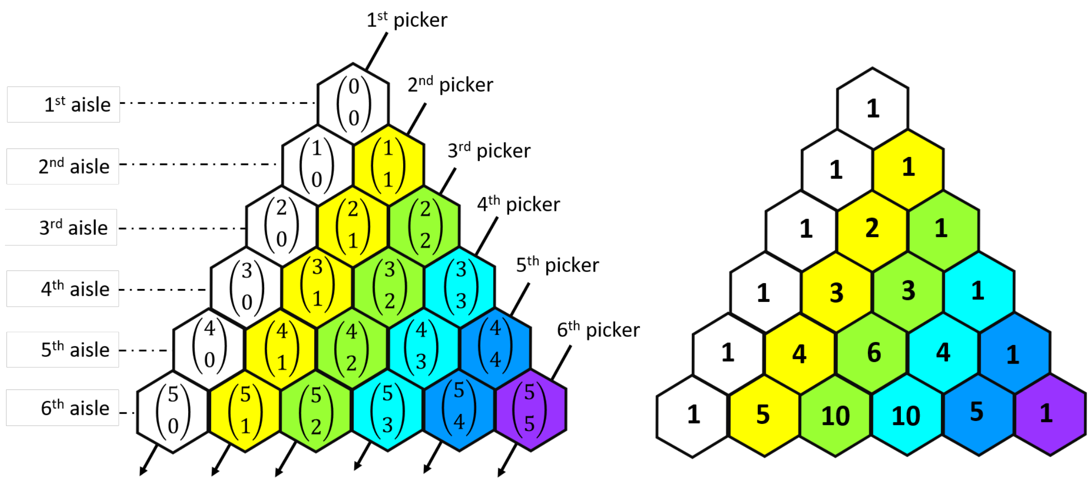


Figure 5.6: Binomial expansion of 6-aisle picking area zone-assignment

Corollary 2. *We can track the pathways of $\binom{|M|-1}{k-1}$ possible zone configurations to assign k pickers to $|M|$ aisles using Pascal's Triangle graph representation.*

For example, there are $\binom{5}{2}$ different pathways to go to the entry at aisle 6, picker 3 in the Pascal's Triangle as shown on Figure 5.7.

We can now state the main theorem of this study and Algorithm 3. With this algorithm, we can track the optimal combination of zone assignments that minimizes the longest-time tour length for a given number of pickers.

Theorem 1. *The min-max OPP in synchronized dynamic zone-picking systems can be solved in polynomial time using DP.*

Proof. $F(i,k)$ is the cost-to-go function which is equal to the lead time of the picking wave that optimally assigns k pickers to i aisles, where $i \geq k$; $i \in M$. As the initial step, for the single picker case where $k = 1$, it is clear that $F(i,1) = RR(1,i)$. Moreover, when the number of pickers equals to the number of aisles where $k = i$, the cost-to-go function returns $F(i,i) = \max\limits_{j=1\ldots i}\Big(RR(j,j)\Big)$. Then, the general DP recursion can be formalized as follows:

$$F(i,k) = \min_{j=k-1\ldots i-1}\left(\max\Big(F(j,k-1), RR(j+1,i)\Big)\right) \tag{5.23}$$

Here, the problem of finding the optimal combination of zone assignment for $|M|$ aisles and k pickers is broken down into the subproblem of introducing the k^{th} picker and combining it with memorized optimal zone assignment subproblem solutions for $k-1$ pickers. In total, we have $(k-1)\Big(|M|-(k-1)\Big)$ subproblems to assign k pickers to $|M|$ aisles where each of them has a constant running time thanks to the memoization . This implies that the recursion has an overall computational complexity of $\mathcal{O}(k|M|)$. $\square$

As an example, Figure 5.8 depicts a visual representation of Algorithm 3 to find the minimum lead time of a picking wave among $\binom{5}{2}$ possible zone configurations with 3 pickers and 6 aisles, i.e., $F(6,3)$.

A pseudocode of the algorithm is given in Algorithm 3.

By this solution methodology, we not only minimize the lead time of the picking wave

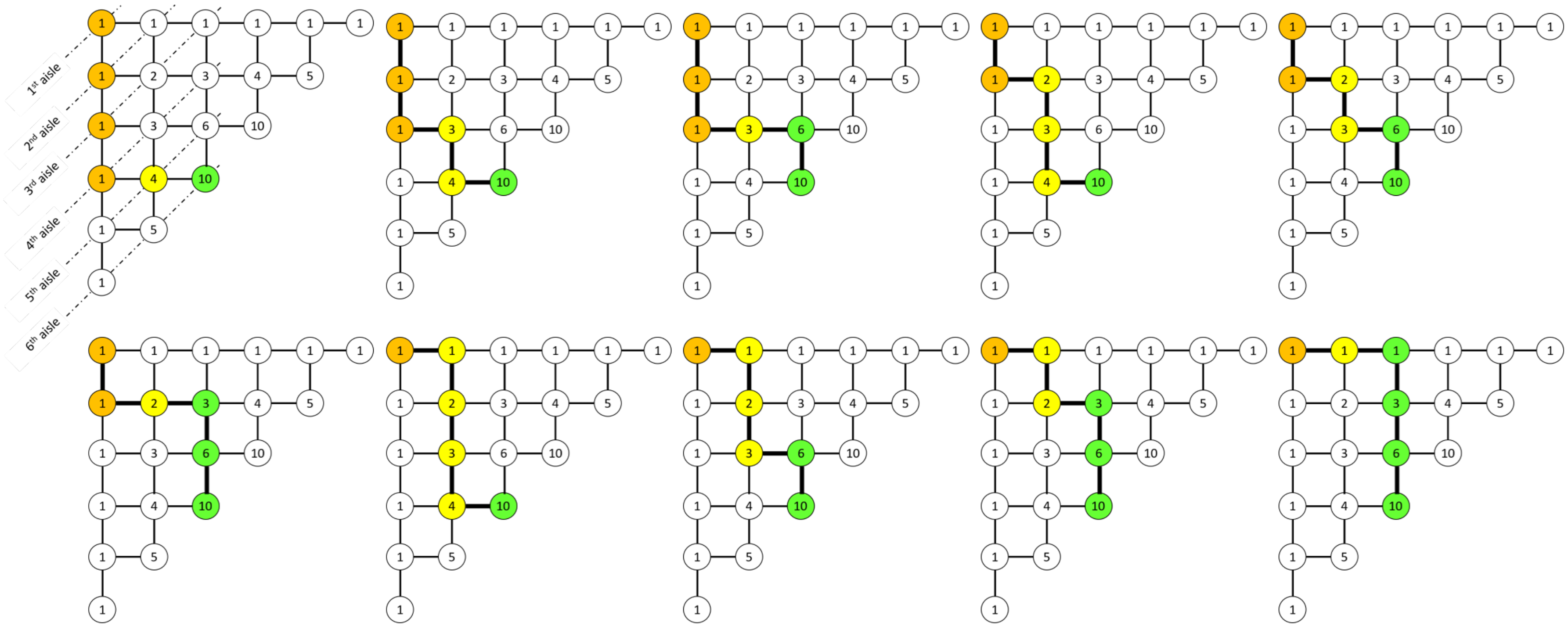
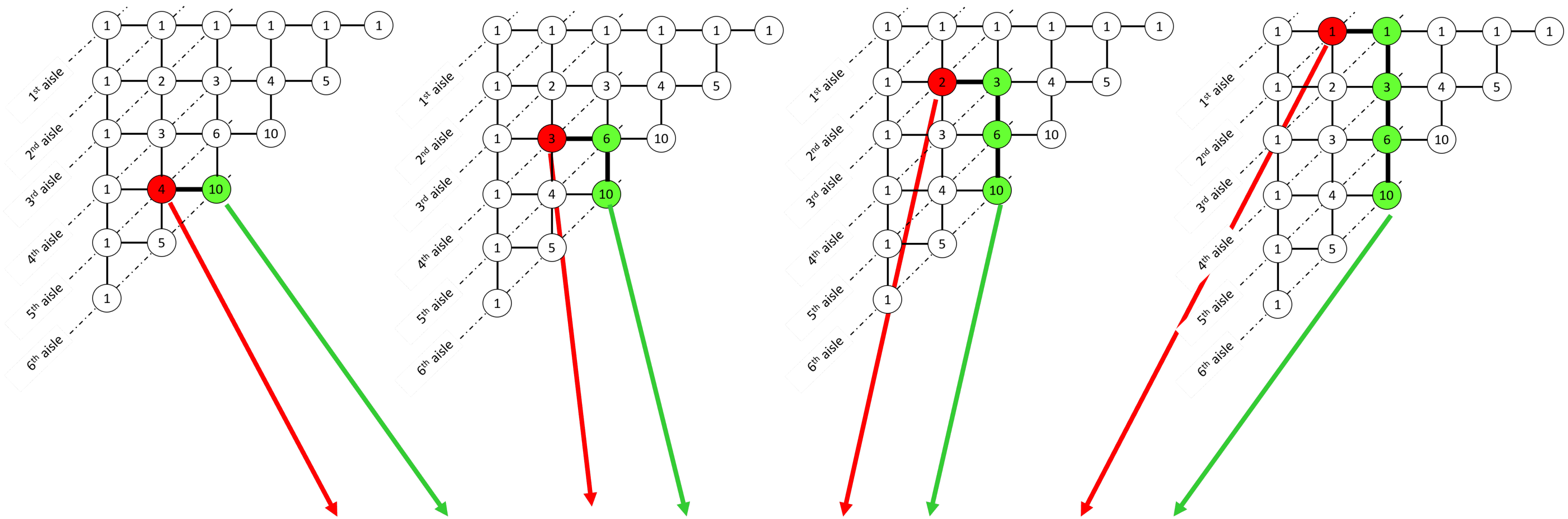


Figure 5.7: The pathways of 10 possible zone configurations for 6 aisles and 3 pickers



$$F(6,3)=\min(\max(F(5,2),RR(6,6)),\ \max(F(4,2),RR(5,6)),\ \max(F(3,2),RR(4,6)),\ \max(F(2,2),RR(3,6)))$$

Figure 5.8: Visual representation of Algorithm 3

Algorithm 3: Dynamic Zone-Picking Algorithm for Multiple Order Pickers

```
for aisle i = 1 to |M| do
    for aisle j = i to |M| do
        return RR(i, j)
    end
end
for aisle i = 1 to |M| do
    F(i, 1) = RR(1, i)
end
for aisle i = 2 to |M| do
    if F(i − 1, i − 1) ≥ RR(i, i) then
        F(i, i) = F(i − 1, i − 1)
    else
        F(i, i) = RR(i, i)
    end
end
for aisle i = 3 to |M| do
    for picker k = 2 to (i − 1) do
        for j = (k − 1) to (i − 1) do
            if temp ≥ max(F(j, k − 1), RR(j + 1, i)) then
                temp ← max(F(j, k − 1), RR(j + 1, i))
            end
        end
        F(i, k) = temp
    end
end
```

for each number of pickers, but we also balance the pickers' travel times.

Proposition 1 allows us to calculate the minimum tour length for each possible zone configuration. Given that there are multiple blocks, heuristic approaches (e.g., S-shape, largest gap, aisle-by-aisle, or combined) can be applied to calculate the routes for each possible zone configuration instead of the RR algorithm. In this way, the multi-picker dynamic zone-picking algorithm can be heuristically solved for a multi-block layout. For the multi-block OPP instance in Figure 5.1(b), Figure 5.9 depicts the resulting 3-picker routes for S-shape, largest gap, aisle-by-aisle, and combined heuristic solutions, with minimum wave-picking lead times of 78, 76, 88, and 76 time units, respectively.

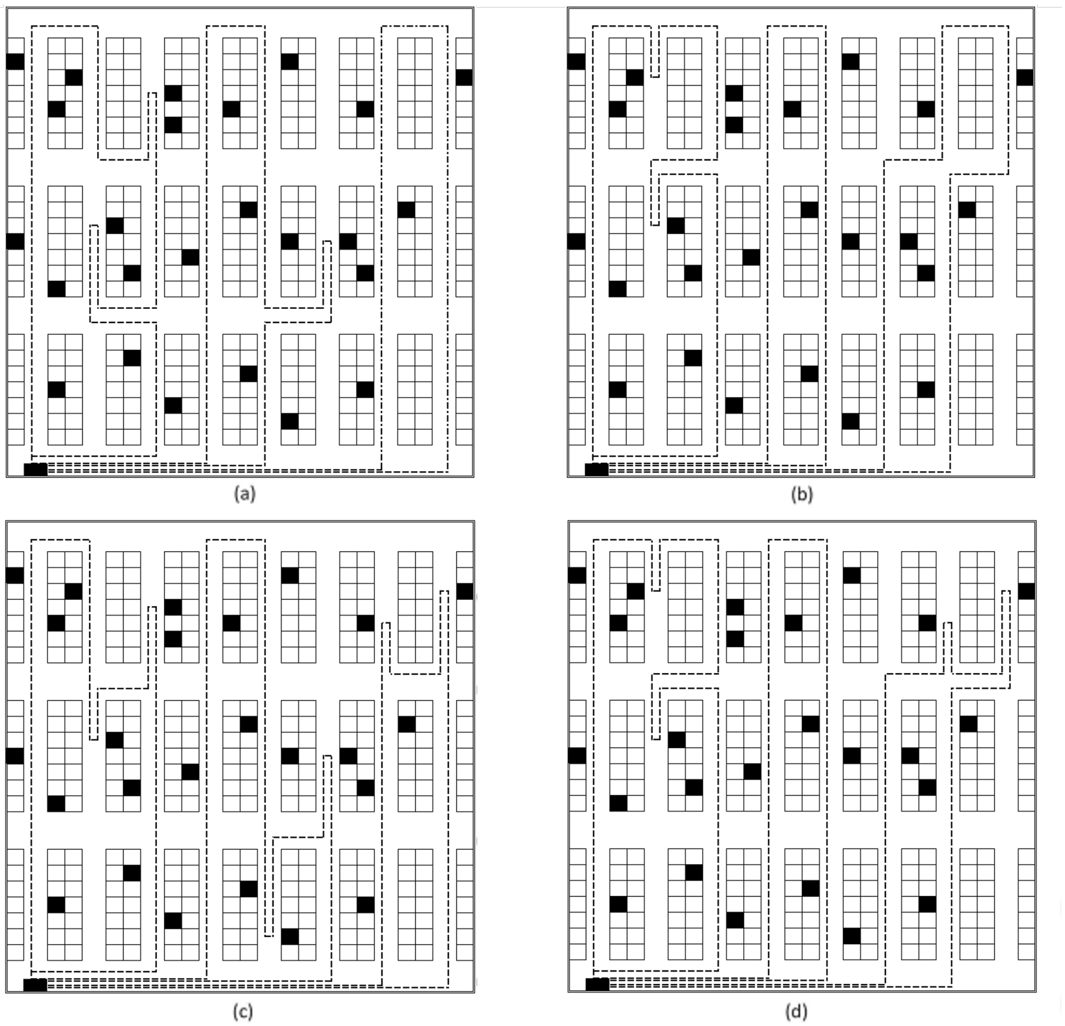


Figure 5.9: Resulting 3-picker heuristic solutions for the instance in Figure 5.1(b). (a) S-shape heuristic solution. (b) Largest gap heuristic solution. (c) Aisle-by-aisle heuristic solution. (d) Combined heuristic solution

