Taylor & Francis
Taylor & Francis Group

International Journal of Production Research

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tprs20

Optimal order picker routing in a conventional warehouse with two blocks and arbitrary starting and ending points of a tour

Makusee Masae , Christoph H. Glock & Panupong Vichitkunakorn

To cite this article: Makusee Masae , Christoph H. Glock & Panupong Vichitkunakorn (2020) 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

To link to this article: https://doi.org/10.1080/00207543.2020.1724342

Published online: 11 Feb 2020.

International Journal of Production Research, 2020
Vol. 58, No. 17, 5337–5358, https://doi.org/10.1080/00207543.2020.1724342

Optimal order picker routing in a conventional warehouse with two blocks and arbitrary starting and ending points of a tour

Makusee Masae[a,b], Christoph H. Glock [a*] and Panupong Vichitkunakorn[b]

[a]*Institute of Production and Supply Chain Management, Technische Universität Darmstadt, Darmstadt, Germany;* [b]*Department of Mathematics and Statistics, Faculty of Science, Prince of Songkla University, Hat Yai, Songkhla, Thailand*

(*Received 9 July 2018; accepted 23 January 2020*)

This paper investigates manual order picking, where workers travel through the warehouse to retrieve requested items from shelves. To minimise the completion time of orders, researchers have developed various routing procedures that guide order pickers through the warehouse. The paper at hand contributes to this stream of research and proposes an optimal order picker routing policy for a conventional warehouse with two blocks and arbitrary starting and ending points of a tour. The procedure proposed in this paper extends an earlier work of Löffler et al. (2018. Picker routing in AGV-assisted order picking systems, Working Paper, DPO-01/2018, Deutsche Post Chair-Optimization of Distribution Networks, RWTH Aachen University, 2018) by applying the concepts of Ratliff and Rosenthal (1983. "Order-picking in a Rectangular Warehouse: a Solvable Case of the Traveling Salesman Problem." *Operations Research* 31 (3): 507–521) and Roodbergen and de Koster (2001a. "Routing Order Pickers in a Warehouse with a Middle Aisle." *European Journal of Operational Research* 133 (1): 32–43) that used graph theory and dynamic programming for finding an optimal picker route. We also propose a routing heuristic, denoted *S*-shape*, for conventional two-block warehouses with arbitrary starting and ending points of a tour. In computational experiments, we compare the average order picking tour length in a conventional warehouse with a single block to the case of a conventional warehouse with two blocks to assess the impact of the middle cross aisle on the performance of the warehouse. Furthermore, we evaluate the performance of the *S*-shape* heuristic by comparing it to the exact algorithm proposed in this study.

Keywords: warehousing systems; warehouse design; logistics; order picking methods; order picker routing; picker-to-parts system

1. Introduction

Order picking is the process of retrieving items from specified storage locations in the warehouse in response to customer orders. Order picking has become an essential component of every supply chain, and it is often considered the most laborious and time-consuming process in warehousing. Some researchers estimated that it accounts for up to 55% of the total warehouse operating cost (e.g. Tompkins et al. 2010), which illustrates that order picking efficiency is an important issue with a strong impact on total warehouse cost. In light of the cost impact of order picking, it is not surprising that prior research has proposed a variety of models that help managers to improve order picking operations and reduce the total cost of warehousing. Researchers have thereby frequently differentiated between five decision problems managers have to solve in order to ensure that an efficient order picking process is realised: order picker routing, order batching, storage assignment, zoning, and layout design (e.g. De Koster, Le-Duc, and Roodbergen 2007; Grosse, Glock, and Neumann 2017). The focus of the paper at hand is on the routing problem, which aims at finding a minimum-length tour for routing the order picker through the warehouse. Order picker routing is especially important in picker-to-parts order picking systems where order pickers travel through the warehouse (afoot or using an electric cart) to retrieve items from the shelves of the warehouse. Order picking tours usually start and end at a so-called depot, where order pickers receive orders and where retrieved items are dropped off for further processing.

A closer look at the literature shows that prior research on order picking assumed in many cases that a single depot exists in the warehouse, such that each order picking tour starts and ends at the depot. Examples include the works of Ratliff and Rosenthal (1983), Roodbergen and de Koster (2001a, 2001b), and Scholz et al. (2016). In practice, however, order picking tours do not necessarily start and end at the depot. De Koster and van der Poort (1998), for instance, discussed a scenario where the order pickers have the opportunity to drop off retrieved items at multiple drop-off locations at the head of each

*Corresponding author. Email: glock@pscm.tu-darmstadt.de

aisle. Another example is the case where order picking tours are updated in real time. In this case, while the order picker is travelling through the warehouse, a new order arrives at the warehouse. The warehouse manager or warehouse system then decides to update the order the picker is currently working on, such that a new tour starts at the position of the order picker where the tour update was received. The ending point of the tour would in such a case still be the depot. Works that investigated this scenario are those of Lu et al. (2016), Giannikas et al. (2017), and Chen, Wei, and Wang (2018), for example. Löffler et al. (2018) described another scenario where the order picker is accompanied by an automated guided vehicle (AGV) that automatically drives back to the depot once an order has been completed. The order picker can then continue with the next tour and use a new AGV without returning to the depot. The starting point of a tour would then be either the depot or the location of the last pick in the previous tour, while the ending point would either be the depot or the location of last pick of the current tour (from where the AGV starts driving back to the depot). The starting and ending points of a tour in this case can then be any locations in the warehouse.

All examples mentioned above investigated a situation where the warehouse is of rectangular shape with parallel picking aisles that are perpendicular to the two cross aisles at both ends of the picking aisles. Such warehouses have frequently been referred to as conventional warehouses with a single block. Order picking warehouses with two blocks have been studied as well (e.g. Roodbergen and De Koster 2001a, 2001b; Jang and Sun 2012), but only for the case where the starting and ending points of a tour are at the depot. The case where order picking tours can start and end at arbitrary positions in the warehouse has not been investigated for the two-block warehouse so far. The paper at hand aims to close this research gap by extending the work of Löffler et al. (2018) to the case where the starting and ending points of a tour can be any location in a conventional warehouse with two blocks. Permitting arbitrary starting and ending points of a tour in the two-block warehouse leads to new equivalence classes for the partial tour subgraphs in addition to the equivalence classes proposed by Löffler et al. (2018). In particular, the number of equivalence classes increases from 14 in Löffler et al. (2018) to 70, which is one of the contributions made by our paper. To address our problem, we apply the concepts of Ratliff and Rosenthal (1983) and Roodbergen and de Koster (2001a) that used graph theory and dynamic programming to find an optimal picker route in a conventional warehouse. The run-time complexity of the proposed algorithm is linear in the number of requested items and picking aisles. In this paper, we also propose a routing heuristic, denoted *S*-shape,* for solving the order picker routing problem in this scenario. This is another contribution of the paper at hand.

The main contributions of this study are therefore to (i) develop an exact algorithm and a routing heuristic for a conventional warehouse with two blocks where the starting and ending points of a tour can be any locations in the warehouse, (ii) assess the impact of the middle cross aisle on the performance of the warehouse using the average tour length obtained by the exact algorithm, and (iii) evaluate the performance of the proposed heuristic compared to the exact algorithm.

The remainder of this paper is organised as follows. Section 2 presents an overview of the related literature. Section 3 describes the problem investigated in this paper. Section 4 introduces and analyses the optimal and heuristic routing of order pickers when starting and ending points of a tour are given. Section 5 presents a numerical example as well as extensive computational experiments. Finally, Section 6 concludes the paper and presents an outlook on future research opportunities.

2. Literature review

Earlier research on order picking mainly focused on four decision problems: layout design, storage assignment, order batching, and order picker routing. These decision problems are briefly discussed in the following.

Layout design evaluates alternative warehouse layouts and may address two sub-problems, namely the facility layout and the aisle configuration problems (De Koster, Le-Duc, and Roodbergen 2007). Facility layout design is related to the locations of various departments in the warehouse, such as receiving, picking, and storage points, for instance. The reader is referred to Kusiak and Heragu (1987) and Meller and Gau (1996) for an overview of the facility layout problem. The aisle configuration problem concerns the dimensioning of the number, length, and width of aisles. Examples include the works of Caron, Marchet, and Perego (2000) and Roodbergen, Vis, and Taylor Jr (2015).

Storage assignment addresses the assignment of products to storage locations in a warehouse. The literature discusses three general storage assignment strategies: (I) random storage assignment, (II) dedicated storage assignment, where items are assigned to shelf locations based on item characteristics (e.g. size, weight, or usage rate), and (III) class-based storage, where the warehouse is divided into zones according to item characteristics, with storage assignment in each zone being random. Works that discussed alternative storage assignment strategies are those of Heskett (1963), Hausman, Schwarz, and Graves (1976), Brynzér and Johansson (1996), Hwang, Hui Oh, and Nam Cha (2003), and Mantel, Schuur, and Heragu (2007), for example.

Order batching combines a set of orders into groups, so-called batches, where each batch can then be processed during a single picking tour. The literature discussed two main batching principles, namely proximity batching and time-window batching (De Koster, Le-Duc, and Roodbergen 2007). Proximity batching combines orders based on their locations in the warehouse (Henn et al. 2010). Examples include the works of Armstrong, Cook, and Saipe (1979), Elsayed and Stern (1983), and Gibson and Sharp (1992). If time-window batching is applied, in contrast, then orders are combined according to their arrival time (Henn et al. 2010). Examples are the works of Le-Duc and de Koster (2007) and van Nieuwenhuyse and de Koster (2009).

The last decision problem is at the core of the paper at hand. *Order picker routing* aims at finding the shortest tour for collecting all items requested in an order. Some researchers presented algorithms that produce an optimal route, while other researchers proposed heuristic solution procedures. Ratliff and Rosenthal (1983), for example, presented an algorithm that produces an optimal route in a conventional warehouse with a single block using graph theory and dynamic programming. The authors further showed that the time required to find an optimal solution using their algorithm increases linearly in the number of aisles in the warehouse. Both the starting and the ending points of a tour were fixed to the depot of the warehouse. An extension of Ratliff and Rosenthal's work was proposed by de Koster and van der Poort (1998), who presented an algorithm that calculates an optimal route for the case where the starting and ending points of an order picking tour can be at the heads of each aisle. Once an order has been completed, the order picker can drop off retrieved items there without returning to the depot. Roodbergen and de Koster (2001a, 2001b) developed algorithms to compute an optimal route in a conventional warehouse with two blocks and compared the average tour length to the case of a single block. For other algorithms that provide optimal solutions to the order picker routing problem, the reader is referred to Goetschalckx and Ratliff (1988), Scholz et al. (2016), Lu et al. (2016), and Chabot et al. (2017), for example. Heuristic algorithms for order picker routing have also attracted the attention of many researchers. Heuristics are widely used in practice, as they are easy to implement and easy to understand by warehouse workers. In addition, while exact algorithms are often dedicated to specific warehouse layouts, heuristics are more adaptive to alternative layouts (De Koster and Van der Poort 1998). Works that evaluate the performance of routing heuristics are those of Hall (1993), Petersen (1997), Daniels, Rummel, and Schantz (1998), and Vaughan and Petersen (1999), for example.

The paper at hand contributes to the further development of algorithms for optimal order picker routing, whose non-availability for many warehouse layouts has frequently been cited as an obstacle in the improvement of order picking efficiency (Elbert et al. 2017). The paper considers a conventional warehouse with two blocks and adapts the solution procedures presented in Ratliff and Rosenthal (1983) and Roodbergen and de Koster (2001a) to this new scenario.

3. Problem description and graph representation

The order picker routing problem considered here assumes a conventional warehouse with two blocks, referred to as the upper and the lower blocks, where each block has r parallel aisles with equal length and width. There are three cross aisles, namely the front, the middle, and the back cross aisles, perpendicular to those parallel aisles. The starting and ending points of the desired order picking tour can be any locations in the warehouse, and each order picking tour is completed by a single order picker using a picking device. Figure 1 presents an example of such a conventional warehouse with two blocks, where the black boxes represent the locations of items to be picked, while the two boxes marked with the symbols e_1 and e_2 indicate the starting and ending points of a picking tour, respectively, which are different locations in this example. Note that the proposed procedure can also be applied to the case where the starting and ending points of a tour are the same location (e.g. the depot). All aisles are assumed to be narrow with items stored on the racks on both sides, such that the picker can retrieve the requested items from both sides of an aisle without having to cross the aisle. The paper at hand also assumes that order pickers working in the same aisle can pass each other, which means that we do not consider picker congestion within aisles. Moreover, the items can be picked directly from the racks without requiring vertical travels. Such systems have been referred to as low-level picker-to-parts systems.

Input to the proposed algorithm is a list of the locations of m items stored in the warehouse as well as a designated starting point e_1 and a designated ending point e_2 of the desired tour, which can be any position in the warehouse. We note that interchanging e_1 and e_2 gives an equivalent order picking problem as we can reverse the optimal tour. We assume that all m items are separately stored in different shelf locations. There are m_u and m_l items in the upper and lower blocks, respectively, such that $m = m_u + m_l$. The proposed algorithm aims to find the shortest order picking tour that starts from e_1, visits all m storage locations, and ends at e_2.

