Hindawi
Complexity
Volume 2022, Article ID 1382558, 24 pages
https://doi.org/10.1155/2022/1382558

Research Article

A Mixed Integer Linear Formulation and a Grouping League Championship Algorithm for a Multiperiod-Multitrip Order Picking System with Product Replenishment to Minimize Total Tardiness

Morteza Farhadi Sartangi[1], **Ali Husseinzadeh Kashan**[2], **Hassan Haleh**[3] **and Abolfazl Kazemi**[1]

[1]*Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*
[2]*Faculty of Industrial and Systems Engineering, Tarbiat Modares University, Tehran, Iran*
[3]*Industrial Engineering Group, Golpayegan College of Engineering, Isfahan University of Technology, Golpayegan 87717-67498, Iran*

Correspondence should be addressed to Ali Husseinzadeh Kashan; a.kashan@modares.ac.ir

Received 18 July 2022; Revised 8 August 2022; Accepted 17 August 2022; Published 24 September 2022

Academic Editor: Reza Lotfi

Order picking, which is collecting a set of products from different locations in a warehouse, has repeatedly been described as one of the most laborious and time-consuming internal logistic processes. Each order is issued to pick some products located at given locations in the warehouse. In this paper, we consider an order picking problem, in which a number of orders with different delivery due dates are going to be retrieved by a limited number of order pickers in multiperiods such that the total tardiness is minimized. The aim is to determine a retrieval plan in terms of order batching and order picker multitrip routing as decision variables. Besides, products are arrived and replenished at the predetermined locations at different periods. Therefore, products sitting in those locations should be delivered soon to provide empty rooms for replenishment. A mixed integer linear programming formulation is proposed for this new problem. The model is optimally solved for small-size problems. For larger instances, grouping metaheuristic algorithms are proposed based on particle swarm optimization and the league championship algorithm that use group-based operators to generate reasonable batches of orders. Improvement heuristics are designed as well. The performance of the MILP formulation and metaheuristic algorithms is analyzed for different problem instances whose designs are based on real data gathered from an auto parts warehouse. Results indicate that our algorithms can stably solve large instances of the problem in a reasonable time.

1. Introduction

Supply chain management aims to efficiently and effectively handle all logistics functions and business activities between companies, as well as meet important supply chain goals, including decreasing costs, improving customer satisfaction, optimal usage of resources, and expanding income/profit and production value. Therefore, the objectives of warehouses and distribution centers are important because they are one of the main parts of the supply chain [1, 2]. Warehouses are required to improve the ability of advanced logistics to meet the responsiveness expected by customers [3]. On the other hand, warehouses are important components of logistics systems, and among the logistic costs, 20% of the total operational costs are due to warehouse operations [4, 5]. The main activities related to a warehouse are purchasing, inventory and storage management, picking, distribution processing, shipping, and delivery. Among these activities, picking is considered as one of the basic operations with the highest labor cost [6]. The decision happens in the picking system at three levels, such as strategic, tactical, and operational [5, 7]. Order picking deals with the retrieval of products from their storage locations to respond to customer requests [8]. Furthermore, it is often analyzed at the operational level, and the main issues at this level are batching, routing, workforce level, workforce allocation, and job assignment. Therefore, the development of efficient picking methods and the optimization of picking operations have special effects on the overall operational efficiency of the warehouse. Most of the order picking operations are practically operated according to the picker-to-parts principle and with the most share of manual work, mainly because humans can more simply react to changes happening in the order picking process compared to automated machines due to their cognitive and motor skills [9]. Two aspects are important in order picking, namely, storage optimization (for picking) and picking optimization, which occurs by responding to customer orders by simulating batching routes and intended times in this regard.

Nowadays, researchers seek to combine the operational-level activities of the warehouse with each other and other levels [7]. On the other hand, maintenance is an important subject in the production industry. Most papers assume pickers are available at all times while this action does not happen. Pickers during planning periods for various factors such as preventive maintenance, random failure, and vacation are unavailable for working. Therefore, the possibility of picker unavailability should be considered during the planning. This fact constitutes one of our assumptions when describing the intended problem.

The warehouse operational planning should be in such a way that it considers the products' entrance and leave the warehouse over time. Thus, the plan is a kind of a multiperiod type in practice. To our best knowledge, this issue, which frequently happens in warehouses, has not been investigated in the literature. In this research, we investigate the multiperiod planning of an order picking system, in which each order can be picked up in one of the operating periods, and products should be delivered by their due date. Warehouses should have replenishment capability and a holding place for the delivered orders that would be filled with other orders as time goes by. Accordingly, for the first time, this study pays attention to the replenishment issue in the field of operational decision-making in order-picking systems (OPSs).

The problem has an application in auto parts warehouses, for which

(i) Multitrip picker routing is enforced due to the limitation in the number of pickers;

(ii) Multiperiod planning makes sense due to the existence of lag times (in days) to pick, pack, and deliver the orders;

(iii) Shelve management is a crucial task because the delivery timeline of the orders should be in such a way that it provides shelve space for the arriving products;

(iv) Order management should be in such a way that it prohibits the assembly line stops and the cost imposed. For such a system, tardiness-based objective functions make more sense than cost-based objective functions.

A new mathematical model is developed for the OPS problem under multiperiod and multitrip assumptions, with the possibility of product replenishment (MPMTR). To solve larger-scale problems, grouping metaheuristic algorithms are proposed based on the problem structure, and their effectiveness is investigated using numerical experiments. Two metaheuristic algorithms, namely, the league championship algorithm (LCA) [10–18] and particle swarm optimization (PSO), are heavily modified and applied to the problem. Then, the solutions of these methods are compared with the solutions obtained using Gams/Cplex software for small-, medium-, and large-scale problems in terms of time and optimality.

The contribution of this research can be bulletized as follows:

(i) Integration of the order batching and multitrip picker routing decisions with the possibility of product replenishment in a multiperiod order picking system to minimize total order tardiness. The notion of product replenishment planning in a multiperiod order picking system is introduced for the first time in the current study. Besides, minimizing total order tardiness has not been used in the previous research carried out on OPS planning.

(ii) Developing a mixed integer linear programming (MILP) formulation for the problem via introducing new constraints enforced by the notion of the multiperiod order retrieval and replenishment planning and the new objective function, i.e., the total order tardiness.

(iii) Adapting the particle swarm optimization (PSO) and league championship algorithms (LCA) and equipping them with constructive heuristic to generate feasible solutions. These algorithms have been heavily adapted to be responsive to the structure of the grouping problems to which the problem under consideration belongs.

The remaining sections of the paper are organized as follows: Section 2 reviews the related research. The MPMTR problem is described and mathematically formulated in Section 3. The proposed metaheuristic algorithms are presented in Section 4. Section 5 is devoted to evaluating the efficiency of the algorithms and computational experiments. Conclusions and future research directions are provided in Section 6.

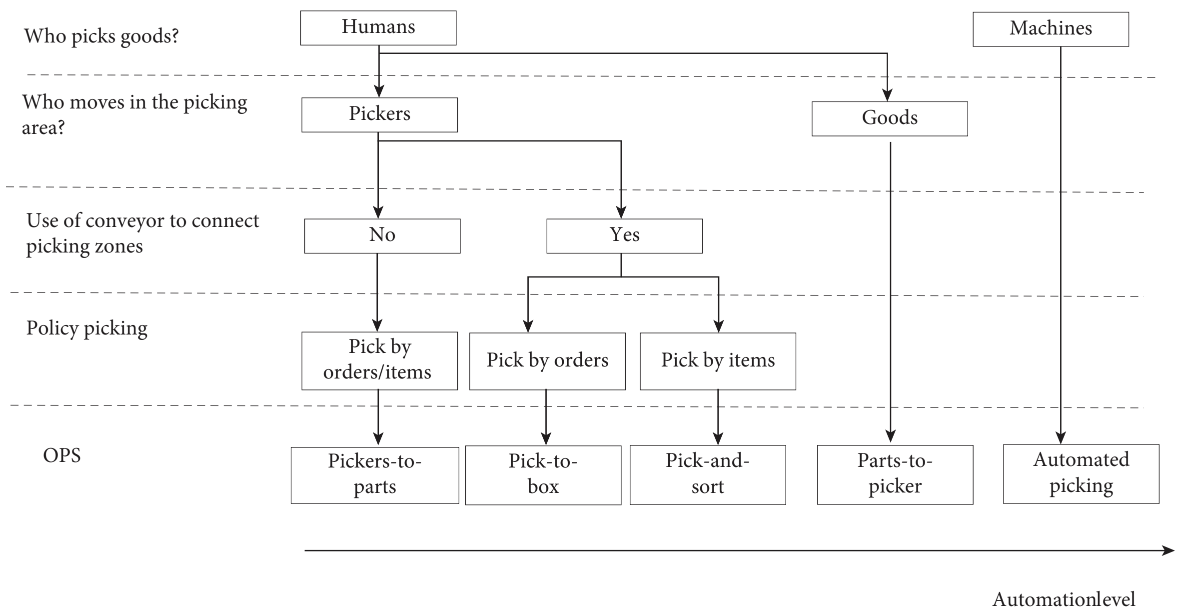
2. Literature Review

2.1. Strategic, Tactical, and Operational Decision in OPSs. Supply chain management consists of several key concepts such as raw materials, suppliers, production, warehousing, transportation, and correctly identifying its boundaries for efficient management, which is a necessary stage. Warehouse management is an initial matter for logistics companies. Therefore, warehouses are one of the main parts of the supply chain. Various decisions help obtain better yield in a warehouse and are divided into three strategic, tactical, and operational levels [7]. In strategic decisions, the focus should be on the layout of the warehouse, the positioning of every zone, the storage, and policy picking. Tactical decisions can consist of the location of products based on their predicted demands in the storage and picking zone. At the operational level, the batches of orders and order-picking routes must be solved optically [7, 19]. In the literature, it talks more about two kinds of warehouses: manual and automated warehouses [20]. The manual warehouse is of superiority worldwide because it requires low cost for investment. For example, nearly 80% of all OPSs in western Europe are manual types [5, 20–22]. Order picking is the mechanism for recovering order items from their storage locations to accomplish customer orders [5]. It has been named as one of the most labor- and time-comprehensive actions in warehouses, typically consisting of more than 50% of the operating costs of warehouses [21–23]. The OPS can be classified into five basic categories by empirical research, and De Koster et al. [5] accepted this issue with a different view. Figure 1 shows a categorization path map of this system [5, 24].

In manual order picking, various subjects exist, including batching, routing, sequencing, congestion, and different layouts for the warehouse. For example, there are high- and low-level warehouses [4, 5, 7]. Thus, this study mainly aims to analyze problems that are related to picking optimization and to combine operational-level decisions for replying optimization to customer demands in manual order picking.

2.2. Decision in Low-Level Picker-to-Parts. Therefore, the rest of the literature review consists of the main subjects in order-picking decisions in low-level picker-to-parts because investigators have approximated that nearly 80% of all order-picking warehouses are managed by a human [5, 22]. Petersen [25] investigated six heuristics strategies for routing pickers in a warehouse, including transversal strategy, return strategy, midpoint strategy, the largest gap, composite, and optimal routing. Optimal routing is the best strategy for routing. Petersen and Schmenner [26] indicated that order picking has a vital role in the supply chain, and warehousing has the highest cost in this process. They analyzed the interaction between order-picking policies (OPP) and storage assignment strategies. Petersen and Aase [27] evaluated three main actions in the warehouse (e.g., picking, routing, and storage policies) and found that the batching of orders results in the best savings, particularly when smaller order sizes are common. Various criteria affect the optimal decision in the warehouse, including the type, size, number of depot locations, order picking equipment, picklist size, and the storage rules of the warehouse [27, 28]. The layout is one of the basic factors that affect order picking. Jan and De Koster [28] first attempted to analyze the order routing picker problem in a three-aisled warehouse and inform a dynamic programming algorithm for solving the order-picking tour of minimal length. Most studies in picker-to-parts regarding the layout have focused on low-level picking, although some efforts exist concerning high-level picking. For example, Chabot et al. [29] defined a mathematical model to respond to customer orders in a real three-dimensional warehouse and solve this capacity vehicle routing problem using large neighborhood search along with Branch and Bound algorithms, and then compared these algorithms against each other to pick orders in that company.

2.3. Integrated Decisions at the Operational Level in the Warehouse. Wäscher [30] considered picker-to-parts systems for reducing costs in the warehouse and analyzed operational decisions such as item location, order batching, and picker routing, especially in order and batching. He presented some solutions in this regard (e.g., priority rule-based, seed, and savings algorithms). Traditional warehousing focuses on improving efficiency operational decisions within the warehouse separately, while Won and Olafsson [31] considered both issues of batching and picking for optimally answering customer requests and showed these issues use the bin-packing problem (BPP) and the TSP since they are both NP-hard problems, thus order picking is an NP-hard problem. There are some efforts to separately attend to the operational-level decisions. For instance, Ho and Tseng [32] presented some rules such as seed- and accompanying-order selection for batching by constant routing policies. Moreover, Tsai et al. [23] used the multiple-genetic algorithm (GA) method consisting of GA_BATCH and GA travelling salesman problem (TSP) algorithms. The optimal batch picking system is established by considering travel distance costs and the earliness and tardiness penalty. Similarly, Henn et al. [33] introduced two metaheuristic approaches for solving order batching, including iterated local search and ant colony optimization. They reached optimal batching by paying attention to the minimum length of the tour, so the focus was on batching rather than routing. Likewise, Henn et al. [34] especially analyzed order batching as one main action in order picking and reviewed some solutions among metaheuristic algorithms (local and tabu search and population-based approaches). They further studied dynamic (time window batching) and static (due dates) batching. Kulak et al. [35] also integrated order batching and routing for multi-aisle warehouses (rectangular, two-block, and low level) and used a novel tabu search algorithm integrated with a novel clustering algorithm to solve this joint at the operational level of the warehouse. Pan et al. [36] reported that most studies refer to warehouses with one picker, while one encounters several pickers in real issues, creating the issue of congestion, which raises the



Picker-to-parts: pickers move around the warehouse to pick the items.