5.6 A Batch-Picking Heuristic: The Modified Clarke & Wright Saving Algorithm with a Min-Max Objective

As discussed at the beginning of the chapter, the other factor increasing the performance of order picking processes is batch-picking. If the number of ordered items is relatively small and the number of orders is large, it is inevitable to partition orders into batches in picking operations. A batch-picking is an order picking operation in which the orders are grouped into batches and those batches are picked simultaneously, therefore, forming a wave. In this section, we present a batch-picking algorithm. In batch-picking problems, we try to partition a given set of orders, not the aisles as in the case of zone-picking, into batches such that a specific objective is optimized. The main difficulty for this problem is to find the optimum routes at each partitioning of the customer order into batches since at each combination we need to solve another picker routing sub-problem. In this sense it is an integrated OPP. For this end, we introduce a modified Clarke & Wright saving algorithm with a min-max objective where we aim to minimize the lead time of an order picking, which is determined by the longest time taken by any of the order pickers collecting each batch. A min-max approach is also a common objective for this type of wave-picking operation as it also balances the workload among pickers while sorting and consolidation processes are not required at the end.

The classical Clarke & Wright saving algorithm, which is widely applied for the vehicle routing problem, computes the savings by merging the locations, hence it is not necessary to solve another optimization problem in the travel time calculation phase of the algorithm. However, to apply Clarke & Wright saving algorithm in the batch-picking OPP, one also needs to solve a picker routing problem at each batch combination to calculate the travel times of the order pickers. Here, orders are not allowed to split into batches. Hence, partitioning a given set of orders into batches complicates the problem more since partition itself is known to be NP-complete (Gademann et al. 2001).

In this regard, we firstly present the formulation of batch-picking OPP with a min-max objective. Let x_s be a binary variable equal to 1 if the batch $s \in S$ is created where S is the set of all possible order-batch combinations. Also let parameter a_{is} be the binary entry stating whether order $i \in Q$ is included in batch $s \in S$ where Q is the set of all customer orders. Then, the BIP formulation for the batch-picking OPP is as follows.

$$
\begin{aligned}
\min \quad & Z && (5.24)\\
s.t. \quad & \sum_{s \in S} a_{is} x_s = 1 \quad i \in Q && (5.25)\\
& d_s x_s \leq Z \qquad s \in S && (5.26)\\
& x_s \in \{0, 1\} \qquad s \in S && (5.27)
\end{aligned}
$$

In this formulation, Constraint 5.25 ensures order $i \in Q$ cannot be split into more than one batch. Additionally, Constraint 5.26 states that the time travelled by each batch-picker cannot be more than the longest tour, Z, which is to be minimized in the objective function. Finally, capacity consideration can be included into the model. The batch configurations which satisfy the capacity constraint 5.28 would constitute the set S. This inequality limits the total order size of the batch $s \in S$ below a specified capacity.

$$
\sum_{i \in Q} a_{is} \leq c \quad s \in S \tag{5.28}
$$

We observe that d_s is assumed to be a parameter although it is an OPP tour required to be solved by itself. Moreover, the possible batch set, S increases exponentially which makes the problem difficult to solve to optimality for instances with large even moderate number of orders. Also, the parameter a_{is} should be predetermined, but could also be a decision variable in an integrated formulation, which would make the formulation non-linear. For this end, we introduce a modified Clarke & Wright saving algorithm with min-max objective. The objective in this batch-picking problem is to minimize the time travelled by the latest picker for each number of batches starting from the largest number of batch to a single batch. The steps of the algorithm are as given in Algorithm 4.

5.6.1 Numerical Example

Next, we illustrate how the algorithm works for a 5-customer order instance. Initially, the algorithm assigns each customer order to a different batch. For each batch, a different picker starts picking. RR algorithm solves the OPP for each batch and the

Algorithm 4: Batch-Picking Algorithm for Multiple Order Pickers

- **Initialization.** Assuming each customer order $i \in Q$ is distributed to a different batch. Here, the number of batches is equal to the number of customer orders. Solve the picker routing problem for each batch using the RR algorithm and record the time travelled by the latest picker as "$z_{|S|}$" where S represents the index set for the created batches.
- **Iterations.**
  - **Step 1.** Combine each batch in pairs and solve the picker routing problem using the RR algorithm for each batch combination.
  - **Step 2.** Merge the minimum time travelled batch combination and record the min-max value, "$z_{|S|}$". Now, we have one less number of batches.
  - **Step 3.** Return to Step 1 until we have a single batch **OR** the capacity of the pickers are consumed.
- **Termination.** Draw the Pareto diagram which shows the latest time travelled for each number of batches.

order pickers arrive in 62, 84, 90, 112, 42 time-units, respectively. The 112 time-units, which is the time travelled by the latest picker, is recorded as z_5.

As the first iteration, the order picking times of all possible batch combinations are solved using the RR algorithm. The 1^{st} and 2^{nd} batches, which gave the shortest time among the candidates, are combined. This new batch is named as batch 1. The order picking time resulting from this combination is 90 time-units. In this case, z_4 remained the same as z_5, which is 112. We note that batch 1 includes the orders of customers 1 and 2.

As the second iteration, the order picking times of all possible batch combinations are solved using the RR algorithm. The 3^{rd} and 5^{th} batches, which gave the shortest time among the candidates, are combined. This new batch is named as batch 3. The order picking time resulting from this combination is 106 time-units. Also in this case, z_3 remained the same as z_5, which is 112. We note that batch 3 includes the orders of customers 3 and 5.

As the third iteration, the order picking times of all possible batch combinations are

```
CUSTOMER ORDER 1 = 62
CUSTOMER ORDER 2 = 84
CUSTOMER ORDER 3 = 90
CUSTOMER ORDER 4 = 112
CUSTOMER ORDER 5 = 42
MIN-MAX BATCH = 112

CUSTOMER ORDER 1 and 2 = 90
CUSTOMER ORDER 1 and 3 = 118
CUSTOMER ORDER 1 and 4 = 140
CUSTOMER ORDER 1 and 5 = 94
CUSTOMER ORDER 2 and 3 = 116
CUSTOMER ORDER 2 and 4 = 118
CUSTOMER ORDER 2 and 5 = 90
CUSTOMER ORDER 3 and 4 = 144
CUSTOMER ORDER 3 and 5 = 106
CUSTOMER ORDER 4 and 5 = 112
MERGED CELLS = 1-2, MERGED BATCH = 90, MIN-MAX BATCH = 112
CUSTOMER ORDER 1 and 3 = 132
CUSTOMER ORDER 1 and 4 = 146
CUSTOMER ORDER 1 and 5 = 116
CUSTOMER ORDER 3 and 4 = 144
CUSTOMER ORDER 3 and 5 = 106
CUSTOMER ORDER 4 and 5 = 112
MERGED CELLS = 3-5, MERGED BATCH = 106, MIN-MAX BATCH = 112
CUSTOMER ORDER 1 and 3 = 158
CUSTOMER ORDER 1 and 4 = 146
CUSTOMER ORDER 3 and 4 = 144
MERGED CELLS = 3-4, MERGED BATCH = 144, MIN-MAX BATCH = 144
CUSTOMER ORDER 1 and 3 = 168
MERGED CELLS = 1-3, MERGED BATCH = 168, MIN-MAX BATCH = 168
```

Figure 5.10: The solution of the batch-picking algorithm for the numerical example

solved using the RR algorithm. The 3^{rd} and 4^{th} batches, which gave the shortest time among the candidates, are combined. This new batch is named as batch 3. The order picking time resulting from this combination is 144 time-units, which is z_2. We note that batch 3 includes the orders of customers 3, 4 and 5.

Finally, as the fourth iteration, remaining two batches, the batches 1 and 3, are combined and the time travelled is solved using the RR algorithm. The order picking time resulting from this last combination is 168 time-units, which is z_1. We note that there is only a single picker exists and the algorithm terminates. The solution flow of the algorithm is given in Figure 5.10. The number of pickers and order picking times obtained in each iteration are visually depicted in Figure 5.11.

So, the decision maker here can decide to go with 3 batches since it is not possible to reduce the wave-picking time further with an increase in the number of pickers.

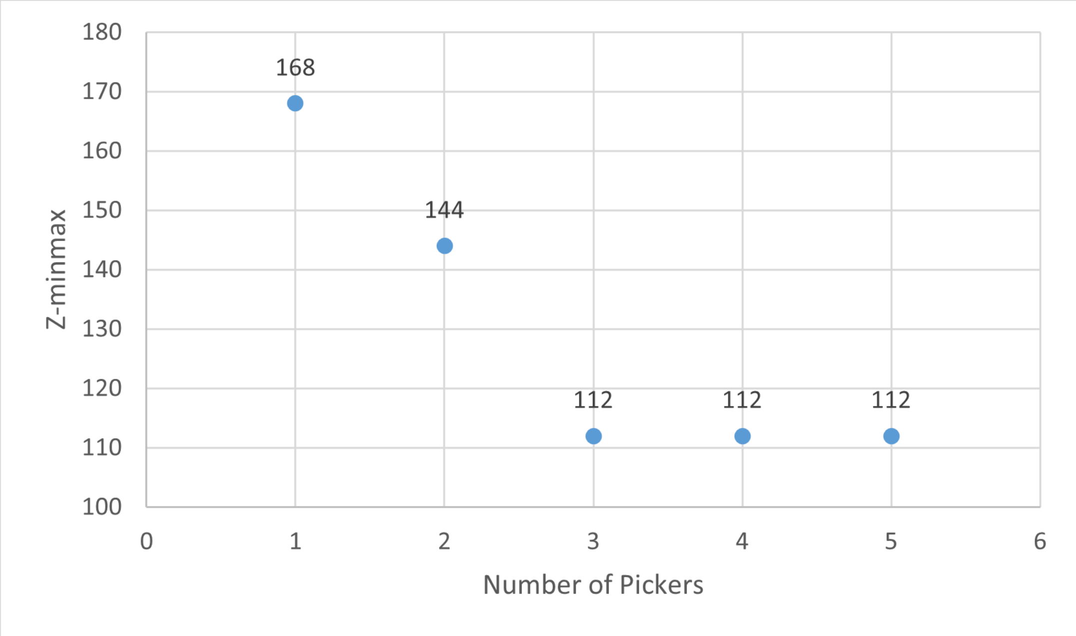


Figure 5.11: Travel times of the latest picker for each number of batches

5.7 Computational Experiments

In this section, we present the computational experiments conducted to test the performance of the algorithms. Experiments are conducted in five parts and the instances are generated in line with those in the literature (Roodbergen & De Koster 2001a, Scholz et al. 2016). First, we test the performance of the two-picker min-max OPP algorithm by analysing the optimality gaps and computational times. Secondly, we aim to observe the impact of the number of pickers on lead time savings. Thirdly, we study the impact of zone-picking on the lead time by comparing the results of zone-picking and no zone-picking policies. As the fourth experiment, we test the performance of the algorithm on multi-block layouts by considering four well-known routing heuristics. Lastly, we compare the performances of zone-picking and batch-picking policies with a min-max objective.

Following Roodbergen & De Koster (2001a), we use a set of 2,000 instances for each combination. We implement the models in AMPL modelling language and the algorithms in C++ in Microsoft Visual Studio 2019. Average run times of algorithms are below 0.1 seconds, hence are not reported. In all settings, the horizontal length between two adjacent aisles set to 2 time units. The end of a sub-aisle (length) of

Table 5.4: Summary of the computational experiments

Aisles	Items	Length	Optimal (%)	Average GAP (%)	Std.Dev. GAP (%)	Savings (%) VRP-MINMAX	Savings (%) Algorithm	Time-Unit Difference after 2000 Picks	VRP-Z % of Solved Instances	VRP-Z Avg Computing Time (s)
7	10	10	90.90	0.51	1.75	34.41	34.09	0	100.0	0.27
7	10	30	69.50	1.59	3.01	39.40	38.47	12	100.0	0.26
15	10	10	**92.20**	**0.28**	1.05	29.46	29.27	4	100.0	2.14
15	10	30	66.55	1.25	2.27	37.77	37.01	6	100.0	2.09
7	15	10	87.10	0.64	1.81	38.23	37.84	6	100.0	2.19
7	15	30	60.75	1.70	2.67	**44.05**	**43.10**	4	100.0	2.14
15	15	10	89.85	**0.28**	**0.90**	33.99	33.80	2	54.0	34.73
15	15	30	60.30	1.14	1.80	42.37	41.71	10	60.3	33.34

a block also refers to the centre of the back cross-aisle of that block and the front cross-aisle of the next block.