Our algorithm can easily be explained in terms of tours on a graph. We define a graph representation G of our order picking problem by associating vertices e_1 and e_2 with the starting and ending points, and vertices $v_{u,i}$, $i = 1, 2, \ldots, m_u$ and $v_{l,i}$, $i = 1, 2, \ldots, m_l$ with the locations of the items stored in the upper and lower blocks of the warehouse, respectively. For $j = 1, 2, \ldots, r$, the vertices a_j, b_j, and c_j represent the rear end, the middle end, and the front end of aisle j, respectively. Assigned to each pair of adjacent vertices in the warehouse is an infinite number of parallel undirected edges. However, from Theorem 3 in Section 4.1, no more than two edges between each pair of adjacent vertices are contained in the solution. Consequently, we can assume that each pair of adjacent vertices in the warehouse is connected by two parallel edges.

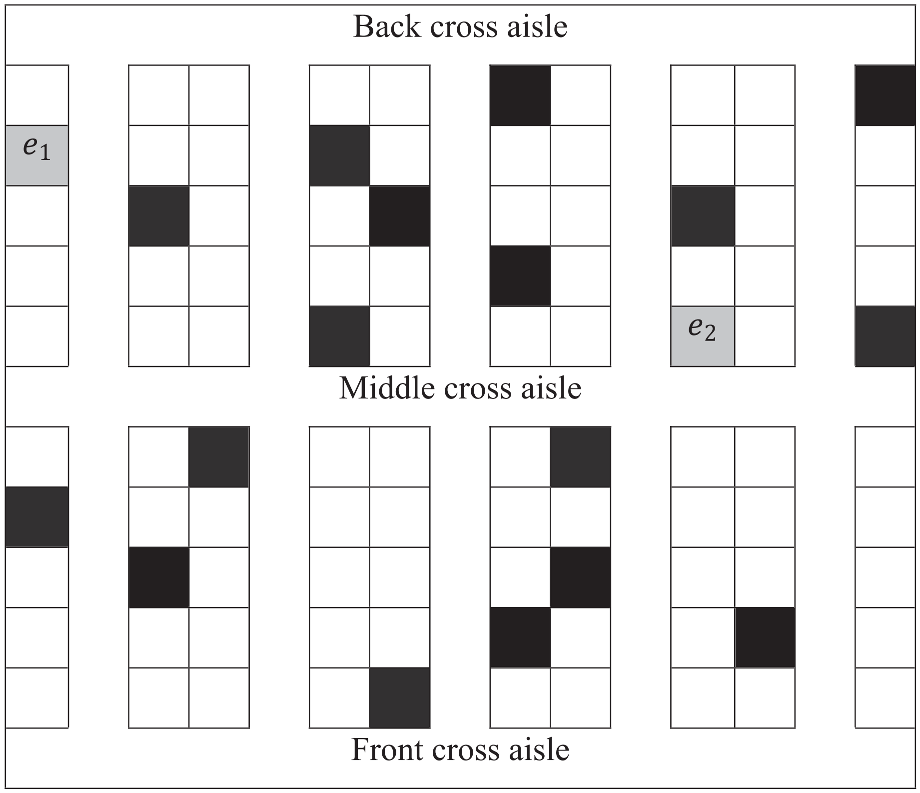

Figure 1. Conventional warehouse with two blocks.

Figure 2 shows a graph G associated with the order picker routing problem in Figure 1 where $r = 5$, $m_u = 9$, and $m_l = 8$. Associated with every edge is a weight that corresponds to the distance between the endpoints of that edge.

An order picking tour in the warehouse then corresponds to a tour, i.e. a directed path, on the graph G, and vice versa. Therefore, the problem of finding the shortest order picking tour is identical to the problem of finding a tour on the graph G containing all required vertices, which will be solved in the next section.

4. Order picker routing with arbitrary starting and ending points of a tour

This section first proposes an optimal order picker routing policy with arbitrary starting and ending points of a tour. Afterwards, we develop a routing heuristic for the same scenario.

4.1. Optimal order picker routing policy

This section presents a solution procedure for the order picker routing problem with arbitrary starting and ending points in a conventional warehouse with two blocks, using the graph representation introduced in the previous section. Löffler et al. (2018) proposed an optimal order picker routing algorithm for arbitrary starting and ending points for the case of a conventional warehouse with a single block. We extend their work to a conventional warehouse with two blocks using the solution procedures proposed in the works of Ratliff and Rosenthal (1983) and Roodbergen and de Koster (2001a).

4.1.1. Outline of the solution procedure

The objective is to find the shortest order picking tour on G. First, we present a procedure for constructing the minimum-length tour subgraph from a graph representation of G. Afterwards, we construct the optimal order picking tour from a minimum-length tour subgraph.

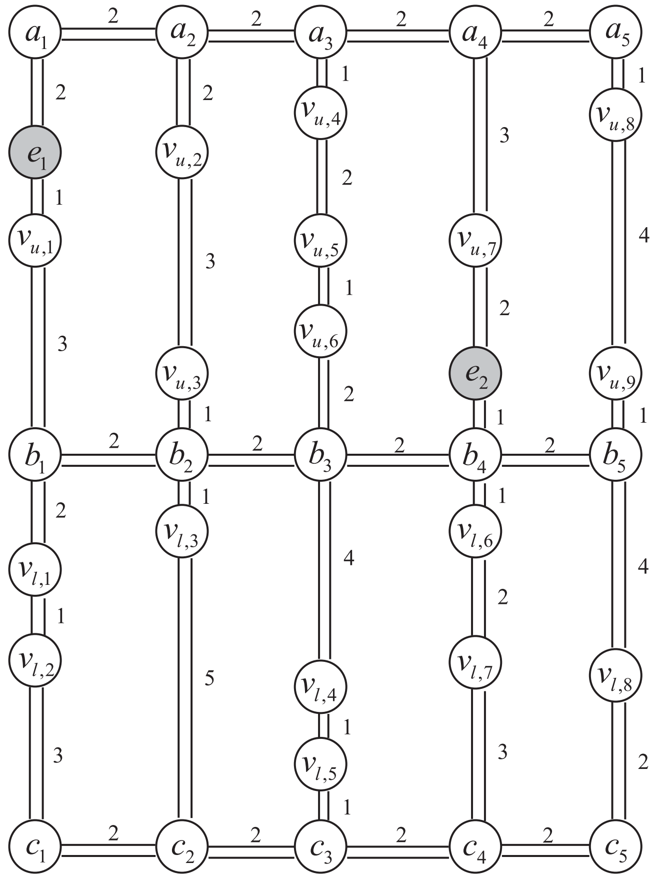

Figure 2. Graph representation G, where $r = 5$, $m_u = 9$, and $m_l = 8$.

4.1.2. *Constructing the minimum-length tour subgraph*

To find the minimum-length tour subgraph of G, we consider a certain sequence of increasing subgraphs of G from the left-most to the right-most picking aisles that contain items to be picked. In each subgraph along the sequence, partial tour subgraphs (PTSs) and their equivalence classes are considered. In each equivalence class, a PTS with minimum length is selected as a candidate for a PTS of the minimum-length tour subgraph. The idea is to build tour subgraphs by iteratively adding edges from left to right and from the lower block to the upper block. Dynamic programming then ensures that we obtain the tour subgraph of G with minimum length. The different steps of the solution procedure are described in the following.

4.1.2.1. Constructing a partial tour subgraph (PTS). We first extend the definition of a tour subgraph in Ratliff and Rosenthal (1983) to the case where the starting and ending points are arbitrarily given.

DEFINITION 1 *Let e_1 and e_2 be the given starting and ending points of a tour. A subgraph T of G is a tour subgraph if there is an order picking tour that starts at e_1, passes through the vertices $v_{u,i}$ and $v_{l,i}$, and ends at e_2, where each edge in T is traversed exactly once.*

We recall that, by definition, an order picking tour must visit every item location, hence the degree of the vertices $v_{u,i}$ and $v_{l,i}$ in T are positive. The starting and ending points e_1 and e_2 can also be interchanged. The interchange will result in the reversed direction of the tour without changing the tour subgraph.

The necessary and sufficient condition for a subgraph T of G to be a tour subgraph is given in Theorem 1.

THEOREM 1 *A connected subgraph of an undirected graph is a tour subgraph with starting and ending points e_1 and e_2 if and only if the degrees of e_1 and e_2 in the subgraph are odd, the degrees of the vertices $v_{u,i}$ and $v_{l,i}$ are positive even, and the degrees of the other vertices are even.*

Proof Assume that a connected subgraph T of G is a tour subgraph with starting and ending points e_1 and e_2. By Definition 1, there is a tour that starts at e_1, passes through the vertices $v_{u,i}$ and $v_{l,i}$, and ends at e_2, where each edge in T is traversed exactly once. For each vertex $v_{u,i}$, the number of edges adjacent to $v_{u,i}$ that the tour uses to visit $v_{u,i}$ is the same as the number of edges used to leave the vertex. Hence, the degree of $v_{u,i}$ must be even. The same argument implies that the degree of $v_{l,i}$ is even as well. On the other hand, the tour T uses one more edge to leave the vertex e_1, so the degree of the vertex is odd. Similarly, the degree of e_2 is also odd.

Conversely, assume that the degrees of e_1 and e_2 in the subgraph are odd, while the degrees of the other vertices are even. From the Eulerian path theorem, there exists a tour that starts at e_1, ends at e_2, and traverses every edge exactly once. Since the degrees of $v_{u,i}$ and $v_{l,i}$ are positive, the tour is an order picking tour. Hence, the subgraph is a tour subgraph. ■

By Definition 1 and Theorem 1, we propose the following characterisation of an order picking tour subgraph.

THEOREM 2 *A subgraph $T \subset G$ is an order picking tour subgraph if and only if all the following conditions hold:*

1. *The vertices $v_{u,i}$ for $i = 1, 2, \ldots, m_u$, $v_{l,i}$ for $i = 1, 2, \ldots, m_l$, e_1, and e_2 all have positive degrees in T.*
2. *The vertices e_1 and e_2 both have odd degrees.*
3. *Excluding vertices with zero degree, T is connected.*
4. *Every vertex in T, except e_1 and e_2, has even degree or zero degree.*

Proof Assume that T is an order picking tour subgraph of G. By Definition 1, there exists a tour that starts at e_1, passes through the vertices $v_{u,i}$ and $v_{l,i}$, and ends at e_2. Hence, the first and the third conditions hold. By Theorem 1, the degrees of e_1 and e_2 in T are odd, while the degrees of all other vertices are even. Hence, the second and the fourth conditions hold.

Conversely, the third condition implies that T is connected. The other three conditions then conclude that T is a tour subgraph with starting point e_1 and ending point e_2 that contains all the vertices $v_{u,i}$ and $v_{l,i}$. Hence, T is an order picking tour subgraph. ■

The following theorem is useful in constructing a minimum-length order picking tour subgraph.

THEOREM 3 *A minimum-length order picking tour subgraph contains no more than two edges between any pair of vertices.*

Proof By contradiction, assume that there are more than two edges between a pair of vertices in a minimum-length tour subgraph T. We can see that deleting two edges between the two vertices from T will still result a tour subgraph, but with a shorter length. Hence, the former tour subgraph T cannot be of minimum length. This is a contradiction. ■

DEFINITION 2 (cf. Ratliff and Rosenthal 1983) *Let L be the subgraph of G, a subgraph T of L is an L-PTS if there exists a subgraph C of $G - L$ such that $T \cup C$ is a tour subgraph of G. The subgraph C is called a completion of T.*

In our algorithm, we only consider three types of the subgraph L, which are described in the following definition.

DEFINITION 3 (cf. Roodbergen and de Koster, 2001a) *Let L_j^- be a subgraph of G containing the vertices a_j, b_j, c_j, and all vertices and edges to the left of them. Let l_j be a subgraph of G containing the vertices b_j, c_j, and all vertices and edges between them. We define $L_j^{+l} = L_j^- \cup l_j$. Similarly, let u_j be a subgraph of G consisting of the vertices a_j and b_j together with all vertices and edges between them and define $L_j^{+u} = L_j^{+l} \cup u_j$. From this point forward, L_j will be used when a result holds if we let $L_j = L_j^-$, $L_j = L_j^{+l}$, or $L_j = L_j^{+u}$.*

To find the minimum-length tour subgraph of G, we consider a sequence of increasing subgraphs of G from aisle $j = 1$ to aisle $j = r$, where L_j partial tour subgraphs (L_j-PTSs) are considered for each aisle j. The following theorem extends the theorem in Roodbergen and de Koster (2001a) and gives the necessary and sufficient conditions for a subgraph of G to be an L_j-PTS.

