European Journal of Operational Research 322 (2025) 1–14

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/eor

Invited review

A survey on the Traveling Salesman Problem and its variants in a warehousing context

Stefan Bock [a], Stefan Bomsdorf [b], Nils Boysen [c,*], Michael Schneider [b]

[a] Bergische Universität Wuppertal, Lehrstuhl für Wirtschaftsinformatik und Operations Research, Gaußstraße 20, 42119 Wuppertal, Germany
[b] RWTH Aachen University, Deutsche Post Chair – Optimization of Distribution Networks, Kackertstraße 7, 52072 Aachen, Germany
[c] Friedrich-Schiller-Universität Jena, Lehrstuhl für Operations Management, Carl-Zeiß-Straße 3, 07743 Jena, Germany

ARTICLE INFO

Keywords:
Routing
Warehousing
Traveling Salesman Problem
Complexity

ABSTRACT

With the advent of e-commerce and its fast-delivery expectations, efficiently routing pickers in warehouses and distribution centers has received renewed interest. The processes and the resulting routing problems in this environment are diverse. For instance, not only human pickers have to be routed but also autonomous picking robots or mobile robots that accompany human pickers. Traditional picker routing, in which a single picker has to visit a given set of picking positions in a picker-to-parts process, can be modeled as the classical Traveling Salesman Problem (TSP). The more involved processes of e-commerce fulfillment, however, require solving more complex TSP variants, such as the clustered, generalized, or prize-collecting TSP. In this context, our paper provides two main contributions: We systematically survey the large number of TSP variants that are known in the routing literature and check whether meaningful applications in warehouses exist that correspond to the respective TSP variant. If they do, we survey the existing research and investigate the computational complexity of the TSP variant in the warehousing context. Previous research has shown that the classical TSP is efficiently solvable in the parallel-aisle structure of warehouses. Consequently, some TSP variants also turn out to be efficiently solvable in the warehousing context, whereas others remain $\mathcal{NP}$-hard. We survey existing complexity results, provide new ones, and identify future research needs.

1. Introduction

One of the most often cited statistics in warehousing research is certainly that 55% of the total warehouse operating expenses are typically attributed to order picking (Frazelle, 2001). Moreover, the order picking process itself typically consists of 50% of travel, in which pickers move unproductively through the aisles of a warehouse to reach their picking positions (Tompkins et al., 2010). Given these figures, it is anything but surprising that seeking efficient picker tours has attracted plenty of scientific research and has become one of the classics of warehousing research.

In the most basic picker-to-parts setup, picker routing can be modeled as the Traveling Salesman Problem (TSP): Given a set of cities (picking positions) and distances between each pair of cities, we seek the shortest tour of the salesperson (order picker) that visits each city exactly once and finally returns to the start city (depot). Different from general graphs, for which the TSP is well-known to be strongly $\mathcal{NP}$-hard, the seminal paper of Ratliff and Rosenthal (1983) shows that this is not true in a warehousing environment. If we have parallel picking aisles with cross aisles at the front and back, which is referred to as *single-block structure*, the specially structured distance matrix allows solving the TSP in polynomial time using a dynamic program. Given that this elementary result is already four decades old, a survey paper on the TSP and its variants in a warehousing environment should provide convincing answers to the following three concerns:

Concern 1: Warehouses have greatly evolved in the past decades, and the progress in automation has diminished the importance of picker routing.

Answer: Especially the advent of e-commerce has transformed many of today's warehouses into technology-enriched, mission-critical fulfillment factories (Boysen et al., 2019). To handle the large number of time-critical orders of a typical e-commerce warehouse with the aging workforces of most industrialized countries, automated and robotized solutions for all elementary warehousing functions have been developed (see Azadeh et al., 2019; Fragapane et al., 2021): Rack-climbing (Chen et al., 2022) and autostore robots (AutoStore, 2023) for

* Corresponding author.
E-mail addresses: sbock@winfor.de (S. Bock), bomsdorf@dpo.rwth-aachen.de (S. Bomsdorf), nils.boysen@uni-jena.de (N. Boysen), schneider@dpo.rwth-aachen.de (M. Schneider).
URLs: https://www.wirtschaftsinformatik.uni-wuppertal.de (S. Bock), https://www.dpo.rwth-aachen.de (S. Bomsdorf), https://www.om.uni-jena.de (N. Boysen), https://www.dpo.rwth-aachen.de (M. Schneider).

https://doi.org/10.1016/j.ejor.2024.04.014
Received 10 July 2023; Accepted 17 April 2024
Available online 21 April 2024

storage, shelf-lifting mobile robots (Amazon Robotics, 2023; Weidinger et al., 2018) for transportation, autonomous mobile robots (AMRs) such as those offered by Magazino (2023) for picking, and robots equipped with a tiltable tray (Zou et al., 2021) for order consolidation. While the general trend towards warehouse automation is evident, autonomous robots also profit from efficient tours and thus require algorithmic routing support (Azadeh et al., 2019). Moreover, automated warehousing systems have the disadvantage that they cannot undergo rapid modifications to adjust for demand peaks, whereas human-centric warehouses can flexibly adjust, e.g., by hiring non-permanent stand-by staff during peak periods such as Singles' Day or Black Friday (Boysen et al., 2019). Thus, it can be projected that large e-commerce retailers will stick to some highly flexible human-centric warehouses on top of their largely automated fulfillment factories for the base load (Schiffer et al., 2022). To keep the costs for the flexibility gains of human-centric warehouses at a manageable level, efficient picker routing, both for humans and robots, remains a vital issue to gain a competitive edge.

Concern 2: Four decades of warehousing research do not leave many open questions regarding the routing of pickers.

Answer: Picker routing has indeed attracted plenty of research; the latest survey paper of Masae et al. (2020a), for instance, has identified 149 papers on this topic. Important contributions are certainly the extension of the dynamic program of Ratliff and Rosenthal (1983) to two-block warehouses with an additional middle cross aisle by Roodbergen and de Koster (2001b) and to multi-block warehouses by Pansart et al. (2018) (based on the results of Cambazard & Catusse, 2018). Furthermore, many routing heuristics have been investigated (e.g., Roodbergen & de Koster, 2001a), mixed-integer programs exploiting the special warehouse structure have been introduced (e.g., Goeke & Schneider, 2021) and many extensions of the basic picker routing problem have been considered (see, e.g., Masae et al., 2020a). Our paper partly surveys and structures these previous research efforts. More importantly, the above-mentioned diversification of warehousing processes, driven by automation and the pressure of discerning online customers, has led to numerous new routing tasks that have not been exhaustively covered by warehousing research, as our further elaborations will show. Thus, we see good arguments for another survey paper on order picking, especially with our special perspective on existing TSP research, that has not been taken before.

Concern 3: Technological and methodological progress has made the differentiation between efficiently solvable and (binary or strongly) $\mathcal{NP}$-hard problems rather unimportant from a practical viewpoint.

Answer: The aforementioned fulfillment factories of large e-commerce retailers are vast properties that store millions of products, i.e., not stock-keeping units (SKUs) but pieces (see Schiffer et al., 2022). The manual picking carts applied by many B2C online retailers have a capacity for multiple bins, each with a capacity for dozens of products, so that the instance sizes of the resulting routing tasks are challenging even for today's (impressively improved) off-the-shelf solvers and state-of-the-art TSP (variant) solvers. Therefore, an exact algorithm with polynomial runtime is still an important contribution (both from a theoretical and practical perspective) to compute optimal solutions quickly. Furthermore, there is a strong trend toward integrated problem settings that involve multiple decision tasks (see van Gils et al., 2018). Picker routing is, for instance, often jointly solved with batching, in which numerous smaller customer orders are to be partitioned into multiple pick lists, each served by a single picker tour. A natural solution approach for such an integrated problem is decomposition, where a routing algorithm is applied to evaluate each batch. In such a setting, an efficient routing algorithm can be the workhorse of a decomposition approach to exhaustively explore the vast batching solution space. In contrast, an $\mathcal{NP}$-hardness proof is a clear sign that another type of algorithm, e.g., a heuristic, is required to solve the respective problem in real-life warehouses. Thus, the complexity status of an optimization problem, which we survey for TSP variants in the warehousing context in this paper, is still an important theoretical result with practical implications.

Given these motivations, our paper makes the following contributions:

- After a thorough review of dozens of TSP variants in the abundant routing literature, we identify ten TSP variants that are especially relevant in the warehousing context. We discuss their warehousing use cases (some of them are highlighted for the first time), identify relevant extensions of the basic problem, survey previous (warehousing) research, and identify future research needs.
- From a theoretical perspective, we survey the complexity status of the TSP and its variants in a block-structured warehousing environment. Furthermore, we provide three new complexity results and identify four open cases, which should be addressed by future research.

The remainder of the paper is structured as follows. Section 2 defines the scope of this survey. Here, we explain the typical block structure of warehouses, describe related research not treated by our survey, and discuss the differences to previous surveys. Section 3 introduces the TSP in a warehousing context, sketches the dynamic program of Ratliff and Rosenthal (1993), and reviews further research in this area. The following ten sections each treat one variant of the TSP, ranging from the TSP with precedence constraints in Section 4 to the Covering Salesman Problem in Section 13. All these sections follow the same structure: We start with the definition of the TSP variant. Then, we describe the respective warehousing use cases, investigate the complexity status in a warehousing environment, survey existing warehousing research, and identify future research needs. Finally, Section 14 concludes the paper.

2. Scope of the survey

This section defines the scope of our survey, characterizes our policy to identify relevant TSP variants, and elaborates on the differences to previous survey papers. First, we offer the following (positive) definition of our paper's scope: We survey single picker routing problems, in which storage positions in a warehousing environment must be visited.

The defining feature of a warehousing environment is the parallel-aisle structure depicted in Fig. 1. Specifically, there are v parallel picking aisles containing storage positions of products to be visited by the picker. Furthermore, there are h cross aisles to move from one picking aisle to another. For $h = 2$, there is one cross aisle at the front and the back, which results in a *single-block* warehouse as depicted in Fig. 1(a). An additional middle cross aisle results in the two-block layout of Fig. 1(b). Naturally, additional cross aisles can also produce three-block and four-block warehouses, and so on. To refer to the general case with $h > 3$ cross aisles, we use the term *multi-block* warehouse.

The main implication of the block layout is a specially structured distance matrix based on rectilinear distances between picking positions and the depot, where picking tours start and end. Due to this special structure, we cannot simply refer to the TSP and its variants when investigating the computational complexity of a specific picker routing problem in a warehouse: The proofs of $\mathcal{NP}$-hardness for the TSP and its variants are based on general graphs with arbitrary distances. Because computational complexity is one of the main issues addressed in this paper, it is important to clearly differentiate between a general TSP (or a variant) and a special TSP with a specific block structure. To denominate the former, we use the well-established abbreviations such as TSP. When referring to the TSP in a single-block layout, instead, we denominate this case as '1B-TSP'. Analogously, '2B-TSP' (MB-TSP) refers to the TSP in a two-block (multi-block) environment with $h = 3$ ($h > 3$) cross aisles. When addressing the routing problem in any of the three above block structures, we use the abbreviation 'W-TSP'.

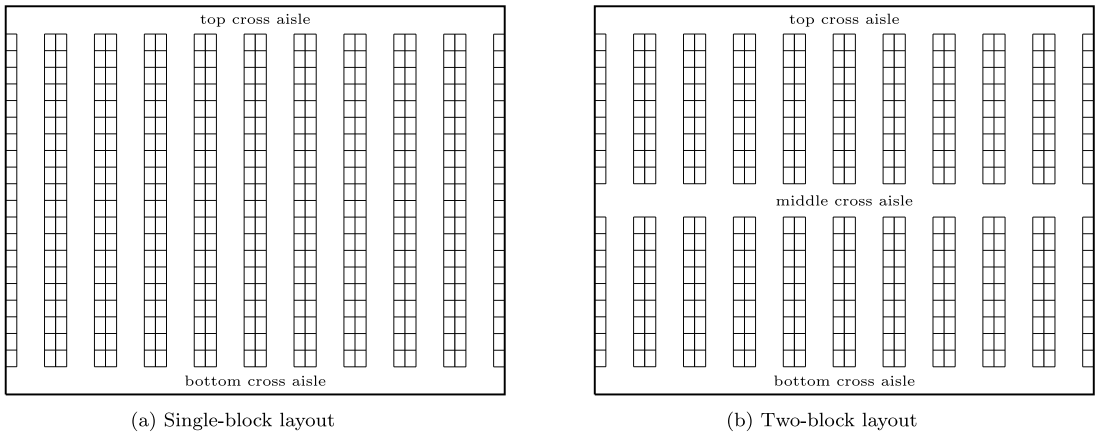

Fig. 1. Parallel-aisle structure of a warehouse environment.