5.7.1 Performance of the Algorithms

In the first experiments, the performance of the two-picker min-max OPP algorithm is analysed using optimality gaps. We use the first set of instances where there are 7 or 15 aisles, each with a length of 10 or 30 time units, and 10 or 15 pick items in the picking list. The summary of the results is given on Table 5.4.

In Table 5.4, Optimal % refers to the percentage of the number of optimal instances out of 2000 instances for each setting. For each instance, the gap is calculated as $GAP = \frac{Z_H - Z^*_{MINMAX}}{Z^*_{MINMAX}}$ where Z^*_{MINMAX} represents the optimal travel time for VRP-MINMAX problem and Z_H represents the travel time of the latest picker calculated by the two-picker OPP algorithm. The results show that the algorithm leads to either optimal solutions or very small gaps from the VRP-MINMAX model. Even though the algorithm is not exact, optimal solutions are produced with a significantly high frequency. Over all settings, a maximum of 92.2% and an average of 77.1% of the solutions are optimal, and the average gap remains below 1.7%. Average Savings represent the average percent time savings when two pickers are used instead of a single picker. We can conclude that using two pickers reduces the order picking time at least about one-third with the best improvement of 44% for the VRP-MINMAX model and 43% for the heuristic. Furthermore, it can be inferred from Table 5.4 that the larger the number of items and the longer the length of an aisle, the more significant is the assigning of a second picker.

For each instance or picking wave, both pickers start collecting the orders together. When a picker arrives back to the depot earlier, s/he is assigned to the longer route for the next picking wave to keep the workload balanced in the long run. The average time difference between the pickers is shown on the next column of Table 5.4. On average, the average time difference between the two pickers is as small as 5.5 units, which shows that the algorithm can balance the workload substantially.

In the last two columns of Table 5.4, we present the average computational times of the VRP-Z formulation to compare its performance with the one of the proposed exact algorithm for the min-max OPP under synchronized dynamic zone-picking. We test the VRP-Z formulation using the same instance set with up to 6 pickers by setting a time limit of 60 seconds. VRP-Z formulation results in significantly large number of unsolved instances when the number of pick locations and the number of aisles is even slightly larger. The results show that there are a maximum of 45% unsolved instances when the number of pick locations is increased from 10 to 15 and the number of aisles is increased from 7 to 15. Hence, we can conclude that the proposed VRP-Z formulation leads to large computing times when the ratio of number of pick locations to the number of aisles is large. On the other hand, the proposed algorithm solves all instances in these settting almost instantly (<0.01 seconds). Even, using the second set of instances (Scholz et al. 2016), as the number of pick locations is increased to 90 and the number of aisles is set as 30, the maximum computation time remains below 0.1 seconds.

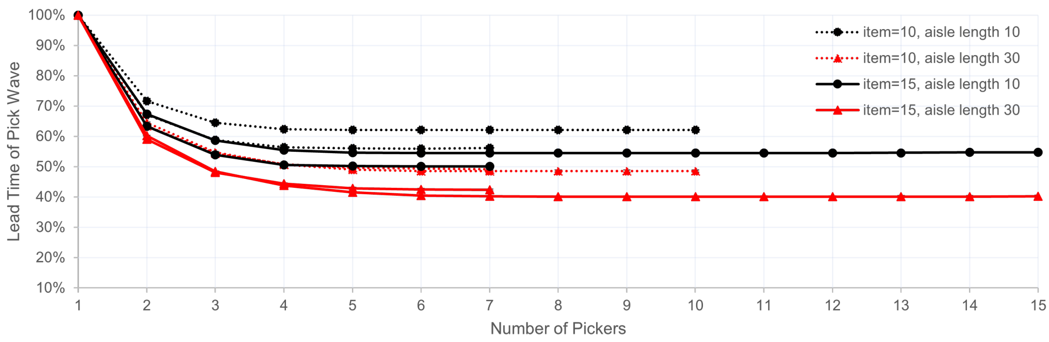


Figure 5.12: Impact of pickers on lead time reduction

5.7.2 Impact of Multiple Pickers on Lead Time of the Picking Wave for the Single Block Layout

To analyse the impact of the multiple pickers on lead time savings, we use the exact algorithm presented in Section 5.5 to find the optimal solutions for the min-max OPP under synchronized dynamic zoning, as it finds the same solutions as the VRP-Z formulation in significantly shorter times. We conduct our experiments on two different sets of randomly generated instances. The first is the same as the one presented in Section 5.7.1, whereas the second set of instances is the subset of the instance set designed by Scholz et al. (2016), where we set the number of aisles as 10, 20 or 30, aisle length as 10, 30 or 50 time units, and number of pick items as 30, 60 or 90.

Results for the first set of instances are summarized on Figure 5.12. Results depict the averages of percentage reductions on the travel time gained by introducing each additional picker for 7-aisle and 15-aisle cases out of 2000 instances for each setting. As a baseline, we set the value of single-picker lead time as 100%. We clearly observe that assigning additional pickers significantly decreases the lead time of a picking wave, but to a certain extent and in a decreasing rate. In other words, the law of diminishing marginal returns applies.

According to the findings from the first set, there is an average of 35% reduction with the introduction of the second picker. However, the impact of reduction decreases to 9% with the 3rd and to 3% with the 4th picker. Although there is still some decrease with the introduction of a new picker, this difference reduces to below 1% for the 6th picker. As Figure 5.12 also shows, relatively larger aisle lengths (triangle markers on the plot) result in more savings with each additional picker. This is also the case when we examine the straight vs. dotted lines, the former of which represent more pick items in the list. The results also suggest that if the number of pick locations is relatively larger, additional pickers yield more savings. The last important finding from Figure 5.12 is that an additional picker brings slightly more benefits if the number of aisles is relatively smaller.

We gain the same insights with the settings by Scholz et al. (2016). The findings are depicted on Figure 5.13, which shows the averages of reductions gained by introducing a new picker for the 10-aisle, 20-aisle, and 30-aisle cases out of 2000 instances for each setting.

From Figure 5.13, there is an average of 43% reduction with the introduction of the second picker. The impact of reduction decreases to 13.5% with the 3rd and to 6.5% with the 4th picker. This difference drops below 1% after the 9th picker for the most extreme test instance. We also notice that assigning 9 pickers yields 82% savings in the most extreme case. Moreover, larger aisle length results in more savings with each additional picker. Figure 5.13 also shows that the most significant impact factors are the aisle length and the number of pick items. Another interesting result is that although the size of the instance set is more than doubled in the most extreme case when it is compared to the most extreme case of the previous instance set, the threshold number of pickers, beyond which no more significant saving is gained, is only increased by half.

To sum up, our overall finding regarding the impact of the multiple pickers is that the length of aisle and the number of pick items have a significantly positive impact on savings gained by assigning more pickers. The savings are more significant if the aisle-number is relatively small as long as the length of an aisle is relatively short. Moreover, there is a threshold value of additional picker beyond which no more saving is received. Thus, it is noteworthy to remind the fact of the law of diminishing marginal returns.

It will be inaccurate if we try to give a balanced configuration between the size of the warehouse and the number of pickers due to significant impact of the number of pick items. In terms of finding a balanced configuration between the warehouse size

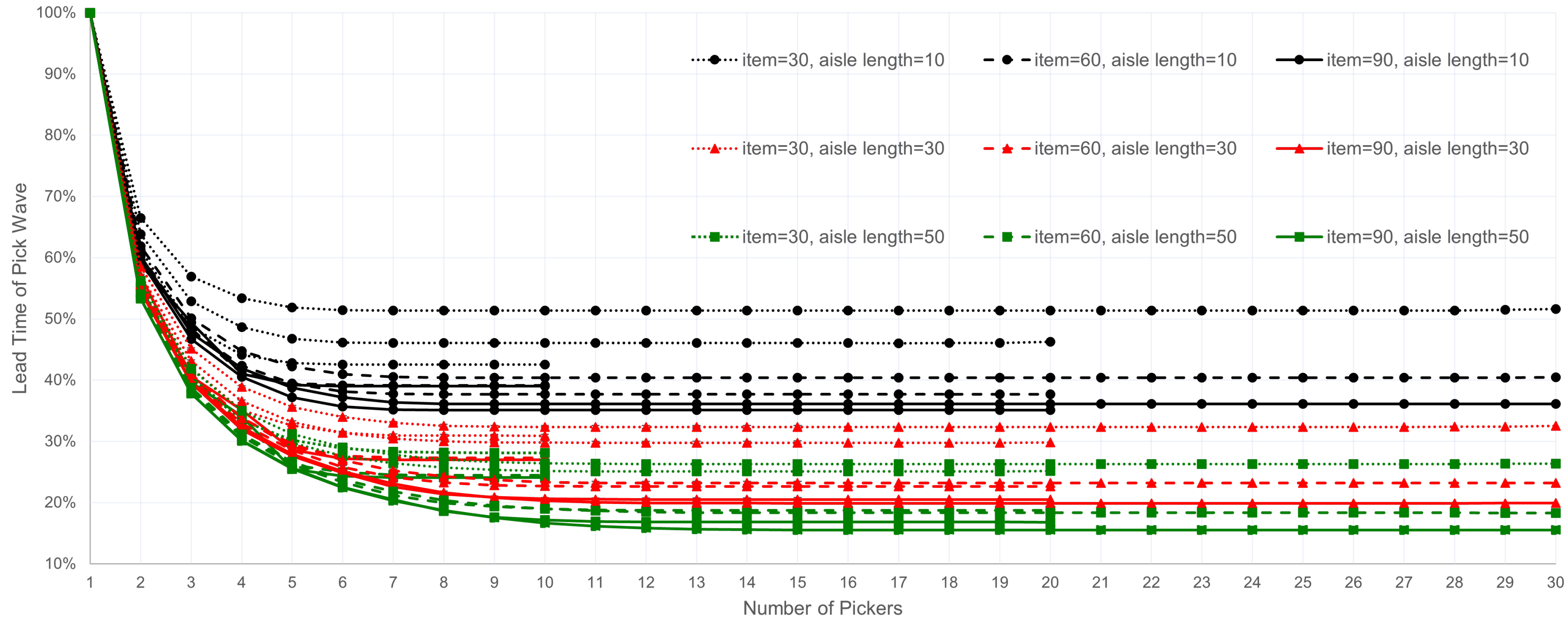


Figure 5.13: Impact of pickers on lead time reduction with Scholz et al. (2016) instances.

and number of pickers, we observe that the number of aisles is not a significant factor. However, the length of aisle is significant in the sense that the deeper the aisles, the more significant is the assigning of an additional picker. Consequently, managers should consider assigning each additional picker more carefully due to decreasing pace in the lead time reduction. As a rule of thumb, managers can keep in my that, for relatively large aisles, assigning the 2nd picker results in about 45% reduction in lead time of a pick wave, independent of the warehouse size. A 3rd picker would lead to 15% reduction in average beyond which it will never reach 10% under synchronized zone-picking systems. So, under strict cost considerations, decision makers can continue with two pickers as it will give the largest reduction in lead time. As we suggested in Subsection 5.4.2, one can further reduce the lead time at the expense of aisle-zone integrity by applying the travel time balancing algorithm.

5.7.3 Comparison of Zone-Picking and No Zone-Picking Cases

Next, we aim to assess the impact of zone-picking on travel time by comparing the gap between the results of the min-max OPP with those without zone-picking. We include 6 pickers in the first set of instances beyond which no value is added by additional pickers. Table 5.5 shows a summary of these gaps. For each of these combinations, best results are indicated in bold. Equal Results % refers to the percentage of the number of instances with the same results out of 2000 instances for each setting. The first inference that can be drawn from Table 5.5 is that the percentage of equal results increases significantly with each additional picker as it increases from 60.9% in average while there are 2 pickers to 99.8% in average with 6 pickers.

Results also show that the average gap between the zone-picking and no zone-picking, i,e, the extent of suboptimality from imposing zone picking decreases significantly with each introduced picker (decrease from 1.86% with 2 pickers to 0.11% with 6 pickers). This can be explained by the fact that the possible movements become more limited when fewer aisles are assigned to each picker.

From Table 5.5, it can also be inferred that aisle-length is also a significant, but negatively associated factor since an increase in the aisle-length leads to a reduction in the number of equal results (from 89.33% to 69.52% in average, p-value < 0.01) and a rise in the average (from 0.50% to 1.53% in average, p-value < 0.01) and standard

Table 5.5: Gap analysis between zone-picking and no zone-picking cases

Aisles	Items	Length	No. of Pickers				
			2	3	4	5	6
			Equal Results %				
7	10	10	79.70	84.00	96.80	99.75	100.00
7	10	30	56.65	66.00	82.60	95.80	99.95
15	10	10	77.05	78.80	95.15	99.70	100.00
15	10	30	48.65	48.75	65.25	86.70	97.15
7	15	10	73.55	79.80	96.15	99.65	100.00
7	15	30	44.80	60.45	73.80	92.60	99.65
15	15	10	68.50	73.65	86.65	97.80	99.85
15	15	30	38.05	38.60	43.60	66.00	85.40
			Average Gap %				
7	10	10	1.26	1.11	0.22	0.02	0.00
7	10	30	2.69	2.06	1.05	0.26	0.00
15	10	10	0.92	0.87	0.19	0.01	0.00
15	10	30	2.33	2.55	1.70	0.65	0.11
7	15	10	1.42	1.19	0.27	0.03	0.00
7	15	30	2.95	2.06	1.36	0.37	0.02
15	15	10	0.92	1.03	0.50	0.09	0.01
15	15	30	2.41	2.92	2.69	1.55	0.77
			Standard Deviation of Gap %				
7	10	10	2.85	2.78	1.24	0.38	0.00
7	10	30	4.07	3.60	3.65	1.47	0.11
15	10	10	1.93	1.84	0.89	0.28	0.00
15	10	30	3.13	3.41	2.91	1.97	0.71
7	15	10	2.66	2.52	1.45	0.42	0.00
7	15	30	3.49	3.18	2.77	1.60	0.33
15	15	10	1.51	1.94	1.41	0.66	0.15
15	15	30	2.60	3.35	3.07	2.66	2.13

deviation of gaps (from 1.25% to 2.50% in average, p-value < 0.01).