THEOREM 4 *A subgraph $T_j \subset L_j$ is an L_j-PTS if and only if*

1. *The degrees of all $v_{u,i}, v_{l,i} \in L_j$ are positive in T_j.*
2. *If L_j contains the vertex e_i ($i = 1$ or 2), the degree of e_i is odd.*
3. *Every vertex in T_j, except possibly for a_j, b_j, c_j, and e_i, has even or zero degree.*
4. *Excluding vertices with zero degree, T_j has either*

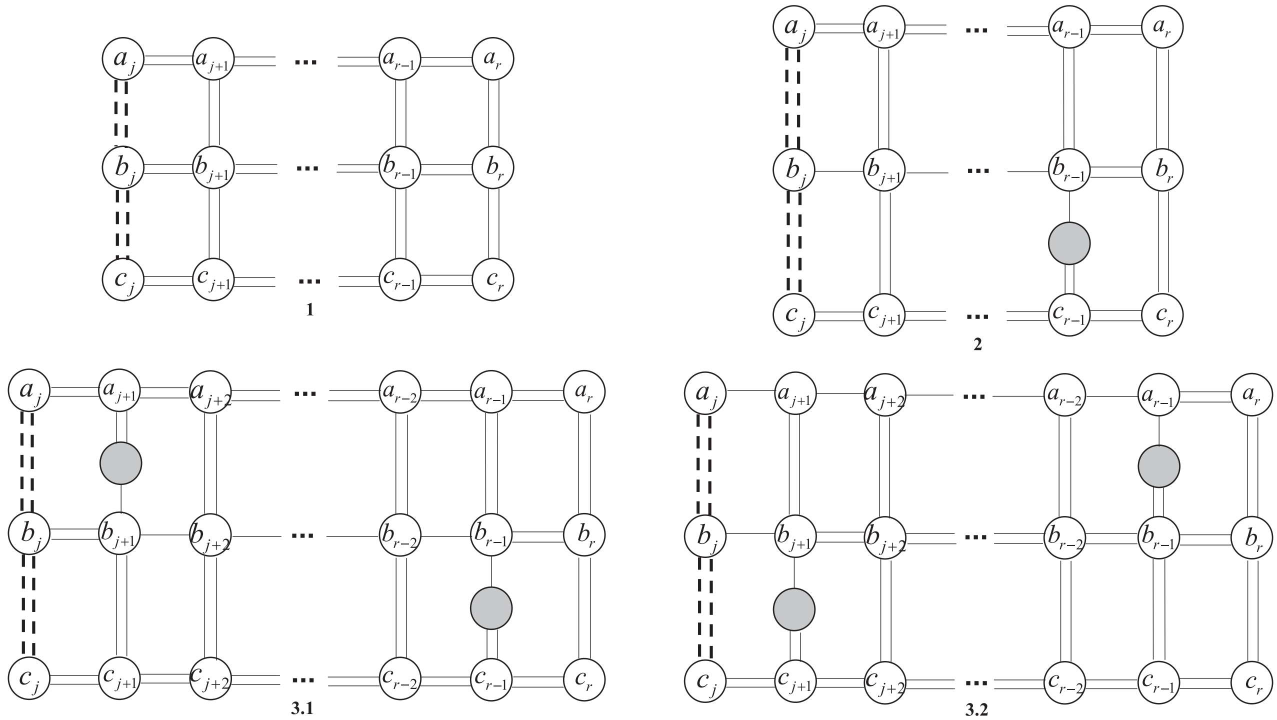

Figure 3. Completions C_j of T_j such that $T_j \cup C_j$ is a tour subgraph.

- *no connected component,*
- *a single connected component containing at least one of a_j, b_j, and c_j,*
- *two connected components, with each component containing at least one of a_j, b_j, and c_j,*
- *three connected components with a_j, b_j, and c_j each in a different component.*

Proof To prove necessity, we extend the proof of Ratliff and Rosenthal (1983). We assume that T_j is an L_j-PTS. By Definition 2, there exists a subgraph C_j of $G - L_j$ such that $T = T_j \cup C_j$ is a tour subgraph. Since the first three conditions hold in T, they consequently hold in T_j as well. We are left to show that the fourth condition holds. Note that, except possibly for a_j, b_j, and c_j, there are no vertices in C_j incident to the vertices in T_j. Since a tour subgraph T is connected, each connected component of T_j must contain at least one of a_j, b_j, and c_j. Hence, the fourth condition holds.

To prove sufficiency, we extend the proof of Ratliff and Rosenthal (1983) and Löffler et al. (2018). We assume that all conditions hold. In order to show that a subgraph T_j is an L_j-PTS, we have to find a subgraph C_j of $G - L_j$ such that $T_j \cup C_j$ is a tour subgraph. From Theorem 2, C_j must have the following properties:

1. It contains all vertices $v_{u,i}, v_{l,i} \in G - L_j$, and their degrees in C_j are positive even.
2. It contains all vertices $e_i \in G - L_j$, and their degrees in C_j are odd.
3. Excluding vertices with zero degree, $T_j \cup C_j$ is connected.

We note that T_j can contain both e_1 and e_2, either e_1 or e_2, or neither e_1 nor e_2. In each case, a completion C_j can be easily constructed as shown in Figure 3 by taking into account the presence of the vertices e_i in T_j. Configuration 1 represents a completion C_j when T_j contains both e_1 and e_2, while configuration 2 is a completion C_j for the case when T_j contains either e_1 or e_2. Both configurations 3.1 and 3.2 represent completions C_j when T_j contains neither e_1 nor e_2; 3.1 is applied when the degree parities of a_j, b_j, and c_j are all even in L_j, whereas 3.2 is used when the degree parities of a_j and b_j are all odd and the degree parity of c_j is even (The other cases can be done similarly). Note that parallel dashed edges in the upper and lower blocks of aisle j are included in C_j for the case $L_j = L_j^-$. Conversely, they are all excluded from C_j when $L_j = L_j^{+u}$. For $L_j = L_j^{+l}$, parallel dashed edges in the upper block are included in C_j, while parallel dashed edges in the lower block are excluded. ■

4.1.2.2. Equivalence class of a partial tour subgraph (PTS). As mentioned in Section 4.1.2, to construct the minimum-length tour subgraph of G, partial tour subgraphs (PTSs) and their equivalence classes have to be considered simultaneously. Suppose that there are two L_j-PTSs, T_j^1 and T_j^2. The equivalence of these two PTSs is specified in Definition 4.

DEFINITION 4 (cf. Ratliff and Rosenthal 1983) *Two L_j-PTSs T_j^1 and T_j^2 are equivalent if for any $C_j \subset G - L_j$ such that $T_j^1 \cup C_j$ is a tour subgraph, $T_j^2 \cup C_j$ is also a tour subgraph, and vice versa.*

The conditions for two L_j-PTSs to be equivalent are described in Theorem 5.

THEOREM 5 (cf. Roodbergen and de Koster, 2001a) *Two L_j-PTSs are equivalent if and only if*

1. *a_j, b_j, and c_j each have the same degree parity (i.e. even (including 0) or odd) in both PTSs.*
2. *Excluding vertices with zero degree, both PTSs have either*
   - *no connected component,*
   - *a single connected component containing at least one of a_j, b_j, and c_j,*
   - *two connected components, with at least one of a_j, b_j, and c_j in each component,*
   - *three connected components, with a_j, b_j, and c_j each in a different component.*
3. *The distribution of a_j, b_j, and c_j over the various components is the same for both PTSs.*

Proof We extend the proof of Ratliff and Rosenthal (1983). Assume that T_j^1 and T_j^2 are equivalent L_j-PTSs. By Definition 2, consider any subgraph C_j of $G - L_j$ such that $T_j^1 \cup C_j$ is a tour subgraph, then $T_j^2 \cup C_j$ is also a tour subgraph. Since $T_j^1 \cup C_j$ is a tour subgraph, the degrees of a_j, b_j, and c_j in $T_j^1 \cup C_j$ must be even. Therefore, the degree parity of these vertices in T_j^1 and C_j are the same. The same argument holds for T_j^2. Hence a_j, b_j, and c_j have the same degree parity in T_j^1 and T_j^2.

If T_j^1 and T_j^2 have different numbers of connected components or different distributions of a_j, b_j, and c_j over the components, we can easily find a subgraph C_j that is a completion of either T_j^1 or T_j^2, but not of the respective other. Hence, all three conditions hold.

Conversely, let T_j^1 and T_j^2 satisfy the three conditions. Let C_j be a subgraph of $G - L_j$ such that $T_j^1 \cup C_j$ is a tour subgraph. Since T_j^1, T_j^2, and C_j are PTSs, the degrees of e_1 and e_2 are odd, while the degrees of the other vertices except possibly for a_j, b_j, and c_j are even. Since a_j, b_j, and c_j have the same degree parity in T_j^1 and C_j, they also have the same degree parity in T_j^2. Thus, the degrees of a_j, b_j, and c_j in $T_j^2 \cup C_j$ are even.

We are left to show that $T_j^2 \cup C_j$ is connected. Observe that shrinking a connected component of a graph does not change its degree parity and the connectivity of the graph. If we shrink the connected components of T_j^1 and T_j^2 to single vertices, the resulting graphs still have the same degree parity and connectivity. Therefore, since $T_j^1 \cup C_j$ is connected, $T_j^2 \cup C_j$ is also connected. By Theorem 2, $T_j^2 \cup C_j$ is a tour subgraph. Hence, T_j^1 and T_j^2 are equivalent. ■

From Theorem 5, an equivalence class of L_j-PTSs can be represented by the degree parity of a_j ($\deg(a_j)$), degree parity of b_j ($\deg(b_j)$), degree parity of c_j ($\deg(c_j)$), and their connectivity in the PTS. The degree parities of a_j, b_j, and c_j can be even (E) (including 0) or uneven (U). In order to decrease the number of equivalence classes in the solution procedure, we do not separate the degree parity between zero (0) and even (E), but instead use the pattern of connectivity to decide on the degree parities of a_j, b_j, and c_j in case their degree parities are even (E).

The connectivity encodes the number of components as well as the distribution of a_j, b_j, and c_j over those components. From Theorem 5, it follows that there are fifteen possible types of connectivity, which are illustrated in Table 1. The connectivity of a PTS that does not have any connected component is indicated by the symbol (-). A single connected component containing at least one of a_j, b_j, and c_j is represented by a, b, c, ab, ac, bc, and abc. Two connected components with at least one of a_j, b_j, and c_j in each component are represented by $a - b$, $a - c$, $b - c$, $a - bc$, $b - ac$, and $c - ab$. Three connected components with a_j, b_j, and c_j each in a different component is represented by $a - b - c$.

For instance, the equivalence class $EUU(a - bc)$ indicates that the degree parities of a_j, b_j, and c_j are even (E), uneven (U), and uneven (U), respectively, and there are two connected components with a_j in one component, while b_j and c_j are in the other. Since $\deg(a_j)$, $\deg(b_j)$, and $\deg(c_j)$ can be either even (E) or uneven (U) and the number of possible types of connectivity is fifteen, there are $2 \times 2 \times 2 \times 15 = 120$ equivalence classes that are relevant for the solution procedure. However, these 120 equivalence classes contain some classes that lead to infeasible solutions. As an example, $EEU(a - b)$ has E, E, and U in a_j, b_j, and c_j, respectively, and consists of two connected components with a_j in one component and b_j in the other. It is impossible that the degree parity of c_j is odd since there is no connected component containing c_j. After deleting all equivalence classes that lead to infeasible solutions, we obtain 70 possible equivalence classes. From Theorem 4, all possible equivalence classes of an L_j-PTS can be categorised as described in Theorems 6, 7, and 8. From Theorem

Table 1. Degree parities of a_j, b_j, c_j, and connectivity of L_j^{+l}-PTS equivalence class.

		Vertical components that can be used for traversing aisle j of the lower block (between b_j and c_j)												
		when aisle j of the lower block contains neither e_1 nor e_2 or both of them							when aisle j of the lower block contains either e_1 or e_2					
L_j^--PTS Equivalence class		1	2,14	3,15	4,17	5,16	6	13	7	8	9	10	11	12
deg(a_j)	***E***	*E*	*E*	*E*	*E*	*E*	*E*	*E*	*E*	*E*	*E*	*E*	*E*	*E*
	U	*U*	*U*	*U*	*U*	*U*	*U*	*U*	*U*	*U*	*U*	*U*	*U*	*U*
deg(b_j)	***E***	*U*	*E*	*E*	*E*	*E*	*E*	*U*	*U*	*E*	*U*	*E*	*U*	*E*
	U	*E*	*U*	*U*	*U*	*U*	*U*	*E*	*E*	*U*	*E*	*U*	*E*	*U*
deg(c_j)	***E***	*U*	*E*	*E*	*E*	*E*	*E*	*U*	*E*	*U*	*E*	*U*	*E*	*U*
	U	*E*	*U*	*U*	*U*	*U*	*U*	*E*	*U*	*E*	*U*	*E*	*U*	*E*
Connectivity	**-**	*bc*	*b*	*c*	*b-c*	*bc*	-	*b-c*	*b*	*c*	*bc*	*bc*	*b-c*	*b-c*
	a	*a-bc*	*a-b*	*a-c*	*a-b-c*	*a-bc*	*a*	*a-b-c*	*a-b*	*a-c*	*a-bc*	*a-bc*	*a-b-c*	*a-b-c*
	b	*bc*	*b*	*b-c*	*b-c*	*bc*	*b*	*b-c*	*b*	*b-c*	*bc*	*bc*	*b-c*	*b-c*
	c	*bc*	*b-c*	*c*	*b-c*	*bc*	*c*	*b-c*	*b-c*	*c*	*bc*	*bc*	*b-c*	*b-c*
	ab	*abc*	*ab*	*c-ab*	*c-ab*	*abc*	*ab*	*c-ab*	*ab*	*c-ab*	*abc*	*abc*	*c-ab*	*ab-c*
	ac	*abc*	*b-ac*	*ac*	*b-ac*	*abc*	*ac*	*b-ac*	*b-ac*	*ac*	*abc*	*abc*	*b-ac*	*b-ac*
	bc	*bc*	*bc*	*bc*	*bc*	*bc*	*bc*	*bc*	*bc*	*bc*	*bc*	*bc*	*bc*	*bc*
	abc	*abc*	*abc*	*abc*	*abc*	*abc*	*abc*	*abc*	*abc*	*abc*	*abc*	*abc*	*abc*	*abc*
	a-b	*a-bc*	*a-b*	*a-b-c*	*a-b-c*	*a-bc*	*a-b*	*a-b-c*	*a-b*	*a-b-c*	*a-bc*	*a-bc*	*a-b-c*	*a-b-c*
	a-c	*a-bc*	*a-b-c*	*a-c*	*a-b-c*	*a-bc*	*a-c*	*a-b-c*	*a-b-c*	*a-c*	*a-bc*	*a-bc*	*a-b-c*	*a-b-c*
	b-c	*bc*	*b-c*	*b-c*	*b-c*	*bc*	*b-c*	*b-c*	*b-c*	*b-c*	*bc*	*bc*	*b-c*	*b-c*
	a-bc	*a-bc*	*a-bc*	*a-bc*	*a-bc*	*a-bc*	*a-bc*	*a-bc*	*a-bc*	*a-bc*	*a-bc*	*a-bc*	*a-bc*	*a-bc*
	b-ac	*abc*	*b-ac*	*b-ac*	*b-ac*	*abc*	*b-ac*	*b-ac*	*b-ac*	*b-ac*	*abc*	*abc*	*b-ac*	*b-ac*
	c-ab	*abc*	*c-ab*	*c-ab*	*c-ab*	*abc*	*c-ab*	*c-ab*	*c-ab*	*c-ab*	*abc*	*abc*	*c-ab*	*c-ab*
	a-b-c	*a-bc*	*a-b-c*	*a-b-c*	*a-b-c*	*a-bc*	*a-b-c*	*a-b-c*	*a-b-c*	*a-b-c*	*a-bc*	*a-bc*	*a-b-c*	*a-b-c*

2, since a tour subgraph must be connected and possess even degree, the shortest length from $EEE(a)$, $EEE(b)$, $EEE(c)$, $EEE(ab)$, $EEE(ac)$, $EEE(bc)$, and $EEE(abc)$ of L_r^{+u} in the last aisle will be the minimum-length tour subgraph.

