Applied Soft Computing Journal 100 (2021) 106953

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

Mitigating the risk of infection spread in manual order picking operations: A multi-objective approach

Ehsan Ardjmand [a,*], Manjeet Singh [b], Heman Shakeri [c], Ali Tavasoli [d], William A. Young II [a]

[a] Department of Analytics and Information Systems, College of Business, Ohio University, OH, 45701, USA
[b] Solutions Design, DHL Supply Chain, Westerville, OH, USA
[c] School of Data Science, University of Virginia, Charlottesville, VA, USA
[d] Department of Mechanical Engineering, Payame Noor University, Tehran, Iran

ARTICLE INFO

Article history:
Received 1 July 2020
Received in revised form 20 November 2020
Accepted 25 November 2020
Available online 30 November 2020

Keywords:
Order picking
Order batching
Picker routing
Multi-objective optimization model
Multi-objective metaheuristics
Evolutionary methods
Physical distancing
Supply chain disruption

ABSTRACT

In the aftermath of the COVID-19 pandemic, supply chains experienced an unprecedented challenge to fulfill consumers' demand. As a vital operational component, manual order picking operations are highly prone to infection spread among the workers, and thus, susceptible to interruption. This study revisits the well-known order batching problem by considering a new overlap objective that measures the time pickers work in close vicinity of each other and acts as a proxy of infection spread risk. For this purpose, a multi-objective optimization model and three multi-objective metaheuristics with an effective seeding procedure are proposed and are tested on the data obtained from a major US-based logistics company. Through extensive numerical experiments and comparison with the company's current practices, the results are discussed, and some managerial insights are offered. It is found that the picking capacity can have a determining impact on reducing the risk of infection spread through minimizing the picking overlap.

1. Introduction

In December 2019, a novel coronavirus named COVID-19 (SARS-CoV-2) was first documented in Wuhan, China. By early March 2020, 114 countries reported cases of the virus contractions, and the World Health Organization declared the rapidly spreading outbreak a pandemic. While writing this article, there are more than 10,000,000 reported confirmed cases of the infection worldwide [1], and current projections anticipate approximately 175,000 deaths by the end of August 2020 in the US [2]. Promptly after the outbreak, it became evident that the supply chains will be severely disrupted, and depending on the outbreak size, demand disruption, and recovery synchronicity of the firms, the performance of the supply chains will be proportionally deteriorated [3].

In the context of supply chains, epidemic outbreaks are a form of disruption risk that impacts operational performance measures. Operational activities ordinarily involve humans and are prone to interruption if the employees are in close physical contact. Physical distancing is advocated by epidemiologists as an infection spread mitigation strategy [4–6], and can be adapted to labor-intensive operational tasks for minimizing the likelihood of personnel contracting the infection, and hence, maintaining the performance.

Manual order picking operations are highly labor-intensive, and due to their criticality, cannot be suspended during a crisis. Furthermore, since the pickers may work in close vicinity, an order picking environment may bring about an elevated risk of virus transmission among the personnel. Thus, physical distancing policies can be utilized to reduce the likelihood of virus propagation among the pickers. Depending on the type of picking environment, such policies may have different consequences and must be selected in accordance with parameters such as picking objectives and warehouse design. For example, a warehouse whose operations are centered around wave picking may need to implement physical distancing practices that have the least conflict with its picking makespan.

The objective of this paper is to revisit the well-known order picking problem in manual order picking warehouse while considering physical distancing practices and aims to investigate the performance and managerial implications of implementing such practices. For this purpose, a tri-objective formulation of the order picking problem is proposed where the total picking time, makespan, and the time over which the workers are picking closer than a minimum physical distance are to be minimized. The third objective, henceforth referred to as picking overlap, is

[*] Corresponding author.
E-mail addresses: ardjmand@ohio.edu (E. Ardjmand), manjeet.singh3@dhl.com (M. Singh), hs9hd@virginia.edu (H. Shakeri), a.tavasoli@albpnu.ac.ir (A. Tavasoli), youngw1@ohio.edu (W.A. Young II).

https://doi.org/10.1016/j.asoc.2020.106953
1568-4946/© 2020 Elsevier B.V. All rights reserved.

computed by measuring the time each pair of pickers are stationed at the proximity of each other as controlled by a distance parameter Δ. Thus, a more substantial overlap must be calculated if more than two pickers are positioned close to each other during picking. Indeed, a more congested area has a higher infection transmission potential. As defined in this study, picking overlap is mainly concerned with the picking time when the pickers have limited motions due to their physical position in relation to the pick locations. One has to draw a distinction between the picking overlap and walking overlap where at least one picker is walking. This study is primarily concerned with the picking overlap. Thus, walking pickers are exempt from the overlap computations. This assumption works best for many warehouses where the picking aisles are wide enough for the pickers to respect a physical distance while passing by each other. It is noteworthy that the picking overlap can provide the decision-makers with a proxy measure of the picking plus walking overlap. Thus, it can be utilized in warehouses with narrow aisles where the aisles' distance does not allow for physical distancing. Additionally, the problem is parametrized by a distance constant Δ that can be adjusted to favor the situations where it is preferred to have a single picker in an aisle.

This study is modeled after the warehouse operations of a major logistics company located in the US and uses the order level data provided by the company. Moreover, the company's order picking practices are assessed to draw a comparison between the current state of affairs and solutions that acknowledge physical distancing. The company's current order picking operations are highly geared towards minimizing the total picking time, and as will be illustrated, improving the physical distancing measure while maintaining the status quo's level of picking efficiency is possible. In fact, there seems to be some degree of positive synergy among picking overlap and total picking time objectives. However, makespan demonstrates intense conflict with the other two objectives. This observation has unpropitious implications for wave picking environments, where makespan is typically prioritized. In other words, physical distancing practices seem to be more challenging to implement in wave picking warehouses when compared to the warehouses, where the objective is to minimize the total picking time.

Conventionally, Order picking problems consist of several sub-problems. In order picking with physical distancing (OPPD) considerations, one is interested in batching a set of orders and devising a picking route for each batch to optimize a set of predefined objectives while minimizing picking overlap amid the stationary pickers. The warehouse of this study utilizes the S-shape routing policy. Thus, the main results are derived using the same procedure. However, to explore the effect of routing policy on the results, the Midpoint policy is assessed in the numerical experiment section as well. While some evidence shows that the S-shape routing policy has anti-congestion properties [7,8], the results of applying the Midpoint policy in the context of the presented problem, does not confirm this hypothesis.

As previously shown in the literature, order batching is NP-hard and challenging to solve for large instances using exact methods [9]. This challenge is more amplified when multiple objectives are considered. This study proposes a mathematical tri-objective model along with three evolutionary algorithms, namely, Non-dominated Sorting GA-II (NSGAII) [10], strength Pareto evolutionary algorithm (SPEA2) [11], and Non-dominated Sorting GA-III (NSGAIII) [12], to tackle the problem. The primary motivation for using three evolutionary metaheuristics comes from some earlier evidence in the literature where evolutionary methods have yielded promising results for multi-objective order picking problems [13,14]. Thus, this study will delve deeper into assessing evolutionary algorithms' applications in such problems as a side objective. There are numerous well-known approaches within the realm of multi-objective evolutionary methods, each utilizing a different strategy. While being far from being an exhaustive list of algorithms, NSGAII (with a non-dominated sorting strategy), NSGAIII (with a reference-point-based procedure), and SPEA2 (with a density estimation technique) represent three important approaches in multi-objective metaheuristics. Thus, evaluating these methods can potentially guide future direction on using multi-objective techniques in order picking. In this study, the company's order batching method is utilized to seed the proposed metaheuristics and significantly improve the results. The proposed methods' solutions are compared against the current practices of the studied company to draw managerial insights. This study contributes to the current body of the literature by considering a picking overlap objective whose utilization can mitigate the risk of infectious disease spread, such as the one caused by COVID-19. Additionally, this study's methods can be applied to order picking operations during flu seasons to avoid downtime in the warehouses due to personnel sickness.

The remainder of this paper is structured as follows. In Section 2, related literature is reviewed. In Section 3, the structure of the problem is explained. Section 4 outlines the proposed optimization model. In Section 5, the order batching method of the case study's company and the proposed metaheuristics are introduced. Section 6 is dedicated to the numerical experiments. Section 7 discusses the managerial implications of the findings, and finally in Section 8 overall conclusion is stated.

2. Literature review

The presented study's central contribution revolves around mitigating the risk of contracting infectious diseases in warehouses by reevaluating the order picking problems and incorporating a picking overlap objective. The rationale behind the picking overlap objective is that being positioned reasonably apart while picking, abates the chance of infection spread. However, the extent of order picking performance deterioration, if any, in the presence of physical distancing is not fully investigated in the literature. Furthermore, to the best of authors' knowledge, picking overlap has not been recognized as an independent objective previously. This section examines the existing order picking literature that is most relevant to the subject of this study.

Structurally, and from an operational standpoint, order picking problems are composed of four sub-problems of order batching, batch assignment, batch sequencing, and picker routing [15–17], among which order batching and picker routing are the subjects of this study. Order batching is primarily revolving around grouping a set of customer orders in such a manner to achieve a set of operational objectives such as total travel time or tardiness minimization [18–20]. Typically, batching alone is not sufficient to accomplish the objectives, and typically is accompanied by a routing decision. The literature offers some evidence of performance independence between batching and routing sub-problems in particular cases [21]. However, an advantageous bilateral effect between these two sub-problems has been observed in several instances [8,15,22,23].

In the picker routing problem, the objective is to form the shortest possible path for picking the batches. Warehouse routing problems are usually modeled as a special case of the traveling salesman problem (TSP), known as Steiner TSP, where each visiting node has a maximum degree of four [17]. Since it was shown that Steiner TSPs encountered in rectangular warehouses with two cross-aisles are solvable in polynomial time [24], several solvable cases of warehouse TSPs have been introduced, and efficient methodologies are developed for more general warehouse architectures [25–27]. A major disadvantage of exact routing methods is the complexity of their produced paths for pickers to follow [8]. Thus, warehouses prefer to employ heuristic methods that generate paths with predictable patterns. The most widely used routing policies in the warehouses are S-shape, midpoint, return, and largest gap [28]. Besides exact and heuristic approaches, some studies have utilized metaheuristic methods for routing [16,29–31].

When integrated, order batching and picker routing can result in significant efficiency gains [15]. However, the integrated problem poses computational challenges. To this end, exact and scalable (both exact and heuristic) methods of order batching and picker routing are extensively studied [67,68]. To tame the complexity of the integrated problem, and replicate a more realistic scenario, some studies have concentrated on exact batching solutions in the presence of heuristic routing policies [7,9,51,61]. On the other hand, some studies have exclusively applied heuristics and metaheuristics [29,30,46,47,57,62,69], or a different combination of exact and heuristic approaches [17].

From a methodological point of view, the literature on multi-objective order picking problems is limited [68], and few studies tackle two objectives [13,14,39,70–74]. To the best of the authors' knowledge, with the exception of NSGAII [13,14], SPEA2 and NSGAIII have not been applied to multi-objective manual order picking problems before. Moreover, as discussed earlier, these three methods represent three notable approaches (i.e., non-dominated sorting, reference points, and density estimation) in the multi-objective evolutionary algorithms. Thus, their application in the context of the proposed tri-objective model and the results of numerical experiments can offer an initial investigation of utilizing these multi-objective methods in the manual order picking literature.

One distinctive feature of order picking problems is their choice of the objective function. Three objectives of total travel time, tardiness, and makespan are the most frequently utilized objectives in the literature, among which total travel time and makespan are the most and least regularly considered objectives [17]. While at least one of these objectives seem to fulfill the requirements of any warehouse, the recent supply chain disruptions pertaining to coronavirus outbreak (COVID-19/SARS-CoV-2) pandemic points to a possible gap in the order picking operational objectives. In a practical context, it is essential to protect the pickers from infectious diseases by assigning them picking tours that have less overlap and are more compliant with physical distancing recommendations. This study addresses the existing literature gap by introducing an overlap objective that acts as a measure of physical distancing practices in conjunction with total travel time and makespan objectives. This research intends to shed light on the performance implications of the overlap objective in a real setting by utilizing efficient metaheuristic designs.