5.7.4 Multiple Order Pickers for Multi-Block Layout and a Comparison of Heuristics

As the fourth experiment, we would like to analyse the performance of the multi-picker dynamic zone-picking algorithm on multi-block layouts by applying four well-known heuristics on the first set of instances with up to 5 blocks and 7 pickers. The S-shape, largest gap, and combined heuristics are improved at the same extent by forcing that the entering aisle for the closest block to the depot should be the right-most filled picking sub-aisle (Roodbergen & De Koster 2001a). For each random instance, dynamic zone-picking algorithm is solved using S-shape, largest gap, aisle-by-aisle and combined heuristics for multi-block layout. Table 5.6 and Table 5.7 are the products of this experiment.

Table 5.6 gives the average minimum lead times of the picking waves out of 2000 instances for each setting. For each combination, the best heuristic is indicated in bold-underlined. Table 5.6 shows that the combined heuristic significantly outperforms other heuristics for multi-block layouts and the performance of the heuristics converges with the increase in the number of pickers.

The combined heuristic has the best performance in 259 of the 280 settings and is only outperformed where the aisle-length/aisle-number ratio is large. Moreover, performance of the aisle-by-aisle heuristic quickly approaches the combined heuristic with the increase of the number of pickers. Table 5.6 also shows that the S-shape heuristic never has the best performance alone. The largest gap strategy solely has the best performance in 14 settings, each of which emerges when the aisle-length/aisle-number ratio is large. The aisle-by-aisle heuristic solely has the best performance in seven settings and performs the same as combined for the single block layout as in line with the literature (Roodbergen & De Koster 2001a). Moreover, it behaves increasingly poorly with the increase in the number of blocks, even worse than S-shape heuristic. However, employing more pickers quickly improves the performance of these heuristics.

Finally, we examine the impact of each additional picker on savings. Table 5.7 shows

Table 5.6: Minimum wave-picking lead times with each additional picker for multi-block layout settings

Method	No. of Aisles	No. of Items	Aisle Length	Number of Blocks													
				1							2						
				No. of Pickers							No. of Pickers						
				1	2	3	4	5	6	7	1	2	3	4	5	6	7
S-Shape	7	10	10	79.2	49.2	40.0	36.9	**36.4**	**36.6**	**37.2**	100.4	63.3	54.8	52.9	**52.7**	**52.9**	**53.5**
	7	10	30	191.9	112.6	80.9	73.6	70.4	69.7	**70.2**	255.9	152.6	129.5	123.9	122.8	**122.8**	**123.3**
	15	10	10	127.3	83.2	69.9	65.3	**64.0**	**63.9**	**63.9**	143.5	94.4	82.7	79.6	**79.0**	**79.0**	**79.0**
	15	10	30	278.0	161.9	123.5	105.8	98.6	95.6	94.7	325.0	195.5	159.2	148.6	145.5	145.0	**144.9**
	7	15	10	88.5	54.2	42.7	38.8	**37.8**	**37.8**	**38.1**	121.5	73.1	59.1	55.4	54.8	**54.8**	**55.0**
	7	15	30	218.5	127.3	90.0	77.3	73.5	**72.4**	**72.6**	318.8	184.1	139.9	130.6	127.6	**126.9**	**127.1**
	15	15	10	151.3	94.8	77.4	70.0	67.1	66.5	**66.4**	178.0	110.6	90.7	84.3	82.6	**82.2**	**82.2**
	15	15	30	345.4	195.9	147.6	121.7	108.4	102.3	99.4	425.9	244.0	185.1	162.3	154.1	151.4	**150.7**
Largest Gap	7	10	10	76.0	46.7	39.4	36.8	**36.4**	**36.6**	**37.2**	95.8	61.6	54.6	52.9	**52.7**	**52.9**	**53.5**
	7	10	30	164.1	98.8	80.0	73.6	70.4	69.7	**70.2**	**218.8**	145.1	129.1	123.8	122.8	**122.8**	**123.3**
	15	10	10	123.7	79.8	69.2	65.0	**64.0**	**63.9**	**63.9**	146.0	92.2	82.2	79.5	**79.0**	**79.0**	**79.0**
	15	10	30	228.2	141.0	114.1	103.9	98.2	95.5	94.7	280.8	180.6	156.6	148.2	145.5	145.0	**144.9**
	7	15	10	88.5	52.2	42.1	38.6	**37.8**	**37.8**	**38.1**	116.3	69.7	58.5	55.4	54.8	**54.8**	**55.0**
	7	15	30	197.5	**113.3**	85.6	77.2	73.5	**72.4**	**72.6**	**269.0**	**165.5**	138.1	130.4	127.6	**126.9**	**127.1**
	15	15	10	146.8	91.1	75.1	69.2	66.9	**66.4**	**66.4**	179.3	106.9	89.2	84.0	82.5	**82.2**	**82.2**
	15	15	30	284.0	167.8	129.9	113.4	105.8	101.5	99.2	353.3	214.3	173.8	159.5	153.5	151.3	**150.7**
Aisle-by-Aisle	7	10	10	**70.8**	**45.5**	**38.5**	**36.6**	**36.4**	**36.6**	**37.2**	97.1	62.9	54.4	**52.8**	**52.7**	**52.9**	**53.5**
	7	10	30	**163.5**	**97.6**	**77.3**	**71.6**	**69.7**	**69.6**	**70.2**	244.6	151.5	127.4	123.2	**122.7**	**122.8**	**123.3**
	15	10	10	**111.2**	**76.5**	**67.2**	**64.4**	**64.0**	**63.9**	**63.9**	137.8	94.0	82.4	79.5	**79.0**	**79.0**	**79.0**
	15	10	30	**226.6**	**139.6**	**110.8**	**100.1**	**95.9**	**94.8**	**94.6**	307.3	194.6	157.8	147.4	145.3	**144.9**	**144.9**
	7	15	10	**80.7**	**50.5**	**41.3**	**38.3**	**37.8**	**37.8**	**38.1**	116.7	72.5	58.5	55.2	**54.7**	**54.8**	**55.0**
	7	15	30	**193.0**	113.9	**84.5**	**75.9**	**73.0**	**72.4**	**72.6**	301.6	181.7	137.5	128.9	127.1	**126.9**	**127.1**
	15	15	10	**132.3**	**86.7**	**73.2**	**68.2**	**66.6**	**66.4**	**66.4**	168.9	109.1	90.4	84.1	82.5	**82.2**	**82.2**
	15	15	30	**283.8**	**167.5**	**129.4**	**110.9**	**103.1**	**99.6**	**98.4**	394.6	237.5	184.3	160.3	152.8	151.0	**150.7**
Combined	7	10	10	**70.8**	**45.5**	**38.5**	**36.6**	**36.4**	**36.6**	**37.2**	**88.8**	**60.1**	**54.0**	**52.8**	**52.7**	**52.9**	**53.5**
	7	10	30	**163.5**	**97.6**	**77.3**	**71.6**	**69.7**	**69.6**	**70.2**	**218.8**	**141.7**	**126.5**	**123.1**	**122.7**	**122.8**	**123.3**
	15	10	10	**111.2**	**76.5**	**67.2**	**64.4**	**64.0**	**63.9**	**63.9**	**126.9**	**89.6**	**81.2**	**79.3**	**79.0**	**79.0**	**79.0**
	15	10	30	**226.6**	**139.6**	**110.8**	**100.1**	**95.9**	**94.8**	**94.6**	**272.1**	**176.9**	**153.0**	**146.5**	**145.2**	**144.9**	**144.9**
	7	15	10	**80.7**	**50.5**	**41.3**	**38.3**	**37.8**	**37.8**	**38.1**	**107.1**	**67.8**	**57.5**	**55.1**	**54.7**	**54.8**	**55.0**
	7	15	30	**193.0**	113.9	**84.5**	**75.9**	**73.0**	**72.4**	**72.6**	271.4	165.6	**135.1**	**128.6**	**127.0**	**126.9**	**127.1**
	15	15	10	**132.3**	**86.7**	**73.2**	**68.2**	**66.6**	**66.4**	**66.4**	**154.2**	**101.7**	**87.4**	**83.3**	**82.3**	**82.2**	**82.2**
	15	15	30	**283.8**	**167.5**	**129.4**	**110.9**	**103.1**	**99.6**	**98.4**	**348.1**	**213.1**	**171.0**	**156.4**	**151.9**	**150.8**	**150.7**

Table 5.6 (Continued)

Method	No. of Aisles	No. of Items	Aisle Length	Number of Blocks													
				3							4						
				No. of Pickers							No. of Pickers						
				1	2	3	4	5	6	7	1	2	3	4	5	6	7
S-Shape	7	10	10	122.6	79.8	71.9	70.3	**70.1**	**70.3**	**70.9**	141.0	97.5	89.4	87.8	**87.7**	**87.8**	**88.2**
	7	10	30	309.1	199.8	180.8	177.4	**176.8**	**176.8**	**177.3**	354.4	249.7	233.6	231.2	231.0	**231.2**	**231.4**
	15	10	10	174.2	111.3	98.5	95.1	**94.6**	**94.5**	**94.5**	194.1	129.0	114.9	111.7	**111.1**	**111.0**	**111.0**
	15	10	30	381.0	239.2	207.1	199.1	197.6	197.5	**197.4**	425.3	286.9	258.2	251.8	250.8	**250.7**	**250.7**
	7	15	10	154.3	91.0	76.4	73.1	**72.5**	**72.5**	**72.8**	174.3	108.8	94.4	91.1	**90.5**	**90.5**	**90.8**
	7	15	30	393.3	231.1	192.1	184.6	182.8	**182.5**	**182.7**	445.7	281.0	245.7	239.3	**238.1**	**238.0**	**238.2**
	15	15	10	223.5	129.9	107.6	100.9	99.2	**98.8**	**98.8**	243.0	147.9	124.7	117.7	116.0	**115.7**	**115.7**
	15	15	30	505.3	290.5	231.6	212.0	206.3	204.8	**204.6**	550.4	336.1	280.9	264.2	259.9	**259.2**	259.2
Largest Gap	7	10	10	114.2	78.4	71.8	70.3	**70.1**	**70.3**	**70.9**	132.8	96.3	89.2	87.8	**87.7**	**87.8**	**88.2**
	7	10	30	**268.3**	194.5	180.5	177.3	**176.8**	**176.8**	**177.3**	**317.4**	246.0	233.3	231.1	231.0	**231.2**	**231.4**
	15	10	10	168.7	109.1	97.9	95.0	**94.6**	**94.5**	**94.5**	188.9	127.0	114.6	111.6	**111.1**	**111.0**	**111.0**
	15	10	30	332.6	227.9	205.5	199.0	197.6	197.5	**197.4**	382.0	278.4	257.2	251.7	250.8	**250.7**	**250.7**
	7	15	10	140.9	87.1	76.0	73.1	**72.5**	**72.5**	**72.8**	161.2	105.4	94.1	91.1	**90.5**	**90.5**	**90.8**
	7	15	30	**329.2**	215.7	191.1	184.5	182.8	**182.5**	**182.7**	**382.5**	267.7	244.8	239.2	**238.1**	**238.0**	**238.2**
	15	15	10	212.9	124.5	106.1	100.6	99.1	**98.8**	**98.8**	237.1	142.5	123.4	117.4	116.0	**115.7**	**115.7**
	15	15	30	**419.9**	263.5	223.8	210.4	206.0	204.8	**204.6**	**474.1**	312.7	275.3	263.1	259.8	**259.2**	259.2
Aisle-by-Aisle	7	10	10	126.5	81.9	71.6	**70.2**	**70.1**	**70.3**	**70.9**	157.4	101.6	**88.8**	**87.7**	**87.7**	**87.8**	**88.2**
	7	10	30	333.7	209.2	179.7	**177.0**	**176.8**	**176.8**	**177.3**	424.9	267.6	233.1	**231.0**	**230.9**	**231.2**	**231.4**
	15	10	10	170.1	114.4	98.8	95.1	**94.6**	**94.5**	**94.5**	204.1	135.7	115.7	111.6	**111.1**	**111.0**	**111.0**
	15	10	30	403.0	256.3	209.4	198.9	**197.5**	**197.4**	**197.4**	505.9	323.1	262.1	251.9	**250.7**	**250.7**	**250.7**
	7	15	10	154.0	96.2	76.1	**72.9**	**72.5**	**72.5**	**72.8**	192.9	120.4	94.4	**90.7**	**90.5**	**90.5**	**90.8**
	7	15	30	415.3	253.2	191.5	**183.4**	**182.6**	**182.5**	**182.7**	528.1	324.0	247.2	**238.6**	**238.1**	**238.0**	**238.2**
	15	15	10	211.3	134.9	109.9	101.3	99.1	**98.8**	**98.8**	256.8	162.2	130.5	118.5	116.0	**115.7**	**115.7**
	15	15	30	525.3	319.7	245.6	213.9	206.1	**204.7**	**204.6**	658.8	400.9	309.0	268.5	259.9	**259.2**	**259.1**
Combined	7	10	10	**111.3**	**77.5**	**71.4**	**70.2**	**70.1**	**70.3**	**70.9**	**130.6**	**95.5**	88.9	**87.7**	**87.7**	**87.8**	**88.2**
	7	10	30	272.1	**192.1**	**179.0**	**177.0**	**176.8**	**176.8**	**177.3**	321.3	**243.9**	**232.6**	**231.0**	**230.9**	**231.2**	**231.4**
	15	10	10	**158.7**	**107.4**	**97.2**	**94.8**	**94.6**	**94.5**	**94.5**	**180.7**	**125.8**	**114.1**	**111.5**	**111.1**	**111.0**	**111.0**
	15	10	30	**332.4**	**225.8**	**203.4**	**198.2**	**197.5**	**197.4**	**197.4**	**381.9**	**276.6**	**255.7**	**251.3**	**250.7**	**250.7**	**250.7**
	7	15	10	**137.4**	**86.1**	**75.3**	**72.9**	**72.5**	**72.5**	**72.8**	**157.6**	**104.5**	**93.5**	90.9	**90.5**	**90.5**	**90.8**
	7	15	30	338.9	**214.8**	**188.2**	**183.4**	**182.6**	**182.5**	**182.7**	391.7	**266.8**	**242.8**	**238.6**	**238.1**	**238.0**	**238.2**
	15	15	10	**198.8**	**121.9**	**104.9**	**100.1**	**99.0**	**98.8**	**98.8**	**219.8**	**140.9**	**122.5**	**117.0**	**115.9**	**115.7**	**115.7**
	15	15	30	425.4	**263.3**	**220.9**	**208.5**	**205.3**	**204.7**	**204.6**	475.5	**312.4**	**272.9**	**261.9**	**259.5**	**259.2**	**259.1**