In addition to the aforementioned (positive) definition of our paper's scope, we also provide a (negative) demarcation of the TSP variants in a warehousing environment that are not treated in our paper:

(i) Warehousing research also investigates alternative layouts that violate the block structure defined above. The most prominent examples are the flying-V (from a bird's-eye view, the middle cross aisle cuts a 'V' through the parallel picking aisles with the lowest point of the 'V' being the depot, see Gue & Meller, 2009), the fishbone (the picking aisles below a 'V'-shape of the cross aisle are shifted by 90°, see Çelk & Süral, 2014), and the discrete cross aisle design (the middle cross aisle does not cut through the parallel picking aisles on a line but intersects at various positions, see Öztürkoğlu et al., 2012). These alternative layouts, however, are rather an academic playground, and the vast majority of real-world warehouses follows the block structure. Therefore, these alternatives are not treated in our paper.

(ii) We only consider single picker problems. We neglect interactions among multiple pickers, e.g., when competing for specific products (e.g., Ardjmand et al., 2018) or blocking each other in narrow aisles (e.g., Schrotenboer et al., 2017). In practical warehousing, coordination issues are typically addressed on an upstream planning level (e.g., during batching, when it is decided which customer orders end up on the same pick lists, see Boysen et al., 2019) or handled by the involved workers on-site.

(iii) We only consider static and deterministic variants of the TSP with a single objective. Foremost, this is a pragmatic choice to reduce the content to a manageable level and to focus on those problem types directly accessible to an analysis of computational complexity.

Next, we briefly specify our policy to identify the TSP variants that are relevant in a warehousing environment. We started the process with a thorough literature and database search on TSP variants. This search resulted in 54 different TSP variants that have been considered in at least one English-written paper published in a peer-reviewed scientific journal (see Appendix A). As a first step, we filtered out the articles violating our scope defined above. In a brainstorming meeting with seven researchers and a warehousing consultant, each remaining TSP variant was assessed with regard to potential warehousing applications. The results were documented and sent out to five warehousing researchers who had not been previously involved. Only if the majority of them voted that the warehousing application is plausible, the respective TSP variant is treated in this paper. However, despite our attempts to objectify this process, we have to openly admit that the selection is certainly biased by the authors' subjective assessment.

Finally, several survey papers on related issues have already been written. There are, for instance, the survey papers on warehousing in general by de Koster et al. (2007), van Gils et al. (2018), and Boysen et al. (2019). Because picker routing is one of the classics of warehousing research, each of these surveys treats picker routing as one important topic. Furthermore, there are focused surveys specifically on picker routing by Masae et al. (2020a) as well as Vanheusden et al. (2023). However, none of these surveys takes our special TSP perspective or focuses on computational complexity. Vice versa, surveys from the routing domain that focus on the TSP and its variants (e.g., Applegate et al., 2007; Gutin & Punnen, 2007) usually do not consider the special impact of the block structure of warehouses and potential applications in this area. We, instead, combine both perspectives, which was not done before.

In the next sections, we turn to the TSP and its variants in a warehousing context. Specifically, we start with the classical TSP in Section 3 and then elaborate on ten TSP variants in Sections 4 to 13. As mentioned above, each section follows the same structure: We start with a problem definition and refer to important research contributions as well as survey papers on the general problem outside the warehousing domain. To give a rough indication of problem instances that can be solved, we add performance information for the most representative results in recent work on the TSP variants. It is clear that an in-depth analysis of instance characteristics and computing environments would be required to reliable compare different methods and variants. We still believe that this information is useful to provide a rough idea of what is computationally achievable. Then, we describe the warehousing use cases of the respective problem, explore the status of computational complexity if a block structure is present, and finally, highlight future research needs.

3. The classical TSP and its application to warehousing

Definition. We introduce a mathematical programming formulation for the classical TSP and adapt it to unambiguously define the respective TSP variants in the following sections. Introducing these formulations aims to simplify the differentiation between the problem definitions. They are, however, not the basis for state-of-the-art solution methods. Let $G = (V, A)$ be a digraph with nodes $V = \{1, \dots, n\}$ and arcs $A = \{(i, j) : i, j \in V, i \neq j\}$. Each arc $(i, j) \in A$ is assigned a fixed cost c_{ij}. The goal of the asymmetric TSP (ATSP) is to find a Hamiltonian circuit T in G such that the sum of the arc costs in T is minimum. Binary variables x_{ij} indicate if an arc (i, j) is part of T ($x_{ij} = 1$) or not ($x_{ij} = 0$). Continuous variables u_i assign a unique index to each node i, indicating the sequence position in the tour. The ATSP can thus be formulated as the following integer linear program proposed by Miller et al. (1960):

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{1}$$

$$\text{s.t.} \sum_{i\in V, i\neq j} x_{ij} = 1 \qquad j \in V \tag{2}$$

$$\sum_{j\in V, j\neq i} x_{ij} = 1 \qquad i \in V \tag{3}$$

$$u_i - u_j + n x_{ij} \leq n - 1 \qquad (i,j) \in A; j \neq 1 \tag{4}$$

$$x_{ij} \in \{0,1\} \qquad (i,j) \in A \tag{5}$$

$$u_i \geq 0 \qquad i \in V \tag{6}$$

The objective function (1) minimizes the total cost of all arcs contained in the tour. Constraints (2) and (3) ensure that each node is visited exactly once. Constraints (4) are the famous subtour elimination constraints of Miller et al. (1960). The decision variables are defined in (5) and (6).

We briefly summarize the state-of-the-art exact and heuristic methods for the symmetric TSP (STSP) and the ATSP. For the STSP, the solver Concorde implements the currently best exact approach using the branch-cut-and-price procedure of Applegate et al. (2007). A comparison of the best exact approaches can be found in Laporte (2010). Concorde can consistently solve STSP instances with 500 randomly distributed nodes in less than a minute. For the ATSP, the branch-and-cut of Fischetti et al. (2003) is among the best exact methods, and a comparison is available in Roberti and Toth (2012). Fischetti et al. (2003) solve almost all instances with up to 1000 nodes in a couple of seconds or minutes while only a few instances run up to 29 min or are not solved to optimality within the time limit of 2.78 h. The best heuristics to address both the ATSP and the STSP are enhancements of the Lin-Kernighan algorithm (Helsgaun, 2017), stem-and-cycle methods (Rego et al., 2011), and the matheuristic POPMUSIC (Taillard & Helsgaun, 2019). For benchmark instances with 1000 nodes, they consistently achieve runtimes of a few seconds.

Warehouse use case: The single picker routing problem in warehouses corresponds to the W-TSP. A good example for the direct applicability of the W-TSP is the following most basic order fulfillment process: In many warehouses, high-rise racks are partitioned into a ground-level pick face and an upper reserve area. In the upper reserve area, unit loads are stored to resupply the ground-level pick face, typically by forklifts. The pick face is stocked with unit loads (e.g., pallets with identical pieces of the same SKU), which are accessed by pickers who either walk while pushing manual picking carts or drive motorized picking carts between storage positions defined on their pick lists. Once pick lists are assigned to pickers, optimizing each tour individually corresponds to solving a W-TSP. Pickers start and end their tours at a single depot, where orders are prepared for shipping. In the most basic setup, order consolidation at the depot (i.e., sorting by customer orders) is not required because either each pick list only contains a single customer order (order-by-order picking) or the picking cart carries multiple bins, each associated with a specific customer order for a pick-while-sort process (de Koster et al., 2007). The default case in the majority of warehouses is certainly an STSP. One reason for an ATSP setup, however, could be one-way picking aisles, which are applied for safety reasons in some warehouses with plenty of motorized picking cart traffic (Boysen et al., 2019). Nonetheless, if not explicitly stated otherwise, we assume symmetric distances for all TSP variants in the warehousing context in the remainder of the paper.

Alternatively, the single picker routing problem can also be modeled as a Steiner TSP (Rodríguez-Pereira et al., 2019). The Steiner TSP extends the TSP by two aspects: The underlying graph does not have to be complete, and some nodes can be visited but are not obligatory. The former property can be used to condense the distance matrix, so that only nodes representing adjacent picking positions in the same picking aisle are directly connected. The first (last) node within each picking aisle can then be connected with a node representing the crossing point between picking and cross aisle, which is optional to visit. This problem representation is, for instance, used by Letchford et al. (2013) to derive compact mathematical programming formulations for the W-TSP.

Complexity status: (a) The 1B-TSP is solvable in polynomial time (see Ratliff & Rosenthal, 1983). Note that Heßler and Irnich (2022) have recently shown that this algorithm is actually linear in the sum of the number of aisles and number of picking positions if the cost coefficients of the transitions of the dynamic program are computed from an unsorted list of picking positions. (b) The 2B-TSP is solvable in polynomial time (see Roodbergen & de Koster, 2001b). (c) The MB-TSP is solvable in polynomial time if the number of cross aisles h is bounded by a constant (see Cambazard & Catusse, 2018; Pansart et al., 2018). More details on these and all following complexity results are provided in Appendix B.

Future research: Regarding the complexity status, the classical TSP in a warehousing environment is fully explored. Hence, we see future research needs rather in relation to the TSP variants that are addressed in the following sections. As a substitute, we comment on the application of heuristics for the W-TSP instead, which, we believe, have lost their prior importance. Because optimal solutions can efficiently be obtained, from a methodological perspective, there is no need to accept the suboptimal solutions of heuristics. At the same time, routing heuristics have a long-lasting tradition in warehousing research. The most prominent examples are the S-shape (meander through the aisles in an S-shaped tour), return (never use the back cross aisle), mid-point (always return before crossing the middle of a picking aisle), and largest-gap (in each picking aisle, do not traverse the two storage positions having the largest distance between them) heuristics (for detailed descriptions and performance tests, see Petersen, 1997; Roodbergen & de Koster, 2001a). Gademann and Velde (2005), for instance, argue that the tours of these heuristics are more intuitive for human pickers, whereas optimal tours "seem illogical or suboptimal to the order pickers who then, as a result, deviate from the specified routes". Empirical support for the existence of such a *maverick picking* is surveyed by Glock et al. (2017). It would be interesting to see whether the empirical results listed there are still valid today with most order pickers being digital natives, who are used to algorithmic advice. To avoid maverick picking, most modern warehouses apply navigation tools such as pick-by-voice solutions. Hence, routing heuristics have certainly lost some of their importance over the years.

4. TSP with precedence constraints

Definition. The TSP with precedence constraints (TSPPC) is a generalization of the ATSP which introduces additional constraints on the sequence of visits (first introduced by Balas et al., 1995). For any pair of nodes $i, j \in V$, a precedence rule $i \prec j$ can be added, stating that i must be visited prior to j in the solution but not necessarily directly before j. Without loss of generality, let the start and end node of the tour be the node excluded in Constraints (4), such that the tour is closed and precedence constraints are well-defined. In the following, node 1 is considered the start and end node. Based on the TSP formulation (1)–(6), the TSPPC can be modeled by adding constraints

$$u_i \leq u_j - 1 \qquad (i,j) \in A : i \prec j; i, j \neq 1. \tag{7}$$

The path version of the TSPPC, i.e., finding a Hamiltonian path with precedence constraints from a start to an end node in G, is also known as sequential ordering problem (Escudero, 1988). The currently best exact methods are branch-and-cut (Gouveia & Ruthmair, 2015), branch-and-bound using beam search (Libralesso et al., 2020), and dynamic programming (Salii & Sheka, 2020). Salii and Sheka (2020) solve TSPPC instances with 7–16 nodes in less than 80 ms using multiple CPUs, while larger instances with up to 253 nodes run in the range of a minute up to an hour. The best heuristic is the hybrid of ant colony optimization and local search of Skinderowicz (2017). They solve instances with up to 700 nodes with a runtime limit of 600 s.

Warehouse use case: Precedence constraints among storage positions can be utilized to influence the arrangement of products on roll containers and pallets for the customers. This is especially important in distribution centers supplying brick-and-mortar stores, where most orders are large because the stores bundle the demand of multiple customer households (Boysen et al., 2021). In such a setting, precedence constraints can support adherence to weight (heavy products at the bottom of the roll containers), fragility (light products on top), stackability, and stability restrictions (big boxes at the bottom), as well as preferred unloading sequences that mirror the store layout (Matusiak et al., 2014).

Complexity status: The complexity statuses of the 1B-TSPPC, the 2B-TSPPC, and the MB-TSPPC are still open.