Parts-to-picker:an automated device brings loads to pickers that are in charge of picking the right quantity required by the orde

Picker-to-box: pickers are assigned to different zones and boxes containing customer orders are movedby means of a conveyor through different zones in order to be filled.

Pick and sort: orders are batched and pickers collect items of a certain product to satisfy theorders of a batch, then a conveyor brings all items to the sorting area wherethe orders are formed.

Completely automated: Where humans do not intervene in the process of picking.

Figure 1: Categorization of the OPS.

waiting time. Thus, they thought to minimize the waiting and travel time for order picking. To solve an integrated sequencing and batching model, Henn [8] used variable neighborhood descent and variable neighborhood search by focusing on the multipicker in a warehouse. Oncan [37] first introduced mixed-integer linear programming for three routing policies (i.e., traversal, return, and midpoint policies) and solved them by the local search algorithm with the Tabu threshold, the accuracy and correctness of which have been proven previously. Chen et al. [38] provided an integrated nonlinear and mixed-integer linear model for categorizing, batching, sequencing, and routing orders in order-picking problems that solve these three issues simultaneously. To solve this model, a heuristic method was used, including an integrated heuristic algorithm based on genetic (for batching and sequencing) and ant colony for the routing problem. In another study, Scholz and Wäscher [20] evaluated iterated local search for joint batching and routing. Similarly, Valle et al. [39] optimally solved the joint order batching and picker routing problems for small size problems (means the number of orders) by valid inequalities (cut), and it was noticeable that up to 5000 orders, batching was solved by the heuristic method but the routing was solved optimally. Chen et al. [40] are among the ones who attempted to accept the assumption of split-off orders and introduced a nonparametric heuristic method (i.e., green area) for online order batching. Additionally, Menéndez et al. [41] applied variable neighborhood descent for integrating batching and routing. However, Scholz et al. [42] were the first to integrate all decisions at the operational level by composition batching, job assignment, and routing and to use a variable neighborhood descent algorithm for large instances. Žulj et al. [43] also applied tabu search algorithms in combination with an adaptive large neighborhood search. They reported that order batching is a nondeterministic polynomial (NP)-hard problem. Their hybrid algorithm was able to solve large-scale instances. Van Gils et al. [44] first examined and described the relations among the storage, batching, zone picking, and routing planning problems and indicated that warehouses can reach notable yields by simultaneously considering storage, batching, zone picking, and routing decisions. Furthermore, Moons et al. [45] found that integrated order picking by the vehicle routing problem leads to increased service levels (e.g., it allows for decreasing the time between placing an order and receiving the products or goods). To outline the main contributions and approaches of order picking systems with attention to operational decisions, an overview of the operational-level decisions in the OPS; Table 1 summarizes the relevant research studies.

2.4. The Review Research Gap. From the literature review, it can be concluded that most OPSs related researches mainly target a picker-to-parts in the manual OPS in the static picking environment. Moreover, routing methods, limited

Table 1: Summary of the related literature.

Reference	Abbreviation	Case study	Mathematical model	Heterogeneous/ homogenous pickers	Layout	Solution method	Routing	Batching	Job assignment	Multi-period	Limited access to picker	Multi-trip routing	Congestion
Petersen [25]	OPRP	—	—	—	Rectangular, low-level	Statistic (ANOVA)	✓	—	—	—	—	—	—
Ii [46]	OPPP	✓	—	—	Rectangular, low-level	Simulation	—	✓	—	—	—	—	—
Jan and De Kosher [28]	OPRMA	—	—	—	Rectangular, two blocks, low-level	Dynamic programming algorithm	✓	—	—	—	—	—	—
Petersen and Aase [27]	PRSPOP	—	—	—	Rectangular, low-level	Simulation	✓	✓	—	—	—	—	—
Wäscher [30]	OP	—	—	—	Rectangular, low-level	Review a few solutions	✓	✓	—	—	—	—	—
Won and Olafsson [31]	JOBOP	—	IP	—	—	Sequential batching and picking (SBP) algorithm	✓	✓	—	—	—	—	—
Ho and Tseng [32]	OB	—	—	—	Rectangular, low-level	Statistic (ANOVA)	✓	✓	—	—	—	—	—
Tsai et al. [23]	MG-OB	—	IP	Hom	Rectangular, low-level	Multiple-GA	✓	✓	✓	—	—	—	—
Dallari et al. [24]	DOPS	✓	—	—	—	Analytical method	—	—	—	—	—	—	—
Henn et al. [33]	MOBP	—	IP	—	Rectangular, low-level	Iterated local search, ant colony	—	✓	—	—	—	—	—
Theys et al. [47]	TSPHORP	—	—	—	Rectangular, two blocks, low-level	TSP-heuristic	✓	—	—	—	—	—	—
Henn and Schmid [48]	OBSP	—	MIP	—	Rectangular, two blocks, low-level	Iterated local search, attribute-based hill climber	—	✓	✓	—	—	—	—
Kulak et al. [35]	JOBPR	—	IP	—	Rectangular, two blocks, low-level	Tabu search integrated with a novel clustering algorithm	✓	✓	—	—	—	—	—
Pan et al. [36]	SAP-TDABC	—	—	Hom	Rectangular, low-level	Heuristic algorithm	—	—	—	—	—	—	✓
Hong et al. [49]	OPB_PC	—	MIP	Hom	Rectangular, low-level	Simulated annealing algorithm	✓	✓	—	—	—	—	✓

Table 1: Continued.

Reference	Abbreviation	Case study	Mathematical model	Heterogeneous/ homogenous pickers	Layout	Solution method	Routing	Batching	Job assignment	Multi-period	Limited access to picker	Multi-trip routing	Congestion
Henn [8]	OBSPMP	—	MILP	Hom	Rectangular, low-level	Variable neighborhood descent and search algorithms	✓	✓	✓	—	—	—	—
Oncan [37]	OPB	—	MILP	Hom	Rectangular, low-level	Iterated local search algorithm with tabu search	✓	✓	—	—	—	—	—
Chen et al. [38]	IOBSRP	—	MINLP	Het	Rectangular, low-level	Ant colony, genetic algorithm	✓	✓	✓	—	—	—	—
Chabot et al. [29]	CAN-OPP	✓	MILP	—	3D narrow aisle. High-level	Large neighborhood search, branch &bound	—	✓	—	—	—	—	—
Chen et al. [50]	ORP_MC	—	IP	Hom	Rectangular, two blocks, low-level	Online ant colony optimization	✓	—	—	—	—	—	✓
Su & Hwang [51]	OPRP_FS	—	—	—	Rectangular, multi block, low-level	Fuzzy clustering	✓	—	—	—	—	—	—
Valle et al. [39]	OS-_JOBRP	✓	IP	Hom	Rectangular, two blocks, low-level	Exact, heuristic	✓	✓	—	—	—	—	—
Scholz and Wäscher [20]	OBI RP	—	ILP	—	Rectangular, two blocks, low-level	Iterated local search	✓	✓	—	—	—	—	—
Scholz et al. [42]	JOBA SRP	—	MILP	Hom	Rectangular, two blocks, low-level	Variable neighborhood descent algorithm	✓	✓	✓	—	—	—	—
Menéndez et al. [41]	OBP	—	ILP	—	Rectangular, low-level	Variable neighborhood search	✓		—	—	—	—	—
Van Gils et al. [44]	ISBZPRPD	✓	—	—	—	Statistic (ANOVA)	✓	✓	✓	—	—	—	—
Žulj et al. [43]	OBP	—	IP	—	—	Adaptive large neighborhood search and tabu search	—	✓	—	—	—	—	—
Chen et al. [40]	OOBA	—	ILP	Hom	Rectangular, two blocks, low-level	Nonparametric heuristic method, green area	—	—	—	—	—	—	—
Moons et al. [45]	I-OP-VRP	—	MILP	Het	—	Record-to-record travel algorithm	✓	✓	—	—	—	—	—

Table 1: Continued.

Reference	Abbreviation	Case study	Mathematical model	Heterogeneous/ homogenous pickers	Layout	Solution method	Routing	Batching	Job assignment	Multi-period	Limited access to picker	Multi-trip routing	Congestion
Kuhn et al. [52]	IOBVRPGR	✓	—	—	Rectangular, three block, low-level	General adaptive large neighborhood search (GALNS)	✓	✓	—	—	—	—	—
Briant et al. [19]	JOBPR	✓	—	—	Rectangular, two blocks, low-level	Column generation	✓	—	—	—	—	—	—
The current study	MPMTR	✓	MILP	Het	Rectangular, two blocks, low-level	Grouping algorithms	✓	✓	—	✓	✓	✓	—

OPRP: order picking routing policies OPPP: order picking policies problem Oprma: order picker routing with middle aisle PRSPOP: picking routing storage policies order picking OP: order picking Jobop: joint order batching and order picking OB: order batching MG-OB: a multiple-GA method to solve order batching DOPS: design of order picking system MOBP: metaheuristic for order batching problem TSPHORP: travel salesman problem heuristic for order routing problem OBSP: order batching sequencing problem JOBPR: joint order batching and picker routing SAP-TDABC: storage assignment problem with travel distance and blocking congestion OPB_PC: order picking batching _ picker congestion OBSPMP: order batching sequencing problem with multiple pickers Iobsrp: integrated order batching, sequencing, and routing problem CAN-OPP: capacity narrow aisle order picking problem ORP_MC: order routing problem_ multi picker congestion OPRP_FS: order picking routing problem-fuzzy set OS-_JOBRP: optimally solve_ joint order batching and routing problem OBI RP: order batching and integrated routing problem JOBA SRP: the joint order batching, assignment, sequencing and routing problem ISBZPRPD: integrating storage batching, zone picking, and routing policy decisions OOBA: online order batching and assigning I-OP-VRP: integrated order picking-vehicle routing problem MPMTR: multi-period, multi-trip assumptions with the possibility of product replenishment IP: integer programming MIP: mixed integer programming NMLIP: nonlinear mixed integer programming.

capacity of pickers, and, recently, joint order batching with routing are commonly assumed in these studies. Few studies have addressed nontypical assumptions/constraints in joint order batching and routing. Among these assumptions is the allowance of multitrips for pickers. However, none of them has assumed multiperiod picking or product replenishment at an integrated operational level. Accordingly, the current study attempts to contribute to the following issues:

(i) Unlike most researchers that assume an unlimited number of pickers, the case of limited pickers, which is realistic for auto parts warehouses considered here, enforces multitrip picker routing.

(ii) Multiperiod planning of order picking systems has not been addressed seriously in the literature. This is where, in auto part warehouses, there would be a lag time (in days) to pick, pack, and deliver the orders.

(iii) When the multiperiod planning comes into practice, delivery due dates of the orders become crucial. Specially, auto parts order management should be in such a way that it prohibits the assembly line stops and the cost imposed. For such a system, tardiness-based objective functions make more sense than cost-based objective functions. In this sense, minimizing total order tardiness is assumed rather than minimizing total costs, which has been considered in the majority of research studies.

(iv) There is no research effort to consider product replenishment in multiperiod planning. When orders are delivered, product shelves become empty and ready to be nested by new products that constitute future orders. Shelve management is a crucial task in warehousing because the delivery timeline of the orders should be in such a way that it provides shelve spaces for the arriving products. This issue is addressed as a product replenishment and is considered for the first time in the order picking planning in this research.

We propose a mathematical model and algorithms for solving the joint order batching and order routing problem (referred to as order picking in general) under the above considerations. Given the computational complexity of the problem, which is forced by the combination of various subproblems, we propose two metaheuristic algorithms that do not only a simple searching mechanism but also fit the structure of the problem via employing heuristic methods for the generation of the new solutions.