Table 5.6 (Continued)

Method	No. of Aisles	No. of Items	Aisle Length	Number of Blocks						
				5						
				No. of Pickers						
				1	2	3	4	5	6	7
S-Shape	7	10	10	159.5	115.3	106.8	105.3	**105.2**	**105.4**	**106.0**
	7	10	30	403.4	301.9	287.4	285.5	**285.4**	**285.5**	**285.8**
	15	10	10	217.2	147.7	132.6	129.1	128.5	**128.4**	**128.4**
	15	10	30	476.8	338.5	311.2	305.6	**305.0**	**305.0**	**305.0**
	7	15	10	195.3	127.0	112.7	109.4	**108.8**	**108.8**	**109.1**
	7	15	30	499.5	333.1	301.4	295.5	**294.6**	**294.6**	**294.7**
	15	15	10	267.7	166.8	142.9	135.6	133.7	**133.4**	**133.4**
	15	15	30	603.2	385.9	332.5	318.3	314.8	**314.3**	314.3
Largest Gap	7	10	10	152.0	114.3	106.8	105.3	**105.2**	**105.4**	**106.0**
	7	10	30	**369.1**	299.2	287.3	285.5	**285.4**	**285.5**	**285.8**
	15	10	10	211.7	145.8	132.3	129.0	128.5	**128.4**	**128.4**
	15	10	30	**436.7**	331.6	310.4	305.6	**305.0**	**305.0**	**305.0**
	7	15	10	181.0	123.9	112.4	109.3	**108.8**	**108.8**	**109.1**
	7	15	30	**435.6**	322.3	300.4	295.4	**294.6**	**294.6**	**294.7**
	15	15	10	260.3	162.1	141.7	135.3	133.7	**133.4**	**133.4**
	15	15	30	**527.7**	**365.3**	328.3	317.5	314.7	**314.3**	314.3
Aisle-by-Aisle	7	10	10	188.2	121.3	**106.2**	**105.2**	**105.2**	**105.4**	**106.0**
	7	10	30	517.9	327.0	287.0	**285.4**	**285.4**	**285.5**	**285.8**
	15	10	10	238.4	157.5	133.3	**128.9**	**128.4**	**128.4**	**128.4**
	15	10	30	610.0	388.4	317.2	305.7	**305.0**	**305.0**	**305.0**
	7	15	10	231.5	144.5	112.8	**109.0**	**108.8**	**108.8**	**109.1**
	7	15	30	650.8	400.0	303.9	**294.9**	**294.6**	**294.6**	**294.7**
	15	15	10	303.1	190.1	151.3	136.6	133.7	**133.4**	**133.4**
	15	15	30	795.4	485.2	371.9	323.8	315.1	**314.3**	314.3
Combined	7	10	10	**150.6**	**113.6**	106.5	105.3	**105.2**	**105.4**	**106.0**
	7	10	30	373.3	**297.3**	**286.6**	**285.4**	**285.4**	**285.5**	**285.8**
	15	10	10	**205.5**	**145.1**	**131.9**	**128.9**	128.5	**128.4**	**128.4**
	15	10	30	438.0	**330.3**	**309.3**	**305.4**	**305.0**	**305.0**	**305.0**
	7	15	10	**178.5**	**123.3**	**112.0**	109.2	**108.8**	**108.8**	**109.1**
	7	15	30	445.1	**320.7**	**298.9**	295.0	**294.6**	**294.6**	**294.7**
	15	15	10	**246.1**	**160.8**	**141.0**	**135.1**	**133.6**	**133.4**	**133.4**
	15	15	30	530.4	365.5	**326.6**	**316.6**	**314.5**	**314.3**	**314.2**

the average percentage savings with each additional picker out of 2000 instances for each setting where the best heuristic is indicated in bold-underlined. For the single block layout, average percentage savings are slightly higher when compared to the optimal cases resulted from the RR algorithm in Section 5.7.2. When we compare the heuristics, additional pickers yield better savings for S-shape and largest gap heuristics at single block layouts. With the increase on the number of blocks, each additional picker results in more savings for the aisle-by-aisle heuristic. The largest gap strategy performs poorly in terms of percentage savings with each additional picker. The combined heuristic never has the best performance alone for the percentage savings, although it still maintains its overall best performance on the minimum lead times. This can be explained by the fact that underperforming heuristics lead to more savings with each additional picker since the performance of the heuristics converges with the increase in the number of pickers. Overall, we observe that the combined heuristic performs relatively the best for the multi-picker multi-block dynamic zone-picking problem, while the performance of largest gap heuristic increases with the number of blocks and that of aisle-by-aisle heuristics increases with the number of pickers.

5.7.5 Comparison of Zone-Picking and Batch-Picking Policies

Lastly, we analyse the performances of the zone-picking and batch-picking policies and also present the average percentage gaps using the first set of instances where there are 7 or 15 aisles, each with a length of 10 or 30 time units, and 10 or 15 pick items in the picking list and up to 5 groups of zones or batches. Table 5.8 summarizes the average minimum lead times of the picking waves and the average percentage gaps between them out of 2000 instances for each setting resulted using the proposed approaches. For each setting, the better policy and also the best setting in terms of average percentage gaps are indicated in bold.

The table shows that the zone-picking policy produces lower lead times. However, zone-picking policy requires sorting/consolidation after completion while batch picking policy returns undivided customer orders at the depot. The average gap in percentage is below 0.5% in average for the worst case and this gaps seems to be quickly closed at the sorting/consolidation process although zone-picking still have the advan-

Table 5.7: Percentage savings with each additional picker for multi-block layout settings

Method	No. of Aisles	No. of Items	Aisle Length	Number of Blocks											
				1						2					
				No. of Pickers						No. of Pickers					
				2	3	4	5	6	7	2	3	4	5	6	7
S-Shape	7	10	10	37.88	**18.70**	**7.75**	**1.36**	**0.00**	**0.00**	**36.95**	13.43	**3.47**	**0.38**	**0.00**	**0.00**
	7	10	30	**41.32**	**28.15**	**9.02**	**4.35**	**0.99**	**0.00**	**40.37**	15.14	**4.32**	**0.89**	**0.00**	**0.00**
	15	10	10	34.64	**15.99**	**6.58**	**1.99**	**0.16**	**0.00**	34.22	**12.39**	**3.75**	**0.75**	**0.00**	**0.00**
	15	10	30	**41.76**	**23.72**	**14.33**	**6.81**	**3.04**	**0.94**	**39.85**	18.57	**6.66**	**2.09**	**0.34**	**0.07**
	7	15	10	38.76	**21.22**	**9.13**	**2.58**	**0.00**	**0.00**	39.84	19.15	**6.26**	**1.08**	**0.00**	**0.00**
	7	15	30	41.74	**29.30**	**14.11**	**4.92**	**1.50**	**0.00**	**42.25**	24.01	**6.65**	**2.30**	**0.55**	**0.00**
	15	15	10	37.34	**18.35**	**9.56**	**4.14**	**0.89**	**0.15**	37.87	**17.99**	**7.06**	**2.02**	**0.48**	**0.00**
	15	15	30	**43.28**	**24.66**	**17.55**	**10.93**	**5.63**	**2.83**	**42.71**	**24.14**	12.32	**5.05**	**1.75**	**0.46**
Largest Gap	7	10	10	**38.55**	15.63	6.60	1.09	**0.00**	**0.00**	35.70	11.36	3.11	**0.38**	**0.00**	**0.00**
	7	10	30	39.79	19.03	8.00	**4.35**	**0.99**	**0.00**	33.68	11.03	4.11	0.81	**0.00**	**0.00**
	15	10	10	**35.49**	13.28	6.07	1.54	**0.16**	**0.00**	**36.85**	10.85	3.28	0.63	**0.00**	**0.00**
	15	10	30	38.21	19.08	8.94	5.49	2.75	0.84	35.68	13.29	5.36	1.82	**0.34**	**0.07**
	7	15	10	**41.02**	19.35	8.31	2.07	**0.00**	**0.00**	**40.07**	16.07	5.30	**1.08**	**0.00**	**0.00**
	7	15	30	**42.63**	24.45	9.81	4.79	**1.50**	**0.00**	38.48	16.56	5.58	2.15	**0.55**	**0.00**
	15	15	10	**37.94**	17.56	7.86	3.32	0.75	0.00	**40.38**	16.56	5.83	1.79	0.36	**0.00**
	15	15	30	40.92	22.59	12.70	6.70	4.06	2.27	39.34	18.90	8.23	3.76	1.43	0.40
Aisle-by-Aisle	7	10	10	35.73	15.38	4.94	0.55	**0.00**	**0.00**	35.22	**13.51**	2.94	0.19	**0.00**	**0.00**
	7	10	30	40.31	20.80	7.37	2.65	0.14	**0.00**	38.06	**15.91**	3.30	0.41	**0.00**	**0.00**
	15	10	10	31.21	12.16	4.17	0.62	**0.16**	**0.00**	31.79	12.34	3.52	0.63	**0.00**	**0.00**
	15	10	30	38.39	20.63	9.66	4.20	1.15	0.21	36.67	**18.91**	6.59	1.42	0.28	0.00
	7	15	10	37.42	18.22	7.26	1.31	**0.00**	**0.00**	37.87	**19.31**	5.64	0.91	**0.00**	**0.00**
	7	15	30	40.98	25.81	10.18	3.82	0.82	**0.00**	39.75	**24.33**	6.25	1.40	0.16	**0.00**
	15	15	10	34.47	15.57	6.83	2.35	0.30	0.00	35.41	17.14	6.97	1.90	0.36	**0.00**
	15	15	30	40.98	22.75	14.30	7.03	3.39	1.20	39.81	22.40	**13.02**	4.68	1.18	0.20
Combined	7	10	10	35.73	15.38	4.94	0.55	**0.00**	**0.00**	32.32	10.15	2.22	0.19	**0.00**	**0.00**
	7	10	30	40.31	20.80	7.37	2.65	0.14	**0.00**	35.24	10.73	2.69	0.32	**0.00**	**0.00**
	15	10	10	31.21	12.16	4.17	0.62	**0.16**	**0.00**	29.39	9.37	2.34	0.38	**0.00**	**0.00**
	15	10	30	38.39	20.63	9.66	4.20	1.15	0.21	34.99	13.51	4.25	0.89	0.21	0.00
	7	15	10	37.42	18.22	7.26	1.31	**0.00**	**0.00**	36.69	15.19	4.17	0.73	**0.00**	**0.00**
	7	15	30	40.98	25.81	10.18	3.82	0.82	**0.00**	38.98	18.42	4.81	1.24	0.08	**0.00**
	15	15	10	34.47	15.57	6.83	2.35	0.30	0.00	34.05	14.06	4.69	1.20	0.12	0.00
	15	15	30	40.98	22.75	14.30	7.03	3.39	1.20	38.78	19.76	8.54	2.88	0.72	0.07

Table 5.7 (Continued)