Future research: First and foremost, future research should resolve the open complexity statuses. Beyond that, precedence constraints do not directly translate into a specific packing pattern of the roll containers and pallets for customers. For instance, a fragile product does not need to be placed on top of a heavy product. They may also be placeable next to each other on the same level of a packing pattern, and, to realize this, the fragile product may still be retrieved before the heavy product. Thus, future research should also consider more holistic routing problems that include the packing pattern and its successive realization on a tour in detail. Such an approach can then be utilized to explore the performance loss caused by the modeling error of precedence constraints.

Beyond the traditional picking process, precedence constraints can also be used to model pickup and delivery processes in warehouses. If product demands change over time, rearranging the storage assignment may become necessary to relocate new fast-moving products closer to the depot. To model such a process, precedence constraints between the storage locations of each pickup (i.e., old position of the SKU) and the corresponding delivery (i.e., its new position) can be introduced. Naturally, the W-TSPPC neglects vehicle capacity. Hence, if a forklift is applied in the rearrangement process, which only has capacity for a single pallet, a multi-commodity one-to-one pickup-and-delivery traveling salesman problem with limited vehicle capacity (Hernández-Pérez & Salazar-González, 2009) must be solved. Future research should explore whether the block structure of warehouses can be exploited to solve this problem more efficiently.

5. Clustered TSP

Definition. In the clustered TSP (CTSP), sets of nodes (clusters) are introduced, in which the contained nodes must be visited contiguously (first introduced by Chisman, 1975). Thus, the visiting order of clusters and the order of nodes within the same cluster is optimized simultaneously. While this setting is referred to as the CTSP with open cluster sequence (CTSP-OCS), the sequence of clusters is predefined in the CTSP with given cluster sequence (CTSP-GCS). To formally introduce the CTSP-OCS, let $V_k, k \in K = \{1, \dots, l\}$, be l disjoint clusters with $V = \bigcup_{k \in K} V_k$, $V_k \cap V_{k'} = \emptyset; k, k' \in K, k \neq k'$. We extend the TSP formulation (1)–(6) by adding

$$\sum_{i \in V_k} \sum_{j \in V_k} x_{ij} = |V_k| - 1 \qquad k \in K \tag{8}$$

to ensure that all nodes belonging to the same cluster are visited consecutively. To model the CTSP-GCS, instead of (8), we extend the TSP by

$$\sum_{k'=1}^{k-1} |V_{k'}| + 1 \le u_i \le \sum_{k'=1}^{k} |V_{k'}| \qquad k \in K; i \in V_k, i \neq 1 \tag{9}$$

assuming ordered sets V_k, i.e., for $k < k'$ $(k, k' \in K)$ it holds: $i \prec j, i \in V_k, j \in V_{k'}$. Note that $\sum_{k'=1}^{k-1} |V_{k'}| = 0$ for $k = 1$. Research on solution methods for the CTSP is sparse. Exact methods for the CTSP-OCS are the branch-and-bound procedure of Lokin (1979) and the Lagrangian relaxation of Jongens and Volgenant (1985). Among the best heuristics for the CTSP-OCS are the memetic algorithm of Alsheddy (2017) and the hybrid algorithm of greedy randomized adaptive search procedure, iterated local search, and variable neighborhood descent of Mestria (2018). Alsheddy (2017) solve instances with up to 105 nodes and 50 clusters to optimality in less than 0.5 s; Mestria (2018) solve instances with up to 500 nodes and 6 clusters within a runtime limit of 15 s obtaining a gap of around 5%. The latter compare to Concorde that solves these instances in around 40 s. Only few publications focus on the CTSP-GCS (e.g., see Anily et al., 1999; Potvin & Guertin, 1998).

Warehouse use case: There are two potential use cases for the W-CTSP: (1) AMR-assisted picking and (2) picking multiple orders (clusters) with multiple depots. We elaborate on both use cases in the following.

(1) AMR-assisted picking: To reduce the unproductive picker travel from and to the depot in each picking tour, recent technological advances enable cooperation between human pickers who pair up with AMRs. Especially, the gripping process itself still is a challenge for automation and restricted in the product range it can be applied to (see, e.g., Correll et al., 2016). The Toru robot of Magazino (2023), for instance, can only process rectangular products (e.g., shoe boxes) with a lower performance than human pickers. As a bridging technology, AMR-assisted picking, thus, still relies on human pickers, who place picked products into bins carried by the respective AMR that accompanies them. Once a picker and an AMR have paired up to collect a new pick list, they jointly proceed through the warehouse until all storage positions defined on the pick list have been visited. Then, the AMR returns to the depot with the picked products, while the human picker travels to the first storage position of the next pick list to meet another AMR for the next pick list (and so on). This work protocol, which Löffler et al. (2023a) call the *fixed assignment policy*, directly corresponds to the W-CTSP if the AMR fleet is not a bottleneck resource so that the picker never has to wait for an AMR at the meeting points. In this case, the clusters of the W-CTSP correspond to pick lists, which must be processed sequentially by a picker and its paired-up AMR before the former switches to the next pick list and pairs up with the next AMR. If the sequence in which pick lists are processed by the picker is unrestricted, we obtain the W-CTSP-OCS. However, varying urgency of orders can also imply a given processing sequence, so that the W-CTSP-GCS must be solved. Both problem settings are investigated by Löffler et al. (2021).

(2) Picking with multiple depots: Some warehouses do not only use a single central depot as the unique start and end point of each tour. Instead, they provide multiple access points to a central conveyor system, which transports the bins to a consolidation area. In such a multi-depot setting, each access point is a potential starting point for the next tour, where the bins full of picked products of the previous tour are handed over and new empty bins are obtained. For a given set of pick lists, the optimal tour of a picker who processes multiple pick lists (clusters) sequentially can also be modeled as a W-CTSP. To do so, the distances between storage positions of the same pick list can directly be derived from the real-world warehouse layout. Distances between storage positions of different pick lists, instead, must include the minimum detour via one of the depots to feasibly switch to a new pick list. Again, there are potential warehousing use cases for both versions, namely W-CTSP-GCS and W-CTSP-OCS, if the processing sequence of pick lists is either given or part of the decision, respectively.

Complexity status: (a) The 1B-CTSP-OCS (and, thus, also the 2B-CTSP-OCS and the MB-CTSP-OCS) is strongly $\mathcal{NP}$-hard (see Löffler et al., 2021). (b) The MB-CTSP-GCS (and, thus, also the 1B-CTSP-GCS and the 2B-CTSP-GCS) is solvable in polynomial time (see Appendix B and Löffler et al., 2021).

Future research: The application of the W-CTSP to *AMR-assisted picking (1)* is subject to two basic prerequisites: the application of the fixed-assignment policy and the exclusion of waiting times for AMRs. Instead of a fixed assignment of picker to AMR for each pick list, the AMR fleet can also act under the *free-floating policy*. This means that AMRs move autonomously between the storage positions of their current pick lists, where they are supported by arbitrary pickers without fixed assignment. This policy, which is considered by Löffler et al. (2023a), promises more flexibility and, thus, a higher picking performance. However, this policy requires synchronization among all pickers and AMRs, which complicates the planning process and makes the system vulnerable to spillover effects of delays. Thus, more flexible work protocols and the inclusion of stochastic influences such as unexpected delays make AMR-assisted picking a challenging field for future research.

More involved routing tasks should also be investigated in the context of *multiple depots (2)*. Quite a few warehouses, especially in e-commerce, use picking carts with a capacity for multiple bins that are processed in parallel. In this case, it can occur that some of a cart's bins are already completed, whereas others still lack products. Then, completed bins can already be handed over at a depot that is passed by during picking, while the uncompleted bins remain on the cart. This results in a dynamic batching process (for first approaches, see Schiffer et al., 2022; Weidinger et al., 2019), which offers manifold research opportunities.

6. Generalized TSP

Definition. The generalized TSP (GTSP, see, e.g., Laporte & Nobert, 1983) also considers clusters of nodes. However, instead of visiting each node exactly once, at least one node per cluster must be visited. Thus, an additional decision must be made, i.e., selecting the nodes to be visited. Let $y_i, i \in V$, be a binary variable which is set to 1 if node i is visited, and 0 otherwise. We exchange (2) and (3) in the TSP formulation (1)–(6) with (10) and (11), and add variable definitions (12):

$$\sum_{i \in V} x_{ij} = y_j \qquad j \in V \tag{10}$$

$$\sum_{j \in V} x_{ij} = y_i \qquad i \in V \tag{11}$$

$$y_i \in \{0,1\} \qquad i \in V. \tag{12}$$

Further, let $V_k, k \in K$, be l clusters with $V = \bigcup_{k \in K} V_k$. Note, that in contrast to the CTSP, the clusters are not necessarily disjoint. Then, we impose the selection of at least one node per cluster by extending the model with

$$\sum_{i \in V_k} y_i \geq 1 \qquad k \in K. \tag{13}$$

The GTSP is also known as set TSP, group TSP, or international TSP. A stricter variant is known as equality GTSP, in which exactly one node per cluster must be visited. Consequently, equality must hold in constraints (13). Note that if the triangle inequality holds for a GTSP instance, which is given for the block structure of warehouses, in an optimal solution exactly one node per cluster is chosen (Laporte & Nobert, 1983). Hence, no differentiation between the GTSP and the equality GTSP is necessary in the warehousing context. Among the best exact approaches for the GTSP are the branch-and-cut algorithm of Fischetti et al. (1997) and the Lagrangian-based branch-and-bound algorithm of Noon and Bean (1991). Fischetti et al. (1997) optimally solve instances with up to 442 nodes and 89 clusters in 16.3 h. The state-of-the-art heuristic methods are the Lin-Kernighan-Helsgaun heuristic (Helsgaun, 2015) and the recently introduced iterated local search of Schmidt and Irnich (2022). Both solve instances with up to 1084 nodes and 217 clusters within a runtime limit of 335 s. Note, however, that the GTSP can also be transformed into an ATSP (see Noon & Bean, 1993) and an ATSP into an STSP (e.g., Ben-Arieh et al., 2003), so that solvers for these problems can also be used to solve GTSPs. Pop et al. (2024) extensively survey the GTSP.

Warehouse use case: Many warehouses, especially large facilities of e-commerce retailers, apply scattered storage (see Boysen et al., 2019). Instead of putting unit loads, commonly pallets of identical products into storage, these warehouses break up the unit loads and store individual pieces in many different positions of the warehouse. The main promise of scattered storage is that whatever products end up jointly on hardly predictable pick lists, there is an increased probability that somewhere in the huge warehouses these products are stored close together and can be picked without excessive picker travel. This advantage on the picking side comes at the price of a more laborious stowing process. Instead of merely putting a unit load into a shelf (e.g., with a single forklift move), an additional stowing workforce has to travel through the warehouse to stow individual products into open shelf positions.

Thus, the picking process in a scattered storage warehouse must handle alternative storage positions from where a specific SKU on a pick list can be obtained. Including this additional selection problem, can directly be modeled as a W-GTSP. Clusters are formed by the set of storage positions of a specific SKU, and at least one storage position from each cluster must be visited to fulfill the product demand of a pick list. Daniels et al. (1998) were the first to consider product availability at multiple positions. A decomposition heuristic, which selects the storage positions per cluster first and solves the resulting W-TSP second, is presented by Weidinger (2018).

Complexity status: The 1B-GTSP (and, thus, also the 2B-GTSP and the MB-GTSP) is strongly $\mathcal{NP}$-hard (see Weidinger, 2018).

Future research: The W-GTSP can represent picking in scattered storage warehouses without modeling error only if merely a single piece of any SKU is requested on a pick list. For higher demands per SKU, the family TSP becomes relevant. Here, the number of nodes per cluster that must be visited to fulfill the demand for more than a single piece is also part of the input. This problem has been introduced by Morán-Mirabal et al. (2014) for general graphs and recently got extended by Bernardino and Paias (2022) to include incompatibility constraints. Note that the W-family TSP is a generalization of the W-GTSP and, thus, also strongly $\mathcal{NP}$-hard. The family TSP, however, is still not general enough to cover all real-world situations. It may very well occur that storage positions do not just contain a single piece of a SKU but multiple ones. This can be modeled within the realm of the family TSP by introducing multiple virtual storage positions with zero distances between each other. However, this approach produces a lot of extra storage positions if many pieces of an SKU are stored at the same storage position. To avoid extra storage positions, an even more general problem setting that includes bookkeeping of the number of pieces per visited storage position and SKU is required. Despite the high practical relevance of scattered storage, this problem has not been investigated yet.