3. Problem Description and Formulation

As shown in Figure 2, most studies in the literature on the manual picker-to-part warehouse with a narrow pick aisle have used a special layout [5]. This layout is commonly investigated in the related literature and practically used by changing the number of blocks, the length and width of aisles, cross aisles, depots, and low-and high-level picking, e.g., [19, 29, 37, 39–41, 48, 51]. The warehouse layout considered in this research is the one typically used in the

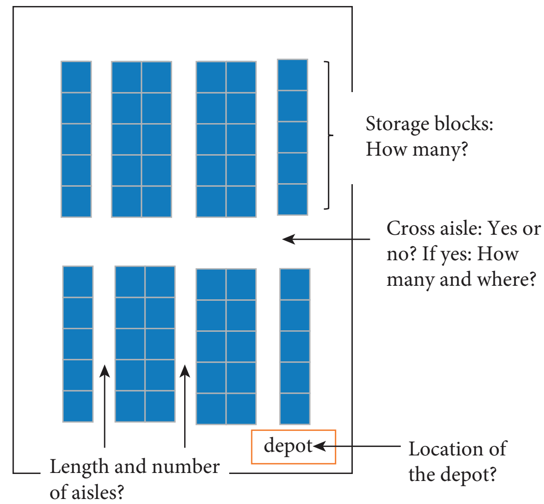


Figure 2: The layout used in manual order picking systems. Source: De Koster et al. [5].

auto parts industry. It has been demonstrated in Figure 6. Order pickers can traverse pick aisles in both directions and can change the orientation inside the pick aisles. The warehouse is available in cellular form, and all orders reaching the warehouse are planned for taking place in a storage position. Pickers use batching policies for order picking.

Every order constitutes delivery of parts/products pallets, each of which has a certain number and specified weight. Each picker b picker has a unique capacity. Identical pickers have an identical weight capacity. It is assumed that there are multipickers that are available at certain time intervals within the planning period. Orders have a due date on which their inclusive products should be delivered. Any delivery delay is counted in the objective function, which is minimizing total order tardiness. At the same time, new products that will arrive are stored in their planned location. Therefore, the order delivery planning should be in such a way that there exists empty room for the arrived product. In other words, the product sitting in the same place should be delivered as soon as possible, such that their location is being replenished with the arrived products. Pickers can provide service multitimes (as a multitrip) when they are present in the warehouse. Thus, the problem is considered as a multiperiod-multitrip order picking problem. This problem integrates order batching and picker multitrip routing with additional constraints on picker availability, picker capacity, and product replenishment.

3.1. Assumptions. The following conditions should be considered in the model:

(i) The objective function is minimizing total tardiness of all picked orders.

(ii) Each order is a parts pallet whose weight may differ from other pallets.

(iii) Order picking planning is of multiperiod type. In auto part warehouses, there would be a lag time (in days) to pick, pack, and deliver the order contents. In this sense, the multiperiod planning is inevitable.

(iv) The number of pickers is limited and given.

(v) The picker capacity is limited.

(vi) There is no continuous access to pickers and they may not be available at some times during the planning period. The availability interval of picker b in period t is [Sbt, Fbt].

(vii) Pickers serve in a multitrip picking system when they are available.

(viii) Warehouse cells are capable of being replenished by new products during the periods.

3.2. Notations and MILP Formulation. The sets, parameters, and decision variables used in the proposed MILP formulation are as follows:

3.2.1. Sets

O Set of orders/products $\left\{o, o_p \epsilon O\right\}$ whit cardinality n.

B Set of pickers $\{\mathrm{b} \epsilon B\}$.

L Set of storage location $\{i, j \epsilon L\}$. $i=1$ indicates the pickers initial position from which they always start their trip for picking.

R Set of trips $\{r, r_p \epsilon R\}$.

T Set of time periods $\{t \epsilon T\}$.

3.2.2. Parameters

n_i The number of times that storage location i is replenished by products.

c_b Picker weight capacity.

p_b Picker capacity in terms of the number of pallets.

w_o Weight of order o.

d_o Delivery due date of order o.

e_{ij} Travel time between location i and j.

s_{bt} Work start time of picker b in period t.

f_{bt} Work finish time of picker b in period t.

B_{bt} Takes value equal to 1 if picker b is available (even partly) in period t. Otherwise, it takes 0.

t_o The first time by which order o arrives to the warehouse.

l_o Location of order/product o.

MA positive integer.

3.2.3. Decision Variables

X_{obtr} Takes value equals to 1 if order o is assigned to picker b in period t in trip r. Otherwise, it takes 0.

Y_{ijbtr} Takes value equal to 1 if picker b in period t traverses from location i to j in trip r. Otherwise, it takes 0.

Z_{ibtr} Takes value equal to 1 if location i is met by picker b at period t in trip r. Otherwise, it takes 0.

R_{btr} Takes value equal to 1 if picker b in period t traverses trip r. Otherwise, it takes 0.

W_{btri} The weight carried by picker b in period t and trip r after leaving location i.

S_{btr} Start time of trip r by picker b in period t.

F_{btr} Finish time of trip r by picker b in period t.

Based on the problem statement and taking the parameters and variables definition, the proposed mathematical formulation of the problem is as follows:

$$\min \sum_{0} \max\left(\sum_{b} \sum_{t, t \geq t_o} \sum_{r} t X_{obtr} - d_o, 0\right). \tag{1}$$

Subject to

$$Z_{ibtr} \leq B_{bt,} \forall i, b, t, r, i \neq 1, \tag{2}$$

$$X_{obtr} \leq Z_{ibtr,} \forall o, b, t, r, i = l_O, t \geq t_o, \tag{3}$$

$$Z_{ibtr} = 0, \forall i, b, t, r, n_i = 0, \tag{4}$$

$$\sum_{b} \sum_{t} \sum_{r} z_{ibtr} = n_{i,} \forall i, i \neq 1, \tag{5}$$

$$\sum_{b} \sum_{t, t \geq t_o} \sum_{r} X_{obtr} = 1, \forall o, \tag{6}$$

$$X_{obtr} = 0, \forall o, b, t, r, t < t_o, \tag{7}$$

$$\sum_{j, j \neq i} Y_{ijbtr} = Z_{ibtr}, \forall b, t, r, i, i \neq 1, \tag{8}$$

$$\sum_{i, i \neq j} Y_{ijbtr} = Z_{jbtr,} \forall b, t, r, j, j \neq 1, \tag{9}$$

$$\begin{aligned} W_{btri} + \sum_{\substack{O, t \geq t_o, \\ j = L_o}} w_O * X_{obtr} \leq W_{btrj} \\ + M\left(1 - Y_{ijbtr}\right), \forall b, t, r, i, j, i \neq j, j \neq 1, \end{aligned} \tag{10}$$

$$W_{btri} \leq c_b * B_{bt,} \forall i, b, t, r, \tag{11}$$

$$\sum_{0} X_{obtr} \leq p_{b,} \forall b, t, r, \tag{12}$$

$$S_{btr} + \sum_{i} \sum_{j} e_{ij} * Y_{ijbtr} \leq F_{btr}, \forall b, t, r, \tag{13}$$

$$S_{btr} \geq F_{bt, r-1}, \forall b, t, r, r \neq 1, \tag{14}$$

$$S_{bt1} = s_{bt,} \forall b, t, \quad (15)$$

$$F_{btr} \le f_{bt,} \forall b, t, r, \quad (16)$$

$$\sum_{b} \sum_{t,t\ge t_o} \sum_{r} t * X_{obtr} \le t_{o_p} - 1, \ \forall i, o, o_P, i = l_o, i = l_{o_p}, t_{o_p} > t_o, \quad (17)$$

$$Z_{ibtr} \le R_{btr,} \forall i, b, t, r, \quad (18)$$

$$R_{btr} \ge R_{btr_{p,}} \forall b, t, r, r_p, r < r_p, \quad (19)$$

$$\sum_{i,i\ne 1} Z_{ibtr} \ge R_{btr,} \forall b, t, r, \quad (20)$$

$$\sum_{b} \sum_{r_p} R_{btr_p} = \sum_{j} \sum_{b} \sum_{r} Y_{1jbtr,} \forall t, \quad (21)$$

$$\sum_{b} \sum_{r_p} R_{btr_p} = \sum_{i} \sum_{b} \sum_{r} Y_{1jbtr,} \forall t, \quad (22)$$

$$X_{obtr}, Y_{ijbtr}, Z_{ibtr}, R_{btr} \in \{0, 1\} W_{btri}, S_{btr}, F_{btr} \ge 0. \quad (23)$$

The objective function in (1) is the total tardiness, which is to be minimized. Constraint (2) indicates that a picker can visit a storage location only when it is available. Constraints (3) and (4) force that when an order is picked, its location should be visited.

Constraint (5) indicates that a location should be visited multiple times in the case of replenishment. $n_i > 1$ indicates that replenishment occurs. Constraint (6) guarantees that all received orders must be picked up. Constraint (7) guarantees the elimination of picking for orders that have not arrived at the warehouse. Constraints (8) and (9) are used to enforce one-time arrival and departure from a storage location in a given trip in a given period by a given picker. Constraint (10) avoids subtours and updates the total weight carried by the picker after leaving an order location. Constraint (11) is the picker's capacity constraint. Constraint (12) is the picker's capacity in terms of the number of orders it can pick in each trip. Constraint sets (13)–(16) are used to plan the picker's operation within its available time. Recall that picker b can be available within [Sbt, Fbt] in a given period t. The start and finish time of each trip is controlled by Constraint sets (13)–(16). Constraint (17) allows multiple replenishment of a storage location. An order should be picked before the arrival of the order, which will take its place. Constraints (18) and (19) arrange the trip counter for each picker in each period. Constraint (20) forces the visit of at least one location during a trip. Constraints (21) and (22) manage the number of departures and arrivals from the pickers' initial location. The statement of variables and their types are provided in (23).

Since the total tardiness as the objective function is a nonlinear function, its linear form can be simply given by replacing (1) with (24) and adding (25) to the set of constraints (2)–(23).

$$\text{Min} \sum_{0} Z_o, \quad (24)$$

$$Z_o \ge \sum_{b} \sum_{t,t\ge t_o} \sum_{r} t X_{obtr} - d_{o,} \forall o. \quad (25)$$

4. The Solution Method

The MPMTR problem consists of two sub-problems of order batching and multitrip routing, both of which are NP-hard problems [6]. Therefore, the MPMTR problem is also classified into the category of NP-hard problems, and finding efficient methods is necessary for solving it.

Only small-size instances of the MPMTR problem can be solved efficiently using the proposed mathematical model, and suitable algorithms are needed for real-world problems. The MPMTR problem resembles grouping problems where a number of orders are up picked in every trip by the picker and grouped into one group. In grouping problems, the aim is to group items into disjoint groups. The structure of grouping problems is such that it admits to developing more effective operators for metaheuristic methods. Because the building blocks that should be preserved in an evolutionary or swarm intelligence method should be the groups or the group segments, focusing on items isolated may have little impact during the search. The most well-adapted grouping evolutionary algorithms for grouping problems are the grouping genetic algorithm and the grouping evolution strategy algorithm [53–56].

Given the grouping nature of the problem, in this study we seek to develop a solution method for solving large-size problems via developing the grouping versions of two well-established algorithms, e.g., particle swarm optimization (PSO) and the league championship algorithm (LCA). These algorithms are called the grouping particle swarm optimization (GPSO) and grouping the league championship algorithm (GLCA) and are described in the next sessions.

We have used the LCA and PSO algorithms in the sense that they have proven effective in many contexts because of the clever rationale behind their mechanisms designed for managing intensification and diversification. The PSO is a classic and popular algorithm for solving optimization problems. To include a relatively new and modern algorithm, we used the LCA. The LCA has proven as an effective algorithm. Therefore, we think it may be useful to compare the results of an older algorithm like the PSO with those of a newer one. However, we heavily adapted these algorithms to be able to solve the problem at hand. Specifically, we developed them such that they are responsive to the structure of grouping problems to which the order picking problem belongs.

4.1. Grouping the PSO and LCA Algorithms. The PSO and the LCA are population-based stochastic optimization techniques inspired by social behaviors. These algorithms were first developed for continuous problems. The similarity of the GPSO/GLCA with the PSO/LCA lies not in the fact that it uses the same idea, followed by the PSO/LCA, but in the discrete search space of the grouping problems. The building blocks of a solution for a grouping problem are the groups to which items are assigned. Therefore, PSO's/LCA's updating equations, which typically work in continuous space, are modified in the GPSO/GLCA to work on groups. It is therefore required to devise suitable mathematical operators which are employed on groups. Before giving the mathematical statements of the GPSO's/GLCA's equations, we first put forward the way in which a solution to the MPMTR problem is represented.

4.1.1. Solution Representation. The first step for developing an algorithm for a discrete problem like the MPMTR is to represent the solution. Since the MPMTR is a grouping problem, we first give our definition of a group structure. For each picker in each period and for each trip, a group is assigned. The content of the group are the orders that the picker in that period and in that trip is responsible for picking them. Figure 3, represents the grouping structure of a sample solution. In this figure, there are two pickers, two periods, and three possible trips. Therefore, there are at most 12 groups. Separators are depicted for periods (orange line) and pickers (green lines). The structure in Figure 3 can be translated in terms of a given solution to the problem as follows: Picker 1 in its first trip in Period 1, picks order 1. He is responsible for picking Orders 4 and 5 in his second trip. The first picker in his first trip picks orders 2 and 3 in Period 2. Orders 5 and 6 are then picked in his second trip. In parallel, Picker 2 picks order 7 in his first trip in Period 2. For the sake of simplicity, we use a threefold notation $t/b/r$ to tag the groups.

The solution represented in Figure 3 only represents the order batching and job assignment parts of the solution but does not say anything about picker routing. To determine the routing for the picker associated to each group, one needs to solve an instance of the travelling salesperson problem (TSP) composed of the location nodes of the orders assigned to the group.