There is a vast literature on order batching and picker routing, and this study does not intend to examine the existing body of research fully. Thus an interested reader is referred to [8,75,76], and Table 1 in which some of the most notable order batching and picker routing studies, sub-problems they have considered, methods applied, and the objectives utilized are summarized.

3. Problem statement

This study's warehouse is modeled after a real warehouse belonging to a well-known logistics company in the USA. The warehouse operates in a wave picking setting where the profile of the next wave is known beforehand. Waves are ordinarily large, and each day is dedicated to picking a single wave. Warehouse's shape is rectangular with multiple parallel aisles, and two front and rear cross aisles. The walking speed of pickers is assumed constant, and the crossing time between two sides of an aisle is negligible. Each shelf can hold multiple products, and each product is located on only one shelf. There are N orders, each consisting of multiple products, to be picked. The warehouse stocks M different products, and there are a sufficient number of pickers to pick the orders. Picking commences with batching the orders and retrieving their products using S-shape routing policy starting from origin v_0. Each picker is assigned a single batch, and there are a sufficient number of pickers to manage all batches. All tours begin simultaneously and at time 0. Orders cannot split among multiple picking tours, and orders assigned to a batch must not exceed pickers' capacity Q. Following physical distancing practices, pickers are preferred to be positioned less than Δ (i.e., minimum social distance) units of a length away while in picking position. Thus, physical distancing considerations are only applied to stationary pickers. Products' dimensions are reasonably similar (i.e., there is no unusually large/small product), and instead of total volume, the capacity is determined as the number of items that a picker can handle. All products are located within reach of the pickers, and therefore, vertical distance is insignificant. Picking time at a location comprises the duration of searching, picking, and checking, and is estimated based on the number of units picked at the location. Fig. 1 depicts the order of operations in OPPD, in which initially a list of orders, each containing multiple items, is received and compiled. Next, the orders are grouped into batches. Finally, each batch is picked by traversing a route through the warehouse and by a picker. As presented in Fig. 1, routing depends on batching decisions, and these two create an intertwined problem.

While the presented study is primarily designed based on the logistics' company warehouse, it can be extended to other manual order picking environments as well. This extension is mainly viable due to the structure of the overlap objective function, which is based on the time pickers operate in proximity to each other. Almost in every manual order picking system, the pickers are bound to work close to each other occasionally and, thus, are prone to the risk of infection spread. Considering the pick overlap as an assessable picking performance that estimates the time pickers are closer than a minimum physical distance, it can be universally applied to different settings in the manual order picking literature.

4. Model

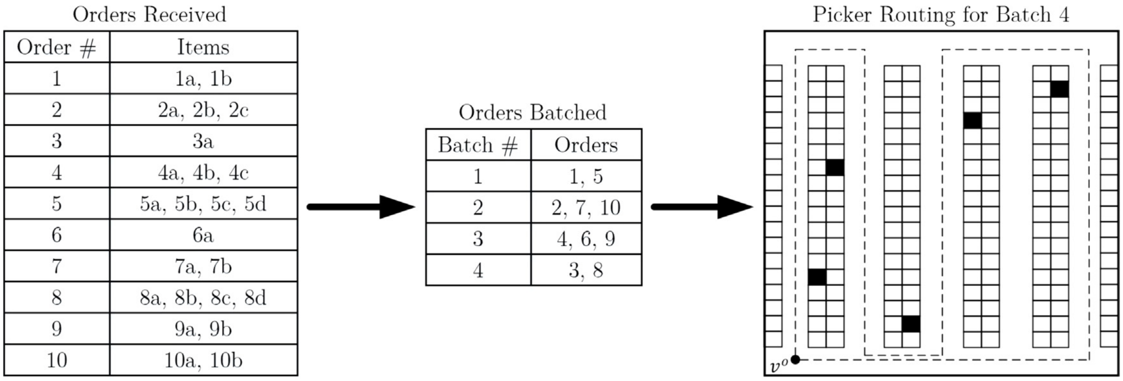
In order picking, warehouse graph representation is a central modeling component. Depending on the routing policy, it is possible to exploit the warehouse graph for obtaining optimum routing. For example, if an exact routing policy is of interest, it suffices to consider the picking locations solely, and the distance between two picking nodes will be the shortest distance between the nodes. In this manner, the warehouse graph will be reduced to a considerably smaller graph that is more convenient to be manipulated by exact methods [24]. Furthermore, warehouse graphs lend them to efficient valid inequalities that are demonstrated to be highly effective for modeling order picking problems [61]. When using heuristic routing policies, it is not necessary to solve for the sequence of the picking nodes, and thus, a model does not need to obtain the nodes' enter and exit times and hence, a less complicated model [7].

When modeling OPPD with an S-shape routing policy, the traversed distance between two nodes is not necessarily the shortest. Thus, the problem's graph cannot be treated as in the exact methods. On the other hand, the entrance and the exit times of each picking location need to be accurately determined in the model to calculate the picking overlap. Therefore, the benefits of sequence-less models cannot be utilized. This structure of OPPD

Table 1
Related studies.

Study	Assignment	Batching	Routing	Sequencing	Methods	Objective(s)
[24]			✓		Dynamic programming	Total distance/time
[32]		✓			Branch and bound	Makespan
[33]			✓		Dynamic programming	Total distance/time
[34]			✓		Heuristics	Total distance/time
[35]		✓			Cluster analysis	Order similarity
[36]		✓			Association rule mining	Order similarity
[9]		✓			Column generation	Total distance/time
[37]		✓	✓		Heuristic	Response time
[38]		✓			Heuristic	Total distance/time
[21]		✓	✓		Heuristic	Total distance/time
[39]		✓	✓		Multiple genetic algorithm	Total distance time and due date
[40]		✓			Variable neighborhood search	Total distance/time
[41]		✓	✓		K-means self-organizing map	Total distance/time Picking vehicle utility
[42]		✓			Heuristic	Makespan
[43]		✓			Hill climbing Tabu search	Total distance/time
[44]		✓			Heuristic	Total distance/time
[45]		✓	✓		Tabu search	Total distance/time
[46]		✓	✓		Association rule mining Genetic algorithm	Total tardiness
[47]		✓		✓	Heuristic	Total tardiness
[48]			✓		Heuristic	Total distance/time
[49]		✓	✓		A*-algorithm simulated annealing	Total distance/time
[31]		✓	✓	✓	Genetic algorithm Ant colony optimization	Total tardiness
[30]		✓	✓		Particle swarm optimization Ant colony optimization	Total tardiness
[50]		✓		✓	Variable neighborhood search	Total tardiness
[51]		✓	✓		Column Generation	Travel cost
[7]		✓			Heuristic Tabu Search	Total distance/time
[52]		✓			Group genetic algorithm	Workload balance
[53]			✓		Heuristic	Total distance/time
[54]		✓	✓		Particle swarm optimization	Total distance/time
[55]			✓		Heuristic	Total distance/time U-turns
[14]		✓	✓		NSGA-II	Total distance/time
[56]			✓		Variable neighborhood search	Total distance/time
[57]			✓		Tabu search	Total distance/time
[23]		✓	✓		Ant colony optimization	Total distance/time
[58]		✓		✓	Variable neighborhood search	Total tardiness
[19]		✓			Variable neighborhood search	Total distance/time
[59]	✓	✓			Parallel Variable neighborhood search	Batch retrieval time
[26]			✓		Dynamic programming	Total distance/time
[15]	✓	✓	✓	✓	Variable neighborhood search	Total tardiness
[60]			✓		Genetic algorithm	Total distance/time
[61]		✓	✓		Branch & bound	Total distance/time
[16]	✓	✓	✓		Lagrangian decomposition-particle swarm optimization Parallel simulated annealing-ant colony optimization	Makespan
[62]			✓		A Memetic Algorithm Simulated annealing	Total time
[63]			✓		Branch & bound	Makespan
[64]	✓			✓	Heuristic	Total tardiness
[18]		✓			Variable neighborhood search Tabu search	Total distance/time
[65]		✓			Heuristic	Makespan delivery cost
[66]	✓	✓	✓		Heuristic	Total distance/time
[17]	✓	✓	✓		Exact/Heuristic	Makespan
[13]	✓	✓	✓		Genetic algorithm Coevolutionary genetic algorithm Archived multi-objective simulated annealing	Total distance/time Makespan

formulation poses a unique challenge: determining the entry times of each picking node while traversing a path that is not necessarily the shortest. To resolve this issue, it is necessary to ascertain the entry times of all intermediary nodes where no picking takes place. This necessity itself poses a new challenge. If the entry times of each node (intermediary and picking) are to be determined, then the typical sub-tour elimination constraints cannot be applied because some nodes are visited more than once. Fig. 2 depicts an S-shape picking tour where some nodes are visited twice. Typical sub-tour elimination constraints cannot



Orders Received

Order #	Items
1	1a, 1b
2	2a, 2b, 2c
3	3a
4	4a, 4b, 4c
5	5a, 5b, 5c, 5d
6	6a
7	7a, 7b
8	8a, 8b, 8c, 8d
9	9a, 9b
10	10a, 10b

Orders Batched

Batch #	Orders
1	1, 5
2	2, 7, 10
3	4, 6, 9
4	3, 8

Fig. 1. Order of operations in OPPD.

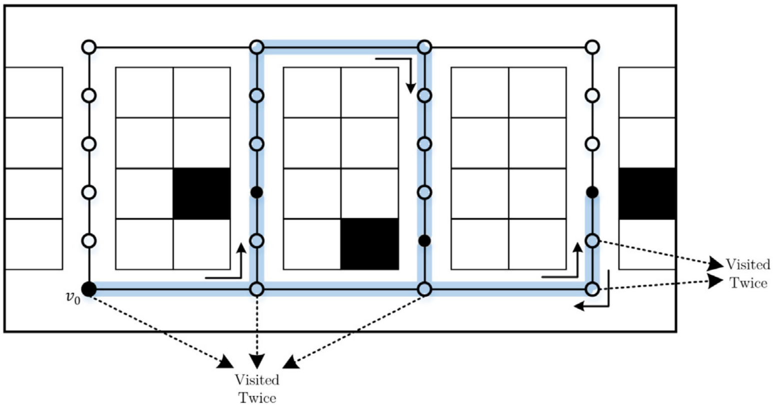


Fig. 2. An example of a tour with nodes that are visited more than once.

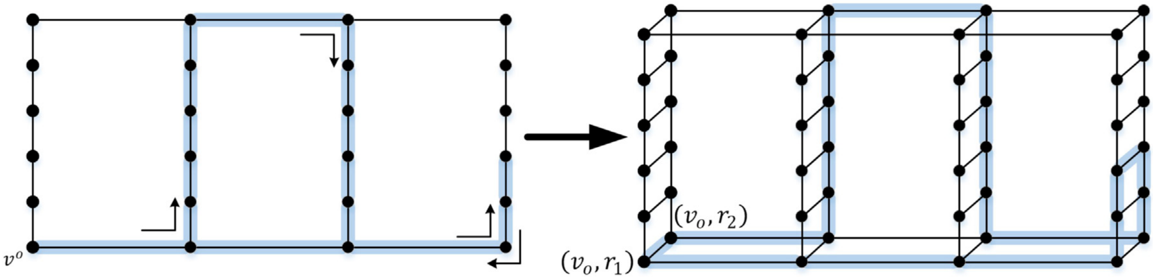


Fig. 3. Transforming a tour to visit each location at most once.

handle the situations where a location is visited more than once. Thus, the Steiner graph of a warehouse is inadequate for modeling OPPD.