7. TSP with backhauls

Definition. The TSP with backhauls (TSPB) is the special case of the CTSP-GCS defined in Section 5 with exactly two clusters (first introduced by Gendreau et al., 1996). The first cluster is referred to as linehaul nodes V_L and the second one as backhaul nodes V_B. Linehaul nodes must be visited prior to backhaul nodes, i.e., $V = V_L \cup V_B; i \prec j, i \in V_L, j \in V_B$. Research on the TSPB remains limited: multiple heuristics are proposed in Gendreau et al. (1996) and improved in Mladenović and Hansen (1997). Further, Gendreau et al. (1997) introduce a 3/2-approximation algorithm. Mladenović and Hansen (1997) solve instances with 500 linehaul and 500 backhaul nodes in around 30 min.

Warehouse use case: Scattered storage warehouses (see Section 6) can apply separate workforces for stowing and picking. It is, however, also possible to combine both processes. In this case, a worker receives a bin containing products to be stowed at a depot. The worker then travels through the warehouse to first stow all these products and, once the bin is empty, switches to picking. This saves the intermediate return to the depot and promises performance gains. If the storage positions, where products are to be stowed and from where products must be picked, have been preselected in an upstream planning process, the resulting routing problem can directly be modeled as a W-TSPB. The linehaul and backhaul nodes are the storage positions where products must be stowed and picked, respectively. Note that the 1B-TSPB is treated by Žulj et al. (2018). They, however, apply the problem to another warehouse use case. They differentiate heavy (linehaul) products that must be picked first to end up on the bottom of the pallets for the customers. Only then a switch to the picking of light and fragile (backhaul) products is allowed.

Complexity status: The MB-TSPB (and, thus, also the 1B-TSPB and the 2B-TSPB) is solvable in polynomial time (see Appendix B and Žulj et al., 2018).

Future research: The W-TSPB can only cover the most basic version of a combined stowing and picking process in scattered storage warehouses. For instance, the selection among alternative stowing and picking positions, multiple depots, as well as multiple bins on a cart, which induces switches between stowing and picking for individual bins at different times, can be relevant. These more general combined stowing and picking processes have not been treated in the literature yet.

8. Prize-collecting TSP

Definition. The prize-collecting TSP (PCTSP) is a generalization of the TSP, in which profits $w_i, i \in V$, are assigned to each node (first introduced by Balas, 1989). A penalty p_i is due if a node is not part of the tour. The objective is to minimize the tour costs while collecting a minimum total profit W. With $y_i, i \in V$, defined like in the GTSP (see Section 6), the PCTSP objective function is

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij} + \sum_{i\in V} p_i(1-y_i). \tag{14}$$

Besides constraints (4)–(6) and (10)–(12), we add

$$\sum_{i\in V} w_i y_i \geq W \tag{15}$$

to induce the lower bound on the collected profits. Note that the penalties are set to 0 in many applications; i.e., the objective function (14) reduces to the classical TSP objective function (1) (e.g., see Feillet et al., 2005). Clímaco et al. (2021) propose a branch-and-cut algorithm and two MIP-based heuristics for the symmetric case. Their branch-and-cut algorithm solves instances with 500 nodes in around 120 s. The same instances are solved using their heuristics in 9 to 110 s obtaining the same solutions. Pantuza and de Souza (2022) introduce a Lagrangian relaxation approach for the asymmetric case.

Warehouse use case: If stowing and picking are not combined in a scattered storage warehouse (see Section 7), the W-PCTSP can be used to direct the process of a single stower, whose (exclusive) task is to store all products on the cart in open storage positions as fast as possible. Thus, the nodes are open storage positions, and the profit associated with each of these positions is defined by the number of products that can be stored there. Penalties for unvisited open storage positions do not exist. In total, storage positions must be selected such that all products on the cart (modeled as the minimum profit of the PCTSP) can be stowed, while the tour cost to visit all selected positions on a tour starting and ending at the depot is minimized.

Complexity status: The study of Bock and Boysen (2023) shows that the 1B-PCTSP is binary $\mathcal{NP}$-hard, even if penalties for non-visited nodes are excluded. This complexity result also applies to the non-restricted problems 1B-PCTSP, 2B-PCTSP and MB-PCTSP. Furthermore, Appendix B sketches an exact branch-and-bound approach for the MB-PCTSP that attains a pseudo-polynomial asymptotic runtime if the number of cross aisles is bounded by $h \in \mathcal{O}(\log n)$. This resolves the complexity status of the W-PCTSP.

Future research: Again, the W-PCTSP only captures the basic stowing process in scattered storage warehouses. Beyond that, products to be stowed can be of different sizes or shapes, so that an additional bin packing problem to fit products into storage positions may become relevant. Multiple stowers operating in parallel can compete for open storage space, so that the team version of the W-PCTSP must be solved. Since warehouse data typically only hold the number of pieces per SKU that are stored in a specific shelf but not their detailed stowing pattern within the shelf, the exact number of pieces per SKU that can actually be stored in an open storage positions often is non-deterministic. Thus, a stochastic problem version arises in which the number of stowable products per position is uncertain and only revealed after the arrival at a storage position. These extended problem versions of the W-PCTSP constitute challenging tasks for future research.

9. Orienteering problem

Definition. In the Orienteering Problem (OP), again, each node $i \in V$ has a nonnegative weight w_i. The objective is to maximize the collected weights of visited nodes without exceeding a given maximum tour cost (first introduced by Golden et al., 1987). Thus, the objective function is

$$\max \sum_{i\in V} w_i y_i \tag{16}$$

subject to constraints (4)–(6) and (10)–(12), plus

$$\sum_{(i,j)\in A} c_{ij} x_{ij} \leq C \tag{17}$$

to induce the upper bound C on the tour cost. The OP is also known as selective TSP (Laporte & Martello, 1990) and closely related to the PCTSP; both problems belong to the class of TSPs with profits (see Feillet et al., 2005). Gunawan et al. (2016) survey the OP and multiple variants. The best exact approach is the branch-and-cut algorithm of Fischetti et al. (1998). A recent high-quality heuristic approach is the adaptive large neighborhood search of Santini (2019). Using a time limit of 5 min, their heuristic solves instances with up to 400 nodes with an average gap of 0.001% to the best-known solution found by the branch-and-cut algorithm of Fischetti et al. (1998). The latter finds optimal solution for all mentioned instances within a time limit of 5 h.

Warehouse use case: Especially e-commerce warehouses face tight due dates to meet the next-day, same-day, or even within-the-hour delivery promises made to their customers (Boysen et al., 2019). When the fixed departure time of a delivery vehicle (also denoted as *cutoff time*) approaches, the maximum tour cost that is acceptable to still reach this departure time can be precomputed. Given this maximum tour cost and a profit per storage position (e.g., capturing the number, value, or urgency of the products stored there), the W-OP can maximize the profit associated with those products still reaching the vehicle departure.

Complexity status: The 1B-OP (and, thus, also the 2B-OP and the MB-OP) is binary $\mathcal{NP}$-hard, which we prove in Appendix B. Furthermore, we provide an exact solution approach that solves the OP in pseudo-polynomial time if the number of cross aisles is bounded by $h \in \mathcal{O}(\log n)$. This resolves the complexity status of the W-OP.

Future research: Varying, storage position-specific picking times and individual due dates for different products extend the W-OP and should be considered by future research. The latter leads to the deadline TSP (see Bansal et al., 2004) in the warehouse, where each storage position is associated with a different deadline. This problem as well as the team orienteering problem (e.g., Gunawan et al., 2016) have not been considered in the warehousing context yet.

10. Traveling Repairman Problem

Definition. The goal of the Traveling Repairman Problem (TRP) is to find a Hamiltonian cycle H in G that minimizes the sum of arrival times at each node in H (first introduced by Afrati et al., 1986). Let $t_{ij}, (i,j) \in A$, be the given travel time between two nodes. Further, let $t_i, i \in V$ be the arrival time at node i ($t_1 = 0$). Thus, the objective function of the TRP can be formulated as

$$\min \sum_{i \in V} t_i. \tag{18}$$

To state the TRP model, besides the TSP constraints (2)–(6), we add

$$t_i + t_{ij} \leq t_j + M x_{ij} \qquad (i,j) \in A; j \neq 1 \tag{19}$$

$$t_i \geq 0 \qquad i \in V \tag{20}$$

with a large number M, e.g., $M = \sum_{(i,j)\in A} t_{ij}$, to set the arrival times. The TRP is also known as the minimum latency problem, the delivery man problem, or the cumulative TSP. The currently best performing exact method is the branch-and-price algorithm of Bulhões et al. (2018). They solve instances with around 50 nodes in less than 11 s. Some instances with around 150 nodes could not be solved to optimality within the time limit of 2 days. Among the best heuristics is the metaheuristic of Silva et al. (2012). Their heuristic solves instances with around 100 nodes in less than 10 s providing optimal solutions and yields new best-known solutions for instances with 200, 500, and 1000 nodes in around 70 s, 25 min, and 8 h.

Warehouse use case: In warehouses dealing with temperature-sensitive products, maintaining the cold chain is essential to not jeopardize the products like flowers, pharmaceuticals, or groceries handled there. In such a setting, reducing the uncooled travel times, i.e., after a picked product has been removed from its refrigerated shelf by a picker until it reaches the depot (where it is put, e.g., into a cooled trailer) could be a suitable measure for preserving the cold chain. To operationalize this, optimizing the picking tours according to the W-TRP is one valid option. The TRP, however, minimizes the sum of travel times from the picker's departure at the depot to the arrivals at the storage positions. In our context of maintaining the cold chain, the picker has to follow the optimal tour returned by the solution of the W-TRP in reverted sequence. Since distances in warehouses are symmetric, this yields the optimal picker tour that minimizes the sum of uncooled travel times until a picked product arrives at the depot.

Complexity status: The complexity statuses of the 1B-TRP, the 2B-TRP, and the MB-TRP are still open.

Future research: Foremost, future research should resolve the open complexity statuses. Beyond that, the basic W-TRP could be extended by non-negligible (storage position-specific) picking times and products with diverging cooling requirements. The latter could also be a reason for another objective function. For instance, each product could be assigned an individual maximum acceptable uncooled travel time, and it is the aim to maximize the minimum difference of actual uncooled travel times from the target times among all products.

11. *k*-best TSP

Definition. The goal of the k-best TSP is to find the set of k-best TSP solutions, i.e., a set of k tours for which the tour costs of the tour with the highest cost is smaller or equal to the costs of any other feasible tour which is not part of the set. van der Poort et al. (1999) introduce and solve the k-best TSP, and compare their results to a branch-and-bound-algorithm. They report the runtimes for solving the k-best TSP compared to the 1-best TSP. For example, solving the 8-best TSP for an instance with 21 nodes doubles the runtime.

Warehouse use case: As elaborated in Section 2, this paper focuses on single-picker problems, which is justified by the basic assumption that neglecting the coordination aspect among the workforce is often pardonable (and can, e.g., be resolved on a local level by the workers). Especially when pickers compete for specific products (Ardjmand et al., 2018) or block each other in narrow aisles (Schrotenboer et al., 2017), it can, however, be preferable to include the coordination aspect into the team version of a routing problem. A straightforward approach to trade off individual picking performance and coordination aspects has recently been introduced by Löffler et al. (2023b) to reduce gatherings of human pickers in an infection-plagued warehouse. They determine the k-best W-TSP tours per picker and apply a straightforward MIP to select one tour per picker to minimize meetings in the aisles. Note that, strictly speaking, Löffler et al. (2023b) only consider the case of $k = 2$ optimal solutions, because they obtain the second optimal solution by simply reverting the tour. Analogously, the k-best TSP can be applied to other team coordination problems in warehouses without wasting too much picking performance.

Complexity status: The complexity statuses of the 1B-k-best TSP, the 2B-k-best TSP, and the MB-k-best TSP are still open.

Future research: Foremost, future research should resolve the open complexity status. Beyond that, the k-best versions of all other W-TSP variants elaborated in this paper are valid fields of research to coordinate teams for the respective use cases.

12. TSP with time windows

Definition. In the TSP with time windows (TSPTW), starting service at a node is only allowed during a given time window $[e_i, l_i], i \in V$ (first introduced by Savelsbergh, 1985). The most studied variant of the TSPTW, which minimizes the total tour cost, is modeled using the objective function (1) plus constraints (2)–(6), (19)–(20) and the additional time window constraints

$$e_i \leq t_i \leq l_i \qquad i \in V. \tag{21}$$

Among the best exact approaches for the TSPTW with cost minimization are approaches based on dynamic programming or branch-and-cut (Ascheuer et al., 2001; Baldacci et al., 2012; Boland et al., 2017). Baldacci et al. (2012), for example, solve 24 out of 25 instances with 152 or 202 nodes to optimality with an average runtime of 399.8 s. The best performing heuristics are the general variable neighborhood searches of da Silva and Urrutia (2010) and Mladenović et al. (2013), the hybrid of beam search and ant colony optimization of López-Ibáñez and Blum (2010), and the modified variable neighborhood search of Karabulut and Tasgetiren (2014). The former consistently provide solutions to the above-mentioned instances in 26 to 40 s. For the TSPTW with further objectives, see Ye et al. (2024).