THEOREM 6 *In case L_j contains neither e_1 nor e_2, the sum of degree parities of a_j, b_j, and c_j of an L_j-PTS is even, and the possible equivalence classes are: $EEE(-)$, $EEE(a)$, $EEE(b)$, $EEE(c)$, $EEE(ab)$, $EEE(ac)$, $EEE(bc)$, $EEE(abc)$, $EEE(a-b)$, $EEE(a-c)$, $EEE(b-c)$, $EEE(a-bc)$, $EEE(b-ac)$, $EEE(c-ab)$, $EEE(a-b-c)$, $EUU(bc)$, $EUU(abc)$, $EUU(a-bc)$, $UEU(ac)$, $UEU(abc)$, $UEU(b-ac)$, $UUE(ab)$, $UUE(abc)$, $UUE(c-ab)$.*

THEOREM 7 *In case L_j contains exactly one of e_1 and e_2, the sum of degree parities of a_j, b_j, and c_j of an L_j-PTS is odd, and the possible equivalence classes are: $EEU(c)$, $EEU(ac)$, $EEU(bc)$, $EEU(abc)$, $EEU(a-c)$, $EEU(b-c)$, $EEU(a-bc)$, $EEU(b-ac)$, $EEU(c-ab)$, $EEU(a-b-c)$, $EUE(b)$, $EUE(ab)$, $EUE(bc)$, $EUE(abc)$, $EUE(a-b)$, $EUE(b-c)$, $EUE(a-bc)$, $EUE(b-ac)$, $EUE(c-ab)$, $EUE(a-b-c)$, $UEE(a)$, $UEE(ab)$, $UEE(ac)$, $UEE(abc)$, $UEE(a-b)$, $UEE(a-c)$, $UEE(a-bc)$, $UEE(b-ac)$, $UEE(c-ab)$, $UEE(a-b-c)$, $UUU(abc)$, $UUU(a-bc)$, $UUU(b-ac)$, $UUU(c-ab)$.*

THEOREM 8 *In case L_j contains both e_1 and e_2, the sum of degree parities of a_j, b_j, and c_j of an L_j-PTS is even, and the possible equivalence classes are: $EEE(-)$, $EEE(a)$, $EEE(b)$, $EEE(c)$, $EEE(ab)$, $EEE(ac)$, $EEE(bc)$, $EEE(abc)$, $EEE(a-b)$, $EEE(a-c)$, $EEE(b-c)$, $EEE(a-bc)$, $EEE(b-ac)$, $EEE(c-ab)$, $EEE(a-b-c)$, $EUU(bc)$, $EUU(abc)$, $EUU(b-c)$, $EUU(a-bc)$, $EUU(b-ac)$, $EUU(c-ab)$, $EUU(a-b-c)$, $UEU(ac)$, $UEU(abc)$, $UEU(a-c)$, $UEU(a-bc)$, $UEU(b-ac)$, $UEU(c-ab)$, $UEU(a-b-c)$, $UUE(ab)$, $UUE(abc)$, $UUE(a-b)$, $UUE(a-bc)$, $UUE(b-ac)$, $UUE(c-ab)$, $UUE(a-b-c)$.*

4.1.2.3. Dynamic programming procedure. In the proposed dynamic programming procedure, we define the states as the equivalence classes of PTSs as well as the three different transitions between states that consist of adding vertices and edges to L_j-PTSs, L_j^- to L_j^{+l}, L_j^{+l} to L_j^{+u}, and L_j^{+u} to L_{j+1}^- (Roodbergen and de Koster, 2001a). Since a minimum-length tour subgraph contains no more than two edges between any two adjacent vertices (see Theorem 3), we consider only the vertical and horizontal components with a single edge and/or double edges between any pair of vertices in each transition. The three possible types of transitions are described in the following.

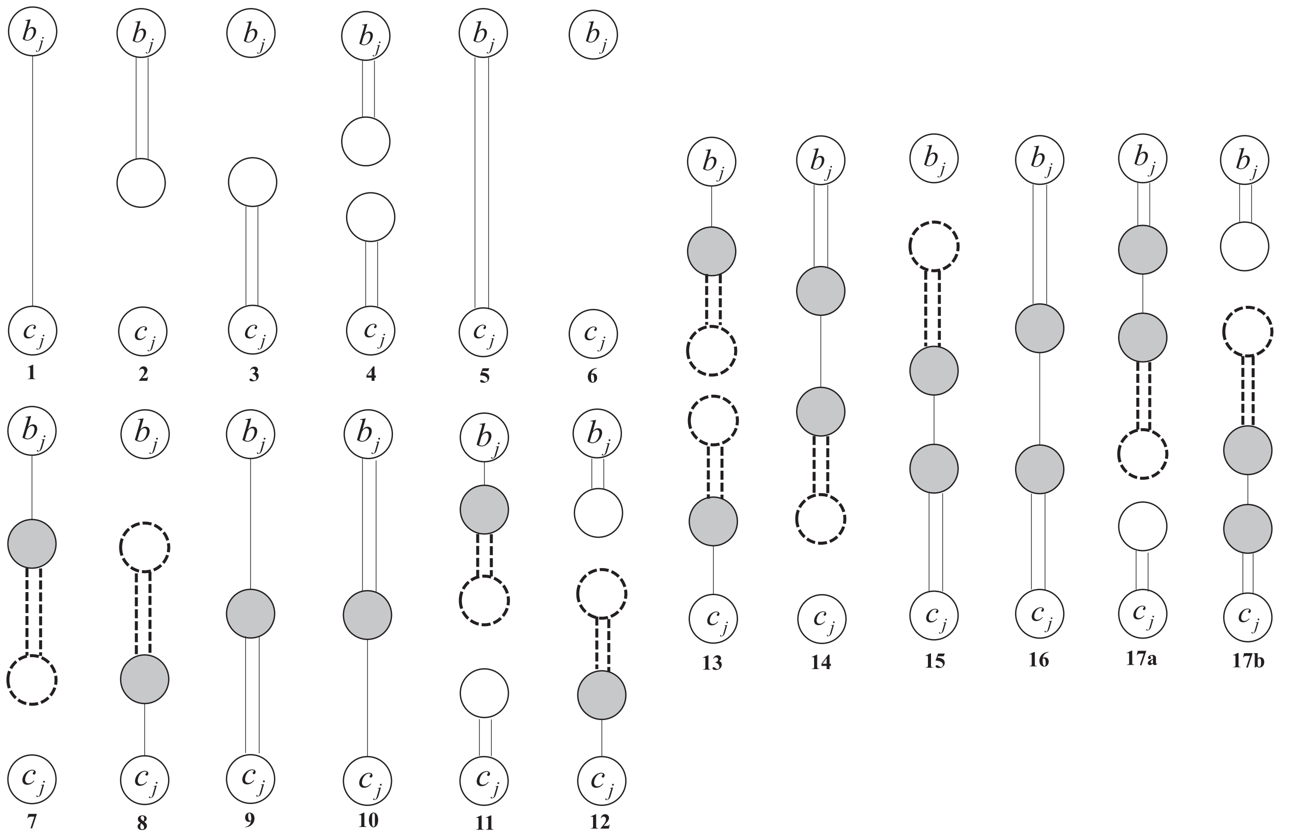

Figure 4. Vertical components for aisle j in the lower block. Dashed parts are optional.

1. *Transition from* L_j^- *to* L_j^{+l}

This transition transforms an L_j^--PTS to an L_j^{+l}-PTS by adding a vertical component for traversing aisle j of the lower block to the L_j^--PTS. The recursive formula (1) provides a connection between the objective value of the previous and the current state. The value $\omega(z, e)$ denotes the objective value of state (z, e), i.e. the shortest length z PTS of equivalence class e, where e belongs to one of the equivalence classes (Eq) described in Theorems 6–8. The value $\omega(v)$ represents the added distance from the vertical transition corresponding to the vertical components in Figure 4, where v belongs to one of the vertical components (Ver) in Figure 4.

$$\omega(L_j^{+l}, e) = \min_{v \in \{Ver\}, e \in \{Eq\}} \{\omega(L_j^-, e) + \omega(v)\} \tag{1}$$

Figure 4 shows all seventeen possible vertical components between the vertices b_j and c_j that are candidates for the optimal solution. They can be categorised into three main groups. Configurations 1–6 represent the vertical components for the case where aisle j of the lower block contains neither the starting point e_1 nor the ending point e_2. If aisle j contains either e_1 or e_2, configurations 7–12 are applied. Configurations 13–17 are finally used when aisle j contains both e_1 and e_2. Configurations 17a and 17b are shown for clarity, and both will be referred to as configuration 17 in the following. In configuration 4, the largest gap between two adjacent items is not traversed. For configurations 11, 12, and 17, the largest gap between any two adjacent items on either side of e_1 or e_2 is not traversed. In configuration 13, the largest gap between e_1 and e_2 is not traversed. Configuration 6 can only be selected if aisle j in the lower block is empty. The connectivity of the L_j^{+l}-PTS equivalence classes as well as $\deg(a_j)$, $\deg(b_j)$, $\deg(c_j)$ that result from adding vertical components illustrated in Figure 4 to the L_j^--PTS equivalence classes are given in Table 1. For instance, if $\deg(a_j)$, $\deg(b_j)$, $\deg(c_j)$, and connectivity of an L_j^--PTS are U, E, E, and a, respectively, the L_j^--PTS belongs to the $UEE(a)$ equivalence class. If it is expanded with vertical component configuration 3, an L_j^{+l}-PTS that results from this transition belongs to the class $UEE(a-c)$. An example for the case where $j = 3$ is illustrated in Figure 5.

2. *Transition from* L_j^{+l} *to* L_j^{+u}

The transition is similar to the previous transition, but a vertical component for traversing between a_j and b_j is added to an L_j^{+l}-PTS instead. Thus, the same seventeen vertical components described earlier can be used. The connectivity of the

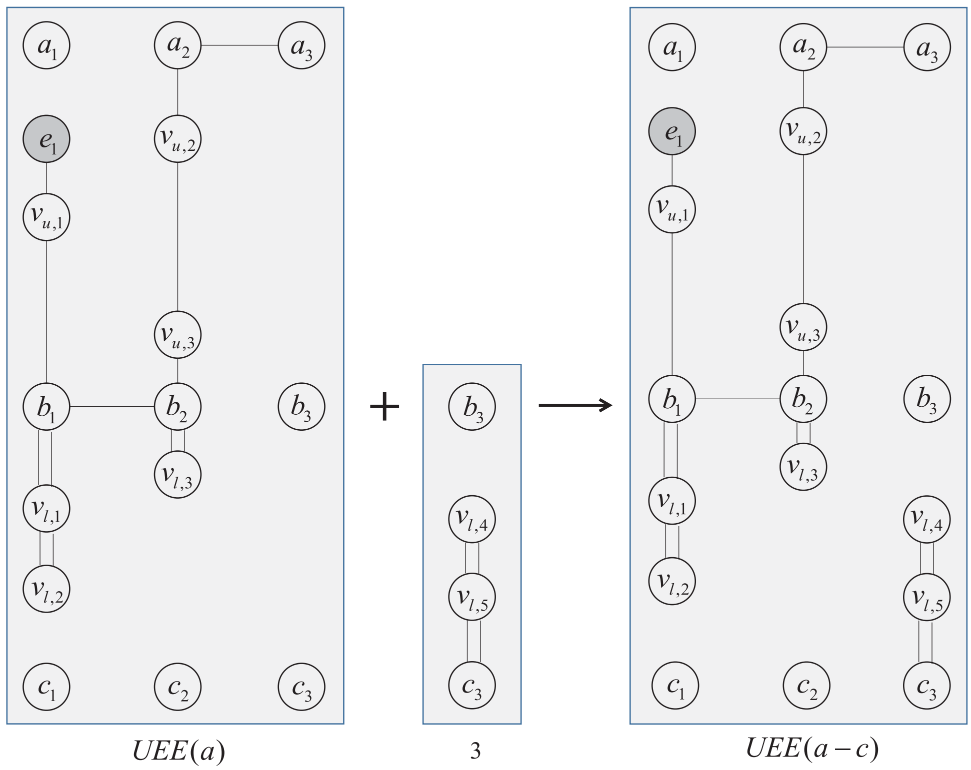

Figure 5. Adding vertical component configuration 3 to an L_3^--PTS of type $UEE(a)$ results in an L_3^{+l}-PTS of type $UEE(a-c)$.