According to the classification of grouping problems demonstrated by Kashan et al. [53]; the MPMTR problem is a problem with non-identical groups. That is, groups differ in their characteristics (different periods, different picker capacity) and if we exchange the whole content of two groups in a given solution, the resultant grouping differs from the original grouping. In terms of the number of groups in a given solution, the MPMTR is a variable grouping problem in which the number of groups is not known in advance and differs between different solutions.

4.1.2. Updating Equations in GPSO and GLCA. Before starting this section, we invite readers to refer to the lots of papers, which introduce the original PSO and LCA algorithms. This section attempts to reform position equations of PSO (equations (27) and (28)) and LCA (equations (31)–(34)) to achieve comparable equations that work with the groups of items rather than scalars. Following Kashan et al. [57, 58], the main idea would be to use suitable operators instead of arithmetic operators. Especially, the "–" operator is substituted with a group dissimilarity measure. Similar to the "–" operator, which quantifies the magnitude of the difference between the two scalars, a group dissimilarity measure quantifies the distance between the two groups. In this study, "distance" is used to show such an operator. If G and G' are two groups with cardinality $|G|$ and $|G'|$, the Jaccard's coefficient is one of the best methods for representing the degree of similarity or difference between G and G'

$$\text{Distance}(G, G') = 1 - \frac{|G \cap G'|}{|G \cup G'|}, \text{Similarity}(G, G') = \frac{|G \cap G'|}{|G \cup G'|}. \tag{26}$$

Given that the original PSO equations are as follows:

$$v_{id}^{t+1} = wv_{id}^{t} + c_1 r_1 \left(p_{id}^{t} - x_{id}^{t}\right) + c_2 r_2 \left(p_{gd}^{t} - x_{id}^{t}\right) \forall i, d, \tag{27}$$

$$x_{id}^{t+1} - x_{id}^{t} = v_{id}^{t+1} \forall i, d. \tag{28}$$

And substituting "–" with "Distance," the updating equations of GPSO are obtained as follows:

$$\begin{aligned} v_{id}^{t+1} &= wv_{id}^{t} + c_1 r_1 \text{Distance}\left(p_{id}^{t}, x_{id}^{t}\right) \\ &\quad + c_2 r_2 \text{Distance}\left(p_{gd}^{t}, x_{id}^{t}\right), \ \forall i, d, \end{aligned} \tag{29}$$

$$\text{Distance}\left(x_{id}^{t+1}, x_{id}^{t}\right) \approx v_{id}^{t+1}, \ \forall i, d. \tag{30}$$

For LCA, given that the original equations are as follows (please refer to [10–12]):

If both i and its opponent have won their matches at week t, then

$$x_{i\,d}^{t+1} - b_{i\,d}^{t} = -\psi_1 r_1 \left(b_{k\,d}^{t} - b_{i\,d}^{t}\right) - \psi_1 r_2 \left(b_{j\,d}^{t} - b_{i\,d}^{t}\right), \ \forall i, d \tag{31}$$

Else if i has won and its opponent has lost, then

$$x_{id}^{t+1} - b_{id}^{t} = \psi_2 r_1 \left(b_{kd}^{t} - b_{id}^{t}\right) - \psi_1 r_2 \left(b_{jd}^{t} - b_{id}^{t}\right), \ \forall i, d. \tag{32}$$

Else if i has lost and its opponent has won, then

$$x_{id}^{t+1} - b_{id}^{t} = -\psi_1 r_2 \left(b_{kd}^{t} - b_{id}^{t}\right) + \psi_2 r_1 \left(b_{jd}^{t} - b_{id}^{t}\right), \ \forall i, d. \tag{33}$$

Else if both i and its opponent have lost their matches at week t, then

$$x_{id}^{t+1} - b_{id}^{t} = \psi_2 r_2 \left(b_{kd}^{t} - b_{id}^{t}\right) + \psi_2 r_1 \left(b_{jd}^{t} - b_{id}^{t}\right), \ \forall i, d. \tag{34}$$

End, and substituting "–" with "Distance," and taking "—Distance" as "Similarity" the updating equations of GLCA are obtained as follows:

If both i and its opponent have won their matches at week t, then

$$\begin{aligned} \text{Distance}\left(x_{id}^{t+1}, b_{id}^{t}\right) &\approx \psi_1 r_1 \text{Similarity}\left(b_{id}^{t}, b_{kd}^{t}\right) \\ &\quad + \psi_1 r_2 \text{Similarity}\left(b_{id}^{t}, b_{jd}^{t}\right) \forall i, d. \end{aligned} \tag{35}$$

Else if i has won and its opponent has lost, then

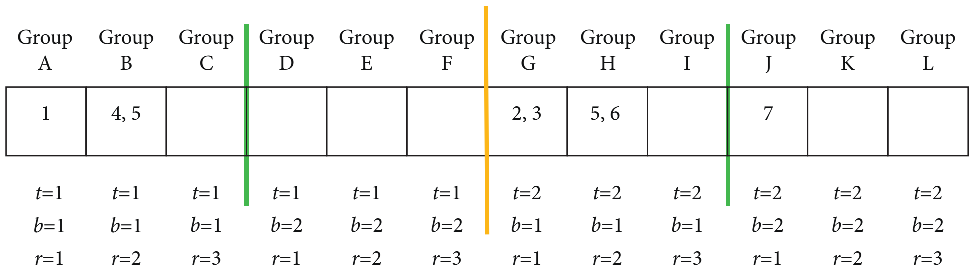


Figure 3: A sample grouping representation of the solution.

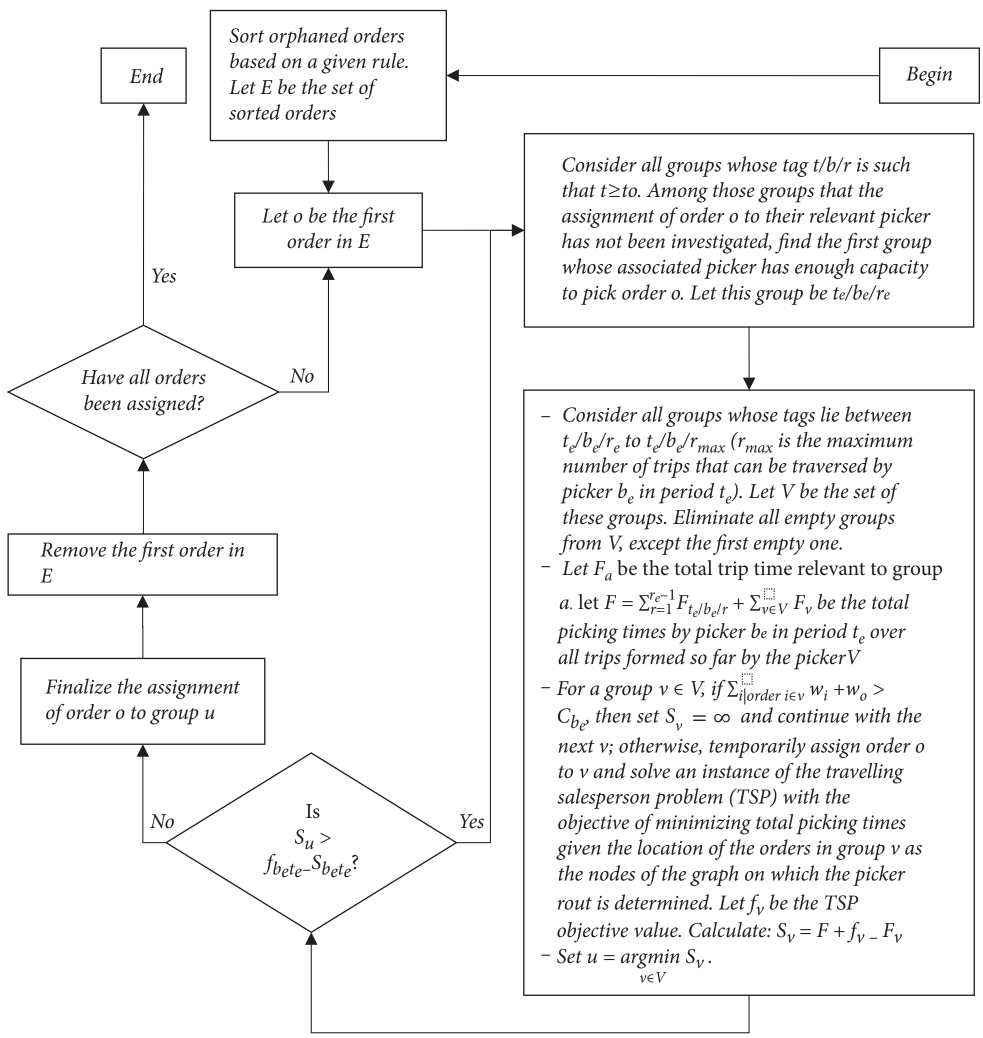


Figure 4: The first-period-first-picker-best-trip algorithm.

$$\text{Distance}\left(x_{id}^{t+1}, b_{id}^{t}\right) \approx \psi_2 r_1 \text{Distance}\left(b_{kd}^{t}, b_{id}^{t}\right) + \psi_1 r_2 \text{Similarity}\left(b_{id}^{t}, b_{jd}^{t}\right), \ \forall i, d. \quad (36)$$

Else if i has lost and its opponent has won, then

$$\text{Distance}\left(x_{id}^{t+1}, b_{id}^{t}\right) \approx \psi_1 r_2 \text{Similarity}\left(b_{id}^{t}, b_{kd}^{t}\right) + \psi_2 r_1 \text{Distance}\left(b_{jd}^{t}, b_{id}^{t}\right), \ \forall i, d. \quad (37)$$

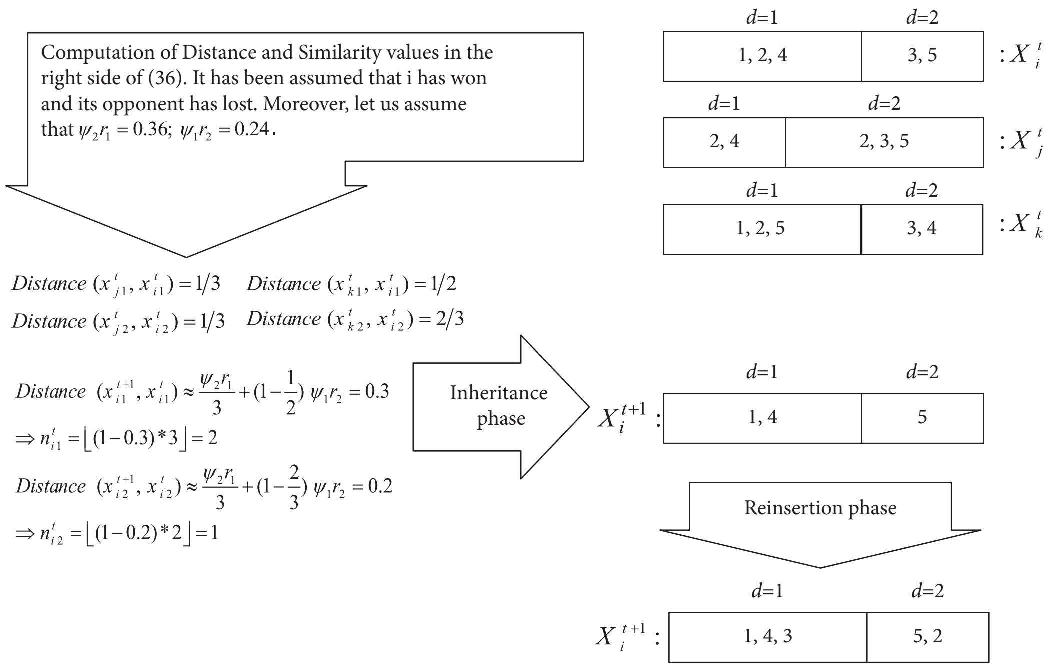


Figure 5: A simple example to demonstrate the mechanism for generating the new solution in GLCA.

Else If both i and its opponent have lost their matches at week t, then

$$\begin{aligned}\text{Distance}\left(x_{id}^{t+1}, b_{id}^t\right) \approx{}& \psi_2 r_2 \text{Distance}\left(b_{kd}^t, b_{id}^t\right) \\ &+ \psi_2 r_1 \text{Distance}\left(b_{jd}^t, b_{id}^t\right), \ \forall i, d.\end{aligned} \tag{38}$$

End.

In (29) and (30) and (35)–(38), d shows the group index and $i = 1, \ldots, N$ is the individual index, where N denotes the size of the population. It is important to note that in these equations all of $x_{id}^t, b_{id}^t, p_{id}^t$ and p_{gd}^t are stand for groups of orders.

4.1.3. Generating a New Solution in GPSO and GLCA. With the aid of adapted equations (29) and (30) for GPSO and (35)–(38) for GLCA, we can generate new feasible solutions and hence propose search-based methods for the MPMTR problem. The whole process is simple. Given a feasible parent solution i (X_i^t in GPSO and B_i^t in GLCA), using the mentioned equations, some orders are removed from groups in the first phase and some others remain in the groups (inheritance), and then the removed orders are backed to groups in the second phase (reinsertion).