Warehouse use case: Even in the largest facilities, picking tours rarely exceed an hour. Hence, it seldom occurs that out-of-stock situations restrict access to storage locations to specific time windows. Products are either available during the whole planning horizon or not at all, so that time windows for storage positions do not occur naturally in most warehouses. However, they can still be used to model two aspects mentioned in previous sections: (a) Given cooled products obtained from refrigerated shelves (see Section 10), the end (start) of a time window for a specific storage location can be set to the maximum acceptable uncooled travel time of the product stored there (to the travel time to reach the depot from there). Solving the resulting W-TSPTW instance and following the resulting tour in reverted sequence provides a minimum-cost picking tour without risking the safety of products until they reach their refrigerated environment at the depot. (b) Time windows can also be applied to coordinate a team of pickers via a straightforward two-stage decomposition approach (see Section 11): In a first stage, each picker is assigned non-overlapping time windows for shelf (or aisle) access, so that blockings or gatherings are excluded. Solving the resulting W-TSPTW instances per picker on the second stage, then ensures that team coordination does not cost too much picking performance.

Complexity status: The 1B-TSPTW (and, thus, also the 2B-TSPTW and the MB-TSPTW) is strongly $\mathcal{NP}$-hard, which we prove in Appendix B by a reduction from the Line-TSPTW that is shown to be strongly $\mathcal{NP}$-hard by Tsitsiklis (1992).

Future research: First, the basic W-TSPTW requires extension if picking times are non-negligible (compared to the typically much longer travel times) and depend on the number and types of products picked at each storage position. Second, the performance loss of the simple team coordination approaches sketched above (see warehouse use case b) and in Section 11 should be benchmarked against more sophisticated multi-picker routing problems directly including the coordination aspect.

13. Covering Salesman Problem

Definition. The Covering Salesman Problem (CSP) is a generalization of the TSP in which not every node needs to be visited but must be in the coverage of a node that is part of the tour (first introduced by Current & Schilling, 1989). The coverage of a node is defined by a given radius R. Thus, $V_i^{cov} = \{j : c_{ji} \leq R\}$ is the set of nodes that cover node $i \in V$. Using the TSP objective (1) with GTSP constraints (4)–(6) and (10)–(13), the CSP can be formulated by adapting constraints (13) as follows:

$$\sum_{k \in V_i^{cov}} y_k \geq 1 \qquad i \in V. \tag{22}$$

Among the best heuristic approaches for the CSP are the local search algorithms of Golden et al. (2012), the hybrid of tabu search, large neighborhood search and Lin–Kernighan heuristic of Lu et al. (2021), and the parallel variable neighborhood search of Zang et al. (2022). The latter solve instances with up to 783 nodes in less than 40 s with an average gap to the best-known solution of around 1%.

Warehouse use case: Autonomous mobile robots are also applied for automated stock-taking in warehouses (Fragapane et al., 2021). If products on the shelves are tagged with RFID chips, a mobile robot equipped with an RFID reader can conveniently register physical inventory even if deep-lane storage is applied (Morenza-Cinos et al., 2019). Because RFID readers have a certain range corresponding to radius R of the CSP, finding the shortest tour such that all relevant storage positions are covered for automated stock-taking can be modeled by W-CSP. The same problem can be applied for alternative systems based on automated product recognition, in which a mobile robot is equipped with a camera system (see Santra & Mukherjee, 2019). When computing the covering node sets V_i^{cov}, however, it must be considered that the camera's line of sight can be blocked by shelves.

Complexity status: The complexity statuses of the 1B-CSP, the 2B-CSP, and the MB-CSP are still open.

Future research: Foremost, future research should resolve the open complexity status. Beyond that, the team version of the W-CSP certainly demands consideration because especially large e-commerce warehouses may use multiple stock-taking robots in parallel.

14. Conclusions

This paper surveys the application of the TSP and its variants in warehousing. Specifically, we describe (known and novel) warehousing use cases for different TSP variants and investigate their complexity status in the block structure of warehouses. As a substitute for a more detailed verbal summary, Table 1 lists our findings.

From a general perspective, beyond the specific warehousing use cases treated in the previous section, we see future research needs in the following areas:

Decomposition methods: Nowadays, warehouses are predominantly massive facilities where multiple pickers interact concurrently. In such environments, pickers vie for products, stowers seek available storage space, and all workers may impede each other in narrow aisles. The focus of this paper is on individual worker problems, overlooking these interactions and merely outlining potential decomposition methods (refer to Sections 11 and 12). A viable avenue for future research involves conducting a systematic evaluation of these concepts and benchmarking them against novel decomposition methods that leverage the extensive repertoire of efficient single-worker W-TSP variants as subproblems.

Dynamic and stochastic problems: Undoubtedly, warehouses offer a comparatively controlled environment that is less susceptible to dynamic influences and stochastic variations compared to most other stages of the supply chain. However, it is important to acknowledge the presence of dynamic impacts (e.g., returned products affecting ongoing picking processes, as discussed in Section 7) and stochastic influences (e.g., stowers needing to determine the number of products that can be stowed at a specific storage position based on the packing pattern of mixed shelves, as discussed in Section 8). Neglecting these factors would be imprudent. Therefore, future research should focus on investigating how the existing solution methods for the static and deterministic W-TSP variants discussed in this paper can be effectively integrated into comprehensive solution frameworks (e.g., multi-scenario approaches) to address the challenges posed by dynamic and stochastic elements.

Computational complexity: Instead of having to independently establish the complexity status of each variant of the W-TSP, it would be advantageous to develop overarching criteria (such as those based on a structured problem hierarchy) that determine the circumstances under which even the parallel-aisle structure of warehouses does not permit an efficient solution.

We conclude with the following final remark: It is amazing to see that such an old-established field like routing in warehouses still offers so many unexplored use cases and unresolved methodological research challenges. Hence, it seems safe to project that routing in warehouses will remain a vividly researched field in the foreseeable future.

Table 1
Summary of results.

Variant	Use case	Complexity	Reference
1B-TSP	Basic picker routing	Polynomial	Ratliff and Rosenthal (1983)
2B-TSP	Basic picker routing	Polynomial	Roodbergen and de Koster (2001b)
MB-TSP	Basic picker routing	Polynomial	Pansart et al. (2018)
1B-, 2B-, MB-TSPPC	Picker routing with precedence constraints	Open	–
1B-, 2B-, MB-CTSP-OCS	AMR-assisted picking, multiple depots	Strongly $\mathcal{NP}$-hard	Löffler et al. (2021)
1B-, 2B-, MB-CTSP-GCS	AMR-assisted picking, multiple depots	Polynomial	Löffler et al. (2021)
1B-, 2B-, MB-GTSP	Picking in scattered storage	Strongly $\mathcal{NP}$-hard	Weidinger (2018)
1B-, 2B-, MB-TSPB	Combined stowing and picking	Polynomial	Žulj et al. (2018)
1B-, 2B-, MB-PCTSP	Stowing in scattered storage	Binary $\mathcal{NP}$-hard	This paper
1B-, 2B-, MB-OB	Picking with cutoff time	Binary $\mathcal{NP}$-hard	This paper
1B-, 2B-, MB-TRP	Picking perishable goods	Open	–
1B-, 2B-, MB-k-best TSP	Picker team coordination	Open	–
1B-, 2B-, MB-TSPTW	Picking perishable goods, team coordination	Strongly $\mathcal{NP}$-hard	This paper
1B-, 2B-, MB-CSP	Stock taking with autonomous robots	Open	–

CRediT authorship contribution statement

Stefan Bock: Formal analysis, Methodology, Validation, Writing – review & editing. **Stefan Bomsdorf:** Conceptualization, Data curation, Validation, Visualization, Writing – review & editing. **Nils Boysen:** Conceptualization, Methodology, Supervision, Validation, Writing – original draft. **Michael Schneider:** Conceptualization, Supervision, Validation, Writing – review & editing.

Acknowledgments

Stefan Bock and Nils Boysen thankfully acknowledge the support of the German Science Foundation (DFG) by the grant "Routing of human and automated order pickers in modern warehouses" (BO 1972/2-1 and BO 3148/14-1). All authors thank a dedicated review team of three anonymous reviewers whose insightful comments helped to improve the paper significantly.

Appendix A. Description of TSP variant identification

Table 2 reports the discussed TSP variants and the result of the selection process. The variants are labeled as follows:

(0) treated in this work
(1) violates scope definition (e.g., multi-objective, multiple workers, or stochastic)
(2) no valid warehousing use case found during brainstorming session
(3) identified use case failed to convince the independent experts.

Appendix B. Description of (known and new) complexity results

Because we see no additional value in providing detailed complexity proofs and algorithm descriptions that are already presented in detail in the original papers, our appendix explains existing results verbally. For precise descriptions, we refer to the original sources. New results of this paper are presented in more detail.

TSP: The 1B-TSP with v picking aisles and n storage positions to be visited can be solved in linear time (i.e., in $\mathcal{O}(v+n)$, see Heßler & Irnich, 2022) using the dynamic program (DP) of Ratliff and Rosenthal (1983). In a nutshell, this DP proceeds as follows. It constructs and extends *partial tour subgraphs* (PTSs) by adding arcs and nodes representing horizontal movements in the cross aisles and vertical traversals of picking aisles starting with the left-most picking aisle. Each PTS can be described by one out of seven possible states, describing the degree of the vertices at the top and bottom of the right-most picking aisle and the number of components of the PTS. Based on its state, possible transitions are derived. Applying a transition to the current PTS generates a successor PTS. Two subsequent picking aisles i and $i+1$ can be connected by the following options of horizontal movement: using the top cross aisle between i and $i+1$ twice, using the bottom cross aisle twice, using both cross aisles once, using both cross aisles twice, or using none of them. Six options for traversing a picking aisle i are possible: traversing a picking aisle completely once or twice, not entering a picking aisle at all, entering and leaving a picking aisle from the top and turning at the bottom-most picking position, entering and leaving a picking aisle from the bottom and turning at the top-most picking position, or entering and leaving a picking aisle from both cross aisles creating a gap between the two consecutive picking positions with the largest distance. The optimal picking tour is represented by a PTS that includes the right-most picking aisle of the warehouse, with one out of four possible states of a feasible tour with the shortest tour length, i.e., the smallest tour costs.

Roodbergen and de Koster (2001b) extend the DP of Ratliff and Rosenthal (1983) to the *2B-TSP*, in which there is an additional middle cross aisle (see Fig. 1(b)). Parts of picking aisles that are separated by the middle cross aisles are referred to as subaisles. The extended case is still solvable in polynomial time with 25 possible states of PTSs. Transitions between states are first applied to include a bottom subaisle, then the top subaisle, before adding possible transitions to the next aisle. The six options for vertical traversal as introduced in Ratliff and Rosenthal (1983) are now applied to subaisles. For horizontal movement, 14 possible configurations are considered due to the additional middle cross aisle.

Based on the results of Cambazard and Catusse (2018), Pansart et al. (2018) introduce a DP for the even more general case of *MB-TSP* with an arbitrary number of blocks. The runtime complexity is in $\mathcal{O}(hv7^h)$, which is still polynomial if the number of cross aisles h is bounded by a constant. In real-world warehouses, the number of cross aisles rarely exceeds five cross aisles because their positive effect of providing additional shortcuts to the neighboring aisles comes at the price of wasted space that is not available for product storage. Similarly to the above-introduced DPs, the algorithm of Cambazard and Catusse (2018) builds PTSs by applying vertical and horizontal transitions to states, processing the cross aisles from bottom to top and the picking aisles from left to right.

CTSP: Löffler et al. (2021) prove that the *1B-CTSP-OCS* is strongly $\mathcal{NP}$-hard. This result also transfers to the more general cases of 2B-CTSP-OCS and MB-CTSP-OCS. Their transformation is from the Hamiltonian path problem (i.e., find a path through a graph that visits all nodes exactly once), in which nodes and edges represent pick lists and aisles that must be accessed by both adjacent pick lists, respectively. If a switch from one pick list to the next proceeds in an aisle which both related pick lists must access (i.e., utilizing an edge in the Hamiltonian path problem), then an aisle visit is saved. Hence, finding a Hamiltonian path in a graph is equivalent to a picking tour saving the maximum number of aisle visits.