Table 2. Degree parities of a_j, b_j, c_j, and connectivity of L_j^{+u}-PTS equivalence class.

		Vertical components that can be used for traversing aisle j of the upper block (between a_j and b_j)												
		when aisle j of the upper block contains neither e_1 nor e_2 or both of them							when aisle j of the upper block contains either e_1 or e_2					
L_j^{+l}-PTS Equivalence class		1	2,14	3,15	4,17	5,16	6	13	7	8	9	10	11	12
deg(a_j)	**E**	U	E	E	E	E	E	U	U	E	U	E	U	E
	U	E	U	U	U	U	U	E	E	U	E	U	E	U
deg(b_j)	**E**	U	E	E	E	E	E	U	E	U	E	U	E	U
	U	E	U	U	U	U	U	E	U	E	U	E	U	E
deg(c_j)	**E**	E	E	E	E	E	E	E	E	E	E	E	E	E
	U	U	U	U	U	U	U	U	U	U	U	U	U	U
Connectivity	**-**	ab	a	b	a-b	ab	-	a-b	a	b	ab	ab	a-b	a-b
	a	ab	a	a-b	a-b	ab	a	a-b	a	a-b	ab	ab	a-b	a-b
	b	ab	a-b	b	a-b	ab	b	a-b	a-b	b	ab	ab	a-b	a-b
	c	c-ab	a-c	b-c	a-b-c	c-ab	c	a-b-c	a-c	b-c	c-ab	c-ab	a-b-c	a-b-c
	ab	ab	ab	ab	ab	ab	ab	ab	ab	ab	ab	ab	ab	ab
	ac	abc	ac	b-ac	b-ac	abc	ac	b-ac	ac	b-ac	abc	abc	b-ac	b-ac
	bc	abc	a-bc	bc	a-bc	abc	bc	a-bc	a-bc	bc	abc	abc	a-bc	a-bc
	abc	abc	abc	abc	abc	abc	abc	abc	abc	abc	abc	abc	abc	abc
	a-b	ab	a-b	a-b	a-b	ab	a-b	a-b	a-b	a-b	ab	ab	a-b	a-b
	a-c	c-ab	a-c	a-b-c	a-b-c	c-ab	a-c	a-b-c	a-c	a-b-c	c-ab	c-ab	a-b-c	a-b-c
	b-c	c-ab	a-b-c	b-c	a-b-c	c-ab	b-c	a-b-c	a-b-c	b-c	c-ab	c-ab	a-b-c	a-b-c
	a-bc	abc	a-bc	a-bc	a-bc	abc	a-bc	a-bc	a-bc	a-bc	abc	abc	a-bc	a-bc
	b-ac	abc	b-ac	b-ac	b-ac	abc	b-ac	b-ac	b-ac	b-ac	abc	abc	b-ac	b-ac
	c-ab	c-ab	c-ab	c-ab	c-ab	c-ab	c-ab	c-ab	c-ab	c-ab	c-ab	c-ab	c-ab	c-ab
	a-b-c	c-ab	a-b-c	a-b-c	a-b-c	c-ab	a-b-c	a-b-c	a-b-c	a-b-c	c-ab	c-ab	a-b-c	a-b-c

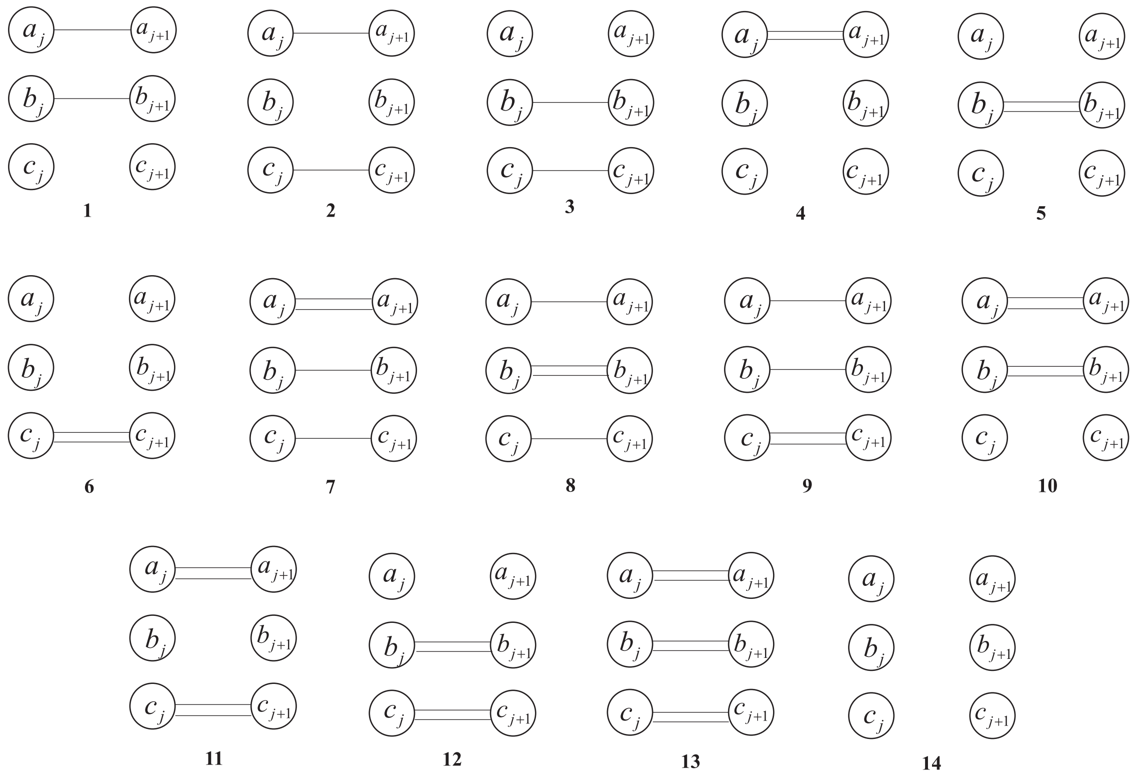

Figure 6. Horizontal components for travelling from aisle j to aisle $j+1$ when an L_j^{+u}-PTS contains neither e_1 nor e_2 or both of them.

L_j^{+u}-PTS equivalence classes as well as $\deg(a_j)$, $\deg(b_j)$, $\deg(c_j)$ are given in Table 2. The following Equation (2) is the recursive formula for this transition.

$$\omega(L_j^{+u}, e) = \min_{v \in \{Ver\}, e \in \{Eq\}} \{\omega(L_j^{+l}, e) + \omega(v)\} \tag{2}$$

3. *Transition from L_j^{+u} to L_{j+1}^{-}*
This transition transforms an L_j^{+u}-PTS to an L_{j+1}^{-}-PTS by connecting aisle j to aisle $j+1$ by a horizontal component between a_j and a_{j+1}, b_j and b_{j+1}, and c_j and c_{j+1}. If an L_j^{+u}-PTS contains neither e_1 nor e_2 or both of them, the horizontal components 1–14 in Figure 6 are applied. If the L_j^{+u}-PTS contains either e_1 or e_2, horizontal components 1–13 in Figure 7 are used in this transition. All horizontal components shown in Figures 6 and 7 can be grouped by their connectivity into eight main types: -, a, b, c, $a-b$, $a-c$, $b-c$, and $a-b-c$. The symbol (-) represents the empty horizontal component shown as component 14 in Figure 6. The recursive formula (3) provides a connection between the objective value of the previous states and the current state, where $\omega(h)$ represents the added distance from the horizontal transition corresponding to the horizontal components in Figures 6 and 7, and where h belongs to one of the horizontal components (*Hor*) in Figures 6 and 7.

$$\omega(L_{j+1}^{-}, e') = \min_{h \in \{Hor\}, e \in \{Eq\}} \{\omega(L_j^{+u}, e) + \omega(h)\} \tag{3}$$

Notice, for example, that both configuration 4 in Figure 6 and configuration 1 in Figure 7 have the connectivity pattern a, but the former is used only when neither e_1 nor e_2 or both of them are contained in L_j^{+u}-PTSs, while the latter is used when either e_1 or e_2 are contained in L_j^{+u}-PTSs. Adding a horizontal component from Figure 6 or Figure 7 to an L_j^{+u}-PTS creates an L_{j+1}^{-}-PTS. Table 3 shows $\deg(a_j)$ of the resulting L_{j+1}^{-}-PTS equivalence classes. We note that not every horizontal component can be used due to Theorem 4 (condition 3). For example, if $\deg(a_j)$ in an L_j^{+u}-PTS is even (E), $\deg(a_j)$ of a horizontal component must be even, and the resulting L_{j+1}^{-}-PTS equivalence class will be of even degree as well. The tables for the degree of b_j and c_j can be easily derived and are identical to Table 3.

Similarly, Table 4 shows the connectivity of the L_{j+1}^{-}-PTS that results from adding a horizontal component of different patterns of connectivity. For example, if an L_j^{+u} equivalence class of connectivity type $a-bc$ is extended by a horizontal component of type $a-b$, the connectivity of the resulting L_{j+1}^{-}-PTS will be $a-b$.

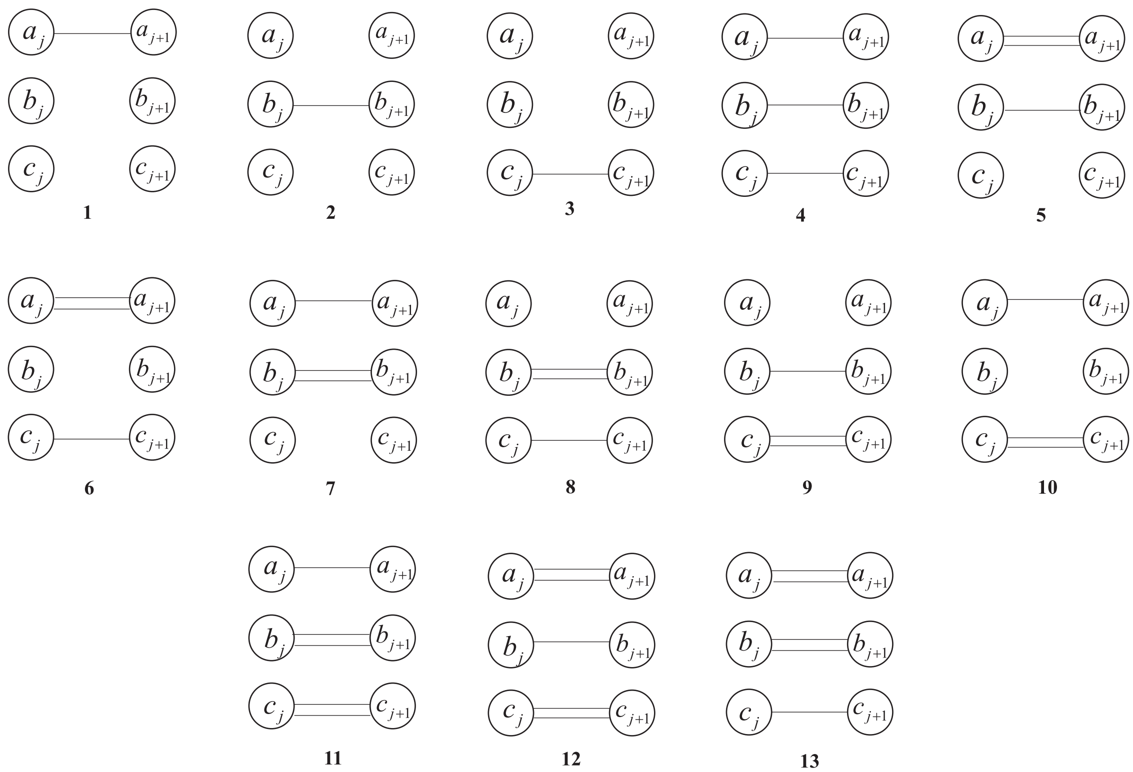

Figure 7. Horizontal components for travelling from aisle j to aisle $j+1$ when an L_j^{+u}-PTS contains either e_1 or e_2.

Table 3. deg(a_j), deg(b_j), and deg(c_j) of the resulting L_{j+1}^{-}-PTS equivalence class.

deg(a_j), deg(b_j), and deg(c_j) in L_j^{+u}	deg(a_j), deg(b_j), and deg(c_j) of horizontal component	
	E	U
E	E	
U		U

In terms of the run-time complexity of the proposed algorithm, it is linear in the number of requested items m and picking aisles r, i.e. $O(m+r)$.

4.1.3. Tour construction algorithm

Section 4.1.2 outlined a procedure for finding the minimum-length tour subgraph that contains the locations of the m required items as well as the starting point e_1 and the ending point e_2. This section presents a procedure to construct a minimum-length order picking tour in the warehouse from the minimum-length tour subgraph of a graph representation G. In the following, we adapt the tour construction procedure presented in Ratliff and Rosenthal (1983) to our problem. The procedure is described in the following:

Step 1. Begin the order picking tour at the starting point e_1 as the first vertex visited.