(1) Inheritance Phase. According to (30), the construction of the new group x_{id}^{t+1} of the offspring solution X_i^{t+1} at iteration $t+1$ should be such that its difference with group x_{id}^t of the parent solution (particle i) be around the value of v_{id}^{t+1}. Let's start from (30)

$$\begin{aligned}\text{Distance}\left(x_{id}^{t+1}, x_{id}^t\right) = 1 - \frac{\left|x_{id}^{t+1} \cap x_{id}^t\right|}{\left|x_{id}^{t+1} \cup x_{id}^t\right|} \approx v_{i\,d}^{t+1} \longrightarrow \left|x_{id}^{t+1} \cap x_{id}^t\right| \\ \approx \left(1 - v_{id}^{t+1}\right)\left|x_{id}^{t+1} \cup x_{id}^t\right|.\end{aligned} \tag{39}$$

Since in the inheritance phase, x_{id}^{t+1} can inherit up to all orders allocated to x_{id}^t, we can replace $|x_{id}^{t+1} \cup x_{id}^t|$ with $|x_{id}^t|$, and arriving at

$$\left|x_{id}^{t+1} \cap x_{id}^t\right| \approx \left(1 - v_{id}^{t+1}\right)\left|x_{id}^t\right|. \tag{40}$$

Relation (40) says that the number of orders shared between x_{id}^{t+1} and x_{id}^t, which is indeed the amount of orders inherited by $x_{i\,d}^{t+1}$ from x_{id}^t should be around the value of $(1 - v_{id}^{t+1})|x_{id}^t|$. This value is represented by n_{id}^t and is set as

$$n_{id}^t = \left(1 - v_{id}^{t+1}\right)\left|x_{id}^t\right|. \tag{41}$$

For GLCA the same approach is also followed. For example, let us consider (35). After calculation of the right side, we get a scalar value. Let this value be called v_{id}^{t+1}. Now the process is same as above. That is, according to (35), the construction of the new group x_{id}^{t+1} of the offspring solution at iteration $t+1$ should be such that its difference with group b_{id}^t of the parent solution B_i^t be around the value of v_{id}^{t+1}, and hence,

$$n_{id}^t = \left(1 - v_{id}^{t+1}\right)\left|b_{id}^t\right|. \tag{42}$$

After calculation of n_{id}^t for group d of the ith individual in the population of GPSO or GLCA at iteration t, n_{id}^t

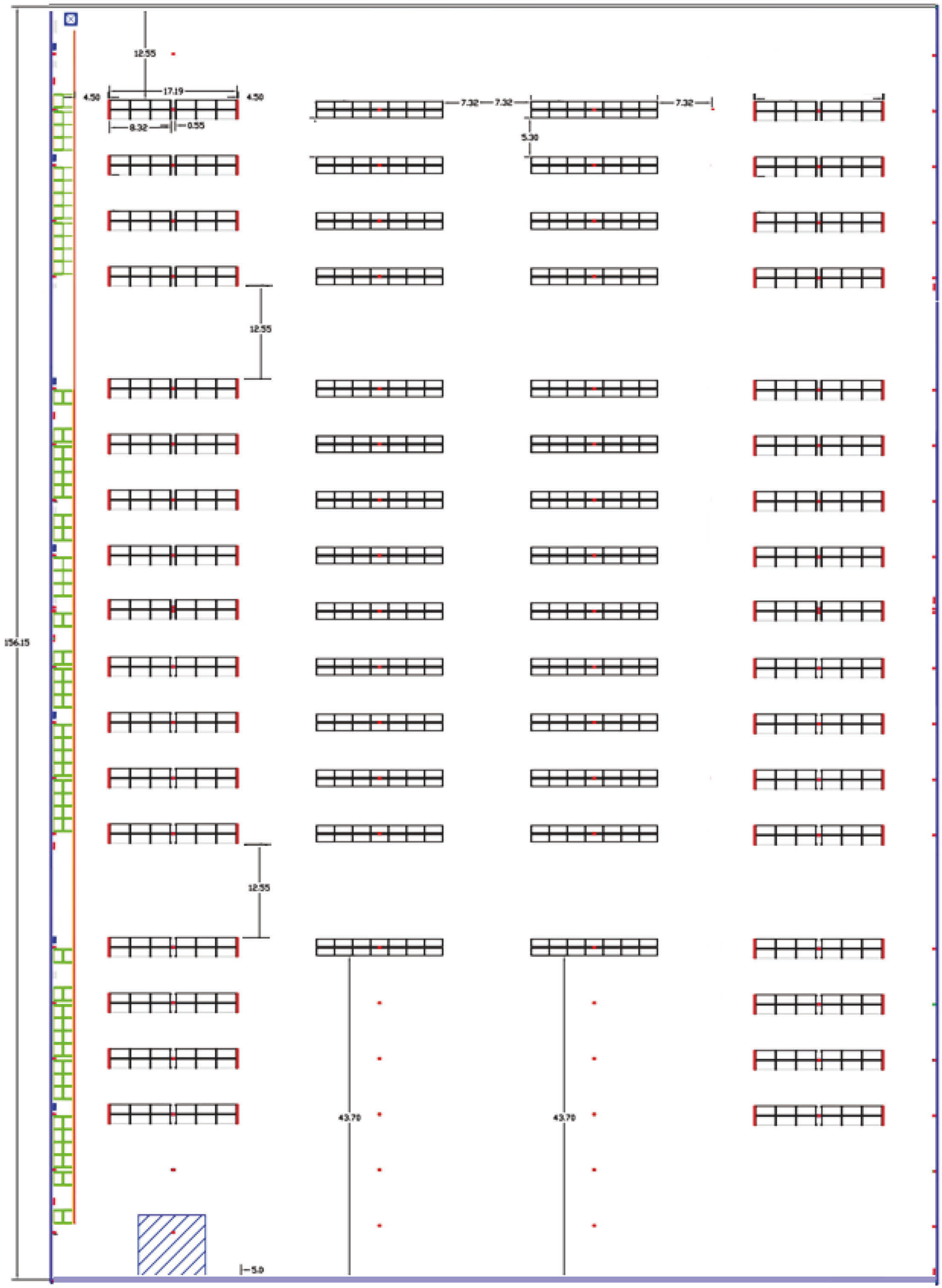


Figure 6: Layout of the intended auto parts warehouse.

Table 2: The raw data.

Stock code	Order code	No. of parts	Supplier code	Delivery destination code	Pallet name	Parts in pallet	Grade	Part weight (g)	part's net weight in pallet (kg)	Weight of full pallet (kg)	Transaction type (delivery/receive)	Delivery due date (in days)	Receiving date (in days)	Location in the warehouse
DN12053650A	1	800	13285	1175	GP5	160	B	1800	288	438	Receive (replenish)	—	3	B5S10111
DN12054650A	2	800	13285	1175	GP5	160	B	1800	288	438	Delivery	2	—	B5S01432
DP13070231	2	480	11293	9909	GP5	480	B	470	226	376	Delivery	2	—	B5S24732
⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 3: Results on instances with 5 orders.

Instance	Periods	No. of picker	Capacity level	Optimal/best solution		GPSO					GLCA				
				Objective function	Time (s)	Objective function			Average times (s)	GAP (%)	Objective function			Average times (s)	GAP (%)
						Min	Avg.	Max			Min	Avg.	Max		
1			1	2	0.81	2	2	2	0.09	0	2	2	2	0.04	0
2		3	2	0	0.22	0	0	0	0.1	0	0	0	0	0.22	0
3			3	0	0.23	0	0	0	0.24	0	0	0	0	0.46	0
4			1	0	0.29	0	0	0	1.27	0	0	0	0	2.39	0
5	3	5	2	0	0.31	0	0	0	0.85	0	0	0	0	2.03	0
6			3	0	0.30	0	0	0	0.83	0	0	0	0	1.45	0
7			1	0	0.40	0	0	0	1.32	0	0	0	0	2.82	0
8		7	2	0	0.43	0	0	0	0.85	0	0	0	0	1.91	0
9			3	0	0.43	0	0	0	1.35	0	0	0	0	1.93	0
10			1	2	0.79	2	2	2	0.78	0	2	2	2	0.9	0
11		3	2	0	0.31	0	0	0	3.46	0	0	0	0	4.27	0
12			3	0	0.31	0	0	0	1.67	0	0	0	0	3.52	0
13			1	0	0.45	0	0	0	3.67	0	0	0	0	3.21	0
14	5	5	2	0	0.47	0	0	0	1.74	0	0	0	0	2.67	0
15			3	0	0.45	0	0	0	1.80	0	0	0	0	2.43	0
16			1	0	0.63	0	0	0	1.34	0	0	0	0	3.36	0
17		7	2	0	0.62	0	0	0	2.34	0	0	0	0	3.32	0
18			3	0	0.64	0	0	0	2.32	0	0	0	0	4.65	0
19			1	2	0.57	2	2	2	1.32	0	2	2	2	2.67	0
20		3	2	0	0.41	0	0	0	2.14	0	0	0	0	2.45	0
21			3	0	0.43	0	0	0	2.95	0	0	0	0	3.51	0
22			1	0	0.61	0	0	0	2.12	0	0	0	0	3.36	0
23	7	5	2	0	0.66	0	0	0	2.72	0	0	0	0	3.59	0
24			3	0	0.65	0	0	0	2.35	0	0	0	0	2.36	0
25			1	0	0.87	0	0	0	3.36	0	0	0	0	3.36	0
26		7	2	0	0.91	0	0	0	2.36	0	0	0	0	3.54	0
27			3	0	0.81	0	0	0	2.40	0	0	0	0	3.14	0
Avg.										0					0

number of orders are selected from x_{id}^t (in GPSO) and b_{id}^t (in GLCA) and are assigned to the same group of the offspring solution. Hence, $n_{id}^t - |x_{id}^t|$ (in GPSO) and $n_{id}^t - |b_{id}^t|$ (in GLCA) a number of orders are orphaned and should be reinserted back into groups during the reinsertion phase.

(2) Reinsertion Phase. As described above, those orders that are not inherited from the groups of X_i^t in GPSO and B_i^t in GLCA are orphaned ones and should be backed into groups during the reinsertion phase. A constructive heuristic is developed to assign the orders to groups. The algorithm, which is called the first-period-first-picker-best-trip algorithm, always creates feasible groups and hence causes both GPSO and GLCA to perform their search in the feasible region. Although it is described in high detail in Figure 4, the algorithm logic is mainly as follows.

Every time among unassigned orders, select an order based on a given attribute and try to put it in the best feasible trip of the first available picker in the first possible period.

To find the "best feasible trip," the algorithm checks the possibility of the assignment of order to different trips a picker traverses in a given period. All possible trips are checked, and an instance of the travelling salesperson problem is solved for each trip to find the best trip, which costs for a less picking time. Once the best trip was found, if the sum of picking times over all picker trips was smaller than the picker presence time in period t (*i.e.*, $s_{bt} - f_{bt}$), the order assignment to the best trip is finalized and the process continues with the next order, until all orders are assigned and a complete solution is obtained.

To generate the initial solution, the first-period-first-picker-best-trip algorithm starts with a random list of all orders. That is, all orders are assumed orphaned and unassigned to groups.

Based on the two phases described above, the pseudo code for generating a new feasible solution is as follows:

Begin.

Step 1 (the inheritance phase)

(i) Let x_{id}^t (for GPSO) and b_{id}^t (for GLCA) be the set of orders allocated to group d in X_i^t and B_i^t at iteration t, respectively (X_i^t and B_i^t are the ith individual in the population of GPSO and GLCA, which are feasible grouping of orders).

Table 4: Results on instances with 10 orders.