To overcome the problem of multiple location visits and implement sub-tour elimination constraints, multiple connected replications of the Steiner graph, henceforth referred to as extended Steiner graph (or extended graph in short), must be utilized. In the example of Fig. 2, an extended graph where sub-tour elimination constraints are fulfilled can be obtained by connecting two replicas of a Steiner graph. In this regard, each node in the original Steiner graph is associated with two nodes (replicas) in the extended graph. The distance between the replicas of a node in the extended graph is assumed to be 0, to preserve the length of a tour. The new higher dimensional, extended graph can be used to create a tour where each node is visited at most once, and sub-tour elimination constraints can be implemented. Fig. 3 illustrates how to utilize an extended Steiner graph to transform the tour of Fig. 2 with two replicas r_1 and r_2 for each node and visit each location at most once. In Fig. 3, instead of backtracking the tour, and for avoiding a location revisitation, the tour stretches to another replica of the Steiner graph in the extended graph. Since this lateral transition has no time associated with it, it does not alter the primary duration of the tour.

Considering the extended Steiner graph representation introduced earlier, the warehouse of this study can be modeled as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes, and edges of the graph. Each node $v \in \mathcal{V}$ has R replicas, corresponding to R Steiner graphs. Storage node of the jth product is denoted by $l(j)$. In other words, l acts as a function whose input is a product and outputs a storage node in the Steiner graph (i.e., $l : \mathcal{M} \rightarrow \mathcal{V}$). Function $a : \mathcal{M} \rightarrow \mathcal{K}$ associates the location of each product

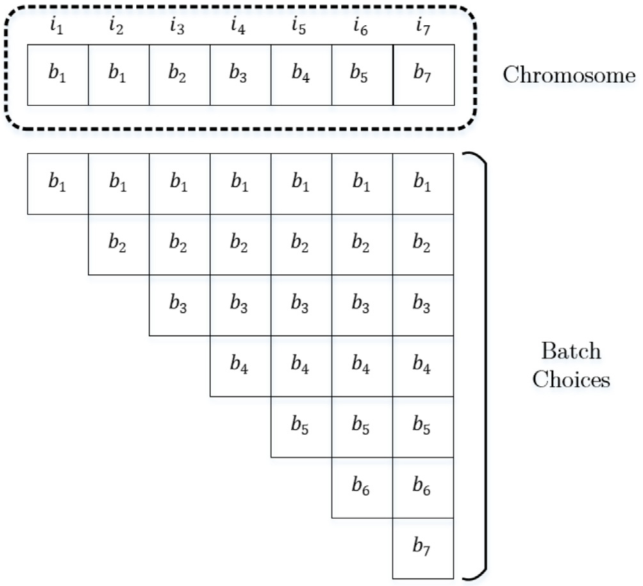

i_1 i_2 i_3 i_4 i_5 i_6 i_7
b_1 b_1 b_2 b_3 b_4 b_5 b_7
Chromosome
b_1 b_1 b_1 b_1 b_1 b_1 b_1
b_2 b_2 b_2 b_2 b_2 b_2
b_3 b_3 b_3 b_3 b_3
b_4 b_4 b_4 b_4
b_5 b_5 b_5
b_6 b_6
b_7
Batch Choices

Fig. 4. A chromosome and its batch choices for seven orders (i.e., $N = B = 7$).

Table 2
Parameters and their levels for NSGAII, SPEA2, and NSGAIII.

Parameter	Level 1	Level 2	Level 3
p_c	0.60	0.70	0.80
p_m	0.05	0.10	0.15
p_g	0.25	0.50	0.75
$\mu+\lambda$	50+150	100+100	150+50

$j \in \mathcal{M}$ to an aisle $k \in \mathcal{K}$. In the proposed model, functions l and a is separated from indices by a comma to avoid confusion. Characteristic function $\bar{q} : \mathcal{N} \times \mathcal{M} \to \{0, 1\}$ outputs 1 if order $i \in \mathcal{N}$ contains product $j \in \mathcal{M}$, otherwise outputs 0. Characteristic function $\delta_{\Delta} : \mathcal{V} \times \mathcal{V} \to \{0, 1\}$ determines whether the distance between two nodes v_1 and v_2 is less than the minimum acceptable physical distance Δ or not. Note that in this study, the terms 'node' and 'location' are often used interchangeably, and when a physical site is intended, the term physical location is preferred. The indices, parameters/functions, and decision variables of the model are defined as follows:

Indices:

$i \in \mathcal{N} = \{1, 2, \dots, N\}$: orders
$j \in \mathcal{M} = \{1, 2, \dots, M\}$: products
$k \in \mathcal{K} = \{1, 2, \dots, K\}$: aisles
$b \in \mathcal{B} = \{1, 2, \dots, B\}$: batches
$v \in \mathcal{V} = \{1, 2, \dots, V\}$: nodes in the Steiner graph
$r \in \mathcal{R} = \{1, 2, \dots, R\}$: node visit (in this study $R = 2$)

Parameters/Functions:

Δ : walking time of the minimum physical distance
$d_{v_1v_2}$: travel time between nodes v_1 and v_2
q_{ij} : units of product j in order i
Q : picking capacity
T : picking time of a unit of a product
Λ : A sufficiently large number
$l(j)$: location of product j in the Steiner graph
$a(v)$: aisle of node v
$\bar{q}_{ij}$: 1 if order i contains product j, otherwise 0
$\delta_{\Delta}(v_1, v_2)$: 1 if $d_{v_1v_2} < \Delta$, 0 otherwise

Decision Variables:

x_{bi} : 1 if order i is assigned to batch b, otherwise 0
$y^{br_1r_2}_{v_1v_2}$: 1 if in batch b, node v_2 is visited for the r_2th time right after node v_1 is visited for the r_1th time, otherwise 0
u_{bk} : 1 if aisle k is visited in batch b, otherwise 0
p_{bk} : 1 if aisle k is the rightmost aisle visited in batch b, otherwise 0
τ_{bv} : picking time spent on node v in batch b
t^{+}_{bvr} : time of entering node v for the rth time in batch b
t^{-}_{bvr} : time of exiting node v for the rth time in batch b
f_b : finishing time of batch b
C_{max} : makespan
$\alpha^{b_1b_2}_{v_1v_2}$: earliest time of exiting node v_1 in batch b_1 or node v_2 in batch b_2
$\beta^{b_1b_2}_{v_1v_2}$: latest time of entering node v_1 in batch b_1 or node v_2 in batch b_2
$\phi^{b_1b_2}_{v_1v_2}$: picking time overlap between node v_1 in batch b_1 and node v_2 in batch b_2

The proposed model $\boldsymbol{P}$ is formulated as follows:

$$\min \quad F = \sum_{b \in \mathcal{B}} f_b \tag{1}$$

$$\min \quad C_{max} \tag{2}$$

$$\min \quad \Phi = \sum_{b_1 \in \mathcal{B}} \sum_{\substack{b_2 \neq b_1 \\ b_2 \in \mathcal{B}}} \sum_{v_1 \in \mathcal{V}} \sum_{v_2 \in \mathcal{V}} \phi^{b_1b_2}_{v_1v_2} \tag{3}$$

subject to Eqs. (4)–(35) which are given in Box I.

In model $\boldsymbol{P}$, (1) to (3) measure the total travel time, makespan and the total overlap, respectively. Constraint (4) assigns each order to exactly one batch and in order to break solution symmetries, it assigns order i_1 to batch b_1, order i_2 to either batch b_1 or batch b_2, and so forth [61]. Batch capacity is imposed in Constraint (5). Constraint (6) ensures that picking activities take place when the first replica of a picking node is visited. Constraint (7) limits the number of nodal visits to one. Constraint (8) makes the caps the number of visiting origin node to one. Constraint (9) equates the number of entries and exits for all nodes replicas. Constraints (10) and (11) guarantee that there is no reversal and double traversal on the back aisle's edges. Constraints (12) allows a movement reversal only on the rightmost aisle of each batch. Constraints (13) to (18) determine the visiting and rightmost aisles of each batch. Constraint (19) calculates the picking time on each node. Constraints (20) to (22) specifies the entry and exit time of each node and eliminates the sub-tours. Constraint (23) measures the finishing time of each batch. Constraint (24) determines the makespan, and constraints (25)–(27) determine the overlap among nodes across different batches. Finally, constraints (28) to (31) specify the type and domain of each decision variable. As can be observed, constraints (25) to (27) in model $\boldsymbol{P}$ are non-linear. However, linearizing these constraints are quite straightforward and commercial solvers such as Gurobi automatically handles them.

5. Methodology

In this section, the company's order batching method is described. Moreover, the application of three metaheuristics, namely, NSGAII, SPEA2, and NSGAIII, in solving the proposed model, $\boldsymbol{P}$, is explained. The proposed metaheuristics use the company's order batching schema as a seeding procedure for initializing the evolution process. As will be illustrated later, using this seeding procedure improves the results significantly. Finally, a full factorial experiment is conducted to determine the best set of parameters for each metaheuristic.

$$\sum_{b \in \{b' \in \mathcal{B}: b' \leq i\}} x_{bi} = 1 \qquad \forall i \in \mathcal{N} \tag{4}$$

$$\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} x_{bi} q_{ij} \leq Q \qquad \forall b \in \mathcal{B} \tag{5}$$

$$\sum_{\substack{v_1 \in \mathcal{R} \\ (v_1, l(j)) \in \mathcal{E}}} y^{b11}_{v_1, l(j)} \geq x_{bi} \bar{q}_{ij} \qquad \forall b \in \mathcal{B}; i \in \mathcal{N}; j \in \mathcal{M}, \bar{q}_{ij} = 1 \tag{6}$$

$$\sum_{r_1 \in \mathcal{R}} \sum_{\substack{v_1 \in \mathcal{R} \\ (v_1, v_2) \in \mathcal{E}}} y^{br_1 r_2}_{v_1 v_2} \leq 1 \qquad \forall b \in \mathcal{B}; v_2 \in \mathcal{V}; r_2 \in \mathcal{R} \tag{7}$$

$$\sum_{r_1 \in \mathcal{R}} \sum_{r_2 \in \mathcal{R}} \sum_{\substack{v_1 \in \mathcal{R} \\ (v_1, 0) \in \mathcal{E}}} y^{br_1 r_2}_{v_1 0} \leq 1 \qquad \forall b \in \mathcal{B} \tag{8}$$

$$\sum_{r_1 \in \mathcal{R}} \sum_{\substack{v_2 \in \mathcal{E} \\ (v_1, v_2) \in \mathcal{E}}} y^{br_1 r_2}_{v_1 v_2} - \sum_{r_3 \in \mathcal{R}} \sum_{\substack{v_2 \in \mathcal{E} \\ (v_2, v_3) \in \mathcal{E}}} y^{br_2 r_3}_{v_2 v_3} = 0 \qquad \forall b \in \mathcal{B}; v_2 \in \mathcal{V}; r_2 \in \mathcal{R} \tag{9}$$

$$\sum_{r_1 \in \mathcal{R}} \sum_{r_2 \in \mathcal{R}} y^{br_1 r_2}_{v_1 v_2} = 0 \qquad \forall b \in \mathcal{B}; (v_1, v_2) \text{ on back aisle}, v_2 < v_1 \tag{10}$$

$$\sum_{r_1 \in \mathcal{R}} \sum_{r_2 \in \mathcal{R}} y^{br_1 r_2}_{v_1 v_2} \leq 1 \qquad \forall b \in \mathcal{B}; (v_1, v_2) \text{ on back aisle}, v_2 > v_1 \tag{11}$$