Step 2. Let v^* be the vertex currently being visited.

Step 3. If there is a pair of unused parallel edges incident to v^*, choose one of them to move to the next vertex, then go back to Step 2. Otherwise, continue to the next step.

Step 4. If there are unused single edges that are not a pair of parallel edges from *Step 3*, incident to v^*, choose one of them to move to the next vertex, then go back to *Step 2*. Otherwise, continue to the next step.

Table 4. Connectivity of L_{j+1}^{-} equivalence class.

Connectivity of L_j^{+u}	Connectivity of the horizontal component added to the L_j^{+u}-PTS							
	–	*a*	*b*	*c*	*a-b*	*a-c*	*b-c*	*a-b-c*
-	-	*	*	*	*	*	*	*
a	#	*a*	&	&	*	*	&	*
b	#	&	*b*	&	*	&	*	*
c	#	&	&	*c*	&	*	*	*
ab	#	*a*	*b*	&	*ab*	*	*	*
ac	#	*a*	&	*c*	*	*ac*	*	*
bc	#	&	*b*	*c*	*	*	*bc*	*
abc	#	*a*	*b*	*c*	*ab*	*ac*	*bc*	*abc*
a-b	&	&	&	&	*a-b*	&	&	*
a-c	&	&	&	&	&	*a-c*	&	*
b-c	&	&	&	&	&	&	*b-c*	*
a-bc	&	&	&	&	*a-b*	*a-c*	&	*a-bc*
b-ac	&	&	&	&	*a-b*	&	*b-c*	*b-ac*
c-ab	&	&	&	&	&	*a-c*	*b-c*	*c-ab*
a-b-c	&	&	&	&	&	&	&	*a-b-c*

* This transition would never give the optimal solution.
# This transition can occur only if there are no items to be picked in $G - L_j^{+u}$.
& This transition would contradict the condition in Theorem 4 (condition 4).

Step 5. If there is a pair of parallel edges incident to v^* including an unused edge, choose it to move to the next vertex, then go back to *Step 2*. Otherwise, continue to the next step.

Step 6. The order picking tour is at the ending point e_2 as the last vertex visited. The minimum-length order picking tour is complete.

4.2. Routing heuristic

This section proposes a routing heuristic, denoted *S*-shape*, that is summarised in the following:

S*-shape: We number the $2r$ picking sub-aisles (each aisle is divided into two sub-aisles; one sub-aisle in each block) from 1 to $2r$ in the following order. Starting in the upper block, the sub-aisles are ordered from left to right. Then we continue with the lower block from right to left, as illustrated in Figure 8. Assume that sub-aisle x contains the starting point e_1 and sub-aisle y contains the ending point e_2. The order picker starts from e_1, moves to the farthest item in the sub-aisle x, then comes back to the middle cross aisle. S/he then moves along the cross aisle to the sub-aisle $x+1$ (where sub-aisle $2r+1$ is sub-aisle 1) if the value of $y - x \bmod 2r$ is among $r, r+1, \ldots, 2r-1$. On the other hand, s/he proceeds to the sub-aisle $x-1$ (where sub-aisle 0 is sub-aisle $2r$) if the value of $y - x \bmod 2r$ is among $0, 1, \ldots, r-1$. If the sub-aisle is not empty, s/he completely traverses the sub-aisle, then continues to the next sub-aisle determined by the value of $y - x$ as before. If the sub-aisle is empty, s/he skips the sub-aisle, and moves to the next sub-aisle. The process is repeated until s/he has visited all picking locations. In the last step, s/he travels from the last picking location to e_2. Figure 8 shows the routing procedure that results from the *S*-shape* heuristic for the example given in Figure 1.

5. Numerical example and computational experiments

5.1. Numerical example

This section illustrates the solution procedure using the example in Figure 2. To find the minimum-length tour subgraph of G, we follow the dynamic programming procedure presented in Section 4.1.2.3 on the sequence of iteratively-built PTSs.

We start from the unique L_1^{-}-PTS of type $EEE(-)$. Since there are items to be picked in the lower block of the left-most picking aisle, and since this aisle neither contains the starting point e_1 nor the ending point e_2, vertical components 1–5 are selected from Table 1 to construct the L_1^{+l}-PTSs and their equivalence classes. The resulting L_1^{+l}-PTSs are $EUU(bc)$, $EEE(b)$, $EEE(c)$, $EEE(b-c)$, and $EEE(bc)$ with their minimum lengths 6, 6, 8, 10, and 12, respectively. To create the L_1^{+u}-PTSs, we add vertical components 7–12 from Figure 4 to all five L_1^{+l}-PTSs that result from the previous transition. The PTSs with their minimum lengths are selected as candidates for the PTSs of the minimum-length tour subgraph. To move from

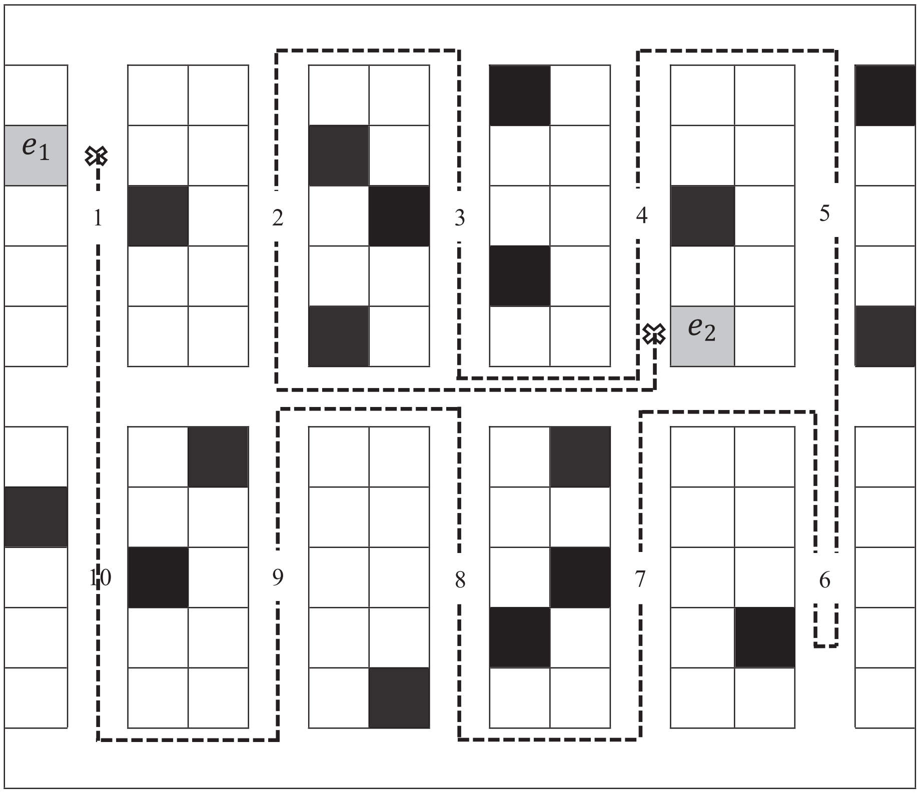

Figure 8. The route resulting from the *S*-shape* heuristic for the example from Figure 1.

aisle 1 to aisle 2, since the L_1^{+u}-PTSs contain e_1, only horizontal components 1–13 in Figure 7 are applicable. For each L_1^{+u}-PTS, we then use Table 3 to rule out some invalid horizontal components and obtain L_2^--PTSs using Table 4. For example, an L_1^{+u}-PTS of type $UUU(a-bc)$ can be extended only by horizontal component 4 in Figure 7. Then, the minimum-length PTSs in L_j^-, L_j^{+l}, and L_j^{+u} for aisles $j = 2, \ldots, 5$ can be calculated in the same manner. This solution procedure was coded in Java, and we obtain the minimum-length tour subgraph of length 65 as shown in Figure 9. To construct a minimum-length order picking tour from this subgraph, the tour construction algorithm proposed in Section 4.1.3 is used.

5.2. Computational experiments

In this section, we conduct computational experiments to first study the impact of the middle cross aisle on the performance of the warehouse in case the starting and ending points of a picking tour can be any location in the warehouse. Secondly, we evaluate the performance of a simple routing heuristic as compared to the exact algorithm for the same scenario. The proposed algorithm and the heuristic were implemented in Java, and all instances were run on a computer with Intel Core i5-7200U 2.50 GHz and 8 GB RAM. Since the average computational time for each instance in Sections 5.2.1 and 5.2.2 is less than 1 s, we conclude that the computational time is less of an issue for the routing procedures we propose in this paper.

5.2.1. Impact of the middle cross aisle on the performance of the warehouse

Roodbergen and de Koster (2001a) conducted numerical experiments to study the impact of the middle cross aisle where a picking tour starts and ends at the depot located at the front of a picking aisle. In order to evaluate the impact of the middle cross aisle on the performance of a warehouse where the starting and ending points of a tour can be any location in the warehouse, we carry out computational experiments to compare the average order picking tour length in the warehouses with and without the middle cross aisle using the exact algorithm proposed in this study. Parameter values varied in the experiments include the size of the warehouse, the layout of the warehouse (with and without the middle cross aisle and the number of aisles), and the size of the pick-list, with some problem sets taken from Roodbergen and de Koster (2001a). We consider two warehouse sizes with a total aisle length of 70 and 450 metres. The distance between two neighbouring

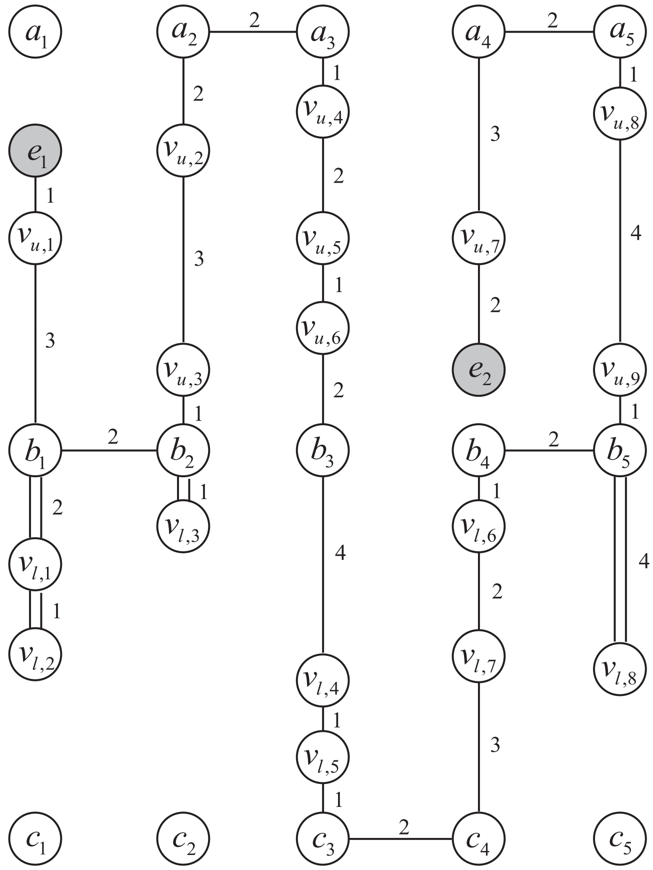

Figure 9. The minimum-length tour subgraph of the graph representation G from Figure 2.

storage locations and two adjacent picking aisles are set to 1 and 3 metre(s), respectively. The distance from the respective last picking positions in the parallel aisles to a cross aisle is set to 1 metre.

To study the impact of the middle cross aisle on the performance of the warehouse, we compare the average order picking tour length of both warehouses (with and without the middle cross aisle) on their optimal layouts. This means that we determine the optimal number of aisles for both warehouse types for each pick-list size. To do so, we calculate the average tour length for a fixed pick-list size by varying the number of aisles from 1 to 50. For example, for the optimal number of aisles in the warehouse with a total aisle length of 450 metres and a pick-list size of 30, we consider warehouses with 1 aisle of 450 metres, 2 aisles of 225 metres, … , and 50 aisles of 9 metres. For each warehouse layout, we then randomly generate 1,000 orders with 30 items per pick-list and calculate the average tour length for the warehouse without the middle cross aisle. After that, the middle cross aisle is added to the warehouse and the average tour length is calculated for the same 1,000 orders. Figure 10 shows the average tour length in both warehouses (with and without the middle cross aisle) with a total aisle length of 450 metres and a pick-list size of 30 items. We find the optimal number of aisles for both warehouse layouts by choosing the minimum of the tour length curve in Figure 10. The optimal number of aisles for the warehouses with and without the middle cross aisle are 25 and 30 with average tour lengths of 293.7 and 358.2 metres, respectively. We apply the same procedure to find the optimal number of aisles and the average tour lengths for all pick-list sizes from 1 to 50.