Instance	Periods	No. of picker	Capacity level	Optimal/best solution		GPSO					GLCA				
				Objective function	Time (s)	Objective function Min	Objective function Avg.	Objective function Max	Average times (s)	GAP (%)	Objective function Min	Objective function Avg.	Objective function Max	Average times (s)	GAP (%)
28			1	4	2.28	4	4	4	3.38	0	4	4	4	4.62	0
29		3	2	0	0.35	0	0	0	2.65	0	0	0	0	4.46	0
30			3	0	0.34	0	0	0	2.98	0	0	0	0	3.91	0
31			1	5	0.83	5	5	5	2.12	0	5	5	5	3.4	0
32	3	5	2	0	0.53	0	0	0	2.14	0	0	0	0	3.55	0
33			3	0	0.54	0	0	0	2.61	0	0	0	0	3.16	0
34			1	3	0.96	3	3	3	2.58	0	3	3	3	3.02	0
35		7	2	0	0.67	0	0	0	3.01	0	0	0	0	3.07	0
36			3	0	0.67	0	0	0	2.78	0	0	0	0	3.98	0
37			1	12	14.76	12	12	12	1.81	0	12	12	12	1.97	0
38		3	2	0	0.51	0	0	0	2.71	0	0	0	0	2.17	0
39			3	0	0.50	0	0	0	2.65	0	0	0	0	3.52	0
40			1	5	0.77	5	5	5	2.53	0	5	5	5	2.29	0
41	5	5	2	0	0.81	0	0	0	3.01	0	0	0	0	3.32	0
42			3	0	0.80	0	0	0	3.18	0	0	0	0	3.31	0
43			1	3	16.31	3	3	3	2.09	0	3	3	3	3.36	0
44		7	2	0	1.13	0	0	0	3.2	0	0	0	0	3.21	0
45			3	0	1.07	0	0	0	3.46	0	0	0	0	4.51	0
46			1	12	2.98	12	12	12	2.18	0	12	12	12	2.48	0
47		3	2	0	0.69	0	0	0	2.14	0	0	0	0	3.36	0
48			3	0	0.68	0	0	0	3	0	0	0	0	3.48	0
49			1	5	3.11	5	5	5	2.92	0	5	5	5	3.61	0
50	7	5	2	0	1.18	0	0	0	3.15	0	0	0	0	3.51	0
51			3	0	1.1	0	0	0	3.45	0	0	0	0	3.65	0
52			1	3	30.25	3	3	3	3.15	0	3	3	3	3.34	0
53		7	2	0	1.57	0	0	0	3.26	0	0	0	0	4.99	0
54			3	0	1.67	0	0	0	3.25	0	0	0	0	4.4	0
Avg.										0					0

(ii) Compute n_{id}^t using (41) for GPSO and (42) for GLCA

(iii) Select n_{id}^t orders from x_{id}^t or b_{id}^t and assign them to x_{id}^{t+1}. With a chance of 50%, the selection is done at random. Otherwise orders are passed based on Earliest Due Date (EDD) rule;

(iv) Repeat the above steps for all ds

Step 2 (the reinsertion phase)

(i) Using first-period-first-picker-best-trip algorithm, assign each of the orphaned orders to groups to obtain the complete offspring solution X_i^{t+1}. With a chance of 50%, at the start of the first-period-first-picker-best-trip algorithm, the orders are sorted randomly. Otherwise, they are sorted based on Earliest Due Date (EDD) rule;

End.

With a simple example depicted in Figure 5, we demonstrate the mechanism for generating the new solution in GLCA (the same figure can be extracted for GPSO algorithm).

Recall that we developed two metaheuristic algorithms for the problem, which are naturally randomized algorithms. With this nature, we expect that if we use each algorithm to solve one instance several times, the obtained results will become different. Parts of the randomness are also enforced by inheritance and reinsertion phases.

5. The Case Study, Experimentations, and Results

In this section, the proposed mathematical formulation and algorithms are tested using data obtained from a real-world warehouse confronted by one of the prominent Iranian automakers, the SAIPA Group. The intended warehouse owns a drive-through racking storage system which is loaded and unloaded from both sides, and its layout has been depicted in Figure 6.

The inbound logistics division at SAIPA Group, which performs on a daily basis, is responsible for providing the required data on the list of suppliers and list of part orders together with their specifications (e.g., size, weight, pallet type, time windows etc). The problem is how to plan for

Table 5: Results on instances with 20 orders.

Instance	Periods	No. of picker	Capacity level	Optimal/best solution		GPSO					GLCA				
				Objective function	Time (s)	Objective function			Average times (s)	GAP (%)	Objective function			Average times (s)	GAP (%)
						Min	Avg.	Max			Min	Avg.	Max		
55			1	15	1500	13	13	13	6.9	15.4	13	13	13	6.2	15.4
56		3	2	4	1500	2	2.6	4	10.6	53.8	2	2	2	11.5	100
57			3	0	1120.2	0	0	0	10.7	0	0	0	0	8.95	0
58			1	12	1500	10	10	10	9.6	20	10	10	10	10.5	20
59	3	5	2	3	1500	0	0	0	9.3	+	0	0	0	8.2	+
60			3	2	1500	0	0	0	18.9	+	0	0	0	15.9	+
61			1	19	971.2	19	19	19	14.7	0	19	19	19	14.3	0
62		7	2	2	1500	0	0	0	20.6	+	0	0	0	20.75	+
63			3	4	1500	0	0	0	24.5	+	0	0	0	22.7	+
64			1	28	1500	24	24	24	11.1	16.7	24	24	24	11.8	16.7
65		3	2	3	1500	2	2	2	25	50	2	2	2	26.2	50
66			3	5	1500	0	0	0	14.5	+	0	0	0	14	+
67			1	31	1500	30	30	30	22.5	3.3	30	30	30	22.6	3.3
68	5	5	2	2	1500	0	0	0	21.7	+	0	0	0	20.85	+
69			3	1	1500	0	0	0	30.9	+	0	0	0	33.8	+
70			1	20	1500	19	19	19	30	5.3	19	19	19	33.1	5.3
71		7	2	5	1500	0	0	0	32	+	0	0	0	34.55	+
72			3	4	1500	0	0	0	62.6	+	0	0	0	55.9	+
73			1	37	1321.6	37	37	37	25.1	0	37	37	37	22	0
74		3	2	3	1500	2	2	2	28.6	50	2	2	2	29.8	50
75			3	5	1500	0	0	0	32.2	+	0	0	0	27.4	+
76			1	34	1500	30	30	30	43.6	13.3	30	30	30	37.3	13.3
77	7	5	2	4	1500	0	0	0	26.0	+	0	0	0	24.25	+
78			3	6	1500	0	0	0	50.7	+	0	0	0	45.9	+
79			1	22	1500	19	19	19	50	15.8	19	19	19	43.5	15.8
80		7	2	4	1500	0	0	0	32.3	+	0	0	0	28.7	+
81			3	3	1500	0	0	0	79.0	+	0	0	0	73.65	+
Avg.															>22.3

picking orders from a central auto parts warehouse and distributing them towards assembly plants.

The daily auto parts logistic problem in SAIPA can be defined as follows. At the start of each day, the number of parts in each order is supplied by the warehouse over the planning horizon, order time windows, and the destination of each order is known. The size, type, and weight of orders are also known. Besides, data on receiving products from suppliers over time is given for replenishing the warehouse over the planning horizon. The problem is to plan for picking orders such that the operational constraints on receiving products and picked orders are met in a way that the total tardiness is minimized. A sample of the raw data has been summarized in Table 2. Based on instance problems, the problem will be solved and the results will be compared using different solution procedures.

Based on described data structure, required problem instances were generated based on real-world situation, whose characteristics have been summarized as follows:

(i) Number of orders: 5, 10, 20, 50, and 100.

(ii) Number of periods: 3, 5, and 7.

(iii) Number of pickers: 3, 5, and 7.

(iv) Picker's weight capacity: 1 (1000 kg), 2 (2000 kg), and 3 (3000 kg).

(v) Picker's capacity in terms of the number of orders it picks in each trip: 1 (2 pallets), 2 (5 pallets), and 3 (8 pallets).

(vi) The layout has been depicted in Figure 6.

(vii) Manhattan distance is used for measuring the travel times.

Based on the above characterization, $5 \times 3 \times 3 \times 3 = 135$ problem instances are generated given the combinations of the number of orders, the number of pickers, the number of periods, and the capacity level for pickers, respectively. The performance of algorithms is evaluated using these problem instances. Both the GPSO and GLCA algorithms were coded in MATLAB and run on a computer with 32 GB of RAM and 3.4 GHz of CPU speed. The proposed mathematical formulation has been coded in GAMS software and solved by the Cplex solver.

The parameters used for GPSO are set as follows:

(i) The population size (NP) = 10;

(ii) The maximum number of iterations = 250;

(iii) Inertia weight (w) = 0.4;

Table 6: Results on instances with 50 orders.

Instance	Periods	No. of picker	Capacity level	Optimal/best solution Objective function	Optimal/best solution Time (s)	GPSO Objective function Min	GPSO Avg.	GPSO Max	GPSO Average times (s)	GLCA Objective function Min	GLCA Avg.	GLCA Max	GLCA Average times (s)	GAP (%)
82			1	*	1500	46	46	46	15.5	46	46	46	15.6	0
83		3	2	*	1500	46	46	46	40.7	46	46	46	39.7	0
84			3	*	1500	23	27.5	35	75.3	23	23	23	74.8	19.6
85			1	*	1500	40	40	40	43.3	40	40	40	36.4	0
86	3	5	2	*	1500	20	20	20	80.4	20	20	20	69.7	0
87			3	*	1500	5	8.75	11	145.2	5	5	5	145.9	75
88			1	*	1500	37	37	37	45.5	37	37	37	45	0
89		7	2	*	1500	8	8.75	11	64	8	8	8	66.8	9.4
90			3	*	1500	0	0.75	3	86.5	0	0	0	60.15	+
91			1	*	1500	80	80	80	25.3	80	80	80	24.2	0
92		3	2	*	1500	46	47.5	49	54.5	46	46	46	57.2	3.3
93			3	*	1500	26	31.75	36	102.3	23	23	23	112	38
94			1	*	1500	100	100	100	43.8	100	100	100	43.1	0
95	5	5	2	*	1500	20	20	20	104.7	20	20	20	115.5	0
96			3	*	1500	5	6.5	9	252.4	5	5	5	228.2	30
97			1	*	1500	66	66	66	109.9	66	66	66	101.4	0
98		7	2	*	1500	8	8	8	163.6	8	8	8	155.8	0
99			3	*	1500	0	0	0	138	0	0	0	128.45	0
100			1	*	1500	115	115	115	56.6	115	115	115	51.1	0
101		3	2	*	1500	46	47.5	50	59.9	46	46	46	66	3.3
102			3	*	1500	25	27.25	31	130.4	23	23	23	135.1	18.5
103			1	*	1500	100	100	100	71.6	100	100	100	76.1	0
104	7	5	2	*	1500	20	20	20	164.3	20	20	20	149.1	0
105			3	*	1500	5	5	5	273.2	5	5	5	266.4	0
106			1	*	1500	66	66	66	72.6	66	66	66	77.7	0
107		7	2	*	1500	8	8	8	195.4	8	8	8	176.3	0
108			3	*	1500	0	0	0	172.3	0	0	0	157.95	0
Avg.														>7.3

*no feasible solution found.

(iv) $c1 = 0.2$;

(v) $c2 = 0.5$;

The parameters used for GLCA are set as follows:

(i) The population size (NP) = 10;

(ii) The maximum number of iterations = 250;

(iii) $\psi_1 = 0.4$;

(iv) $\psi_2 = 0.4$;

Each algorithm was run 10 times on each instance. In each instance, the min, the max, the average performance, the average time for achieving the best solution and the gap values between the objective functions reported by the two algorithms or the gap values between the objective functions reported by the two algorithms versus optimal objective value are presented. Results are reported based on min, max, and average performances. Results have been reported in Tables 3–7.

In Tables 3 and 4, the results of the proposed algorithms are compared with those provided by GAMS/Cplex. Based on the results for small instances, the gaps for GPSO and GLCA are zero, confirming that the use of these algorithms would probably be effective for solving larger-sized problems. As can be seen on small-sized problems, the running times are close to each other. There are several problem instances with 10 orders on which GAMS/Cplex takes a considerable time to report the optimal solution. However, for GPSO and GLCA, time trends are rather constant. From the results, it can be seen that on small problems, the performance of different methods is independent of the number of periods, the number of pickers, and the capacity level of the pickers.

On problems with 20 orders in Table 5, the performance of GPSO and GLCA is the same on 26 out of 27 problems in terms of the average performance. The only exception is problem #56, for which GLCA performs better than GPSO, whose max performance is worse than GLCA. This indicates that GPSO and GLCA exhibit stable performance and achieve the same objective value in each run. Therefore, the standard deviation of the results for each problem category is equal to zero.

On many test problem instances in this table, GAMS/Cplex fails to approve the optimal solution. Only on three instance problems, it approves optimality. In terms of gap values, there exists a significant gap between the output of

Table 7: Results on instances with 100 orders.

Instance	Periods	No. of picker	Capacity level	Optimal/best solution		GPSO				GLCA				GAP (%)
				Objective function	Time (s)	Objective function			Average times (s)	Objective function			Average times (s)	
						Min	Avg.	Max		Min	Avg.	Max		
109			1	*	1500	92	92	92	38.7	92	92	92	41.3	0
110		3	2	*	1500	92	93.5	95	70.2	92	92	92	70.7	1.6
111			3	*	1500	53	65.25	75	150.2	46	47.5	48	142.3	37.4
112			1	*	1500	95	95	95	75.6	95	95	95	71.5	0
113	3	5	2	*	1500	40	44	48	103.2	40	40	40	96.6	10
114			3	*	1500	76	81	85	327.3	65	67.75	69	330.5	19.6
115			1	*	1500	95	95	95	75	95	95	95	71.8	0
116		7	2	*	1500	74	76.75	79	248.5	74	74	74	217.5	3.7
117			3	*	1500	43	46.25	50	298.8	37	37	37	283.8	25
118			1	*	1500	190	190	190	64.5	190	190	190	56.2	0
119		3	2	*	1500	94	99.5	104	161	92	92	92	173.8	8.2
120			3	*	1500	141	163	170	302.4	139	139.25	140	312.4	17.1
121			1	*	1500	200	200	200	147.9	200	200	200	130.7	0
122	5	5	2	*	1500	123	129.25	136	238.4	120	120	120	259.9	7.7
123			3	*	1500	72	81.25	93	456.2	65	65.5	67	412.9	24
124			1	*	1500	190	190	190	199.8	190	190	190	171.8	0
125		7	2	*	1500	77	80.75	83	344.8	74	74	74	341.2	9.1
126			3	*	1500	46	49.25	52	452.3	37	37	37	497.5	33.1
127			1	*	1500	285	285	285	132.1	285	285	285	131.8	0
128		3	2	*	1500	234	253	273	199.7	230	230	230	188	10
129			3	*	1500	162	175	180	332.9	138	138	138	323.2	26.8
130			1	*	1500	285	285	285	158.6	285	285	285	141	0
131	7	5	2	*	1500	121	129	138	401.7	120	120	120	408	7.5
132			3	*	1500	75	81.5	90	469.7	65	65	65	395.6	25.4
133			1	*	1500	190	190	190	287.1	190	190	190	244.5	0
134		7	2	*	1500	75	82	86	651.7	74	74	74	550	10.8
135			3	*	1500	37	45.5	52	825	37	37	37	757.2	23
Avg.														11.11

*no feasible solution found.