Method	No. of Aisles	No. of Items	Aisle Length	Number of Blocks											
				3						4					
				No. of Pickers						No. of Pickers					
				2	3	4	5	6	7	2	3	4	5	6	7
S-Shape	7	10	10	34.91	9.90	**2.23**	**0.28**	**0.00**	**0.00**	30.85	8.31	**1.79**	**0.11**	**0.00**	**0.00**
	7	10	30	35.36	9.51	**1.88**	**0.34**	**0.00**	**0.00**	29.54	6.45	**1.03**	**0.09**	**0.00**	**0.00**
	15	10	10	**36.11**	11.50	3.45	**0.53**	**0.11**	**0.00**	**33.54**	10.93	2.79	**0.54**	**0.09**	**0.00**
	15	10	30	**37.22**	13.42	3.86	**0.75**	0.05	**0.05**	32.54	10.00	2.48	0.40	**0.04**	**0.00**
	7	15	10	**41.02**	16.04	**4.32**	**0.82**	**0.00**	**0.00**	37.58	13.24	3.50	**0.66**	**0.00**	**0.00**
	7	15	30	**41.24**	16.88	3.90	**0.98**	**0.16**	**0.00**	36.95	12.56	2.60	**0.50**	**0.04**	**0.00**
	15	15	10	**41.88**	17.17	6.23	1.68	**0.40**	**0.00**	39.14	15.69	5.61	1.44	**0.26**	**0.00**
	15	15	30	**42.51**	20.28	8.46	2.69	**0.73**	**0.10**	38.94	16.42	5.95	1.63	**0.27**	0.00
Largest Gap	7	10	10	31.35	8.42	2.09	**0.28**	**0.00**	**0.00**	27.48	7.37	1.57	**0.11**	**0.00**	**0.00**
	7	10	30	27.51	7.20	1.77	0.28	**0.00**	**0.00**	22.50	5.16	0.94	0.04	**0.00**	**0.00**
	15	10	10	35.33	10.27	2.96	0.42	**0.11**	**0.00**	32.77	9.76	2.62	0.45	**0.09**	**0.00**
	15	10	30	31.48	9.83	3.16	0.70	0.05	**0.05**	27.12	7.61	2.14	0.36	**0.04**	**0.00**
	7	15	10	38.18	12.74	3.82	**0.82**	**0.00**	**0.00**	34.62	10.72	3.19	**0.66**	**0.00**	**0.00**
	7	15	30	34.48	11.40	3.45	0.92	**0.16**	**0.00**	30.01	8.55	2.29	0.46	**0.04**	**0.00**
	15	15	10	41.52	14.78	5.18	1.49	0.30	**0.00**	**39.90**	13.40	4.86	1.19	**0.26**	**0.00**
	15	15	30	37.25	15.07	5.99	2.09	0.58	**0.10**	34.04	11.96	4.43	1.25	0.23	0.00
Aisle-by-Aisle	7	10	10	**35.26**	**12.58**	1.96	0.14	**0.00**	**0.00**	**35.45**	**12.60**	1.24	0.00	**0.00**	**0.00**
	7	10	30	**37.31**	**14.10**	1.50	0.11	**0.00**	**0.00**	**37.02**	**12.89**	0.90	0.04	**0.00**	**0.00**
	15	10	10	32.75	**13.64**	**3.74**	**0.53**	**0.11**	**0.00**	33.51	**14.74**	**3.54**	0.45	**0.09**	**0.00**
	15	10	30	36.40	**18.30**	**5.01**	0.70	**0.05**	0.00	**36.13**	**18.88**	**3.89**	**0.48**	0.00	**0.00**
	7	15	10	37.53	**20.89**	4.20	0.55	**0.00**	**0.00**	**37.58**	**21.59**	**3.92**	0.22	**0.00**	**0.00**
	7	15	30	39.03	**24.37**	**4.23**	0.44	0.05	**0.00**	**38.65**	**23.70**	**3.48**	0.21	**0.04**	**0.00**
	15	15	10	36.16	**18.53**	**7.83**	**2.17**	0.30	**0.00**	36.84	**19.54**	**9.20**	**2.11**	**0.26**	**0.00**
	15	15	30	39.14	**23.18**	**12.91**	**3.65**	0.68	0.05	**39.15**	**22.92**	**13.11**	**3.20**	**0.27**	**0.04**
Combined	7	10	10	30.37	7.87	1.68	0.14	**0.00**	**0.00**	26.88	6.91	1.35	0.00	**0.00**	**0.00**
	7	10	30	29.40	6.82	1.12	0.11	**0.00**	**0.00**	24.09	4.63	0.69	0.04	**0.00**	**0.00**
	15	10	10	32.33	9.50	2.47	0.21	**0.11**	**0.00**	30.38	9.30	2.28	0.36	**0.09**	**0.00**
	15	10	30	32.07	9.92	2.56	0.35	**0.05**	0.00	27.57	7.56	1.72	0.24	0.00	**0.00**
	7	15	10	37.34	12.54	3.19	0.55	**0.00**	**0.00**	33.69	10.53	2.78	0.44	**0.00**	**0.00**
	7	15	30	36.62	12.38	2.55	0.44	0.05	**0.00**	31.89	9.00	1.73	0.21	**0.04**	**0.00**
	15	15	10	38.68	13.95	4.58	1.10	0.20	**0.00**	35.90	13.06	4.49	0.94	0.17	**0.00**
	15	15	30	38.11	16.10	5.61	1.53	0.29	0.05	34.30	12.64	4.03	0.92	0.12	**0.04**

Table 5.7 (Continued)

Method	No. of Aisles	No. of Items	Aisle Length	Number of Blocks					
				5					
				No. of Pickers					
				2	3	4	5	6	7
S-Shape	7	10	10	27.71	7.37	**1.40**	**0.09**	**0.00**	**0.00**
	7	10	30	25.16	4.80	**0.66**	**0.04**	**0.00**	**0.00**
	15	10	10	32.00	10.22	2.64	**0.46**	**0.08**	**0.00**
	15	10	30	29.01	8.06	1.80	0.20	**0.00**	**0.00**
	7	15	10	34.97	11.26	2.93	**0.55**	**0.00**	**0.00**
	7	15	30	33.31	9.52	1.96	**0.30**	**0.00**	**0.00**
	15	15	10	37.69	14.33	5.11	1.40	**0.22**	**0.00**
	15	15	30	36.02	13.84	4.27	1.10	0.16	0.00
Largest Gap	7	10	10	24.80	6.56	**1.40**	**0.09**	**0.00**	**0.00**
	7	10	30	18.94	3.98	0.63	**0.04**	**0.00**	**0.00**
	15	10	10	31.13	9.26	2.49	0.39	**0.08**	**0.00**
	15	10	30	24.07	6.39	1.55	0.20	**0.00**	**0.00**
	7	15	10	31.55	9.28	2.76	0.46	**0.00**	**0.00**
	7	15	30	26.01	6.79	1.66	0.27	**0.00**	**0.00**
	15	15	10	**37.73**	12.58	4.52	1.18	**0.22**	**0.00**
	15	15	30	30.78	10.13	3.29	0.88	0.13	0.00
Aisle-by-Aisle	7	10	10	**35.55**	**12.45**	0.94	0.00	**0.00**	**0.00**
	7	10	30	**36.86**	**12.23**	0.56	0.00	**0.00**	**0.00**
	15	10	10	**33.93**	**15.37**	**3.30**	0.39	0.00	**0.00**
	15	10	30	**36.33**	**18.33**	**3.63**	**0.23**	**0.00**	**0.00**
	7	15	10	**37.58**	**21.94**	**3.37**	0.18	**0.00**	**0.00**
	7	15	30	**38.54**	**24.03**	**2.96**	0.10	**0.00**	**0.00**
	15	15	10	37.28	**20.41**	**9.72**	**2.12**	**0.22**	**0.00**
	15	15	30	**39.00**	**23.35**	**12.93**	**2.69**	**0.25**	0.00
Combined	7	10	10	24.57	6.25	1.13	**0.09**	**0.00**	**0.00**
	7	10	30	20.36	3.60	0.42	0.00	**0.00**	**0.00**
	15	10	10	29.39	9.10	2.27	0.31	**0.08**	**0.00**
	15	10	30	24.59	6.36	1.26	0.13	**0.00**	**0.00**
	7	15	10	30.92	9.16	2.50	0.37	**0.00**	**0.00**
	7	15	30	27.95	6.80	1.30	0.14	**0.00**	**0.00**
	15	15	10	34.66	12.31	4.18	1.11	0.15	**0.00**
	15	15	30	31.09	10.64	3.06	0.66	0.06	**0.03**

tages in terms of congestion in the aisles and familiarity of pick locations. Furthermore, the results shows that the gap gets larger with more number of picking items and less number of aisles while the length of an aisle does not seem to be significant factor regarding the use of zone- or batch-picking policy. Finally, needless to say, we get the same results when there is only one zone or batch to pick.

5.8 Concluding Remarks

In this chapter, we have focused on the multi-picker OPP in parallel-aisle warehouses under a synchronized dynamic zone-picking policy, with the objective of minimizing order lead time by minimizing the maximum travel time of each picker, while aiming to ensure the balance among pickers. This problem is important in the sense of making practical and effective use of resources while fulfilling the customer orders within increasingly competitive due dates.

After proposing integer programming formulations for the cases without and with zoning policies, we have introduced a knapsack-based DP heuristic for the two-picker min-max OPP without zoning, as well as a novel polynomial-time exact algorithm for the min-max OPP in synchronized dynamic zone-picking systems. This DP algorithm assigns pickers for zone-picking and returns the optimal combination of zone assignments for multiple pickers. Lastly, we focus on the multi-picker OPP from a batch-picking perspective. Throughout the extensive computational experiments, we have tested the performance of the algorithms, analysed the impact of zoning to understand the extent of the suboptimality from imposing zone picking, analysed the impact of multiple pickers on lead time savings and showed how the DP can be adapted to multiple blocks.

Table 5.8: Performance comparison of zone-picking and batch-picking policies

Aisles	Items	Length	No. of zones/batches				
			1	2	3	4	5
			Zone-picking lead times (in seconds)				
7	10	10	**70.58**	**45.97**	**39.12**	**36.65**	**36.24**
7	10	30	**148.90**	**93.20**	**78.57**	**72.37**	**70.04**
15	10	10	**110.51**	**77.07**	**67.89**	**64.73**	**64.19**
15	10	30	**205.25**	**129.95**	**108.99**	**99.84**	**95.76**
7	15	10	**82.11**	**51.04**	**42.24**	**38.44**	**37.64**
7	15	30	**180.32**	**105.61**	**84.18**	**76.92**	**73.38**
15	15	10	**131.53**	**86.40**	**73.79**	**68.59**	**66.82**
15	15	30	**258.61**	**154.35**	**122.76**	**109.63**	**102.99**
			Batch-picking lead times (in seconds)				
7	10	10	70.58	53.86	49.27	48.46	47.46
7	10	30	148.90	108.70	97.46	95.30	92.72
15	10	10	110.51	86.20	78.94	77.58	76.02
15	10	30	205.25	148.23	130.11	126.34	122.27
7	15	10	82.11	63.97	57.17	55.67	55.07
7	15	30	180.32	132.66	115.34	111.39	109.71
15	15	10	131.53	101.84	90.00	87.15	86.21
15	15	30	258.61	184.96	155.68	147.49	144.94
			Average Gap %				
7	10	10	**0.00**	0.17	0.26	0.32	0.31
7	10	30	**0.00**	0.17	0.24	0.32	0.32
15	10	10	**0.00**	**0.12**	**0.16**	**0.20**	**0.18**
15	10	30	**0.00**	0.14	0.19	0.27	0.28
7	15	10	**0.00**	0.25	0.35	0.45	0.46
7	15	30	**0.00**	0.26	0.37	0.45	0.50
15	15	10	**0.00**	0.18	0.22	0.27	0.29
15	15	30	**0.00**	0.20	0.27	0.35	0.41

CHAPTER 6

CONCLUSION

Order picking is the most expensive and labor-intensive warehouse activity. In this thesis, we present compact mathematical models and exact/heuristic algorithms for various order picking problems.

In Chapters 3 and 4, we have approached the OPP from an arc routing perspective. The OPP formulations in the literature depend on the famous TSP and its derivatives, including Scholz et al. (2016) and Pansart et al. (2018). However, taking the TSP as a base case and developing new constraints exploiting the special properties of the parallel-aisle warehouse not only make the formulations more difficult to harness for the researchers but also it prevents to extend the OPP to different variants. In this thesis, we present compact arc routing-based formulations for the single- and two-block OPP in parallel-aisle warehouses. Our approach is important in the sense that it is an arc routing-based approach making use of specifics of the graph structure corresponding to the warehouse layout. The disconnectivity elimination constraints proposed in our studies are the direct implications of the observations we made on this special structure. Since it is also a compact formulation, it can be a base picker routing model for more complex integrated operational warehouse problems or OPPs with multiple blocks or multiple pickers.

Throughout these chapters, we firstly describe the problem in the single- and two-block parallel-aisle warehouse layouts and present the related literature. Afterwards, we constitute the single-block OPP formulation. Here, we define and classify the intra-aisle and cross-aisle movement types, mention how a sequential relation between these movements can also halves the feasible region, and we explain how to formulate the degree constraints by only using the odd-degree movements. At the same time, we present the assignment, sequencing and degree constraints. Afterwards, we introduce the disconnectivity elimination constraints by defining when and how such disconnectivities can occur. Also, we present a lazy constraint approach for the constraints which polynomially increase the size of the problem. Later, in the experiments section, we show the significant contribution of the lazy constraint approach. Next, we extend the binary integer programming formulation in the preceding chapter to the two-block OPP by also explaining how to modify the index sets and parameters, the additional formations in cross-aisle movements and disconnectivities. Lastly, for layouts with more than two blocks, we present an easy-to-implement picker routing heuristic which performs relatively good for the layouts with longer aisle lengths. Finally, we test the efficiency of the proposed models in terms of computation times and integrality gap. In both single- and two-block warehouses, our formulations consist of assignment, sequencing, degree and disconnectivity elimination constraints. while the first three types of constraints occur similarly in both cases, additional disconnectivities, and therefore disconnectivity elimination constraints, are defined for the two-block case. These constraints are formulated as lazy constraints, substantially improving the efficiency of the formulations. The solution approaches and the results of the computational experiments in these two chapters are submitted and are under review in a peer-reviewed journal.

In Chapter 5, we consider the multi-picker OPP under a parallel dynamic zone-picking policy where each picker is assigned to a dedicated zone of aisles at each pick wave. We balance the workload among pickers by minimizing the maximum distance traveled by the latest picker and make them use the minimum-time routes. Considering the balance of the workload distributed among pickers and tight due dates promised to customers, we present mathematical models, and propose exact and heuristic algorithms which minimize the lead time of the wave zone-picking process for a given the number of aisles and a given number of pickers. Firstly, we present VRP-MINMAX formulations by extending compact VRP formulations for cases with and without zoning. However, these formulations perform poorly with the increase in the number of picking items. Thus, we focus on approximate and exact dynamic programming algorithms. In this regard, we consider two-picker OPP as a preliminary step, and then generalize it to multi-picker OPP in parallel-aisle single-block/multi-block warehouses under a zone-picking policy. Thanks to the properties of Pascal's Triangle, the proposed algorithm on dynamic zone-picking for multiple pickers performs significantly quickly and exactly. To the best of our knowledge, this study fills two important gaps regarding the order picking operations. It is the first study dealing with multiple-picker OPP on multi-block layouts. Also, there is a clear gap in integrated OPP literature considering zone picking since it is mostly considered as a tactical level decision. This study fills this gap since it is an integrated zone-picking, picker routing and workforce allocation problem. The solution approaches and the results of the computational experiments of this chapter have been published in a peer-reviewed journal (Saylam et al. 2022).