Figure 11 presents the average tour lengths for each pick-list size ranging from 1 to 50 for warehouses with and without the middle cross aisle with total aisle length of 450 metres. The average tour lengths for the warehouse with the middle cross aisle are lower than the ones obtained for the warehouse without the middle cross aisle for all considered pick-list sizes. This implies that adding the middle cross aisle reduces the average tour length also in the case where picking tours start and end in different locations. This is because the middle cross aisle offers more possibilities for creating tours for the order picker (Roodbergen and De Koster 2001a). To measure the impact of the middle cross aisle, we calculate the percentage savings in average tour length in both warehouse layouts using the formula $(Z^{wo} - Z^{w})/Z^{wo}$, where Z^{w} and Z^{wo} represent the average tour lengths of the warehouse with and without the middle cross aisle, respectively. It can be seen from Figure 12 that the

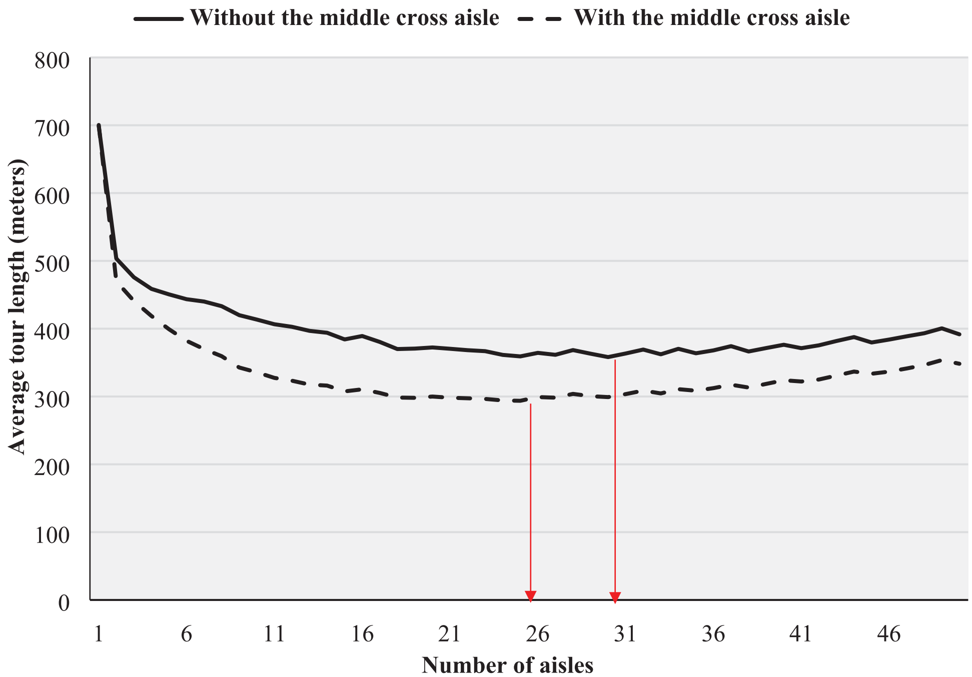

Figure 10. Average tour lengths in the warehouse with and without the middle cross aisle for a total aisle length of 450 metres and a pick-list size of 30.

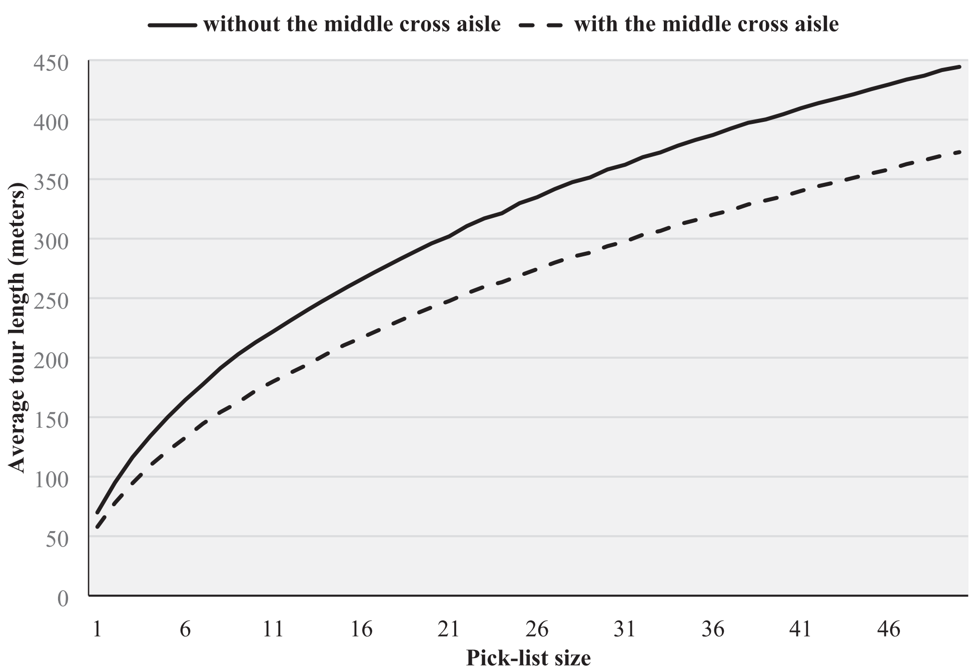

Figure 11. Average tour lengths in the warehouse with and without the middle cross aisle for a total aisle length of 450 metres.

percentage savings in average tour length that result from adding the middle cross aisle to the warehouse are approximately between 16% and 20%. The highest percentage saving was found for a pick-list size of 9 items. It can further be seen that the percentage savings in average tour length get smaller as the pick-list size increases. This is due to the fact that the more items the order picker needs to retrieve in a picking tour, the higher is the chance that the order picker has to traverse all aisles, which reduces the advantage the middle cross aisle brings about. This observation is consistent with the results reported by Roodbergen and de Koster (2001a).

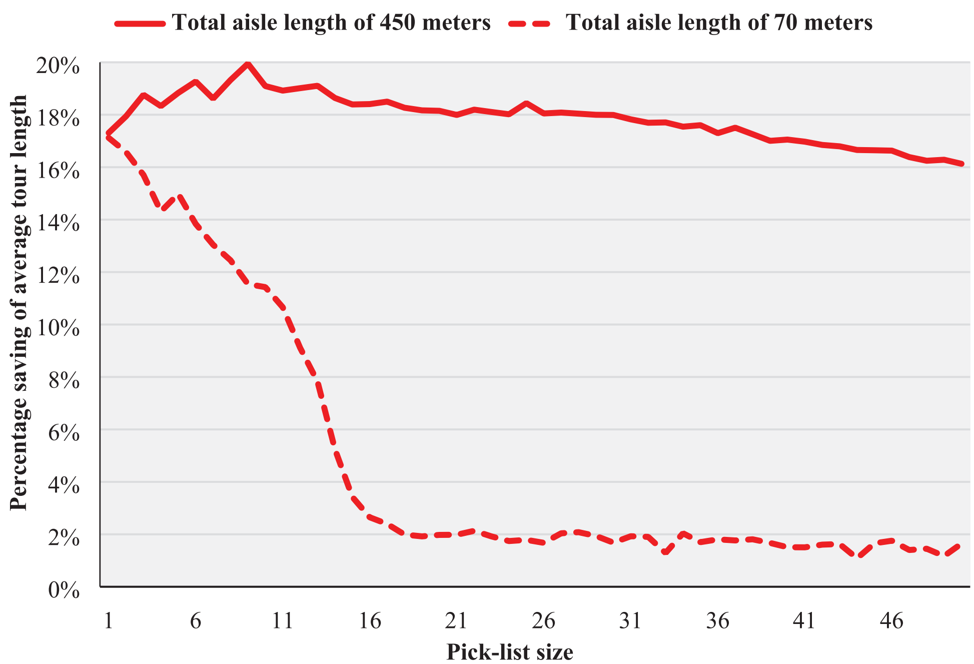

Figure 12. Percentage savings in average tour length when a middle cross aisle is added to the warehouse compared to the case without the middle cross aisle for total aisle lengths of 70 and 450 metres.

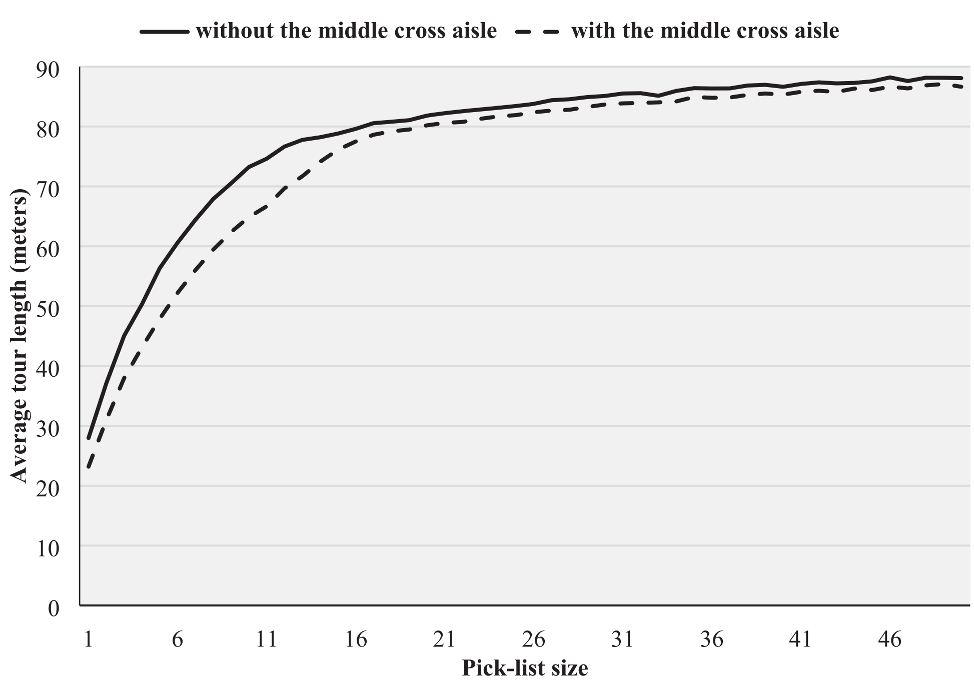

Figure 13. Average tour lengths in the warehouse with and without the middle cross aisle for warehouse with total aisle length of 70 metres.

We use the same experiment for a warehouse with a total aisle length of 70 metres. Figure 13 shows the average tour lengths for each pick-list size. It is obvious that adding the middle cross aisle reduces the average tour length for all considered cases. As can be seen from Figure 12, the percentage savings in average tour length are smaller than for the warehouse with a total aisle length of 450 metres. The percentage savings for pick-list sizes between 1 and 50 vary from 1% to 17%, and seem to become stable at around 1-2% when the pick-list sizes are larger than 17 items. Again, large pick

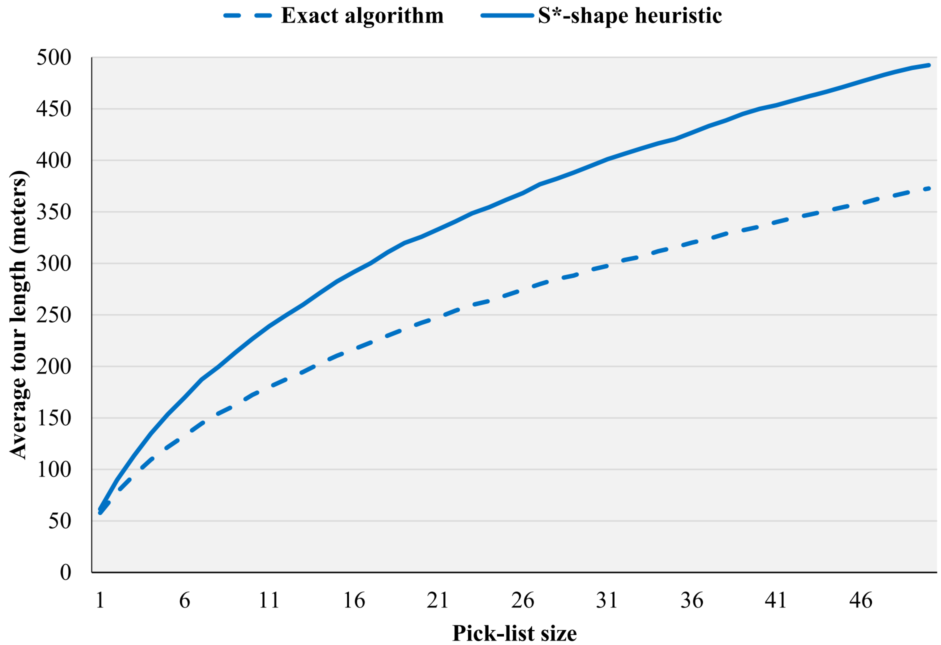

Figure 14. Average tour lengths for the exact algorithm and the *S*-shape* heuristic for a warehouse with a total aisle length of 450 metres.

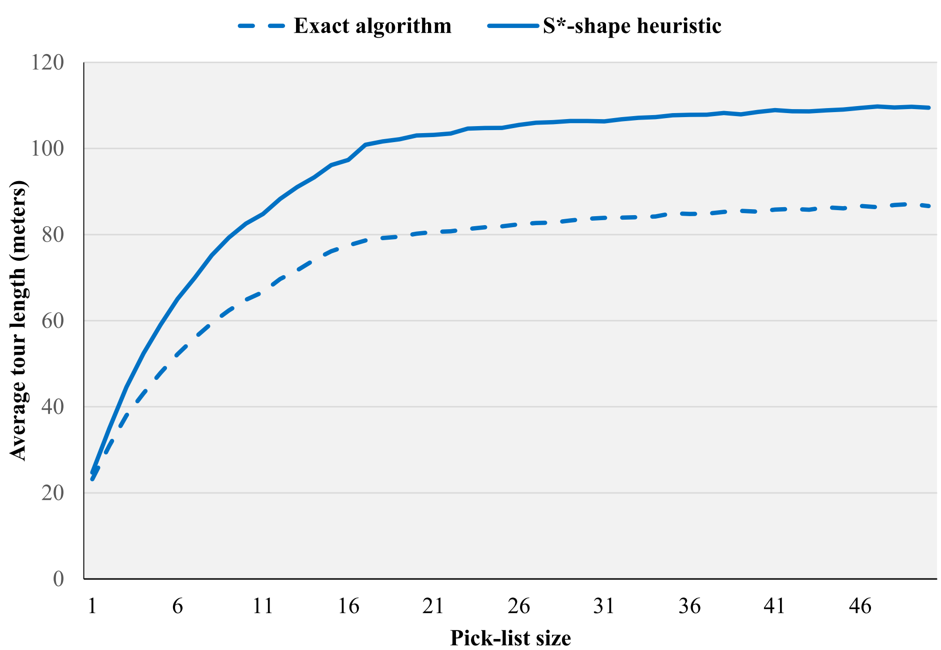

Figure 15. Average tour lengths for the exact algorithm and the *S*-shape* heuristic for a warehouse with a total aisle length of 70 metres.

lists entail that the order picker may have to traverse most of the aisles in an average picking tour, such that the middle cross aisle is only infrequently used in this case.