$$\sum_{r_1 \in \mathcal{R}} \sum_{r_2 \in \mathcal{R}} y^{br_1 r_2}_{v_1 v_2} + \sum_{r_2 \in \mathcal{R}} \sum_{r_1 \in \mathcal{R}} y^{br_2 r_1}_{v_2 v_1} \leq 1 + p_{b, a(v_1)} \qquad \forall b \in \mathcal{B}; (v_1, v_2) \in \mathcal{E}, a(v_1) = a(v_2) \tag{12}$$

$$y^{br_1 r_2}_{v_1 v_2} \leq u_{b, a(v_2)} \qquad \forall b \in \mathcal{B}; (v_1, v_2) \in \mathcal{E}, v_2 \text{ not on cross-aisles}; r_1, r_2 \in \mathcal{R} \tag{13}$$

$$x_{bi} \bar{q}_{ij} \leq u_{b, a(l(j))} \qquad \forall b \in \mathcal{B}; i \in \mathcal{N}; j \in \mathcal{M}, \bar{q}_{ij} = 1 \tag{14}$$

$$\sum_{i \in \mathcal{N}} \sum_{\substack{j \in \mathcal{M} \\ a(l(j)) = k}} x_{bi} \bar{q}_{ij} \geq u_{bk} \qquad \forall b \in \mathcal{B}; k \in \mathcal{K} \tag{15}$$

$$u_{bk} - \sum_{k' \in \{k'' \in \mathcal{K}: k'' \geq k+1\}} u_{bk'} \leq p_{bk} \qquad \forall b \in \mathcal{B}; k \in \mathcal{K} \tag{16}$$

$$p_{bk} \leq u_{bk} \qquad \forall b \in \mathcal{B}; k \in \mathcal{K} \tag{17}$$

$$\sum_{k \in \mathcal{K}} p_{bk} = 1 \qquad \forall b \in \mathcal{B} \tag{18}$$

$$\sum_{i \in \mathcal{N}} \sum_{\substack{j \in \mathcal{M} \\ l(j) = v}} T x_{bi} q_{ij} = \tau_{bv} \qquad \forall b \in \mathcal{B}; v \in \mathcal{V} \tag{19}$$

$$t^{+}_{bv1} + \tau_{bv} = t^{-}_{bv1} \qquad \forall b \in \mathcal{B}; v \in \mathcal{V} \setminus \{v_0\} \tag{20}$$

$$t^{+}_{bvr} = t^{-}_{bvr} \qquad \forall b \in \mathcal{B}; v \in \mathcal{V} \setminus \{v_0\}; r \in \mathcal{R} \setminus \{1\} \tag{21}$$

$$t^{-}_{bv_1 r_1} + d_{v_1 v_2} - \Lambda(1 - y^{br_1 r_2}_{v_1 v_2}) \leq t^{+}_{bv_2 r_2} \qquad \forall b \in \mathcal{B}; (v_1, v_2) \in \mathcal{E}; r_1, r_2 \in \mathcal{R} \tag{22}$$

$$t^{+}_{bvr} \leq f_b \qquad \forall b \in \mathcal{B}; v \in \mathcal{V}; r \in \mathcal{R} \tag{23}$$

$$C_{max} \geq f_b \qquad \forall b \in \mathcal{B} \tag{24}$$

$$\min(t^{-}_{b_1 v_1 1}, t^{-}_{b_2 v_2 1}) = \alpha^{b_1 b_2}_{v_1 v_2} \qquad \forall b_1, b_2 \in \mathcal{B}, b_1 \neq b_2; v_1, v_2 \in \mathcal{V} \tag{25}$$

$$\max(t^{+}_{b_1 v_1 1}, t^{+}_{b_2 v_2 1}) = \beta^{b_1 b_2}_{v_1 v_2} \qquad \forall b_1, b_2 \in \mathcal{B}, b_1 \neq b_2; v_1, v_2 \in \mathcal{V} \tag{26}$$

$$\max(\alpha^{b_1 b_2}_{v_1 v_2} - \beta^{b_1 b_2}_{v_1 v_2}, 0) = \phi^{b_1 b_2}_{v_1 v_2} \qquad \forall b_1, b_2 \in \mathcal{B}, b_1 \neq b_2; v_1, v_2 \in \mathcal{V}, \delta_{\Delta}(v_1, v_2) = 1 \tag{27}$$

$$x_{bi} \in \{0, 1\} \qquad \forall b \in \mathcal{B}; i \in \mathcal{N} \tag{28}$$

$$y^{br_1 r_2}_{v_1 v_2} \in \{0, 1\} \qquad \forall b \in \mathcal{B}; (v_1, v_2) \in \mathcal{E}; r_1, r_2 \in \mathcal{R} \tag{29}$$

$$u_{bk}, p_{bk} \in \{0, 1\} \qquad \forall b \in \mathcal{B}; k \in \mathcal{K} \tag{30}$$

$$\tau_{bv} \in \mathbb{R}_{\geq 0} \qquad \forall b \in \mathcal{B}; v \in \mathcal{V} \tag{31}$$

$$t^{+}_{bv}, t^{-}_{bv} \in \mathbb{R}_{\geq 0} \qquad \forall b \in \mathcal{B}; v \in \mathcal{V}; r \in \mathcal{R} \tag{32}$$

Box I.

$$\alpha^{b_1 b_2}_{v_1 v_2}, \beta^{b_1 b_2}_{v_1 v_2}, \phi^{b_1 b_2}_{v_1 v_2} \in \mathbb{R}_{\geq 0} \qquad \forall b_1, b_2 \in \mathcal{B}, b_1 \neq b_2; v_1, v_2 \in \mathcal{V} \tag{33}$$

$$f_b \in \mathbb{R}_{\geq 0} \qquad \forall b \in \mathcal{B} \tag{34}$$

$$C_{max} \in \mathbb{R}_{\geq 0} \tag{35}$$

Box I. (*continued*).

5.1. Company's order batching heuristic and seeding

The company's order batching heuristic forms a single batch at each step. For this purpose, it starts with determining the most visited aisle by all unassigned orders. Then, it proceeds with selecting the orders that require a minimum extra distance to be traversed if selected into the batch. Selecting new orders is controlled through a penalty parameter that is updated every time a new order is added to the batch. The structure of this heuristic is designed with S-shape routing policy in mind. Parameters and variables used by the company's order batching heuristic are as follows:

$i \in \mathcal{N} = \{1, 2, \ldots, N\}$: orders
$k \in \mathcal{K} = \{1, 2, \ldots, K\}$: aisles
L : aisle length
H : distance between the center of two neighbor aisles
n : number of bays per aisle
$\mathcal{N}^-$: set of unassigned orders in the order pool
i' : a pseudo order for generating a batch
v_{ik} : 1 if order i visits aisle k; otherwise 0
v'_k : 1 if a batch visits aisle k; otherwise 0
k_{max} : index for the most visited aisle by all unassigned orders
$\Psi_{\mathcal{K}} = (\psi_1, \psi_2, \ldots, \psi_K)$: Calibration vector where ψ_k is the penalty for visiting aisle k by a new order
i^{best} : order with least penalty
s_i : total penalty of order i
k_{left} : index of the most left aisle that a batch visits
k_{right} : index of the most right aisle that a batch visits

Algorithm 1 outlines the company's order batching steps. Batches formed by Algorithm 1 create a solution that will be used as a seed in the proposed metaheuristics. As will be discussed later, this type of seeding has a significant positive performance effect on all proposed metaheuristics.

Table 3
Results of company's current method.

N	Q	Δ	F	C_{max}	Φ	CPU (s)
10	50	0	1310	780	0	0.10
10	50	10	1310	780	20	0.09
10	50	30	1310	780	200	0.10
10	100	0	1080	1080	0	0.09
10	100	10	1080	1080	0	0.09
10	100	30	1080	1080	0	0.10
25	50	0	2370	820	0	0.21
25	50	10	2370	820	420	0.21
25	50	30	2370	820	500	0.21
25	100	0	2100	1280	0	0.21
25	100	10	2100	1280	0	0.21
25	100	30	2100	1280	20	0.21
50	50	0	6190	860	0	0.43
50	50	10	6190	860	3330	0.41
50	50	30	6190	860	7270	0.41
50	100	0	4960	1350	0	0.41
50	100	10	4960	1350	2760	0.40
50	100	30	4960	1350	6000	0.41
100	50	0	12 160	870	0	0.82
100	50	10	12 160	870	12 820	0.86
100	50	30	12 160	870	22 670	0.83
100	100	0	9950	1360	0	0.83
100	100	10	9950	1360	8680	0.85
100	100	30	9950	1360	13 390	0.82
200	50	0	13 460	850	0	1.64
200	50	10	13 460	850	17 360	1.63
200	50	30	13 460	850	29 590	1.62
200	100	0	11 660	1350	0	1.62
200	100	10	11 660	1350	7510	1.62
200	100	30	11 660	1350	12 780	1.63
500	50	0	36 430	860	0	4.56
500	50	10	36 430	860	137 960	4.58
500	50	30	36 430	860	250 140	4.58
500	100	0	31 250	1370	0	4.54
500	100	10	31 250	1370	64 270	4.57
500	100	30	31 250	1370	120 190	4.55

5.2. NSGAII, SPEA2 and NSGAIII

The core components of evolutionary algorithms are its solution representation (i.e., chromosome), fitness function, and crossover, mutation, and selection operators. [77,78]. In this section, elements of the proposed NSGAII, SPEA2, and NSGAIII are introduced.

5.2.1. Solution representation and fitness function

A widely-used approach to define a chromosome is by utilizing a binary matrix whose arrays show the assignment of orders to catches. However, matrix representation has two main drawbacks. First, since each order is assigned to precisely one batch, the assignment matrix will be a sizeable sparse matrix and inefficient in terms of memory consumption. Second, assignment matrices can introduce symmetry and, subsequently, degenerate solutions. For example, the batches assigned to two pickers can be exchanged without altering the objective function's value. Having degenerate solutions can create numerous optimal regions in the search space and delay the convergence of algorithms. This study utilizes a vector structure for chromosomes to evade these impediments. For this purpose, the set of chromosomes is defined as $\mathbf{\Omega} = \{\Omega = (\omega_1, \omega_2, \ldots, \omega_N) : \omega_i \in \mathcal{B}_i, \forall i \in \mathcal{N}\}$ where $\mathcal{B}_i = \{b : b \in \mathcal{B}, b \leq i\}$. Each component of chromosome Ω corresponds to an order $i \in \mathcal{N}$ while its value is a batch number $b \in \mathcal{B}_i$. Thus, the first order i_1 can only be assigned to the batch b_1 while the last order i_N can be assigned to any of the batches. For this representation to function properly, it is necessary to have an equal number of batches and orders, i.e., $B = N$. Moreover, this equality is necessary to satisfy the capacity constraint in case each order occupies a single batch. Fig. 4 illustrates a chromosome consisting of seven orders and the batches to which each order belongs. Additionally, Fig. 4 depicts the batch choices that each chromosome element, associated with an order, can have. While forming the initial population, each chromosome is generated in

Table 4
Best objective values and CPU times obtained by Gurobi, NSGAII, SPEA2, and NSGAIII.