6.1 Major Findings

In general, computational experiments on randomly generated instances in line with those in the literature show that (i) the arc-routing based formulations perform at least as good as the ones in the literature in terms of computing times, and (ii) the proposed algorithms can find optimal and near-optimal solutions in negligible computational times.

Our computational experiments from Chapters 3 and 4 show that the performance of the proposed formulation for single-block layout outperforms all TSP-based formulations while it is comparable to that of Goeke & Schneider (2021), and also its straightforward extension to the two-block layout outperforms the best-known approaches for the two-block OPP to date. Other noteworthy findings obtained from our computational experiments include: (i) although the number of constraints is in quadratic order of the number of aisles, applying the multi-aisle disconnectivity elimination constraints as lazy constraints keeps the constraint size linear for the most of the instances and also significantly decreases the actual number of constraints and the computing times, (ii) the integrality gap of the LP relaxation is particularly lower as the size of the instance increases, due to an increase in the number of disconnectivity elimination constraints, (iii) the computing times are significantly shorter as the ratio of the number of pick locations to the number of aisles is larger, (iv) the difference between computing times of single- and two-block formulations increases with a larger number of aisles as the need of more disconnectivity elimination constraints also increases, and (v) the proposed multi-block heuristic outperforms all its counterparts for the layouts with large number of blocks and large size of aisle lengths.

Important findings obtained from our computational experiments in Chapter 5 are that (i) the proposed algorithm for the min-max OPP with two pickers generates efficient solutions very quickly as the optimality gap averages and the standard deviations are low, (ii) for relatively large aisles, assigning a second picker reduces the order picking time around 45% in average while this impact exponentially diminishes with each additional picker, (iii) the proposed synchronized dynamic zone-picking approach with a min-max objective spontaneously leads to the balanced partition of the pickers' workload thus reduces management supervision, and (iv) the travel time gap between "no zoning" and "zoning" strategies quickly disappears with the introduction of additional pickers. It is important to note that we have worked on uniformly distributed pick locations. That is no demand frequency is taken into account. If a class-based storage policy, where fast moving class products are stored in the closest aisles, is applied, the advantage of dynamic zoning policy would be more significant. This is because the possibility of having a pick location in the further aisles would be relatively small. For the same reason, the proposed algorithm will terminate faster.

6.2 Future Research Directions

For further research, it is noteworthy that the proposed arc routing-based models can be embedded as the routing subproblem into the integrated/combined order picking operation problems or can be extended for different variants of the OPP, where each of these variations could yield a significant contribution to the literature. In this regard, one can aim to extend the compact model to such different variants and still keep it compact and efficient with polynomial number of variables and constraints. The proposed formulation can be extended to a general number of blocks. In such a formulation, assignment, sequencing and degree constraints would be straightforward, whereas the increase in the number of disconnectivity elimination constraints would pose the main challenge. But first, one should analyse the solution of the instances in order to better generalize how and when such disconnectivities occur for multi-block layouts.

It is also possible to use the proposed formulations in a matheuristic for the multi-block OPP as the assignment, sequencing and degree constraints in this chapter could lead to a very tight lower bound. The formulation can also be extended to turn restrictions and delays at intersections. In problems with turn penalties, turn costs could be added during pre-processing for cross-aisle keeping movements according to their number of turns. For example, type (2) cross-aisle keeping movement has one turn while type (4) cross-aisle keeping movement has two turns at an aisle. In the same sense, intersection delay costs could be added during pre-processing for each movement according to their number of intersection visits. An optimal route, then, minimizes the sum of the travel time and the turn/intersection delay penalties. Last but not least, a modified RR algorithm can be studied using the directed movements introduced in this thesis.

On the other hand, the integrated work regarding the min-max OPP under synchronized dynamic zone-picking system can also be further improved by considering joint zone- and batch-picking approach. The cost factor is not considered in this study and can be a future research direction. Since each additional picker reduces the zone-picking lead time in a decreasing pace, the optimal configuration of zoning and the number of pickers can be further explored with consideration of the cost and service level objectives. Moreover, a cost-related study should also consider other cost factors that vary with the number of zones. For example, as dynamic zones create a more challenging environment for the pickers, a successful operation requires an intact communication and coordination with the help of supporting systems such as pick-by-voice systems. Also, unique features of warehouses, such as intersection points between picking aisles and cross-aisles, intra-aisle and inter-aisle movement types, can be included in the OPP formulations. Such an extension could strengthen the formulation for the multi-picker OPP. The integer programming model can be adapted to a matheuristic to improve the algorithm. The proposed travel time balancing algorithm could also be embedded into various heuristics for further improvement purposes. Finally, the introduced exact dynamic programming approach for the optimal single-block zone assignment can be further studied for an extension to multiple-block layouts.

REFERENCES

Ardjmand, E., Bajgiran, O. S., & Youssef, E. (2019). Using list-based simulated annealing and genetic algorithm for order batching and picker routing in put wall based picking systems. *Applied Soft Computing*, *75*, 106–119. doi:10.1016/j.asoc.2018.11.019.

Ardjmand, E., Shakeri, H., Singh, M., & Bajgiran, O. S. (2018). Minimizing order picking makespan with multiple pickers in a wave picking warehouse. *International Journal of Production Economics*, *206*, 169–183. doi:10.1016/j.ijpe.2018.10.001.

Atashi Khoei, A., Süral, H., & Tural, M. K. (2022). Energy minimizing order picker forklift routing problem. *(in press) European Journal of Operational Research*. doi:10.1016/j.ejor.2022.08.038.

Bartholdi, J. J., & Eisenstein, D. D. (1996). A production line that balances itself. *Operations Research*, *44*(1), 21–34. doi:10.1287/opre.44.1.21.

Bartholdi, J. J., Eisenstein, D. D., & Foley, R. D. (2001). Performance of bucket brigades when work is stochastic. *Operations Research*, *49*(5), 710–719. doi:10.1287/opre.49.5.710.10609.

Bartholdi, J. J., & Hackman, S. (2019). Warehouse and Distribution Science. Release 0.98.1. https://www.warehouse-science.com/book/editions/wh-sci-0.98.1.pdf. Accessed: June 2022.

Bayram, V., Baloch, G., Gzara, F., & Elhedhli, S. (2022). Optimal order batching in warehouse management: A data-driven robust approach. *INFORMS Journal on Optimization*, *4*(3), 278–303. doi:10.1287/ijoo.2021.0066.

Burkard, R. E., Deineko, V. G., Van Dal, R., van der Veen, J. A., & Woeginger, G. J.

(1998). Well-solvable special cases of the traveling salesman problem: a survey. *SIAM Review*, *40*(3), 496–546. doi:10.1137/S0036144596297514.

Cambazard, H., & Catusse, N. (2018). Fixed-parameter algorithms for rectilinear steiner tree and rectilinear traveling salesman problem in the plane. *European Journal of Operational Research*, *270*(2), 419–429. doi:10.1016/j.ejor.2018.03.042.

Çelik, M., & Süral, H. (2014). Order picking under random and turnover-based storage policies in fishbone aisle warehouses. *IIE Transactions*, *46*(3), 283–300. doi:10.1080/0740817X.2013.768871.

Çelik, M., & Süral, H. (2016). Order picking in a parallel-aisle warehouse with turn penalties. *International Journal of Production Research*, *54*(14), 4340–4355. doi:10.1080/00207543.2016.1154624.

Çelik, M., & Süral, H. (2019). Order picking in parallel-aisle warehouses with multiple blocks: Complexity and a graph theory-based heuristic. *International Journal of Production Research*, *57*(3), 888–906. doi:10.1080/00207543.2018.1489154.

Çelik, M., Archetti, C., & Süral, H. (2022). Inventory routing in a warehouse: The storage replenishment routing problem. *European Journal of Operational Research*, *301*(3), 1117–1132. doi:10.1016/j.ejor.2021.11.056.

Çeven, E., & Gue, K. R. (2017). Optimal wave release times for order fulfillment systems with deadlines. *Transportation Science*, *51*(1), 52–66. doi:10.1287/trsc.2015.0642.

Chabot, T., Lahyani, R., Coelho, L. C., & Renaud, J. (2017). Order picking problems under weight, fragility and category constraints. *International Journal of Production Research*, *55*(21), 6361–6379. doi:10.1080/00207543.2016.1251625.

Chen, F., Wang, H., Qi, C., & Xie, Y. (2013). An ant colony optimization routing algorithm for two order pickers with congestion consideration. *Computers & Industrial Engineering*, *66*(1), 77–85. doi:10.1016/j.cie.2013.06.013.

Chen, F., Wang, H., Xie, Y., & Qi, C. (2016). An ACO-based online routing method for multiple order pickers with congestion consideration in warehouse. *Journal of Intelligent Manufacturing*, *27*(2), 389–408. doi:10.1007/s10845-014-0871-1.

Chen, F., Xu, G., & Wei, Y. (2019). An integrated metaheuristic routing method for multiple-block warehouses with ultranarrow aisles and access restriction. *Complexity*, *Special Issue*, 69–82. doi:10.1155/2019/1280285.

Chen, T.-L., Cheng, C.-Y., Chen, Y.-Y., & Chan, L.-K. (2015). An efficient hybrid algorithm for integrated order batching, sequencing and routing problem. *International Journal of Production Economics*, *159*, 158–167. doi:10.1016/j.ijpe.2014.09.029.

Claus, A. (1984). A new formulation for the travelling salesman problem. *SIAM Journal on Algebraic Discrete Methods*, *5*(1), 21–25. doi:10.1137/0605004.

De Koster, R. (1994). Performance approximation of pick-to-belt orderpicking systems. *European Journal of Operational Research*, *72*(3), 558–573. doi:10.1016/0377-2217(94)90423-5.

De Koster, R. B., Le-Duc, T., & Zaerpour, N. (2012). Determining the number of zones in a pick-and-sort order picking system. *International Journal of Production Research*, *50*(3), 757–771. doi:10.1080/00207543.2010.543941.

De Koster, R. B. M., Le-Duc, R., & Roodbergen, K. J. (2007). Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, *182*(2), 481–501. doi:10.1016/j.ejor.2006.07.009.

De Santis, R., Montanari, R., Vignali, G., & Bottani, E. (2018). An adapted ant colony optimization algorithm for the minimization of the travel distance of pickers in manual warehouses. *European Journal of Operational Research*, *267*(1), 120–137. doi:10.1016/j.ejor.2017.11.017.

van Der Gaast, J. P., De Koster, R. B., Adan, I. J., & Resing, J. A. (2020). Capacity analysis of sequential zone picking systems. *Operations Research*, *68*(1), 161–179. doi:10.1287/opre.2019.1885.

D'Haen, R., Braekers, K., & Ramaekers, K. (2022). Integrated scheduling of order picking operations under dynamic order arrivals. *(in press) International Journal of Production Research*. doi:10.1080/00207543.2022.2078747.

Eglese, R. W., & Letchford, A. N. (2000). Polyhedral theory for arc routing problems. In *Arc Routing* (pp. 199–230). Springer. doi:10.1007/978-1-4615-4495-1_1.

Eiselt, H. A., & Laporte, G. (2000). A historical perspective on arc routing. In *Arc Routing* (pp. 1–16). Springer. doi:10.1007/978-1-4615-4495-1_6.

Feller, W. (2008). *An introduction to probability theory and its applications, vol 2*. John Wiley & Sons.

Gademann, A. N., van den Berg, J. P., & van der Hoff, H. H. (2001). An order batching algorithm for wave picking in a parallel-aisle warehouse. *IIE Transactions*, *33*(5), 385–398. doi:10.1080/07408170108936837.

Gademann, N., & Velde, V. D. S. (2005). Order batching to minimize total travel time in a parallel-aisle warehouse. *IIE Transactions*, *37*(1), 63–75. doi:10.1080/07408170590516917.

Gavish, B., & Graves, S. C. (1978). *The travelling salesman problem and related problems*. Working Paper GR-078-78. Operations Research Center, Massachusetts Institute of Technology. URL: http://hdl.handle.net/1721.1/5363f.

Goeke, D., & Schneider, M. (2021). Modeling single-picker routing problems in classical and modern warehouses. *INFORMS Journal on Computing*, *33*(2), 436–451. doi:10.1287/ijoc.2020.1040.

Gu, J., Goetschalckx, M., & McGinnis, L. F. (2007). Research on warehouse operation: A comprehensive review. *European Journal of Operational Research*, *177*(1), 1–21. doi:10.1016/j.ejor.2006.02.025.

Guo, S., Singh, M., & Goodarzi, S. (2022). Enhance picking viability in e-commerce warehouses under pandemic. *(in press) International Journal of Production Research*. doi:10.1080/00207543.2022.2101400.

Hall, R. W. (1993). Distance approximations for routing manual pickers in a warehouse. *IIE Transactions*, *25*(4), 76–87. doi:10.1080/07408179308964306.

Helsgaun, K. (2000). An effective implementation of the lin–kernighan traveling salesman heuristic. *European Journal of Operational Research*, *126*(1), 106–130. doi:10.1016/S0377-2217(99)00284-2.

Henn, S., Koch, S., & Wäscher, G. (2012). Order batching in order picking warehouses: A survey of solution approaches. In R. Manzini (Ed.), *Warehousing in the Global Supply Chain: Advanced Models, Tools and Applications for Storage Systems* (pp. 105–137). London: Springer London. doi:10.1007/978-1-4471-2274-6_6.