GPSO and GLSA versus GAMS/Cplex. However, in many cases, which are indicated with a "+" sign, the gap values cannot be computed due to zero objective values reported by GPSO and GLCA.

For larger-sized problems with more than 50 orders, GAMS/Cplex is not able to find a feasible solution for any problem instance within the given time limit. We therefore compare the performances of GPSO and GLCA. However, there is no guideline to judge the quality of results achieved by GPSO and GLCA, except for their reported statistics. The results related to the large-sized problem instances are summarized in Tables 6 and 7.

On a problem with 50 jobs, again GLCA performs robustly and achieves the same results in all runs on each problem instance. However, there are fluctuations in the performance of GPSO. On 9 out of 27 instances, GPSO fails to show constant behavior and hence its standard deviation is not zero.

On problems with 100 orders, still GLCA performs very stable compared to GPSO. It achieves the same objective value on all runs for 23 out of 27 problem instances. However, this record for GPSO is only 9 out of 27. Moreover, the worst performance of GLCA is significantly better than GPSO. Such a reliable performance indicates that GLCA is a more dependable algorithm than GPSO.

In terms of the running times, as depicted in Figure 7, the behavior of CPSO and GLCA is close to each other. The reason is mainly because that both algorithms use the same number of population size and iterations and hence generate and evaluate the same number of solutions. Moreover, both algorithms use the same mechanism and heuristics for the generation of new solutions. As can be seen from the figures, as the capacity of the picker increases, it is allowed to form longer tours and visit more locations. Hence, larger instances of TSP should be inevitably solved, and this will increase the computation times. Enlarging the length of the planning horizon in terms of the number of periods and increasing the number of pickers will increase the computation times required to generate a feasible solution. Because the number of groups has increased and checking the suitability of the assignment of orders to more number of groups will take significant time.

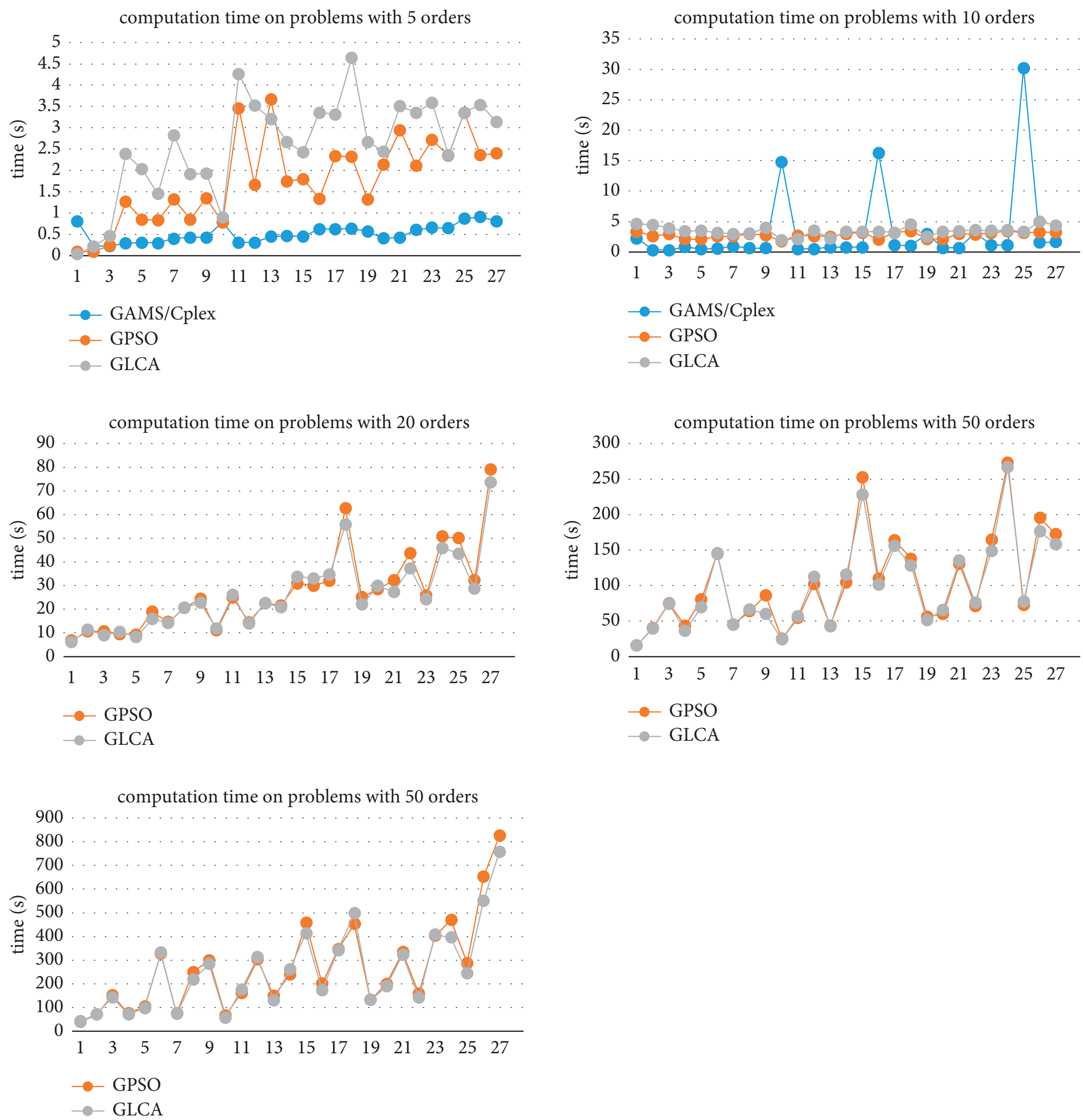


Figure 7: Plot of computation times taken by various methods.

6. Managerial Insights and Practical Implications

In this paper, we considered a real-world application of OPS planning and used mathematical programming techniques (e.g., MILP formulations and solution methods) to deal with it. The problem is mostly related to product warehousing, especially in the automotive industry. The main characteristics of such warehouses are that the number of order pickers in terms of the material handling equipment such as lift trucks, transpallet trucks, reach trucks, etc. is limited and they are used in circulation. This justifies the need for considering multitrip order picking when modeling OPS planning in the automotive industry.

Usually, there is a lag (in days) to pick, pack, and deliver the orders to their destinations. Therefore, planning based on daily span is just an oversimplification that is seen almost in most research studies related to OPS.

Besides retrieval, storage is also an equally important process in warehouse management. When orders are delivered, product shelves become empty and ready to be nested by new products that constitute future orders. Shelves management is a crucial task in warehousing because the delivery timeline of the orders should be in such a way that provides shelve spaces for arriving products. This issue is addressed as product replenishment. Most research studies only consider the retrieval planning. However, due to the limited capacity *o* warehouses, the retrieval plan can affect the replenishment or storage plan. Therefore, both the replenishment and retrieval planning should be done in an integrated fashion. If the planning span becomes multiperiod, both the retrieval and replenishment planning can be done in an integrated manner.

Under multiperiod planning, it is possible to consider order delivery due dates and hence consider responsiveness measures, which are in relation to customers, such as total order tardiness, rather than the conventional total retrieval cost measure, which is internal. Such an objective function is fit for the automotive industry, in which line stops impose huge costs.

With reference to the above issues, this research conducted an integration of the order batching and multitrip picker routing decisions with the possibility of product replenishment in a multiperiod order picking system to minimize total order tardiness. The notion of product replenishment planning in a multiperiod order picking system was introduced for the first time in the current study. Besides, minimizing total order tardiness has not been used in previous research studies.

7. Concluding Remarks and Future Research Directions

This study focused on integrating order batching and multitrip picker routing in an order picking system. The possibility of product replenishment, which was considered for the first time in this research, calls for considering the picking planning as a multiperiod planning. Products are entered into the warehouse and constitute orders that are picked during the time. Therefore, picking planning should be such that the empty rooms for replenishment of new products are provided. With the aim of minimizing total tardiness, as a new criterion for measuring the performance of picking planning, we presented a mixed integer linear formulation for the problem and developed adapted metaheuristic algorithms equipped with constructive heuristic to ensure feasibility.

The proposed mixed integer linear formulation introduces new constraints enforced by the notion of multiperiod order retrieval and replenishment planning and the new objective function, i.e., the total tardiness, which makes more sense for responsive warehouse planning (especially for auto parts order management, that should be in such a way that it prohibits assembly line stops and the cost imposed).

The metaheuristic algorithms were based on grouping particle swarm optimization (GPSO) and grouping league championship algorithm (GLCA). We heavily adapted these algorithms to be responsive to the structure of the grouping problems to which the problem under consideration belongs. Our results indicated that the league championship algorithm can stably solve large instances of the problem in a reasonable time.

The results of the computational experiment indicated that

(i) There were problem instances with 10 orders on which GAMS/Cplex took a considerable time to report optimality. For GPSO and GLCA, time trends were rather constant.

(ii) On problems with 20 orders, the average performance of GPSO and GLCA was almost the same on all problems. In many test problem instances, GAMS/Cplex failed to approve the optimality.

(iii) On problems with more than 50 orders, GAMS/Cplex was not able to find a feasible solution to any problem instance. GLCA performed robust and achieved the same results in all runs on each problem instance. There were fluctuations in the performance of GPSO.

(iv) On problems with 100 orders, GLCA performed very stable compared to GPSO. It achieved the same objective value on all runs for 23 out of 27 problem instances. This record for GPSO was only 9 out of 27. Moreover, the worst performance of GLCA was significantly better than GPSO.

For future research, the inclusion of the congestion of pickers can be considered in the picking process. Our modeling, which considered limited access to pickers, can be extended to relate preventive maintenance of order picker facilities with typical operational decisions of the order picking systems. Multilayer layouts can also be considered. Besides the proposed metaheuristic algorithms, developing the grouping version of other new metaheuristic algorithms such as optics-inspired optimization [59–62]) or F3EA metaheuristic algorithm [63] and measuring their suitability for OPS is recommended. Finally, developing exact solution methods like column generation or enhanced MILP formulations [64] would be worthwhile to consider in future research.

Data Availability

The authors declare that the data are available and can be presented upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. A. Cano, A. A. Correa-Espinal, and R. A. Gómez-Montoya, "Mathematical programming modeling for joint order batching, sequencing and picker routing problems in manual order picking systems," *Journal of King Saud University - Engineering Sciences*, vol. 32, no. 3, pp. 219–228, 2020.

[2] M. R. Islam, S. M. Ali, A. M. Fathollahi-Fard, and G. Kabir, "A novel particle swarm optimization-based grey model for the prediction of warehouse performance," *Journal of Computational Design and Engineering*, vol. 8, no. 2, pp. 705–727, 2021.

[3] V. Giannikas, W. Lu, B. Robertson, and D. McFarlane, "An interventionist strategy for warehouse order picking: evidence from two case studies," *International Journal of Production Economics*, vol. 189, pp. 63–76, 2017.

[4] C. Cergibozan and A. S. Tasan, "Order batching operations: an overview of classification, solution techniques, and future research," *Journal of Intelligent Manufacturing*, vol. 30, no. 1, pp. 335–349, 2016.