Problem			Gurobi				NSGAII				SPEA2				NSGAIII			
N	Q	Δ	F	C_{max}	Φ	CPU (s)	F	C_{max}	Φ	CPU (s)	F	C_{max}	Φ	CPU (s)	F	C_{max}	Φ	CPU (s)
10	50	0	**1230**	**530**	**0**	3600	**1230**	**530**	**0**	**82.42**	**1230**	**530**	**0**	239.01	**1230**	**530**	**0**	112.95
10	50	10	**1230**	**530**	**0**	3600	**1230**	**530**	**0**	**88.31**	**1230**	**530**	**0**	209.85	**1230**	**530**	**0**	113.89
10	50	30	**1230**	**530**	**0**	3600	**1230**	**530**	**0**	**96.37**	**1230**	**530**	**0**	191.58	**1230**	**530**	**0**	113.33
10	100	0	**1080**	**530**	**0**	3600	**1080**	**530**	**0**	**82.93**	**1080**	**530**	**0**	211.05	**1080**	**530**	**0**	113.31
10	100	10	**1080**	**530**	**0**	3600	**1080**	**530**	**0**	**90.27**	**1080**	**530**	**0**	192.91	**1080**	**530**	**0**	111.34
10	100	30	**1080**	**530**	**0**	3600	**1080**	**530**	**0**	**87.23**	**1080**	**530**	**0**	182.75	**1080**	**530**	**0**	110.45
25	50	0	NA	NA	NA	3600	**2350**	**820**	**0**	**93.49**	**2350**	**820**	**0**	372.61	**2350**	**820**	**0**	123.33
25	50	10	NA	NA	NA	3600	**2350**	**820**	120	**94.51**	**2350**	**820**	**100**	302.30	**2350**	**820**	**100**	128.55
25	50	30	NA	NA	NA	3600	**2350**	**820**	200	**93.87**	**2350**	**820**	**130**	227.53	**2350**	**820**	180	128.09
25	100	0	NA	NA	NA	3600	**2050**	**820**	**0**	**96.96**	2100	**820**	**0**	198.13	2100	**820**	**0**	128.90
25	100	10	NA	NA	NA	3600	**2050**	**820**	**0**	**100.53**	**2050**	**820**	**0**	186.52	**2050**	**820**	**0**	132.12
25	100	30	NA	NA	NA	3600	**2100**	**820**	**20**	**101.28**	**2100**	**820**	**20**	173.60	**2100**	**820**	**20**	130.85
50	50	0	NA	NA	NA	3600	**6090**	**700**	**0**	**106.05**	6110	**700**	**0**	254.14	6120	**700**	**0**	145.35
50	50	10	NA	NA	NA	3600	6090	**700**	2140	**113.94**	**6070**	**700**	**2080**	187.57	**6070**	**700**	2220	147.11
50	50	30	NA	NA	NA	3600	6060	**700**	**5050**	**113.39**	**6050**	**700**	5370	195.32	**6050**	**700**	5160	151.78
50	100	0	NA	NA	NA	3600	**4920**	**700**	**0**	**107.71**	**4920**	**700**	**0**	186.37	**4920**	**700**	**0**	140.30
50	100	10	NA	NA	NA	3600	4960	**700**	670	**118.70**	**4920**	**700**	**450**	190.12	**4920**	**700**	480	145.43
50	100	30	NA	NA	NA	3600	**4920**	**700**	2260	**120.15**	**4920**	**700**	**1930**	188.10	4960	**700**	2310	146.98
100	50	0	NA	NA	NA	3600	**12 110**	**820**	**0**	**141.03**	12 150	**820**	**0**	312.13	**12 110**	**820**	**0**	183.39
100	50	10	NA	NA	NA	3600	12 070	**820**	**10 260**	**155.74**	**11 960**	**820**	10 350	242.68	12 070	**820**	10 420	192.68
100	50	30	NA	NA	NA	3600	12 070	**820**	20 140	**158.17**	12 010	**820**	20 030	252.27	**12 000**	**820**	**19 840**	192.95
100	100	0	NA	NA	NA	3600	9950	**820**	**0**	**140.15**	**9910**	**820**	**0**	208.68	9950	**820**	**0**	179.05
100	100	10	NA	NA	NA	3600	9950	**820**	**4240**	**162.50**	**9910**	**820**	4520	233.37	9950	**820**	4750	194.00
100	100	30	NA	NA	NA	3600	9950	**820**	9330	**164.99**	**9890**	**820**	**9050**	233.95	9950	**820**	9130	197.86
200	50	0	NA	NA	NA	3600	**13 400**	**740**	**0**	**213.27**	13 460	**740**	**0**	380.41	13 410	**740**	**0**	267.37
200	50	10	NA	NA	NA	3600	**13 460**	**740**	**15 850**	**245.72**	**13 460**	**740**	15 860	353.61	**13 460**	**740**	16 050	294.82
200	50	30	NA	NA	NA	3600	13 450	**740**	27 180	**233.81**	13 460	**740**	27 390	378.01	**13 410**	**740**	**26 960**	284.34
200	100	0	NA	NA	NA	3600	11 660	**740**	**0**	**212.36**	11 660	**740**	**0**	291.66	**11 650**	**740**	**0**	262.82
200	100	10	NA	NA	NA	3600	**11 660**	**740**	**5590**	**248.58**	**11 660**	**740**	5820	319.38	**11 660**	**740**	5760	296.07
200	100	30	NA	NA	NA	3600	**11 660**	**740**	**11 640**	**249.07**	**11 660**	**740**	11 720	322.02	**11 660**	**740**	11 710	297.65
500	50	0	NA	NA	NA	3600	36 430	**770**	**0**	**607.46**	36 420	**770**	**0**	801.40	**36 350**	**770**	**0**	681.27
500	50	10	NA	NA	NA	3600	36 430	**770**	**131 490**	**702.39**	**36 380**	**770**	132 440	831.20	36 430	**770**	132 170	724.33
500	50	30	NA	NA	NA	3600	36 430	**770**	**241 880**	736.88	**36 350**	**770**	242 910	849.76	36 430	**770**	242 800	**723.90**
500	100	0	NA	NA	NA	3600	**31 250**	**770**	**0**	**605.30**	**31 250**	**770**	**0**	789.48	**31 250**	**770**	**0**	663.50
500	100	10	NA	NA	NA	3600	**31 250**	**770**	**59 870**	**759.21**	**31 250**	**770**	60 350	850.58	**31 250**	**770**	60 370	811.70
500	100	30	NA	NA	NA	3600	**31 250**	**770**	116 350	**763.66**	**31 250**	**770**	116 370	951.07	**31 250**	**770**	**115 950**	868.10

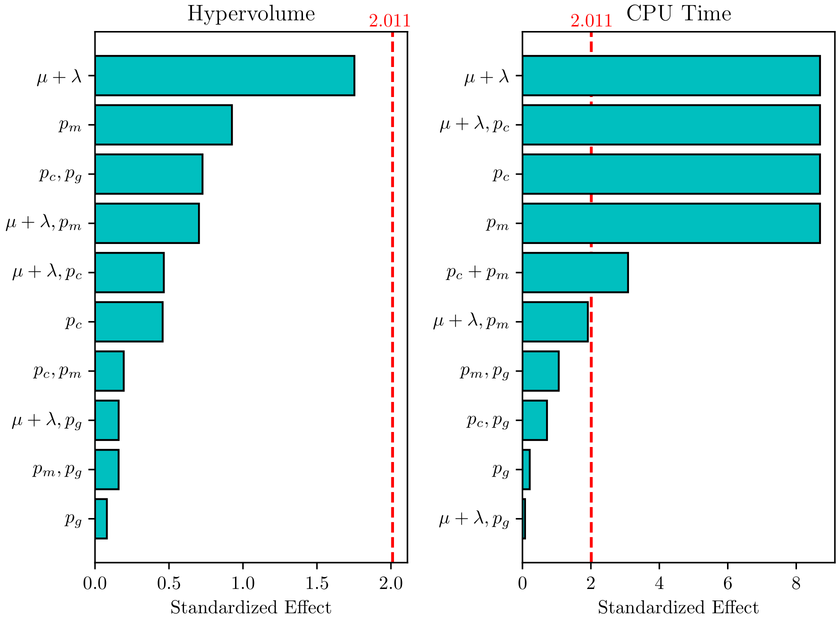
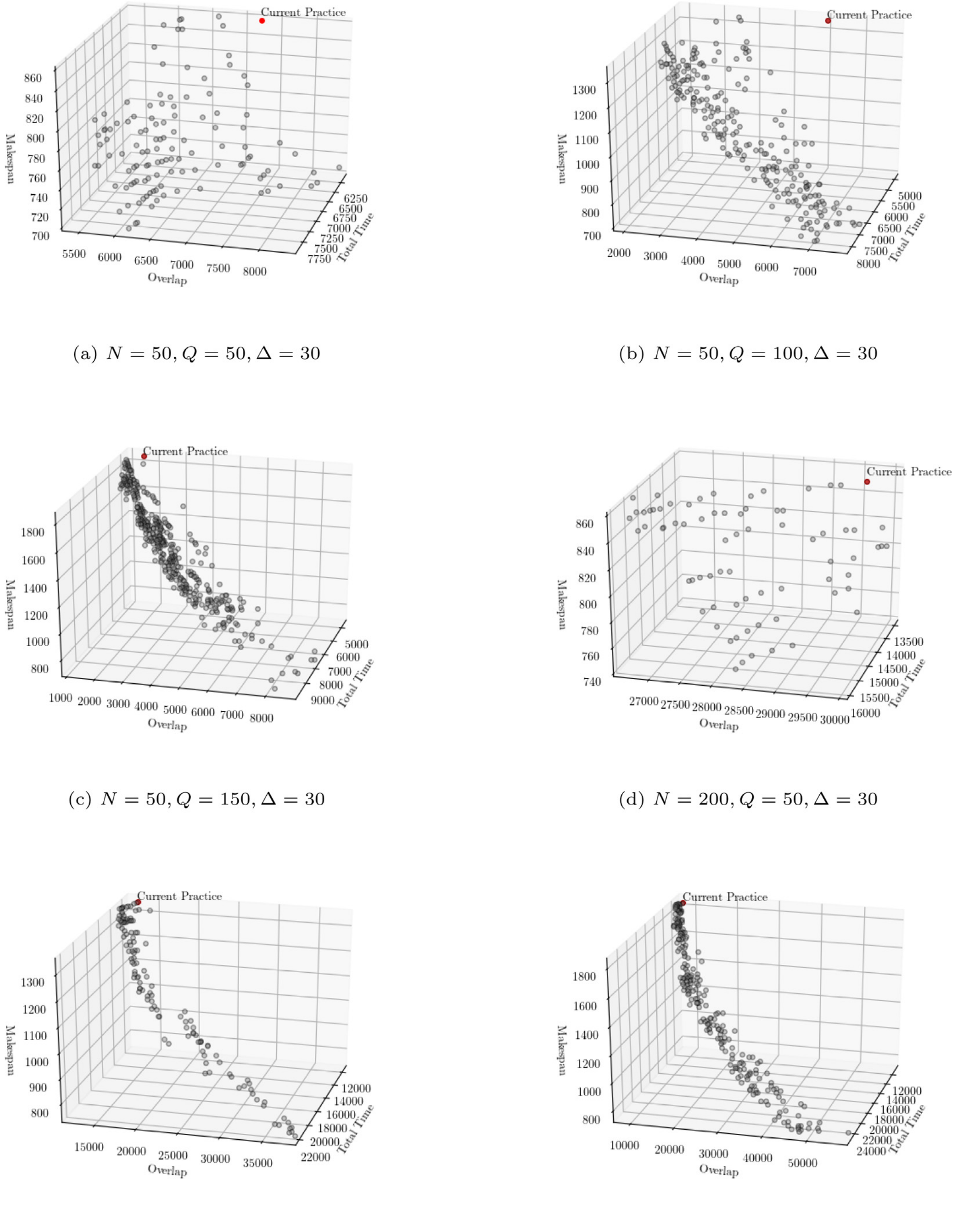


Fig. 5. Standardized effect of parameters on Hypervolume and CPU time for NSGAII with $N = 50$, $Q = 50$, and $\Delta = 30$.

such a way to ensure feasibility. Each chromosome's fitness value is computed by measuring its total travel time, makespan, and total overlap using the S-shape routing policy. Remember that in this study, it is assumed that there are a sufficient number of pickers to pick the batches, and each batch is assigned to a separate picker.