Löffler et al. (2021) also show that the *MB-CTSP-GCS* is solvable in polynomial time. This result is also applicable to the more specific cases of the 1B-CTSP-OCS and the 2B-CTSP-OCS. They do so by introducing a nested DP. The outer DP considers all storage positions of the pick lists (in their given order) as potential locations where the switch from one pick list to the next is executed. Each transition of this outer DP thus faces a fixed starting point (i.e., the state of the previous stage that represents the end point of the previous pick list) and end point (i.e., the state of the current stage that represents a potential end point of the current pick list) of a picking path. To determine the cost of such a transition, the inner DP can be used, which is a straightforward extension of the DPs for the 1B-TSP, 2B-TSP, or MB-TSP elaborated in Section 3. The details of these extensions are explicitly elaborated by Löffler et al. (2021) and Masae et al. (2020b) for the path versions of the 1B-TSP and the 2B-TSP, respectively. However, these extensions also directly transfer to the path version of the MB-TSP. Because both DPs, i.e., the inner and the outer, run in polynomial time, the MB-CTSP-GCS, too, can be solved in polynomial time.

GTSP: Weidinger (2018) proves that the *1B-GTSP* is strongly $\mathcal{NP}$-hard. Obviously, this result also transfers to the more general cases of the 2B-GTSP and the MB-GTSP. The transformation is from the hitting set problem (i.e., given a collection of sets with elements from T, find a subset of T of cardinality h such that at least one element of each set is contained). Each set of hitting set is represented by a specific SKU, which is stored in the middle of the picking aisle that is introduced for each element if the set contains this element. In a scattered storage setting, the storage information of each SKU are represented by a set of the hitting set problem. The storage information just contains the picking aisle that contains the concrete storage positions. In this setup, finding a minimum tour for the 1B-GTSP with just h aisle visits is equivalent to finding a YES-instance of the hitting set problem (and vice versa).

TSPB: Žulj et al. (2018) provide a polynomial time algorithm for the *1B-TSPB*, which simplifies the nested DP procedure of Löffler et al. (2021) for the MB-CTSP-GCS and proceeds as follows. Each linehaul node defines a potential storage position, where all linehauls have

Table 2
Results of the TSP database search and subsequent selection process.

Variant	Objective	Multiple tours	Visit all nodes	Multiple visits	Precedence constraints	Resource constraints	Label
Angle TSP	Min turning angles	No	Yes	No	No	No	(1)
Arc replenishment TSP	Min tour cost	No	Yes	No	No	Capacity of salesperson	(2)
Black and White TSP	Min tour cost	No	Yes	No	Yes	No	(2)
Bottleneck TSP	Min max inter-tour cost	No	Yes	No	No	No	(2)
Capacitated prize-collecting TSP	Min tour cost	No	Yes	No	No	Bounds on collected node weights	(2)
Chebyshev TSP	Min tour cost	No	Yes	No	No	No	(1)
Close-enough TSP	Min tour cost	No	Yes	No	No	No	(3)
Clustered TSP	Min tour cost	No	Yes	No	Yes	No	(0)
Clustered TSP with given cluster sequence	Min tour cost	No	Yes	No	Yes	No	(0)
Clustered TSP with d-relaxed priority rule	Min tour cost	No	Yes	No	Yes	No	(2)
Colored balanced TSP	Min diff. betw. min and max edge cost	Yes	Yes	No	No	No	(1)
Colored TSP	Min tour cost	Yes	Yes	No	No	No	(1)
Constant TSP	Min tour cost	No	Yes	No	No	No	(1)
Covering salesman problem	Min tour cost	No	No	No	No	No	(0)
Covering Tour Problem	Min tour cost	No	No	No	No	No	(3)
Deadline TSP	Max number of visited nodes	No	No	No	No	No	(0)
Equality generalized TSP	Min tour cost	No	No	No	No	No	(1)
Family TSP	Min tour cost	No	No	No	No	No	(0)
Film-copy deliverer problem	Min tour cost	No	Yes	k	No	No	(2)
Generalized covering salesman problem	Min tour cost	No	No	Yes	No	No	(2)
Generalized TSP	Min tour cost	No	No	No	No	No	(0)
k-best TSP	Min tour cost	Yes	Yes	No	No	No	(0)
k-collect TSP	Min tour cost	No	Yes	No	Yes	Capacity of salesperson	(2)
k-delivery TSP	Min tour cost	No	Yes	No	Yes	Capacity of salesperson	(3)
k-peripatetic SP	Min tour cost	Yes	Yes	No	No	No	(1)
k-sum TSP	Min sum of k largest inter-tour cost	No	Yes	No	No	No	(2)
Maximum TSP	Max tour cost	No	Yes	No	No	No	(2)
Maximum-scatter TSP	Max min inter-tour cost	No	Yes	No	No	No	(2)
Minmax multiple TSP	Min max tour cost	Yes	Yes	No	No	No	(1)
Moving-target TSP	Min tour cost	No	Yes	No	No	No	(2)
Multiple TSP	Min tour cost	Yes	Yes	No	No	No	(1)
Orienteering Problem	Max node weights	No	No	No	No	Upper bound on tour cost	(0)
Open-loop TSP	Min tour cost	No	Yes	No	No	No	(2)
Period TSP	Min tour cost	Yes	Yes	k	No	No	(1)
Prize-collecting TSP	Min tour cost	No	Yes	No	No	Lower bound on collected node weights	(0)
Probabilistic TSP	Min tour cost	No	No	No	No	No	(1)
Remote TSP	Max min tour cost	No	Yes	No	No	No	(2)
Resource-constrained TSP	Min tour cost	No	Yes	No	No	Upper bound on resource consumption	(2)
Steiner TSP	Min tour cost	No	No	No	No	No	(0)
Stochastic TSP	Min tour cost	No	Yes	No	No	No	(1)
Time-dependent TSP	Min tour cost	No	Yes	No	No	No	(3)
Traveling purchaser problem	Min (tour cost + purchasing cost)	No	No	No	No	No	(2)
Traveling repairman problem	Min latency	No	Yes	No	No	No	(0)
Traveling salesman location problem	Min tour cost	Yes	Yes	No	No	No	(1)
TSP with backhauls	Min tour cost	No	Yes	No	Yes	No	(0)
TSP with delivery and backhauls	Min tour cost	No	Yes	No	Yes	Capacity of salesperson	(2)
TSP with multiple time windows	Min tour cost	No	Yes	No	No	No	(2)
TSP with multiple visits	Min tour cost	No	Yes	≥ 1	No	No	(2)
TSP with pickup and delivery	Min tour cost	No	Yes	No	Yes	Capacity of salesperson	(2)
TSP with precedence constraints	Min tour cost	No	Yes	No	Yes	No	(0)
TSP with release dates	Min tour cost	No	Yes	No	No	No	(2)
TSP with time slots	Min tour cost	No	Yes	No	No	No	(2)
TSP with time windows	Min tour cost	No	Yes	No	No	No	(0)
TSP with time windows and rejections	Min (tour cost + penalty cost)	No	No	No	No	No	(2)
Tunneling TSP	Max tour cost	No	Yes	No	No	No	(1)

been processed and the switch to processing the backhaul nodes can be executed. For each of these possible switch nodes, the shortest Hamiltonian path that starts at the depot, visits all linehaul nodes and ends at the respective switch node can be determined with the polynomial time algorithm of Löffler et al. (2021) for the path version of the 1B-TSP. Then, the tour can be completed by applying the same algorithm to determine a shortest Hamiltonian path that starts at the respective switch node, visits all backhaul nodes, and ends at the depot. The minimum among all possible switch nodes returns the best 1B-TSPB tour. Analogously to the CTSP, to solve the 2B-TSPB, the algorithm of Löffler et al. (2021) for the path version of the 1B-TSP must be substituted by that of Masae et al. (2020b) for the path versions of the 2B-TSP. Recall that the idea of the algorithm of Löffler et al. (2021) directly transfers to the path version of the MB-TSPC, too. Hence, the MB-TSPB is also solvable in polynomial time.

PCTSP: The study provided by Bock and Boysen (2023) considers a special variant of the W-PCTSP that does not allow penalty costs p_i for unvisited nodes i. We dub this variant prize-collecting-no-penalty TSP (PCNPTSP). Hence, the PCTSP introduced in Section 8 is equivalent to the PCNPTSP by setting $p_i = 0, \forall i \in V$. By a reduction of the well-known knapsack problem, Bock and Boysen (2023) prove that the *1B-PCNPTSP* is at least binary $\mathcal{NP}$-hard. Clearly, this result also applies to the more general variants 2B-PCNTSP/2B-PCTSP and MB-PCNPTSP/MB-PCTSP. The reduction maps each element $i \in \{1, \dots, n\}$ of the knapsack problem with weight $\tilde{w}_i$ and price $\tilde{p}_i$ to a node i in the warehouse with profit $w_i = \tilde{p}_i$ that is located on aisle i. Hence, each picking aisle is represented by exactly one node. The chosen position on this vertical aisle guarantees that node i can be visited in the assumed single-block layout from the bottom cross aisle by a cyclical tour requiring $2n \cdot \tilde{w}_i$ time units. As the depot is located on the bottom cross aisle (at the crossing point with vertical aisle 1), the substantial length of the (vertical) aisles prevents that optimal tour schedules travel along the top cross aisle. Furthermore, due to short cross aisles (neighboring picking aisles are separated by one distance unit), a round trip along the entire bottom cross aisle takes $2(n-1)$ time units. Hence, finding a feasible knapsack allocation with a total prize not less than P and with a total weight lower or equal to C is equivalent to the generation of a tour schedule in the mapped warehouse that requires less than $2n \cdot (C+1)$ time units for collecting a total profit of at least $W = P$. After starting from the depot, this schedule performs a cyclical tour along the bottom cross aisle to collect the corresponding prize of node i if and only if element i is allocated to the knapsack.

The complexity status of the variants 1B-PCTSP, 2B-PCTSP, and *MB-PCTSP* is resolved by the following derivation of an exact solution approach that attains an asymptotic pseudo-polynomial runtime. Specifically, we propose an exact best-first branch-and-bound approach for the MB-PCTSP that guarantees an asymptotic pseudo-polynomial runtime if the number of cross aisles is bounded by $h \in \mathcal{O}(\log(n))$ as is

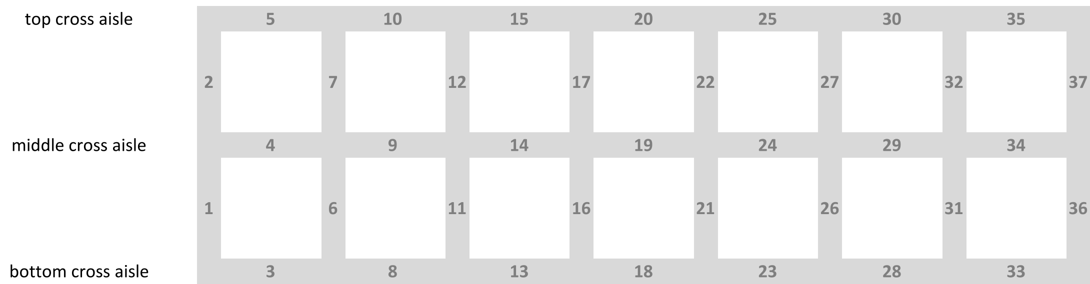

Fig. 2. The indexing of the block-regions for a two-block layout with eight picking aisles.

usually the case in real-world warehouses. For this purpose, we extend the approach by Bock and Boysen (2023) for solving the MB-PCNPTSP to optimality. As the MB-PCTSP additionally includes predetermined penalties for the nodes not visited in a tour, the enumeration of partial tours has to include these penalty costs. Because there is an optimal tour that collects all visited nodes, we can take up the basic idea of the approaches of Ratliff and Rosenthal (1983) and Cambazard and Catusse (2018) to derive an enumeration scheme. Specifically, the enumeration scheme of our branch-and-bound extends partial solutions in an iterative manner by fixing the traversal mode of a block-region (part of a picking aisle between two cross aisles and vice versa), while block-regions are added in a predetermined sequence from bottom to top and from left to right. This sequence is transformed into an indexing of the block-regions that is illustrated by Fig. 2.