5.2.2. Comparison of the exact algorithm to a routing heuristic

This section evaluates the performance of the *S*-shape* heuristic by comparing the average length of the order picking tours produced by this method to the tours obtained using the exact routing algorithm.

We compare the average tour lengths of the *S*-shape* heuristic and the exact algorithm on the optimal two-block warehouse for each pick-list size discussed in Section 5.2.1. Figures 14 and 15 show the average order picking tour lengths

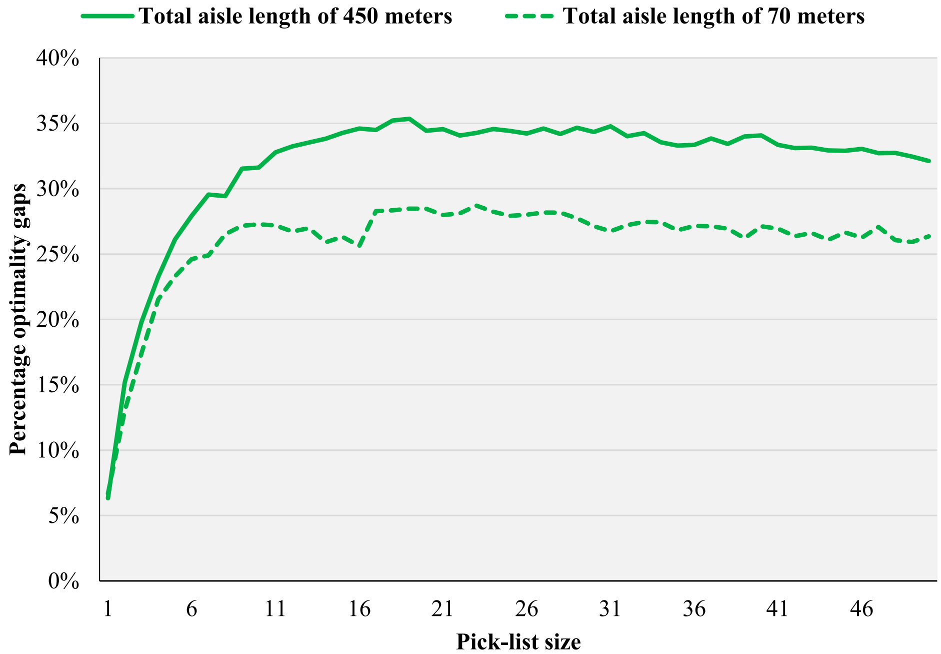

Figure 16. Average percentage optimality gaps of the *S*-shape* heuristic for warehouses with total aisle lengths of 70 and 450 metres.

obtained from the exact algorithm and the *S*-shape* heuristic for two warehouses with a total aisle length of 450 and 70 metres, respectively. In addition, the percentage optimality gap of the *S*-shape* heuristic is calculated as $(Z^h - Z^e)/Z^e$, where Z^h and Z^e represent the average tour lengths resulting from the *S*-shape* heuristic (h) and the exact algorithm (e), respectively. Figure 16 presents the average percentage optimality gaps of the *S*-shape* heuristic for warehouses with total aisle lengths of 70 and 450 metres. There are a few things to point out from the figures. First, the results demonstrate that the exact algorithm clearly outperforms the *S*-shape* heuristic in all considered cases as expected. Secondly, our results indicate that the optimality gap of the *S*-shape* heuristic is between 6.32% and 35.34% for the warehouse with a total aisle length of 450 metres and between 6.68% and 28.70% for the warehouse with a total aisle length of 70 metres. With respect to the warehouse with a total aisle length of 450 metres, the percentage optimality gap of the *S*-shape* heuristic increases when the pick-list size increases, and it reaches a peak of 35.34% when the pick-list size is 19. After that, the percentage optimality gap slightly decreases to values around 32-34%. Similarly, the percentage optimality gap of the *S*-shape* heuristic for the warehouse with the total aisle length of 70 metres increases when the pick-list size increases and reaches 28.70% when the pick-list size is 23. Afterwards, the percentage optimality gap seems to become stable at around 26-27%.

6. Conclusion

This study proposed an efficient algorithm for determining a minimum-length order picking tour as well as a routing heuristic for a conventional warehouse with two blocks where the starting and ending points of the picking tour are not fixed to the depot, but where they can be any locations in the warehouse instead. This paper thus extended an earlier work of Löffler et al. (2018), who studied the case of a conventional warehouse with a single block. The work at hand adapted the solution procedures proposed by Ratliff and Rosenthal (1983) and Roodbergen and de Koster (2001a), both based on graph theory and dynamic programming, for finding a minimum-length order picking tour for this warehouse layout. An example was presented to illustrate how the proposed algorithm works. Furthermore, we conducted computational experiments to investigate the impact of a middle cross aisle by comparing the average length of the optimal order picking tour in a warehouse with and without a middle cross aisle. Finally, we also compared the performance of the proposed routing heuristic to the exact algorithm. In our experiments, the middle cross aisle reduced the average tour length for every problem setting we studied. Moreover, the exact algorithm obtained tours that were between 6.32% and 35.34% shorter than those generated by the heuristic. These results emphasise that optimal order picker routing should be the preferred means of guiding the order picker through the warehouse.

There are several options for extending this work. First, the paper at hand assumed that the warehouse has narrow picking aisles, such that the horizontal travel distance of the order picker within an aisle can be neglected. For future research, it would be interesting to extend the present work to a situation where items are retrieved in a wide-aisle warehouse. In this case, it would also be necessary to calculate an additional horizontal travel distance for picking items from both sides of the aisle. Secondly, it was assumed that no order picker congestion occurs within aisles. Relaxing this assumption would be a natural extension of the present work. Moreover, future research could consider turn penalties, which take into account the time lost whenever the order picker enters or leaves an aisle or when a U-turn is necessary within an aisle. Finally, this study could also be extended to other warehouse layouts, e.g. conventional warehouse with multiple blocks, fishbone or flying-V.

Acknowledgments

The authors are grateful to the anonymous reviewers whose feedback on an earlier version of this paper helped to improve this work substantially.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Christoph H. Glock http://orcid.org/0000-0001-6006-0070

References

Armstrong, R. D., W. D. Cook, and A. L. Saipe. 1979. "Optimal Batching in a Semi-Automated Order Picking System." *Journal of the Operational Research Society* 30 (8): 711–720.

Brynzér, H., and M. I. Johansson. 1996. "Storage Location Assignment: Using the Product Structure to Reduce Order Picking Times." *International Journal of Production Economics* 46: 595–603.

Caron, F., G. Marchet, and A. Perego. 2000. "Optimal Layout in low-Level Picker-to-Part Systems." *International Journal of Production Research* 38 (1): 101–117.

Chabot, T., R. Lahyani, L. C. Coelho, and J. Renaud. 2017. "Order Picking Problems Under Weight, Fragility and Category Constraints." *International Journal of Production Research* 55 (21): 6361–6379.

Chen, F., Y. Wei, and H. Wang. 2018. "A Heuristic Based Batching and Assigning Method for Online Customer Orders." *Flexible Services and Manufacturing Journal* 30 (4): 640–685.

Daniels, R. L., J. L. Rummel, and R. Schantz. 1998. "A Model for Warehouse Order Picking." *European Journal of Operational Research* 105 (1): 1–17.

De Koster, R., T. Le-Duc, and K. J. Roodbergen. 2007. "Design and Control of Warehouse Order Picking: A Literature Review." *European Journal of Operational Research* 182 (2): 481–501.

De Koster, R., and E. Van der Poort. 1998. "Routing Orderpickers in a Warehouse: a Comparison Between Optimal and Heuristic Solutions." *IIE Transactions* 30 (5): 469–480.

Elbert, R. M., T. Franzke, C. H. Glock, and E. H. Grosse. 2017. "The Effects of Human Behavior on the Efficiency of Routing Policies in Order Picking: the Case of Route Deviations." *Computers & Industrial Engineering* 111: 537–551.

Elsayed, E. A., and R. G. Stern. 1983. "Computerized Algorithms for Order Processing in Automated Warehousing Systems." *The International Journal of Production Research* 21 (4): 579–586.

Giannikas, V., W. Lu, B. Robertson, and D. McFarlane. 2017. "An Interventionist Strategy for Warehouse Order Picking: Evidence From two Case Studies." *International Journal of Production Economics* 189: 63–76.

Gibson, D. R., and G. P. Sharp. 1992. "Order Batching Procedures." *European Journal of Operational Research* 58 (1): 57–67.

Goetschalckx, M., and H. D. Ratliff. 1988. "Order Picking in an Aisle." *IIE Transactions* 20 (1): 53–62.

Grosse, E. H., C. H. Glock, and W. P. Neumann. 2017. "Human Factors in Order Picking: a Content Analysis of the Literature." *International Journal of Production Research* 55 (5): 1260–1276.

Hall, R. W. 1993. "Distance Approximations for Routing Manual Pickers in a Warehouse." *IIE Transactions* 25 (4): 76–87.

Hausman, W. H., L. B. Schwarz, and S. C. Graves. 1976. "Optimal Storage Assignment in Automatic Warehousing Systems." *Management Science* 22 (6): 629–638.

Henn, S., S. Koch, K. F. Doerner, C. Strauss, and G. Wäscher. 2010. "Metaheuristics for the Order Batching Problem in Manual Order Picking Systems." *Business Research* 3 (1): 82–105.

Heskett, J. L. 1963. "Cube-per-order Index-a key to Warehouse Stock Location." *Transportation and Distribution Management* 3 (1): 27–31.

Hwang, H., Y. Hui Oh, and C. Nam Cha. 2003. "A Stock Location Rule for a low Level Picker-to-Part System." *Engineering Optimization* 35 (3): 285–295.

Jang, H. Y., and J. U. Sun. 2012. "A Graph Optimization Algorithm for Warehouses with Middle Cross Aisles." *Applied Mechanics and Materials* 145: 354–358.

Kusiak, A., and S. S. Heragu. 1987. "The Facility Layout Problem." *European Journal of Operational Research* 29 (3): 229–251.

Le-Duc, T., and R. M. de Koster. 2007. “Travel Time Estimation and Order Batching in a 2-Block Warehouse.” *European Journal of Operational Research* 176 (1): 374–388.

Löffler, M., N. Boysen, and M. Schneider. 2018. “Picker Routing in AGV-Assisted Order Picking Systems.” Working Paper, DPO-01/2018, Deutsche Post Chair-Optimization of Distribution Networks, RWTH Aachen University.

Lu, W., D. McFarlane, V. Giannikas, and Q. Zhang. 2016. “An Algorithm for Dynamic Order-Picking in Warehouse Operations.” *European Journal of Operational Research* 248 (1): 107–122.

Mantel, R. J., P. C. Schuur, and S. S. Heragu. 2007. “Order Oriented Slotting: a new Assignment Strategy for Warehouses.” *European Journal of Industrial Engineering* 1 (3): 301–316.

Meller, R. D., and K. Y. Gau. 1996. “The Facility Layout Problem: Recent and Emerging Trends and Perspectives.” *Journal of Manufacturing Systems* 15 (5): 351–366.

Petersen, C. G. 1997. “An Evaluation of Order Picking Routeing Policies.” *International Journal of Operations & Production Management* 17 (11): 1098–1111.

Ratliff, H. D., and A. S. Rosenthal. 1983. “Order-picking in a Rectangular Warehouse: a Solvable Case of the Traveling Salesman Problem.” *Operations Research* 31 (3): 507–521.

Roodbergen, K. J., and R. De Koster. 2001a. “Routing Order Pickers in a Warehouse with a Middle Aisle.” *European Journal of Operational Research* 133 (1): 32–43.

Roodbergen, K. J., and R. De Koster. 2001b. “Routing Methods for Warehouses with Multiple Cross Aisles.” *International Journal of Production Research* 39 (9): 1865–1883.

Roodbergen, K. J., I. F. Vis, and G. D. Taylor Jr. 2015. “Simultaneous Determination of Warehouse Layout and Control Policies.” *International Journal of Production Research* 53 (11): 3306–3326.

Scholz, A., S. Henn, M. Stuhlmann, and G. Wäscher. 2016. “A new Mathematical Programming Formulation for the Single-Picker Routing Problem.” *European Journal of Operational Research* 253 (1): 68–84.

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

Van Nieuwenhuyse, I., and R. B. de Koster. 2009. “Evaluating Order Throughput Time in 2-Block Warehouses with Time Window Batching.” *International Journal of Production Economics* 121 (2): 654–664.

Vaughan, T. S., and C. G. Petersen. 1999. “The Effect of Warehouse Cross Aisles on Order Picking Efficiency.” *International Journal of Production Research* 37 (4): 881–897.