5.2.2. Crossover, mutation, and selection

In this study, a standard two-point crossover is utilized. It is noteworthy that the two-point crossover does not violate the structure of a chromosome. Thus, after crossover, for each chromosome component ω_i we have $\omega_i \in \mathcal{B}_i$ where $\mathcal{B}_i = \{b : b \in \mathcal{B}, b \leq i\}$. In other words, each order is assigned to a batch



(a) $N = 50, Q = 50, \Delta = 30$

(b) $N = 50, Q = 100, \Delta = 30$

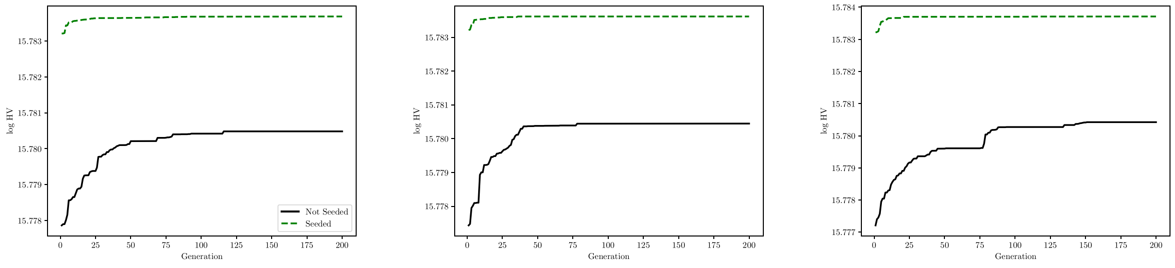
(c) $N = 50, Q = 150, \Delta = 30$

(d) $N = 200, Q = 50, \Delta = 30$

(e) $N = 200, Q = 100, \Delta = 30$

(f) $N = 200, Q = 150, \Delta = 30$

Fig. 6. Company's current practices vs. the Pareto frontier obtained by NSGAII for six problem instances.



(a) NSGAII

(b) SPEA2

(c) NSGAIII

Fig. 7. Log HV convergence plot of NSGAII, SPEA2, and NSGAIII with seeded and not seeded initialization (instance $N = 100, Q = 50, \Delta = 30$).

Table 5
Log hypervolume and CPU times of Gurobi, NSGAII, SPEA2, and NSGAIII.

Problem			Gurobi		NSGAII		SPEA2		NSGAIII	
N	Q	Δ	log^{HV}	CPU (s)	log^{HV}	CPU (s)	log^{HV}	CPU (s)	log^{HV}	CPU (s)
10	50	0	15.8191727	3600	15.8253209	**89.60**	**15.8253241**	255.09	15.8253204	117.99
10	50	10	15.8182753	3600	15.8253109	**94.79**	15.8253242	222.81	**15.8253242**	119.06
10	50	30	15.8187047	3600	**15.8253203**	**98.50**	15.8253175	202.58	15.8253142	117.05
10	100	0	15.8172561	3600	15.8254391	**93.79**	**15.8254392**	218.92	15.8254387	120.85
10	100	10	15.8226942	3600	15.8254392	**97.60**	15.8254391	210.99	**15.8254393**	116.18
10	100	30	15.8190604	3600	15.8254391	**94.42**	15.8254391	195.94	**15.8254393**	116.21
25	50	0	NA	3600	15.7952111	**103.28**	**15.7952181**	381.96	15.7952111	128.27
25	50	10	NA	3600	15.7952018	**101.69**	15.7952065	330.64	**15.7952065**	131.39
25	50	30	NA	3600	15.7951908	**101.12**	**15.7951952**	309.60	15.7951937	132.02
25	100	0	NA	3600	**15.7954343**	**102.35**	15.7954271	221.98	15.7954236	133.46
25	100	10	NA	3600	**15.7954434**	**105.54**	15.7954360	192.49	15.7954367	134.71
25	100	30	NA	3600	15.7954247	**109.00**	**15.7954254**	186.49	15.7954253	134.64
50	50	0	NA	3600	**15.8042021**	**110.50**	15.8041769	302.37	15.8041647	148.79
50	50	10	NA	3600	15.8038670	**120.11**	**15.8038729**	203.25	15.8038628	150.87
50	50	30	NA	3600	**15.8034525**	**121.19**	15.8034051	206.46	15.8034147	154.32
50	100	0	NA	3600	15.8051488	**110.68**	**15.8051700**	196.02	15.8050592	143.38
50	100	10	NA	3600	15.8049673	**121.70**	**15.8050502**	193.39	15.8050272	148.63
50	100	30	NA	3600	15.8047441	**125.80**	**15.8047962**	193.45	15.8047428	152.21
100	50	0	NA	3600	**15.7865895**	**144.63**	15.7865725	352.66	15.7865850	185.14
100	50	10	NA	3600	15.7850301	**157.79**	**15.7851035**	262.64	15.7850668	195.59
100	50	30	NA	3600	15.7836593	**162.96**	**15.7837096**	262.75	15.7836866	202.55
100	100	0	NA	3600	15.7865373	**146.74**	**15.7883620**	215.66	15.7876362	181.68
100	100	10	NA	3600	15.7873039	**166.32**	**15.7875933**	239.52	15.7873425	197.42
100	100	30	NA	3600	15.7867690	**167.51**	**15.7868749**	238.80	15.7866130	202.26
200	50	0	NA	3600	**15.7936651**	**219.02**	15.7936348	414.61	15.7936373	270.96
200	50	10	NA	3600	**15.7913076**	**250.77**	15.7912973	378.19	15.7912801	297.72
200	50	30	NA	3600	15.7896198	**248.07**	**15.7896268**	395.48	15.7896164	300.49
200	100	0	NA	3600	15.7945664	**221.31**	**15.7948829**	296.14	15.7942809	269.21
200	100	10	NA	3600	15.7936919	**257.19**	**15.7938265**	321.07	15.7937663	300.73
200	100	30	NA	3600	15.7927048	**265.03**	**15.7929294**	325.23	15.7928241	304.20
500	50	0	NA	3600	15.7695024	**622.13**	**15.7695513**	821.97	15.7694678	687.54
500	50	10	NA	3600	**15.7499714**	**715.82**	15.7498042	840.32	15.7498477	750.72
500	50	30	NA	3600	**15.7328319**	757.31	15.7327861	872.92	15.7327527	**756.10**
500	100	0	NA	3600	15.7711374	**620.77**	**15.7726792**	804.34	15.7707146	671.20
500	100	10	NA	3600	15.7621031	**810.71**	**15.7625451**	876.23	15.7625302	836.04
500	100	30	NA	3600	15.7527633	**814.52**	15.7534157	977.34	**15.7536448**	899.88

whose index is less than the order index to break the solutions' symmetry. To ensure feasibility and in the case of generating infeasible offspring, crossover operation is repeated up to five times. If no feasible offspring is generated after the fifth time, the crossover operation is aborted.

To perform a mutation, a chromosome's component ω_i is randomly chosen, and its value is changed to a random value $\omega_i' \in \mathcal{B}_i$. In case the mutant is infeasible, it is discarded, and a new mutant is generated. Similar to the crossover, this process is repeated up to a maximum of five times to find a feasible mutant, and if no feasible solution is generated after the fifth time, mutation operation is aborted. A distinctive feature of NSGAII, SPEA2, and NSGAIII is their selection process. In this study, the works of [10–12] are closely followed for implementing the selection operator.

5.3. Parameter tuning

Parameters of the metaheuristics have a determining impact on their performance. In this study, a full factorial method was utilized for tuning the parameters of the metaheuristics. Tuning parameters of the NSGAII, SPEA2, and NSGAIII include the probability of crossover p_c, probability of mutation p_m, probability of altering a gene during mutation process p_g, number of selected individuals for the next generation μ, and number of offsprings at each generation λ. Since a higher number of generations positively impacts the results, it was omitted from the tuning process and was set to 200 for all experiments. For each parameter, three levels were considered, as shown in Table 2, and all experiments were conducted on a test problem with $N = 50$, $Q = 50$, and $\Delta = 30$. Two response factors of Pareto frontier Hypervolume and CPU time were investigated.

The factorial experiments found no significant relationship between the parameters and their interaction with the performance of the algorithms in terms of Hypervolume. However, a significant relationship was observed when the CPU time was considered as the response variable. Fig. 5 depicts the standardized effect of parameters on Hypervolume and CPU time for NSGAII. As can be observed, none of the parameters' standardized effect reaches the threshold to be considered significant for the Hypervolume. However, three parameters and their interaction were determined significant when considered with the CPU time. Since no performance difference was detected, the parameters were set to minimize the CPU time ($\mu + \lambda = 50 + 150$, $p_c = 0.6$, $p_m = 0.05$, and $p_g = 0.25$).

6. Numerical experiments

This section presents the results of applying NSGAII, SPEA2, and NSGAIII to a set of problems sampled from the order level data of the studied logistics company, and compares them against the company's current practices. On average, each order contains 4.32 units of products. In the less frequent instances of orders containing unusually large numbers (i.e., larger than capacity), they are modified to fit the batches' capacity. This modification was deemed essential to investigate the effect of batch size on the solutions. Number of orders in each problem is selected from $\{10, 25, 50, 100, 200, 500\}$. Capacity Q and minimum physical distance Δ are chosen from $\{50, 100\}$ and $\{0, 10, 30\}$, respectively. The maximum allowed CPU time for all instances is set to 3600 s. The warehouse structure used for numerical experiments

Table 6
NPS, MID, SNS, and RAS obtained by Gurobi, NSGAII, SPEA2, and NSGAIII.