As a consequence, each resulting partial solution determines a connected set of block-regions with fixed traversal modes. Because there exists an optimal tour that collects all visited nodes, each chosen traversal mode unambiguously defines the additional travel times and penalties to be paid by determining the nodes visited. Specifically, a traversal mode either visits inner nodes of a block-region from both ends, from one chosen end, or does not visit any inner node. By doing so for some block-regions in the predetermined sequence illustrated by Fig. 2, analogously to Cambazard and Catusse (2018) or Bock and Boysen (2023), the partial tour defines a set of h edge nodes (i.e., crossing points of vertical aisles and cross aisles) that separate the fixed part from the non-explored block-regions. Due to the predetermined sequence of block-regions, there is exactly one edge node per cross aisle, which is why their node degrees and connectivity in the partial tour unambiguously define its extensibility status. Because there is an optimal tour schedule that collects all visited nodes, the number of traversal modes to be considered for a block-region with $\tilde{n}$ open storage positions, can be upper bounded by $\frac{1}{2}\tilde{n}^2 + \frac{3}{2}\tilde{n} + 3$. By additionally dropping dominated modes, the resulting branching degree can be further reduced. As two solutions with identical extensibility status are directly comparable by the number of stored products and the total costs of needed tour length plus paid penalties, the asymptotic number of partial solutions to be explored during the enumeration process is upper bounded by $\mathcal{O}(nh7^hTW)$, with h defining the number of cross aisles, T giving an upper bound of the total tour costs (i.e., total tour length plus penalties of non-visited nodes), and W being the minimum profit to be collected by the tour. Together with a pseudo-polynomial branching time and a polynomial branching degree, the branch-and-bound attains a pseudo-polynomial asymptotic runtime if $h \in \mathcal{O}(\log n)$ applies.

OP: The OP is closely related to the prize-collecting-no-penalty TSP (PCNPTSP) considered by Bock and Boysen (2023). Specifically, the PCNPTSP seeks to generate a tour with minimal length that collects at least a profit of W, whereas the OP pursues to find a tour with a total length not exceeding C that maximizes the total weight of visited nodes. Therefore, we conclude that the decision or feasibility variant of both problems is identical. By considering given threshold values C and W, this problem asks whether a tour in the warehouse exists that does not exceed the maximum length C while collecting a minimum profit of W. Consequently, the complexity results derived by Bock and Boysen (2023) for the PCNPTSP also apply to the OP. By exchanging the roles of tour length and profit (or total weight/capacity) of collected items with each other during the enumeration process, the branch-and-bound approach proposed by Bock and Boysen (2023) for optimally solving the PCNPTSP can be adapted to be also applicable to the OP. The modified branch-and-bound approach maintains a heap of partial solutions that guarantees that the solution with the maximum collected profit is accessible first. In each enumeration step, this solution is taken from the heap to be subsequently extended (branched) to new partial solutions. These solutions have to keep the prescribed tour length threshold C. Consequently, partial solutions that exceed this threshold are deleted after being generated by a branching step. Because this modified enumeration does not change the maximum number of enumerated non-dominated partial solutions derived by Bock and Boysen (2023), we also obtain an asymptotic runtime of $\mathcal{O}(nh7^hTW)$, with T and W being upper bounds on the total tour duration and the total profit that can be collected, respectively. Thus, likewise to the PCNPTSP/PCTSP, the OP is binary $\mathcal{NP}$-hard and can be solved to optimality in asymptotic pseudo-polynomial time if the number of cross aisles is bounded from above by $h \in \mathcal{O}(\log n)$.

TSPTW: The study of Tsitsiklis (1992) analyzes the complexity status of the Line-TSP (and the Line-TRP) under different time window restrictions. The Line-TSP assumes that all nodes to be served and the depot are located along a single line. Therefore, the position of a node i is unambiguously defined by its distance x_i from the origin of the line, whereas travel times between two nodes i and j are proportional to the respective differences $|x_i - x_j|$. Despite its simple transportation network, depending on the assumed time window configurations and delivery times, there are several strongly or binary $\mathcal{NP}$-hard variants (see the overviews provided by Bock, 2015; Tsitsiklis, 1992). Specifically, by a reduction from 3-SAT, Tsitsiklis (1992) shows that the Line-TSP with time windows (Line-TSPTW) is strongly $\mathcal{NP}$-hard. As defined in Section 12, the time windows $[e_i, l_i]$ require that the start of service at node i is only possible at time t_i such that $e_i \le t_i \le l_i$ holds. Because the line structure can be mapped to a single picking aisle, the transportation network of the Line-TSP is obviously a sub-network of a block-structured warehouse. Hence, complexity results derived for the Line-TSPTW may be transferable to the 1B-TSPTW. However, the Line-TSPTW as defined by Tsitsiklis (1992) seeks an open tour; i.e., the final node of the tour is not predetermined and a return to the depot (the starting point of the tour) is not included. We prove the strong $\mathcal{NP}$-hardness of the 1B-TSPTW by the following reduction from the Line-TSPTW: We consider an instance of the Line-TSPTW defined by a tuple (x_i, e_i, l_i) for each node $i \in \{1, \dots, n\}$, a position x_0 of the depot indexed 0 and a time threshold T. The given instance of the Line-TSPTW is feasible, if and only if there exists an open tour with a total duration not exceeding T that feasibly serves all nodes. We map the given Line-TSPTW with identical node positions to a single picking aisle. Moreover, we copy the values e_i and l_i for each node i $(1 \le i \le n)$, while reducing the deadline l_i to T whenever $l_i > T$

applies. If the latter leads to an empty time window; i.e., $e_i > T$, we know that the instance is not feasible. We claim that there exists a cyclic tour in the warehouse for this transformed instance with total cost not exceeding $T + \max\{c_{j,0} \mid 1 \leq j \leq n\}$, if and only if the Line-TSPTW instance is feasible. Clearly, if the given Line-TSPTW is feasible, there exists an open tour that serves all nodes within their time windows while terminating at some node i not later than T. Thus, by adding a return to the depot, i.e., to node 0, we obtain a cyclic tour with total cost $T + c_{i,0} \leq T + \max\{c_{j,0} \mid 1 \leq j \leq n\}$ that serves all nodes in their time windows and not later than T. Conversely, we assume that there is a feasible cyclic tour with total cost not greater than $T + \max\{c_{j,0} \mid 1 \leq j \leq n\}$. Because the found tour is feasible due to the modified deadlines, each node i, $1 \leq i \leq n$, is served at time $t_i \leq T$ such that $e_i \leq t_i \leq l_i$ holds. We consider the node i with the latest service time, i.e., $i = \arg\max\{t_j \mid 1 \leq j \leq n\}$, and erase the travel from i back to the depot. The resulting open tour leads from 0 to i and does not require more than T time units while all nodes are serviced within their time window. Therefore, the original instance of the Line-TSPTW is feasible.

References

Afrati, F., Cosmadakis, S., Papadimitriou, C. H., Papageorgiou, G., & Papakostantinou, N. (1986). The complexity of the travelling repairman problem. *RAIRO - Theoretical Informatics and Applications*, *20*(1), 79–87.

Alsheddy, A. (2017). Solving the free clustered TSP using a memetic algorithm. *International Journal of Advanced Computer Science and Applications*, *8*(8).

Amazon Robotics (2023). Shelf lifting robots. Available online at https://www.aboutamazon.com/news/operations/10-years-of-amazon-robotics-how-robots-help-sort-packages-move-product-and-improve-safety, Last accessed 2023-10-31.

Anily, S., Bramel, J., & Hertz, A. (1999). A 5/3-Approximation algorithm for the clustered traveling salesman tour and path problems. *Operations Research Letters*, *24*, 29–35.

Applegate, D. L., Bixby, R. E., Chvátal, V., & Cook, W. J. (2007). The traveling salesman problem: A computational study. In *Princeton series in applied mathematics*, Princeton University Press.

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.

Ascheuer, N., Fischetti, M., & Grötschel, M. (2001). Solving the asymmetric travelling salesman problem with time windows by branch-and-cut. *Mathematical Programming*, *90*(3), 475–506.

AutoStore (2023). Warehouse robots. Available online at https://www.autostoresystem.com/, Last accessed 2023-10-31.

Azadeh, K., de Koster, R., & Roy, D. (2019). Robotized and automated warehouse systems: Review and recent developments. *Transportation Science*, *53*(4), 917–945.

Balas, E. (1989). The prize collecting traveling salesman problem. *Networks*, *19*(6), 621–636.

Balas, E., Fischetti, M., & Pulleyblank, W. R. (1995). The precedence-constrained asymmetric traveling salesman polytope. *Mathematical Programming*, *68*(1–3), 241–265.

Baldacci, R., Mingozzi, A., & Roberti, R. (2012). New state-space relaxations for solving the traveling salesman problem with time windows. *INFORMS Journal on Computing*, *24*(3), 356–371.

Bansal, N., Blum, A., Chawla, S., & Meyerson, A. (2004). Approximation algorithms for deadline-TSP and vehicle routing with time-windows. In *STOC '04: proceedings of the thirty-sixth annual ACM symposium on theory of computing* (pp. 166–174). Association for Computing Machinery.

Ben-Arieh, D., Gutin, G., Penn, M., Yeo, A., & Zverovitch, A. (2003). Transformations of generalized ATSP into ATSP. *Operations Research Letters*, *31*(5), 357–365.

Bernardino, R., & Paias, A. (2022). The family traveling salesman problem with incompatibility constraints. *Networks*, *79*(1), 47–82.

Bock, S. (2015). Solving the traveling repairman problem on a line with general processing times and deadlines. *European Journal of Operational Research*, *244*(3), 690–703.

Bock, S., & Boysen, N. (2023). Routing replenishment workers: The prize collecting traveling salesman problem in scattered storage warehouses. *INFORMS Journal on Computing*, *36*(1), 3–20.

Boland, N., Hewitt, M., Vu, D. M., & Savelsbergh, M. (2017). Solving the traveling salesman problem with time windows through dynamically generated time-expanded networks. In *Integration of AI and OR techniques in constraint programming: CPAIOR 2017* (pp. 254–262). Springer.

Boysen, N., de Koster, R., & Füßler, D. (2021). The forgotten sons: Warehousing systems for brick-and-mortar retail chains. *European Journal of Operational Research*, *288*(2), 361–381.

Boysen, N., de Koster, R., & Weidinger, F. (2019). Warehousing in the e-commerce era: A survey. *European Journal of Operational Research*, *277*(2), 396–411.

Bulhões, T., Sadykov, R., & Uchoa, E. (2018). A branch-and-price algorithm for the minimum latency problem. *Computers & Operations Research*, *93*, 66–78.

Çelk, M., & Süral, H. (2014). Order picking under random and turnover-based storage policies in fishbone aisle warehouses. *IIE Transactions*, *46*(3), 283–300.

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.

Chen, W., de Koster, R., & Gong, Y. (2022). Analysis and design of rack-climbing robotic storage and retrieval systems. *Transportation Science*, *56*(6), 1409–1722.

Chisman, J. A. (1975). The clustered traveling salesman problem. *Computers & Operations Research*, *2*(2), 115–119.

Clímaco, G., Simonetti, L., & Rosseti, I. (2021). A branch-and-cut and MIP-based heuristics for the prize-collecting travelling salesman problem. *RAIRO - Operations Research*, *55*, 719–726.

Correll, N., Bekris, K. E., Berenson, D., Brock, O., Causo, A., Hauser, K., Okada, K., Rodriguez, A., Romano, J. M., & Wurman, P. R. (2016). Analysis and observations from the first Amazon picking challenge. *IEEE Transactions on Automation Science and Engineering*, *15*(1), 172–188.

Current, J. R., & Schilling, D. A. (1989). The covering salesman problem. *Transportation Science*, *23*(3), 208–213.

da Silva, R. F., & Urrutia, S. (2010). A general VNS heuristic for the traveling salesman problem with time windows. *Discrete Optimization*, *7*(4), 203–211.

Daniels, R. L., Rummel, J. L., & Schantz, R. (1998). A model for warehouse order picking. *European Journal of Operational Research*, *105*(1), 1–17.

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

Escudero, L. (1988). An inexact algorithm for the sequential ordering problem. *European Journal of Operational Research*, *37*(2), 236–249.

Feillet, D., Dejax, P., & Gendreau, M. (2005). Traveling salesman problems with profits. *Transportation Science*, *39*(2), 188–205.

Fischetti, M., González, J. J. S., & Toth, P. (1998). Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing*, *10*(2), 133–148.

Fischetti, M., Lodi, A., & Toth, P. (2003). Solving real-world ATSP instances by branch-and-cut. In G. Goos, J. Hartmanis, J. van Leeuwen, M. Jünger, G. Reinelt, & G. Rinaldi (Eds.), *Lecture notes in computer science*: *vol. 2570*, *Combinatorial optimization — eureka, you shrink!* (pp. 64–77). Springer.