[5] R. De Koster, T. Le-Duc, and K. J. Roodbergen, *Design and Control of Warehouse Order Picking: A Literature Review, ERIM Report Series Reference Number*, vol. 182, no. 2, pp. 481–501, 2006.
[6] C. Y. Cheng, Y. Y. Chen, T. L. Chen, and J. Jung-Woon Yoo, "Using a hybrid approach based on the particle swarm optimization and ant colony optimization to solve a joint order batching and picker routing problem," *International Journal of Production Economics*, vol. 170, pp. 805–814, 2015.
[7] T. Van Gils, K. Ramaekers, A. Caris, and R. B. de Koster, "Designing efficient order picking systems by combining planning problems: state-of-the-art classification and review," *European Journal of Operational Research*, vol. 267, no. 1, pp. 1–15, 2018b.
[8] S. Henn, "Order batching and sequencing for the minimization of the total tardiness in picker-to-Part Warehouses," *Flexible Services and Manufacturing Journal*, vol. 27, no. 1, pp. 86–114, 2012.
[9] M. Masae, C. H. Glock, and E. H. Grosse, "Order picker routing in warehouses: a systematic literature review," *International Journal of Production Economics*, vol. 224, Article ID 107564, 2020.
[10] A. Husseinzadeh Kashan, "League championship algorithm: a new algorithm for numerical function optimization," in *Proceedings of the In SoCPaR 2009 - Soft Computing and Pattern Recognition*, pp. 43–48, Malacca, Malaysia, December 2009.
[11] A. Husseinzadeh Kashan, "An efficient algorithmþ þfor constrained global optimization and application to mechanical engineering design: league championship algorithm (LCA)," *Computer-Aided Design*, vol. 43, no. 12, pp. 1769–1792, 2011.
[12] A. Husseinzadeh Kashan, "League Championship Algorithm (LCA): an algorithm for global optimization inspired by sport championships," *Applied Soft Computing*, vol. 16, pp. 171–200, 2014.
[13] M. R. Alimoradi and A. Husseinzadeh Kashan, "A league championship algorithm equipped with network structure and backward Q-learning for extracting stock trading rules," *Applied Soft Computing*, vol. 68, pp. 478–493, 2018.
[14] N. Alizadeh and A. Husseinzadeh Kashan, "Enhanced grouping league championship and optics inspired optimization algorithms for scheduling a batch processing machine with job conflicts and non-identical job sizes," *Applied Soft Computing*, vol. 83, Article ID 105657, 2019.
[15] A. Husseinzadeh Kashan, S. Karimiyan, M. Karimiyan, and H. Husseinzadeh Kashan, "A modified League Championship Algorithm for numerical function optimization via artificial modeling of the "between two halves analysis"" in *Proceedings of the 6th International Conference on Soft Computing and Intelligent Systems and the 13th International Symposium on Advanced Intelligent Systems, 1944-1949*, Kube, Jappan, 2012.
[16] A. Husseinzadeh Kashan, S. Jalili, and S. Karimiyan, "Optimum structural design with discrete variables using league championship algorithm," *Civil Engineering Infrastructures Journal*, vol. 51, pp. 253–275, 2018a.
[17] A. Hueesinzadeh Kashan, M. Eyvazi, and A. Abbasi-Pooya, "An effective league championship Algorithm for the stochastic multi-period portfolio optimization problem," *Scientia Iranica*, vol. 27, pp. 829–845, 2020.
[18] S. Jalili, A. H. Kashan, and Y. Hosseinzadeh, "League championship algorithms for optimum design of pin-jointed structures," *Journal of Computing in Civil Engineering*, vol. 31, no. 2, pp. 1–17, 2017.
[19] O. Briant, H. Cambazard, D. Cattaruzza, N. Catusse, A. L. Ladier, and M. Ogier, "An efficient and general approach for the joint order batching and picker routing problem," *European Journal of Operational Research*, vol. 285, no. 2, pp. 497–512, 2020.
[20] A. Scholz and G. Wäscher, "Order batching and picker routing in manual order picking systems: the benefits of integrated routing," *Central European Journal of Operations Research*, vol. 25, no. 2, pp. 491–520, 2017.
[21] G. Marchet, M. Melacini, and S. Perotti, "Investigating order picking system Adoption: a case-study-based approach," *International Journal of Logistics Research and Applications*, vol. 18, no. 1, pp. 82–98, 2015.
[22] C. H. Glock, E. H. Grosse, R. M. Elbert, and T. Franzke, "Maverick picking: the impact of modifications in work schedules on manual order picking processes," *International Journal of Production Research*, vol. 55, no. 21, pp. 6344–6360, 2017.
[23] C. Y. Tsai, J. J. H. Liou, and T. M. Huang, "Using a multiple-GA method to solve the batch picking problem: considering travel distance and order due time," *International Journal of Production Research*, vol. 46, no. 22, pp. 6533–6555, 2008.
[24] F. Dallari, G. Marchet, and M. Melacini, "Design of order picking system," *International Journal of Advanced Manufacturing Technology*, vol. 42, no. 1–2, pp. 1–12, 2009.
[25] C. G. Petersen, "An evaluation of order picking routeing policies," *International Journal of Operations & Production Management*, vol. 17, no. 11, pp. 1098–1111, 1997.
[26] C. G. Petersen and R. W. Schmenner, "An evaluation of routing and volume-based storage policies in an order picking operation,," *Decision Sciences*, vol. 30, no. 2, pp. 481–501, 1999.
[27] C. G. Petersen and G. Aase, "A comparison of picking, storage, and routing policies in manual order picking," *International Journal of Production Economics*, vol. 92, no. 1, pp. 11–19, 2004.
[28] R. Jan and R. De Koster, "Theory and methodology routing order pickers in a warehouse with a middle aisle," *European Journal of Operational Research*, vol. 133, pp. 32–43, 2001.
[29] T. Chabot, C. C. Coelho, J. Renaud, and J. F. Côté, "Mathematical Models, Heuristic and Exact Method for Order Picking in 3D-Narrow Aisles," 2015, http://www.cirrelt.ca.
[30] G. Wäscher, "Order picking: a survey of planning problems and methods," *Supply Chain Management and Reverse Logistics*, pp. 323–347, Springer, Berlin, Germany, 2004.
[31] J. Won and S. Olafsson, "Joint order batching and order picking in warehouse operations," *International Journal of Production Research*, vol. 43, no. 7, pp. 1427–1442, 2005.
[32] Y. C. Ho and Y. Y. Tseng, "A study on order-batching methods of order-picking in a distribution centre with two cross-aisles," *International Journal of Production Research*, vol. 44, no. 17, pp. 3391–3417, 2006.
[33] S. Henn, S. Koch, K. F. Doerner, C. Strauss, and G. Wascher, "Metaheuristics for the order batching problem in manual order picking systems," *Business Research*, vol. 3, no. 1, pp. 82–105, 2010.
[34] S. Henn, S. Koch, and G. Wäscher, "Order Batching in Order Picking Warehouses: A Survey of Solution Approaches. Working Paper No.01/2011," 2011, http://www.fww.ovgu.de/femm.
[35] O. Kulak, Y. Sahin, and M. E. Taner, "Joint order batching and picker routing in single and multiple-cross-aisle warehouses using cluster-based tabu search algorithms," *Flexible Services and Manufacturing Journal*, vol. 24, no. 1, pp. 52–80, 2012.

[36] J. C. H. Pan, P. H. Shih, and M. H. Wu, "Storage assignment problem with travel distance and blocking considerations for a picker-to-Part Order picking system," *Computers & Industrial Engineering*, vol. 62, no. 2, pp. 527–535, 2012.

[37] T. Oncan, "MILP formulations and an iterated local search algorithm with tabu thresholding for the order batching problem," *European Journal of Operational Research*, vol. 243, no. 1, pp. 142–155, 2015.

[38] T. L. Chen, C. Y. Cheng, Y. Y. Chen, and L. K. Chan, "An efficient hybrid algorithm for integrated order batching, sequencing and routing problem," *International Journal of Production Economics*, vol. 159, pp. 158–167, 2015.

[39] C. A. Valle, J. E. Beasley, and A. S. Da Cunha, "Optimally solving the joint order batching and picker routing problem," *European Journal of Operational Research*, vol. 262, no. 3, pp. 817–834, 2017.

[40] F. Chen, Y. Wei, and H. Wang, "A heuristic based batching and assigning method for online customer orders," *Flexible Services and Manufacturing Journal*, vol. 30, no. 4, pp. 640–685, 2018.

[41] B. Menéndez, E. G. Pardo, A. Alonso-Ayuso, E. Molina, and A. Duarte, "Variable neighborhood search strategies for the order batching problem," *Computers & Operations Research*, vol. 78, pp. 500–512, 2017.

[42] A. Scholz, D. Schubert, and G. Wäscher, "Order picking with multiple pickers and due dates – simultaneous solution of order batching, batch Assignment and sequencing, and picker routing problems," *European Journal of Operational Research*, vol. 263, no. 2, pp. 461–478, 2017.

[43] I. Žulj, S. Kramer, and M. Schneider, "A hybrid of adaptive large neighborhood search and tabu search for the order-batching problem," *European Journal of Operational Research*, vol. 264, no. 2, pp. 653–664, 2018.

[44] T. Van Gils, K. Ramaekers, K. Braekers, B. Depaire, and A. Caris, "Increasing order picking efficiency by integrating storage, batching, zone picking, and routing policy decisions," *International Journal of Production Economics*, vol. 197, pp. 243–261, 2018.

[45] S. Moons, K. Braekers, K. Ramaekers, A. Caris, and Y. Arda, "The value of integrating order picking and vehicle routing decisions in a B2C E-commerce environment," *International Journal of Production Research*, vol. 57, no. 20, pp. 6405–6423, 2019.

[46] C. G. P. Ii, "An evaluation of order picking policies for mail order companies," *Production and Operations Management*, vol. 9, no. 4, pp. 319–335, 2009.

[47] C. Theys, O. Bräysy, W. Dullaert, and B. Raa, "Using a TSP heuristic for routing order pickers in warehouses," *European Journal of Operational Research*, vol. 200, no. 3, pp. 755–763, 2010.

[48] S. Henn and V. Schmid, "Metaheuristics for Order Batching and Sequencing in Manual Order Picking Systems. Working Paper No.11/2011," 2011, http://www.fww.ovgu.de/femm.

[49] S. Hong, A. L. Johnson, and B. A. Peters, "Batch picking in narrow-aisle order picking systems with consideration for picker blocking," *European Journal of Operational Research*, vol. 221, no. 3, pp. 557–570, 2012.

[50] F. Chen, H. Wang, Y. Xie, and C. Qi, "An ACO-based online routing method for multiple order pickers with congestion consideration in warehouse," *Journal of Intelligent Manufacturing*, vol. 27, no. 2, pp. 389–408, 2016.

[51] T. S. Su and M. H. Hwang, "An efficient order-picking route planning based on a fuzzy set method with a multiple-aisle in a distribution center," *Procedia Manufacturing*, vol. 11, pp. 1856–1862, 2017.

[52] H. Kuhn, D. Schubert, and A. Holzapfel, "Integrated order batching and vehicle routing operations in grocery retail – a general adaptive large neighborhood search algorithm," *European Journal of Operational Research*, vol. 294, no. 3, pp. 1003–1021, 2021.

[53] A. H. Kashan, A. A. Akbari, and B. Ostadi, "Grouping evolution strategies: an effective approach for grouping problems," *Applied Mathematical Modelling*, vol. 39, no. 9, pp. 2703–2720, 2015.

[54] A. Abbasi-Pooya and A. Husseinzadeh Kashan, "New mathematical models and a hybrid Grouping Evolution Strategy algorithm for optimal helicopter routing and crew pickup and delivery," *Computers & Industrial Engineering*, vol. 112, pp. 35–56, 2017.

[55] A. H. Kashan, M. Keshmiry, J. H. Dahooie, and A. Abbasi-Pooya, "A simple yet effective grouping evolutionary strategy (GES) algorithm for scheduling parallel machines," *Neural Computing and Applications*, vol. 30, no. 6, p. 1925, 2018.

[56] M. Ghadiri Nejad, A. Husseinzadeh Kashan, and S. M. Shavarani, "A novel competitive hybrid approach Based on grouping evolution strategy algorithm for solving U-shaped assembly line balancing Problems," *Production Engineering*, vol. 12, no. 5, pp. 555–566, 2018.

[57] A. Husseinzadeh Kashan, M. Jenabi, and M. Husseinzadeh Kashan, "A new solution approach for grouping problems based on evolution strategies," in *Proceedings of the SoCPaR 2009 IEEE International Conference of Soft Computing and Pattern Recognition*, pp. 88–93, Kualalumpur, Malaysia, December 2009.

[58] A. Husseinzadeh Kashan, M. Husseinzadeh Kashan, and S. Karimiyan, "A particle swarm optimizer for grouping problems," *Information Sciences*, vol. 252, pp. 81–95, 2013.

[59] A. Husseinzadeh Kashan, "A new metaheuristic for optimization: optics inspired optimization (OIO)," *Computers & Operations Research*, vol. 55, pp. 99–125, 2015a.

[60] A. H. Kashan, "An effective algorithm for constrained optimization based on optics inspired optimization (OIO)," *Computer-Aided Design*, vol. 63, pp. 52–71, 2015b.

[61] S. Jalili and A. Husseinzadeh Kashan, "Optimum discrete design of steel tower structures using optics inspired optimization method," *The Structural Design of Tall and Special Buildings*, vol. 27, no. 9, Article ID e1466, 2018.

[62] S. Jalili and A. Husseinzadeh Kashan, "An optics inspired optimization (OIO) method for optimal design of truss structures," *The Structural Design of Tall and Special Buildings*, vol. 28Article ID e1598, pp. 1–23, 2019.

[63] A. Husseinzadeh Kashan, R. Tavakkoli-Moghaddam, and M. Gen, "Find-Fix-Finish-Exploit-Analyze (F3EA) meta-heuristic algorithm: an effective algorithm with new evolutionary operators for global optimization," *Computers & Industrial Engineering*, vol. 128, pp. 192–218, 2019.

[64] A. Husseinzadeh Kashan and O. Ozturk, "Improved MILP Formulation Equipped with Valid Inequalities for Scheduling a Batch Processing Machine with Non-identical Job Sizes," *Omega*, vol. 112, Article ID 102673, 2022.