Problem			Gurobi				NSGAII				SPEA2				NSGAIII			
N	*Q*	*Δ*	NPS	MID	SNS	RAS	NPS	MID	SNS	RAS	NPS	MID	SNS	RAS	NPS	MID	SNS	RAS
10	50	0	3	1500	67	1977	**4.6**	1482.1	67.5	1961.6	**4.6**	**1480.3**	67.8	1963.1	**4.6**	1494.3	**73.9**	**1976.7**
10	50	10	3	1538	86	1706	6.8	1539.1	88.7	1447.2	**7.4**	1533.3	**97.9**	**1747.5**	**7.4**	**1528.4**	95.2	1697.4
10	50	30	3	1581	83	564	**19.0**	1582.3	**84.1**	**581.4**	17.8	**1576.5**	75.8	548.0	16.8	1579.3	78.2	554.1
10	100	0	3	1506	66	2043	6.0	1501.1	**67.5**	2033.7	**6.4**	**1500.3**	65.4	2030.0	**6.4**	1506.0	67.2	**2036.8**
10	100	10	3	1540	77	1805	9.0	**1529.2**	79.0	1698.4	9.0	1532.5	83.3	**1827.6**	**9.4**	1534.3	**84.5**	1796.9
10	100	30	3	1575	86	704	19.4	1578.3	**88.5**	636.9	**21.4**	1574.9	73.7	683.3	**21.4**	**1571.4**	73.0	**696.6**
25	50	0	NA	NA	NA	NA	**1.2**	2502.5	2.2	**3182.0**	1.0	**2492.7**	**NA**	3171.0	1.0	2500.3	**NA**	3179.0
25	50	10	NA	NA	NA	NA	5.6	**2593.5**	109.4	10.8	7.0	2618.7	**128.2**	**14.1**	**7.8**	2606.6	126.3	13.5
25	50	30	NA	NA	NA	NA	6.2	**2629.9**	128.2	7.6	**13.4**	2687.6	132.4	**8.9**	8.8	2668.2	**140.0**	8.1
25	100	0	NA	NA	NA	NA	8.6	2490.5	111.4	3253.1	**11.0**	**2469.5**	91.2	3234.6	10.0	2551.7	**133.7**	**3312.2**
25	100	10	NA	NA	NA	NA	77.4	2666.3	**205.1**	344.5	102.6	**2640.3**	183.8	341.7	**107.2**	2648.4	189.1	**347.8**
25	100	30	NA	NA	NA	NA	104.4	2716.3	**252.0**	**37.8**	**118.6**	2669.6	215.3	35.7	93.4	**2624.2**	184.1	37.3
50	50	0	NA	NA	NA	NA	3.6	**6310.9**	148.8	6990.9	**4.2**	6398.9	**205.6**	**7096.3**	**4.2**	6365.3	179.9	7052.1
50	50	10	NA	NA	NA	NA	60.6	7316.9	**266.1**	**10.8**	**66.2**	7106.5	203.8	10.2	57.0	**7102.6**	253.8	10.3
50	50	30	NA	NA	NA	NA	80.2	9034.6	**354.7**	14.6	**97.8**	9037.8	343.7	**14.9**	75.4	**8891.2**	310.6	14.5
50	100	0	NA	NA	NA	NA	17.8	6062.8	766.8	6939.0	**21.8**	**5793.2**	543.0	6684.3	16.4	6488.0	**1148.3**	**7397.2**
50	100	10	NA	NA	NA	NA	149.2	6889.6	**750.0**	**7.9**	**211.4**	6531.0	745.0	7.4	167.6	**6280.6**	720.5	6.5
50	100	30	NA	NA	NA	NA	167.4	8179.3	**1396.3**	**10.7**	**215.6**	**7668.7**	1134.2	9.6	194.0	7678.1	1033.4	9.6
100	50	0	NA	NA	NA	NA	**4.0**	12 631.1	391.2	13 439.0	**4.0**	**12 505.7**	303.9	13 317.0	**4.0**	12 710.4	**605.9**	**13 522.9**
100	50	10	NA	NA	NA	NA	52.2	17 760.2	**486.3**	**27.6**	**81.6**	17 342.6	384.1	26.9	48.4	**17 242.7**	387.2	26.8
100	50	30	NA	NA	NA	NA	59.2	25 269.5	**707.7**	**39.1**	74.8	24 983.6	664.7	38.5	**79.4**	**24 945.6**	677.2	38.5
100	100	0	NA	NA	NA	NA	21.8	12 833.5	2117.0	13 872.8	**26.8**	**12 024.0**	1279.4	13 007.7	19.0	13 648.8	**2913.9**	**14 708.4**
100	100	10	NA	NA	NA	NA	153.4	15 580.2	**2397.7**	**18.7**	**242.2**	14 864.0	1958.4	17.7	181.6	**14 472.3**	2193.7	16.9
100	100	30	NA	NA	NA	NA	168.2	20 131.9	3840.0	**25.2**	**200.8**	19 092.2	3247.5	23.2	179.6	**18 979.9**	**3849.6**	23.1
200	50	0	NA	NA	NA	NA	**7.2**	14 368.9	874.9	15 132.3	7.0	**14 135.4**	497.9	14 897.4	**7.2**	15 175.5	**1498.9**	**15 939.5**
200	50	10	NA	NA	NA	NA	**72.2**	22 167.0	326.5	37.2	69.0	22 159.8	**361.9**	**37.3**	64.4	**22 105.6**	348.5	37.2
200	50	30	NA	NA	NA	NA	50.0	32 042.8	**556.9**	**52.0**	51.6	**31 892.0**	484.3	51.6	**74.0**	31 946.6	433.4	51.4
200	100	0	NA	NA	NA	NA	27.4	16 885.9	4090.6	17 883.3	**34.4**	**15 281.6**	2444.6	16 228.8	24.0	18 380.5	**5143.5**	**19 409.7**
200	100	10	NA	NA	NA	NA	122.0	21 008.7	5263.3	**28.4**	**172.2**	**19 505.1**	4607.1	25.6	152.2	19 746.3	**6125.0**	26.0
200	100	30	NA	NA	NA	NA	107.0	28 731.3	**9553.7**	**39.9**	**152.8**	25 403.6	6895.1	34.0	143.8	**24 855.1**	8698.5	32.6
500	50	0	NA	NA	NA	NA	**6.4**	40 446.8	4041.6	41 244.0	**6.4**	**37 934.5**	1433.2	38 729.5	6.2	43 532.3	**7228.9**	**44 328.4**
500	50	10	NA	NA	NA	NA	73.4	**140 844.6**	2695.4	208.7	74.4	141 654.2	2755.1	**210.3**	**78.4**	141 015.3	**2847.5**	209.0
500	50	30	NA	NA	NA	NA	70.4	**250 776.8**	**4880.3**	345.9	**109.6**	250 908.2	3910.0	**348.3**	89.2	250 810.2	4130.6	346.4
500	100	0	NA	NA	NA	NA	26.4	49 618.9	13 637.6	50 706.6	**40.6**	**45 427.5**	9002.2	46 453.4	22.2	50 168.1	**14 510.3**	**51 268.5**
500	100	10	NA	NA	NA	NA	106.2	119 733.7	**550 42.9**	**154.0**	**131.0**	114 231.7	39 596.5	145.3	128.4	**113 506.1**	43 919.1	143.8
500	100	30	NA	NA	NA	NA	97.4	193 834.3	**86 800.0**	222.5	**115.8**	192 825.9	63 772.7	**226.3**	114.0	**188 492.5**	75 474.3	220.6

closely follows one of the company's warehouses (i.e., rectangular with parallel aisles and two cross-aisles). However, the dimensions, number of aisles, and pick faces are modified to replicate a more controlled experiment environment. Note that this modification does not change the overall performance of the proposed algorithms or the conclusions drawn from results. Each problem is attempted five times, and the results detailed are based on the overall performance of each algorithm over the total number of trials for each problem. All algorithms are implemented in Python 3.7 and executed on a 64-bit Windows operating system with an Intel Core i9-7940X CPU and 64 GB RAM.

6.1. Company's method

Table 3 lists the total time F, makespan C_{max}, and overlap Φ obtained from applying the company's method to the problems. As will be shown later, these results are highly efficient in terms of total travel time. However, the solutions it suggests are not efficient in terms of overlap as they are designed with a single total travel time objective in mind. As will be discussed in the remainder of this section, it is possible to maintain or minimally deteriorate the same level of total time efficiency while improving the overlap considerably. However, the same statement is not necessarily applicable to the makespan due to its negative correlation with both overlap and total travel time.

6.2. Proposed methods

Table 4 lists the best objective values and CPU times obtained by Gurobi, NSGAII, SPEA2, and NSGAIII. As can be observed, even for small instances with 25 orders, Gurobi is unable to find any feasible solution in the allotted one hour. Compared to the company's method in Table 3, all proposed metaheuristics of Table 4 find superior or equal solutions for all instances of the problem. This higher quality is especially notable for makespan and overlap.

The results of Table 4 demonstrate the extent to which each metaheuristic has been able to improve the solutions. However, it does not reveal any information on the relative performance of the methods. A multi-objective algorithm can be examined on several fronts. Some of the deciding performance measures of a multi-objective method are the number, quality, and diversity of the solutions it finds. The following metrics are utilized to compare the performance of the proposed multi-objective algorithms from various aspects [14,79,80]:

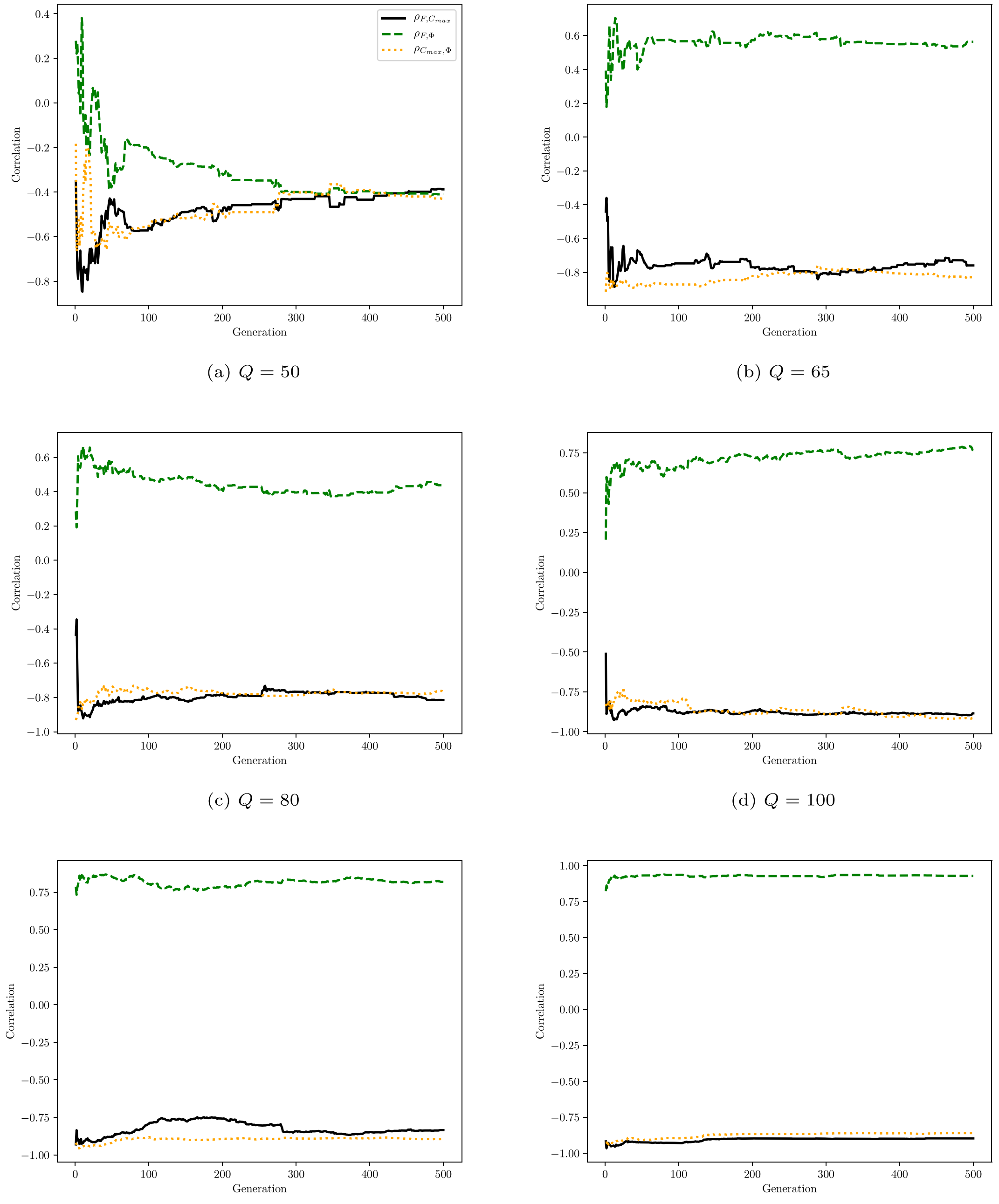
- *Number of Pareto Solutions (NPS):* counts the number of non-dominated solutions found.
- *Mean Ideal Distance (MID):* is the average Euclidean distance between Pareto frontier solutions and the ideal point (0,0), and is measured as:

$$MID = \frac{\sum_{i=1}^{n} c_i}{n} \tag{36}$$

where $c_i = \sqrt{f_{1i}^2 + f_{2i}^2 + f_{3i}^2}$, and f_{1i}^2, f_{2i}^2 and f_{3i}^2 are the value of the objective functions for the ith non-dominated solution.

- *Spread of Non-dominated Solutions (SNS):* assesses the diversity of the non-dominated solutions and is defined as:

$$SNS = \sqrt{\frac{\sum_{i=1}^{n} (MID - c_i)^2}{n - 1}} \tag{37}$$



(a) $Q = 50$

(b) $Q = 65$

(c) $Q = 80$

(d) $Q = 100$

(e) $Q = 200$

(f) $Q = 400$

Fig. 8. Objectives' correlation for different values of Q throughout NSGAII's evolution process ($N = 50$, $\Delta = 30$).