Henn, S., & Schmid, V. (2013). Metaheuristics for order batching and sequencing in manual order picking systems. *Computers & Industrial Engineering*, *66*(2), 338–351. doi:10.1016/j.cie.2013.07.003.

Henn, S., & Wäscher, G. (2012). Tabu search heuristics for the order batching problem in manual order picking systems. *European Journal of Operational Research*, *222*(3), 484–494. doi:10.1016/j.ejor.2012.05.049.

Ho, Y.-C., & Liao, T.-W. (2009). Zone design and control for vehicle collision prevention and load balancing in a zone control agv system. *Computers & Industrial Engineering*, *56*(1), 417–432. doi:10.1016/j.cie.2008.07.007.

Hong, S., Johnson, A. L., & Peters, B. A. (2012). Batch picking in narrow-aisle order picking systems with consideration for picker blocking. *European Journal of Operational Research*, *221*(3), 557–570. doi:10.1016/j.ejor.2012.03.045.

Jane, C.-C., & Laih, Y.-W. (2005). A clustering algorithm for item assignment in a synchronized zone order picking system. *European Journal of Operational Research*, *166*(2), 489–496. doi:10.1016/j.ejor.2004.01.042.

Jang, H. Y., & Sun, J. U. (2012). A graph optimization algorithm for warehouses with middle cross aisles. *Applied Mechanics and Materials*, *145*, 354–358. doi:10.4028/www.scientific.net/AMM.145.354.

Karp, R. M. (1972). Reducibility among combinatorial problems. In R. E. Miller, J. W. Thatcher, & J. D. Bohlinger (Eds.), *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations* (pp. 85–103). Boston, MA: Springer US. doi:10.1007/978-1-4684-2001-2_9.

Koster, M. B. M. D., der Poort, E. V., & Wolters, M. (1999). Efficient orderbatching methods in warehouses. *International Journal of Production Research*, *37*(7), 1479–1504. doi:10.1080/002075499191094.

Lamballais, T., Merschformann, M., Roy, D., de Koster, M., Azadeh, K., & Suhl, L. (2022). Dynamic policies for resource reallocation in a robotic mobile fulfillment system with time-varying demand. *European Journal of Operational Research*, *300*(3), 937–952. doi:10.1016/j.ejor.2021.09.001.

Letchford, A. N., Nasiri, S. D., & Theis, D. O. (2013). Compact formulations of the steiner traveling salesman problem and related problems. *European Journal of Operational Research*, *228*(1), 83–92. doi:10.1016/j.ejor.2013.01.044.

Lin, C.-C., Kang, J.-R., Hou, C.-C., & Cheng, C.-Y. (2016). Joint order batching and picker manhattan routing problem. *Computers & Industrial Engineering*, *95*, 164–174. doi:10.1016/j.cie.2016.03.009.

Masae, M., Glock, C. H., & Grosse, E. H. (2020a). Order picker routing in warehouses: A systematic literature review. *International Journal of Production Economics*, *224*, 107564. doi:10.1016/j.ijpe.2019.107564.

Masae, M., Glock, C. H., & Vichitkunakorn, P. (2020b). Optimal order picker routing in a conventional warehouse with two blocks and arbitrary starting and ending points of a tour. *International Journal of Production Research*, *58*(17), 5337–5358. doi:10.1080/00207543.2020.1724342.

Masae, M., Glock, C. H., & Vichitkunakorn, P. (2020c). Optimal order picker routing in the chevron warehouse. *IISE Transactions*, *52*(6), 665–687. doi:10.1080/24725854.2019.1660833.

Masae, M., Padungwech, W., & Vichitkunakorn, P. (2022). Evaluating the effects of different factors on the order picking efficiency in the single-block warehouse. *Thai Journal of Operations Research: TJOR*, *10*(1), 94–102. URL: https://ph02.tci-thaijo.org/index.php/TJOR/article/view/245983/167415.

Melacini, M., Perotti, S., & Tumino, A. (2011). Development of a framework for pick-and-pass order picking system design. *The International Journal of Advanced Manufacturing Technology*, *53*(9), 841–854. doi:10.1007/s00170-010-2881-2.

Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *J. ACM*, *7*(4), 326–329. doi:10.1145/321043.321046.

Pansart, L., Catusse, N., & Cambazard, H. (2018). Exact algorithms for the order picking problem. *Computers & Operations Research*, *100*, 117–127. doi:10.1016/j.cor.2018.07.002.

Parikh, P. J., & Meller, R. D. (2008). Selecting between batch and zone order picking strategies in a distribution center. *Transportation Research Part E: Logistics and Transportation Review*, *44*(5), 696–719. doi:10.1016/j.tre.2007.03.002.

Pearce, R. H. (2019). *Towards a general formulation of lazy constraints*. Ph.D. thesis The University of Queensland.

Petersen, C. G. (1997). An evaluation of order picking routeing policies. *International Journal of Production & Operations Management*, *17*(11), 1098–1111. doi:10.1108/01443579710177860.

Pinto, A. R. F., & Nagano, M. S. (2022). A comprehensive review of batching problems in low-level picker-to-parts systems with order due dates: Main gaps, trade-offs, and prospects for future research. *Journal of Manufacturing Systems*, *65*, 1–18. doi:10.1016/j.jmsy.2022.08.006.

Ratliff, H. D., & Rosenthal, A. S. (1983). Order-picking in a rectangular warehouse: A solvable case of the traveling salesman problem. *Operations Research*, *31*(3), 507–521. doi:10.1287/opre.31.3.507.

Roodbergen, K. J., & De Koster, R. B. M. (2001a). Routing methods for warehouses with multiple cross aisles. *International Journal of Production Research*, *39*(9), 1865–1883. doi:10.1080/00207540110028128.

Roodbergen, K. J., & De Koster, R. B. M. (2001b). Routing order pickers in a warehouse with a middle aisle. *European Journal of Operational Research*, *133*(1), 32–43. doi:10.1016/S0377-2217(00)00177-6.

Rouwenhorst, B., Reuter, B., Stockrahm, V., van Houtum, G.-J., Mantel, R., & Zijm, W. H. (2000). Warehouse design and control: Framework and literature review. *European Journal of Operational Research*, *122*(3), 515–533. doi:10.1016/S0377-2217(99)00020-X.

Roy, D., Krishnamurthy, A., Heragu, S. S., & Malmborg, C. J. (2012). Performance analysis and design trade-offs in warehouses with autonomous vehicle technology. *IIE Transactions*, *44*(12), 1045–1060. doi:10.1080/0740817X.2012.665201.

Roy, D., Nigam, S., De Koster, R., Adan, I., & Resing, J. (2019). Robot-storage zone assignment strategies in mobile fulfillment systems. *Transportation Research Part E: Logistics and Transportation Review*, *122*, 119–142. doi:10.1016/j.tre.2018.11.005.

Ruberg, Y., & Scholz, A. (2016). *A mathematical programming formulation for the single-picker routing problem in a multi-block layout*. Working Paper 2/2016 Faculty of Economics and Management, Universität Magdeburg, Germany. doi:10.24352/UB.OVGU-2018-550.

Saylam, S., Çelik, M., & Süral, H. (2022). The min–max order picking problem in synchronised dynamic zone-picking systems. *(in press) International Journal of Production Research*. doi:10.1080/00207543.2022.2058433.

Scholz, A. (2016). *An exact solution approach for the single-picker routing problem in warehouses with an arbitrary block layout*. Working Paper 6/2016 Faculty of Economics and Management, Universität Magdeburg, Germany. doi:10.24352/UB.OVGU-2018-554.

Scholz, A., Henn, S., Stuhlmann, M., & Wäscher, G. (2016). A new mathematical programming formulation for the single-picker routing problem. *European Journal of Operational Research*, *253*(1), 68–84. doi:10.1016/j.ejor.2016.02.018.

Scholz, A., Schubert, D., & Wäscher, G. (2017). Order picking with multiple pickers and due dates–simultaneous solution of order batching, batch assignment and sequencing, and picker routing problems. *European Journal of Operational Research*, *263*(2), 461–478. doi:10.1016/j.ejor.2017.04.038.

Scholz, A., & Wäscher, G. (2017). Order batching and picker routing in manual order picking systems: the benefits of integrated routing. *Central European Journal of Operations Research*, *25*(2), 491–520. doi:10.1007/s10100-017-0467-x.

Srinivas, S., & Yu, S. (2022). Collaborative order picking with multiple pickers and robots: Integrated approach for order batching, sequencing and picker-robot routing. *International Journal of Production Economics*, *254*, 108634. doi:10.1016/j.ijpe.2022.108634.

Su, Y., Zhu, X., Yuan, J., Teo, K. L., Li, M., & Li, C. (2022). An extensible multi-block layout warehouse routing optimization model. *(in press) European Journal of Operational Research*. doi:10.1016/j.ejor.2022.05.045.

Theys, C., Bräysy, O., Dullaert, W., & Raa, B. (2010). Using a TSP heuristic for routing order pickers in warehouses. *European Journal of Operational Research*, *200*(3), 755–763. doi:10.1016/j.ejor.2009.01.036.

Tompkins, J. A., White, J. A., Bozer, Y. A., & Tanchoco, J. M. A. (2010). *Facilities Planning*. Hoboken, NJ: John Wiley & Sons.

Tsai, C.-Y., Liou, J. J., & Huang, T.-M. (2008). Using a multiple-GA method to solve the batch picking problem: considering travel distance and order due time. *International Journal of Production Research*, *46*(22), 6533–6555. doi:10.1080/00207540701441947.

Valle, C. A., Beasley, J. E., & da Cunha, A. S. (2017). Optimally solving the joint order batching and picker routing problem. *European Journal of Operational Research*, *262*(3), 817–834. doi:10.1016/j.ejor.2017.03.069.

van der Gaast, J. P., & Weidinger, F. (2022). A deep learning approach for the selection of an order picking system. *European Journal of Operational Research*, *302*(2), 530–543. doi:10.1016/j.ejor.2022.01.006.

van Gils, T., Caris, A., Ramaekers, K., & Braekers, K. (2019a). Formulating and solving the integrated batching, routing, and picker scheduling problem in a real-life spare parts warehouse. *European Journal of Operational Research*, *277*(3), 814–830. doi:10.1016/j.ejor.2019.03.012.

van Gils, T., Caris, A., Ramaekers, K., Braekers, K., & de Koster, R. B. (2019b). Designing efficient order picking systems: The effect of real-life features on the relationship among planning problems. *Transportation Research Part E: Logistics and Transportation Review*, *125*, 47–73. doi:10.1016/j.tre.2019.02.010.

van Gils, T., Ramaekers, K., Braekers, K., Depaire, B., & Caris, A. (2018a). Increasing order picking efficiency by integrating storage, batching, zone picking, and routing policy decisions. *International Journal of Production Economics*, *197*, 243–261. doi:10.1016/j.ijpe.2017.11.021.

van Gils, T., Ramaekers, K., Caris, A., & Koster, R. B. M. D. (2018b). Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review. *European Journal of Operational Research*, *267*(1), 1–15. doi:10.1016/j.ejor.2017.09.002.

Vanheusden, S., van Gils, T., Braekers, K., Ramaekers, K., & Caris, A. (2022a). Analysing the effectiveness of workload balancing measures in order picking operations. *International Journal of Production Research*, *60*(7), 2126–2150. doi:10.1080/00207543.2021.1884307.

Vanheusden, S., van Gils, T., Ramaekers, K., Cornelissens, T., & Caris, A. (2022b). Practical factors in order picking planning: state-of-the-art classification and review. *(in press) International Journal of Production Research*. doi:10.1080/00207543.2022.2053223.

Vanheusden, S., van Gils, T., Caris, A., Ramaekers, K., & Braekers, K. (2020). Operational workload balancing in manual order picking. *Computers & Industrial Engineering*, *141*, 106269. doi:10.1016/j.cie.2020.106269.

Vaughan, T. S., & Petersen, C. G. (1999). The effect of warehouse cross aisles on order picking efficiency. *International Journal of Production Research*, *37*(4), 881–897. doi:10.1080/002075499191580.

Yu, M., & De Koster, R. (2008). Performance approximation and design of pick-and-pass order picking systems. *IIE Transactions*, *40*(11), 1054–1069. doi:10.1080/07408170802167613.

Yu, M., & De Koster, R. B. (2009). The impact of order batching and picking area zoning on order picking system performance. *European Journal of Operational Research*, *198*(2), 480–490. doi:10.1016/j.ejor.2008.09.011.

Zhang, J., Zhang, X., & Zhang, N. (2023). Considering pickers' learning effects in selecting between batch picking and batch-synchronized zone picking for online-to-offline groceries. *Applied Mathematical Modelling*, *113*, 358–375. doi:10.1016/j.apm.2022.09.009.

Zulj, I., Glock, C. H., Grosse, E. H., & Schneider, M. (2018). Picker routing and storage-assignment strategies for precedence-constrained order picking. *Computers & Industrial Engineering*, *123*, 338–347. doi:10.1016/j.cie.2018.06.015.

CURRICULUM VITAE

Serhat Saylam

EDUCATION

- Master of Science, Logistics & Supply Chain Management, Air Force Institute of Technology, Dayton, OH, United States, March 2012.
- Bachelor of Science, Electronics Engineering, Turkish Air Force Academy, İstanbul, Türkiye, August 2006.
- High School, Maltepe Military High School, İzmir, Türkiye, June 2002.

RESEARCH INTERESTS

- Operational Research
- Logistics & Supply Chain Optimisation
- Warehouse Management
- Routing
- Dynamic Programming

PUBLICATIONS

Saylam, S., Çelik, M., & Süral, H. (2022). The min–max order picking problem in synchronised dynamic zone-picking systems. *(in press) International Journal of Production Research.* doi:10.1080/00207543.2022.2058433.