Fischetti, M., Salazar González, J. J., & Toth, P. (1997). A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. *Operations Research*, *45*(3), 378–394.

Fragapane, G., de Koster, R., Sgarbossa, F., & Strandhagen, J. O. (2021). Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. *European Journal of Operational Research*, *294*(2), 405–426.

Frazelle, E. H. (2001). *World-class warehousing and material handling.* Logistics Management Library.

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

Gendreau, M., Hertz, A., & Laporte, G. (1996). The traveling salesman problem with backhauls. *Computers & Operations Research*, *23*(5), 501–508.

Gendreau, M., Laporte, G., & Hertz, A. (1997). An approximation algorithm for the traveling salesman problem with backhauls. *Operations Research*, *45*(4), 639–641.

Glock, C. H., Grosse, E. H., Elbert, R. M., & Franzke, T. (2017). Maverick picking: The impact of modifications in work schedules on manual order picking processes. *International Journal of Production Research*, *55*(21), 6344–6360.

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

Golden, B., Levy, L., & Vohra, R. (1987). The orienteering problem. *Naval Research Logistics*, *34*(3), 307–318.

Golden, B., Naji-Azimi, Z., Raghavan, S., Salari, M., & Toth, P. (2012). The generalized covering salesman problem. *INFORMS Journal on Computing*, *24*(4), 534–553.

Gouveia, L., & Ruthmair, M. (2015). Load-dependent and precedence-based models for pickup and delivery problems. *Computers & Operations Research*, *63*, 56–71.

Gue, K. R., & Meller, R. D. (2009). Aisle configurations for unit-load warehouses. *IIE Transactions*, *41*(3), 171–182.

Gunawan, A., Lau, H. C., & Vansteenwegen, P. (2016). Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, *255*(2), 315–332.

Gutin, G., & Punnen, A. P. (Eds.), (2007). The traveling salesman problem and its variations. In *Combinatorial optimization*: *vol. 12*, Springer.

Helsgaun, K. (2015). Solving the equality generalized traveling salesman problem using the Lin–Kernighan–Helsgaun Algorithm. *Mathematical Programming Computation*, *7*(3), 269–287.

Helsgaun, K. (2017). An extension of the Lin-Kernighan-Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems. *Technical Report*, (pp. 1–60). Denmark: Department of Computer Science, Roskilde University.

Hernández-Pérez, H., & Salazar-González, J.-J. (2009). The multi-commodity one-to-one pickup-and-delivery traveling salesman problem. *European Journal of Operational Research*, *196*(3), 987–995.

Heßler, K., & Irnich, S. (2022). A note on the linearity of Ratliff and Rosenthal's algorithm for optimal picker routing. *Operations Research Letters*, *50*(2), 155–159.

Jongens, K., & Volgenant, T. (1985). The symmetric clustered traveling salesman problem. *European Journal of Operational Research*, *19*(1), 68–75.

Karabulut, K., & Tasgetiren, M. F. (2014). A variable iterated greedy algorithm for the traveling salesman problem with time windows. *Information Sciences*, *279*, 383–395.

Laporte, G. (2010). A concise guide to the traveling salesman problem. *Journal of the Operational Research Society*, *61*(1), 35–40.

Laporte, G., & Martello, S. (1990). The selective travelling salesman problem. *Discrete Applied Mathematics*, *26*(2–3), 193–207.

Laporte, G., & Nobert, Y. (1983). Generalized travelling salesman problem through n sets of nodes: An integer programming approach. *INFOR. Information Systems and Operational Research*, *21*(1), 61–75.

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.

Libralesso, L., Bouhassoun, A.-M., Cambazard, H., & Jost, V. (2020). Tree search for the sequential ordering problem. *vol. 325*, In *ECAI 2020, frontiers in artificial intelligence and applications* (pp. 459–465).

Löffler, M., Boysen, N., & Schneider, M. (2021). Picker routing in AGV-assisted order picking systems. *INFORMS Journal on Computing*, *34*(1), 440–462.

Löffler, M., Boysen, N., & Schneider, M. (2023). Human-robot cooperation: Coordinating autonomous mobile robots and human order pickers. *Transportation Science*, *57*(4), 979–998.

Löffler, M., Schneider, M., & Žulj, I. (2023). Cost-neutral reduction of infection risk in picker-to-parts warehousing systems. *OR Spectrum*, *45*, 151–179.

Lokin, F. (1979). Procedures for travelling salesman problems with additional constraints. *European Journal of Operational Research*, *3*(2), 135–141.

López-Ibáñez, M., & Blum, C. (2010). Beam-ACO for the travelling salesman problem with time windows. *Computers & Operations Research*, *37*(9), 1570–1583.

Lu, Y., Benlic, U., & Wu, Q. (2021). A highly effective hybrid evolutionary algorithm for the covering salesman problem. *Information Sciences*, *564*, 144–162.

Magazino (2023). Autonomous mobile robots. Available online at https://www.magazino.eu/, Last accessed 2023-10-31.

Masae, M., Glock, C. H., & Grosse, E. H. (2020). Order picker routing in warehouses: A systematic literature review. *International Journal of Production Economics*, *224*, Article 107564.

Masae, M., Glock, C. H., & Vichitkunakorn, P. (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.

Matusiak, M., de Koster, R., Kroon, L., & Saarinen, J. (2014). A fast simulated annealing method for batching precedence-constrained customer orders in a warehouse. *European Journal of Operational Research*, *236*(3), 968–977.

Mestria, M. (2018). New hybrid heuristic algorithm for the clustered traveling salesman problem. *Computers & Industrial Engineering*, *116*, 1–12.

Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Association for Computing Machinery*, *7*(4), 326–329.

Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, *24*(11), 1097–1100.

Mladenović, N., Todosijević, R., & Urošević, D. (2013). An efficient general variable neighborhood search for large travelling salesman problem with time windows. *Yugoslav Journal of Operations Research*, *23*(1), 19–30.

Morán-Mirabal, L., González-Velarde, J., & Resende, M. G. (2014). Randomized heuristics for the family traveling salesperson problem. *International Transactions in Operational Research*, *21*(1), 41–57.

Morenza-Cinos, M., Casamayor-Pujol, V., & Pous, R. (2019). Stock visibility for retail using an RFID robot. *International Journal of Physical Distribution and Logistics Management*, *49*, 1020–1042.

Noon, C. E., & Bean, J. C. (1991). A Lagrangian based approach for the asymmetric generalized traveling salesman problem. *Operations Research*, *39*(4), 623–632.

Noon, C. E., & Bean, J. C. (1993). An efficient transformation of the generalized traveling salesman problem. *INFOR. Information Systems and Operational Research*, *31*(1), 39–44.

Öztürkoğlu, Ö., Gue, K. R., & Meller, R. D. (2012). Optimal unit-load warehouse designs for single-command operations. *IIE Transactions*, *44*(6), 459–475.

Pansart, L., Catusse, N., & Cambazard, H. (2018). Exact algorithms for the order picking problem. *Computers & Operations Research*, *100*, 117–127.

Pantuza, G., & de Souza, M. C. (2022). Formulations and a Lagrangian relaxation approach for the prize collecting traveling salesman problem. *International Transactions in Operational Research*, *29*(2), 729–759.

Petersen, C. G. (1997). An evaluation of order picking routing policies. *International Journal of Operations & Production Management*, *17*(11), 1098–1111.

Pop, P. C., Cosma, O., Sabo, C., & Sitar, C. P. (2024). A comprehensive survey on the generalized traveling salesman problem. *European Journal of Operational Research*, *314*(3), 819–835.

Potvin, J.-Y., & Guertin, F. (1998). A genetic algorithm for the clustered traveling salesman problem with a prespecified order on the clusters. In D. L. Woodruff (Ed.), *Operations research/computer science interfaces series*, *Advances in computational and stochastic optimization, logic programming, and heuristic search* (pp. 287–299). Springer.

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.

Rego, C., Gamboa, D., Glover, F., & Osterman, C. (2011). Traveling salesman problem heuristics: Leading methods, implementations and latest advances. *European Journal of Operational Research*, *211*(3), 427–441.

Roberti, R., & Toth, P. (2012). Models and algorithms for the asymmetric traveling salesman problem: An experimental comparison. *EURO Journal on Transportation and Logistics*, *1*, 113–133.

Rodríguez-Pereira, J., Fernández, E., Laporte, G., Benavent, E., & Martínez-Sykora, A. (2019). The steiner traveling salesman problem and its extensions. *European Journal of Operational Research*, *278*(2), 615–628.

Roodbergen, K. J., & de Koster, R. (2001b). Routing order pickers in a warehouse with a middle aisle. *European Journal of Operational Research*, *133*(1), 32–43.

Roodbergen, K. J., & de Koster, R. (2001a). Routing methods for warehouses with multiple cross aisles. *International Journal of Production Research*, *39*(9), 1865–1883.

Salii, Y. V., & Sheka, A. S. (2020). Improving dynamic programming for travelling salesman with precedence constraints: Parallel Morin–Marsten bounding. *Optimization Methods & Software*, *36*(6), 1128–1154.

Santini, A. (2019). An adaptive large neighbourhood search algorithm for the orienteering problem. *Expert Systems with Applications*, *123*, 154–167.

Santra, B., & Mukherjee, D. P. (2019). A comprehensive survey on computer vision based approaches for automatic identification of products in retail store. *Image and Vision Computing*, *86*, 45–63.

Savelsbergh, M. W. P. (1985). Local search in routing problems with time windows. *Annals of Operations Research*, *4*(1), 285–305.

Schiffer, M., Boysen, N., Klein, P., Laporte, G., & Pavone, M. (2022). Optimal picking policies in e-commerce warehouses. *Management Science*, *68*(10), 7497–7517.

Schmidt, J., & Irnich, S. (2022). New neighborhoods and an iterated local search algorithm for the generalized traveling salesman problem. *EURO Journal on Computational Optimization*, *10*, Article 100029.

Schrotenboer, A. H., Wruck, S., Roodbergen, K. J., Veenstra, M., & Dijkstra, A. S. (2017). Order picker routing with product returns and interaction delays. *International Journal of Production Research*, *55*(21), 6394–6406.

Silva, M. M., Subramanian, A., Vidal, T., & Ochi, L. S. (2012). A simple and effective metaheuristic for the minimum latency problem. *European Journal of Operational Research*, *221*(3), 513–520.

Skinderowicz, R. (2017). An improved ant colony system for the sequential ordering problem. *Computers & Operations Research*, *86*, 1–17.

Taillard, É. D., & Helsgaun, K. (2019). POPMUSIC for the travelling salesman problem. *European Journal of Operational Research*, *272*(2), 420–429.

Tompkins, J. A., White, J. A., Bozer, Y. A., & Tanchoco, J. M. A. (2010). *Facilities planning*. John Wiley & Sons.

Tsitsiklis, J. N. (1992). Special cases of traveling salesman and repairman problems with time windows. *Networks*, *22*(3), 263–282.

van der Poort, E., Libura, M., Sierksma, G., & van der Veen, J. A. (1999). Solving the k-best traveling salesman problem. *Computers & Operations Research*, *26*(4), 409–425.

van Gils, T., Ramaekers, K., Caris, A., & de Koster, R. (2018). 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.

Vanheusden, S., van Gils, T., Ramaekers, K., Cornelissens, T., & Caris, A. (2023). Practical factors in order picking planning: State-of-the-art classification and review. *International Journal of Production Research*, *61*(6), 2032–2056.

Weidinger, F. (2018). Picker routing in rectangular mixed shelves warehouses. *Computers & Operations Research*, *95*, 139–150.

Weidinger, F., Boysen, N., & Briskorn, D. (2018). Storage assignment with rack-moving mobile robots in KIVA warehouses. *Transportation Science*, *52*(6), 1479–1495.

Weidinger, F., Boysen, N., & Schneider, M. (2019). Picker routing in the mixed-shelves warehouses of e-commerce retailers. *European Journal of Operational Research*, *274*(2), 501–515.

Ye, M., Bartolini, E., & Schneider, M. (2024). A general variable neighborhood search for the traveling salesman problem with time windows under various objectives. *Discrete Applied Mathematics*, *346*, 95–114.

Zang, X., Jiang, L., Ratli, M., & Ding, B. (2022). A parallel variable neighborhood search for solving covering salesman problem. *Optimization Letters*, *16*(1), 175–190.

Zou, B., de Koster, R., Gong, Y., Xu, X., & Shen, G. (2021). Robotic sorting systems: Performance estimation and operating policies analysis. *Transportation Science*, *55*(6), 1430–1455.

Žulj, 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.