- Rate of Achievement Simultaneously to two objectives (RAS): measures the quality of the non-dominated solutions in relation with the minimum objective function, and is computed as:

$$RAS = \frac{\sum_{i=1}^{n}\left[\left(\frac{(f_{1i}-F_i)}{F_i}+\frac{(f_{2i}-F_i)}{F_i}+\frac{(f_{3i}-F_i)}{F_i}\right)\right]}{n} \tag{38}$$

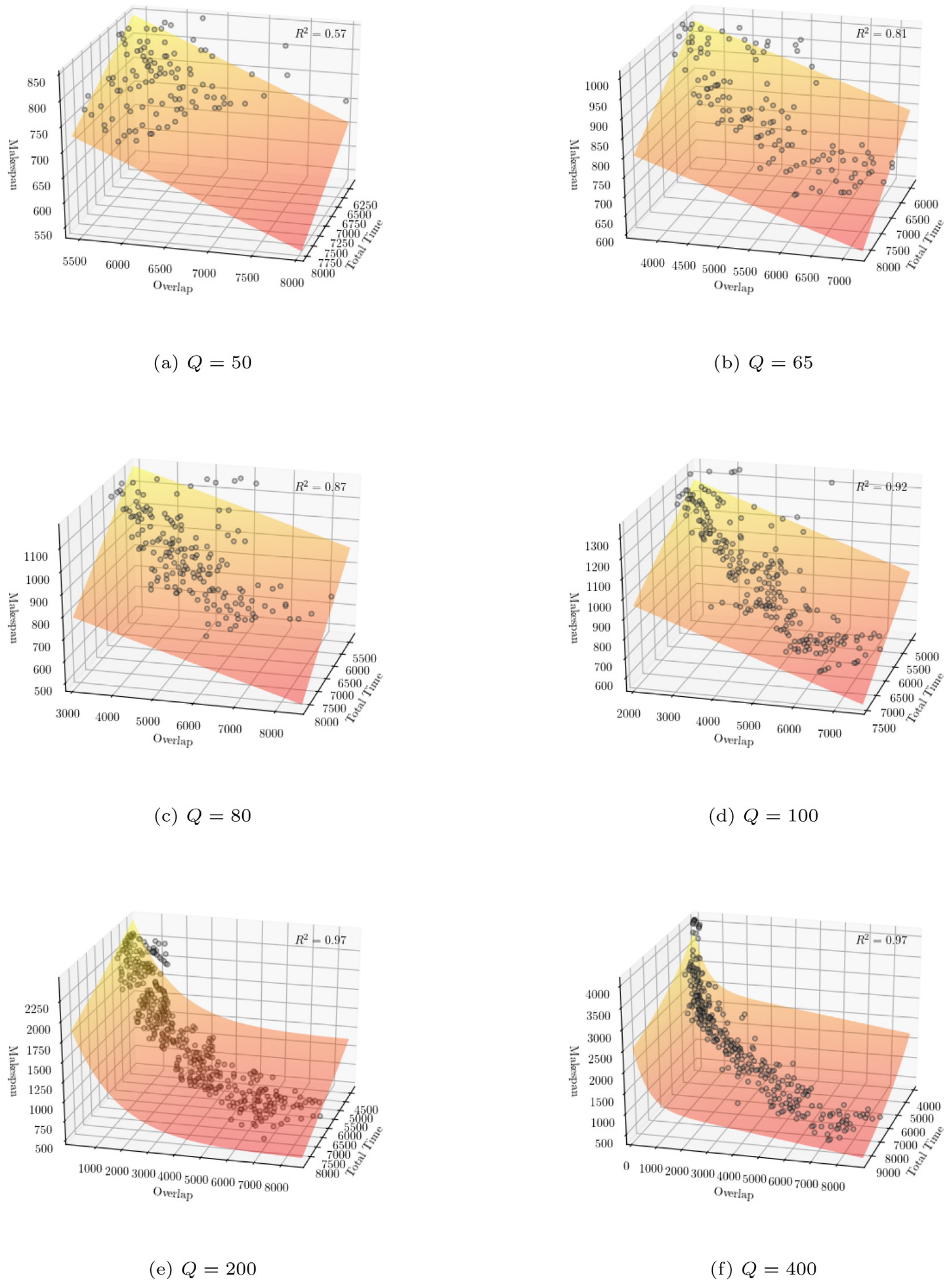
where $F_i = \min\{f_{1i}, f_{2i}, f_{3i}\}$.

- *Hypervolume (HV):* represents both quality and diversity of a set of solutions on a Pareto frontier (P) by calculating the hypervolume of the Pareto front in regards with a reference point, and is computed as follows:

$$HV(P) = \left\{\bigcup_{i=1} A(x_i) \mid \forall x_i \in P\right\} \tag{39}$$

where x_i is a solution in P, and $A(x_i)$ is the rectangular area confined between the points x_i and a reference point.

Table 5 shows the logarithm of hypervolume and CPU times of Gurobi, NSGAII, SPEA2, and NSGAIII. Log hypervolume conveys the same information as hypervolume. However, due to the large size of hypervolume numbers, log hypervolume is used in Table 5. A non-parametric Kruskal–Wallis test on the hypervolume ranks of averages was conducted to compare the results of Gurobi, NSGAII, SPEA2, and NSGAIII. At a 0.95 significance level, it was found that there is no statistically meaningful difference in terms of log HV among the metaheuristics. Due to its small sample size, Gurobi was not involved in the test. However, a general assessment of the Gurobi's log HV results illustrates an inferior performance compared to the other proposed methods. A similar Kruskal–Wallis test was performed to compare the CPU times,



(a) $Q = 50$

(b) $Q = 65$

(c) $Q = 80$

(d) $Q = 100$

(e) $Q = 200$

(f) $Q = 400$

Fig. 9. Pareto frontiers and the best fitted curve for different values of Q ($N = 50$, $\Delta = 30$).

and a significant difference was observed. Subsequently, a Mann–Whitney pairwise test with significance level 0.01 was conducted to find the method with the lowest CPU time, and NSGAII was determined as the fastest approach.

Table 6 tabulates the average NPS, MID, SNS, and RAS values obtained by Gurobi, NSGAII, SPEA2, and NSGAIII rounded to the nearest integer. Similar to the log HV value in Table 5, the performance of the proposed metaheuristics was evaluated using a non-parametric Kruskal–Wallis test, and it was found there is no statistically significant difference between the algorithms. Thus, the only differentiating parameter among the algorithms is their CPU time in which NSGAII outperforms the other algorithms. While there might be several reasons as to why a method such as NSGAIII that is designed for a larger number of objectives cannot outperform NSGAII and SPEA2, one possible explanation may lie in the seeding procedure of algorithms. As was discussed earlier, the seeding process provides high-quality solutions in terms of total travel time. Thus, the algorithms find it easier to explore the feasible space for solutions with lower makespan and overlap at the start. In other words, the evolutionary force of the algorithms is mostly concentrated on the makespan and overlap, and a smaller feasible space needs to be examined. In a smaller search space, the algorithms' performance can converge easier, and hence, equal solution quality.

Table A.7
Best no-overlap objective values obtained by Gurobi, company's batching algorithm, NSGAII, SPEA2, and NSGAIII using S-shape routing policy.

Problem			Gurobi		Company		NSGAII		SPEA2		NSGAIII	
N	Q	Δ	F	C_{max}	F	C_{max}	F	C_{max}	F	C_{max}	F	C_{max}
10	50	0	**1230**	**530**	1310	780	**1230**	**530**	**1230**	**530**	**1230**	**530**
10	50	10	**1230**	**530**	1320	790	**1230**	**530**	**1230**	**530**	**1230**	**530**
10	50	30	**1250**	**535**	1395	780	1250	**535**	**1230**	**535**	1250	**535**
10	100	0	**1080**	**530**	**1080**	1080	**1080**	**530**	**1080**	**530**	**1080**	**530**
10	100	10	**1080**	**530**	**1080**	1080	**1080**	**530**	**1080**	**530**	**1080**	**530**
10	100	30	**1080**	**535**	**1080**	1080	**1080**	**535**	**1080**	**535**	**1080**	**535**
25	50	0	NA	NA	2370	**820**	2370	**820**	**2350**	**820**	**2350**	**820**
25	50	10	NA	NA	2630	990	**2575**	**855**	2585	**855**	2585	**855**
25	50	30	NA	NA	2880	1130	2620	**855**	**2615**	**855**	2630	865
25	100	0	NA	NA	**2100**	1280	**2100**	**820**	**2100**	**820**	**2100**	**820**
25	100	10	NA	NA	**2100**	1280	**2100**	**855**	**2100**	**855**	**2100**	865
25	100	30	NA	NA	**2135**	1285	**2135**	**855**	**2135**	**855**	**2135**	870
50	50	0	NA	NA	6190	860	**6100**	**700**	6110	**700**	6190	**700**
50	50	10	NA	NA	9335	1340	**7800**	**1010**	8500	1030	8630	1035
50	50	30	NA	NA	14 260	2240	11 920	**1850**	11 940	**1850**	**10 705**	**1850**
50	100	0	NA	NA	4960	1350	**4920**	**700**	**4920**	**700**	**4920**	**700**
50	100	10	NA	NA	7750	2465	**6015**	**1015**	6630	**1015**	6590	1040
50	100	30	NA	NA	9505	2615	7315	**1850**	**6995**	**1850**	7380	**1850**
100	50	0	NA	NA	12 160	870	**12 100**	**820**	**12 100**	**820**	12 160	**820**
100	50	10	NA	NA	23 790	2075	21 230	**1830**	**20 895**	**1830**	21 555	**1830**
100	50	30	NA	NA	34 210	3180	28 755	**2730**	28 305	**2730**	**28 300**	**2730**
100	100	0	NA	NA	9950	1360	**9940**	**820**	9950	**820**	9950	**820**
100	100	10	NA	NA	19 735	3155	15 375	**1830**	**15 220**	**1830**	15 515	**1830**
100	100	30	NA	NA	26 625	4355	18 580	**2730**	18 540	**2730**	**17 735**	**2730**
200	50	0	NA	NA	13 460	850	**13 410**	**740**	13 460	**740**	13 460	**740**
200	50	10	NA	NA	28 295	2690	**25 430**	**2030**	25 565	2070	25 455	2045
200	50	30	NA	NA	37 065	3650	31 435	**2890**	31 230	**2890**	**31 130**	**2890**
200	100	0	NA	NA	**11 660**	1350	**11 660**	**740**	**11 660**	**740**	**11 660**	**740**
200	100	10	NA	NA	17 385	2430	17 195	2025	17 185	2035	**17 105**	**2000**
200	100	30	NA	NA	22 985	3845	**20 805**	**2890**	20 965	**2890**	20 835	**2890**
500	50	0	NA	NA	**36 430**	860	**36 430**	**770**	**36 430**	**770**	**36 430**	**770**
500	50	10	NA	NA	129 775	5415	120 905	5140	118 555	**5015**	**117 645**	5290
500	50	30	NA	NA	211 780	8530	188 835	8250	190 885	8265	**187 060**	**8120**
500	100	0	NA	NA	**31 250**	1370	**31 250**	**770**	**31 250**	**770**	**31 250**	**770**
500	100	10	NA	NA	91 160	6160	84 465	5995	85 980	**5830**	**83 250**	5945
500	100	30	NA	NA	123 035	8085	110 560	7750	109 665	7885	**107 235**	**7670**

6.3. Comparison with company's solution

In this section, the results of the proposed metaheuristics are compared against the case study's current practices. Fig. 6 depicts the current practices of the company against the Pareto frontier obtained by NSGAII. As can be observed, NSGAII offers a more diverse array of solutions. As was argued earlier, the company's method is quite efficient in terms of total travel time. However, as Fig. 6 illustrates, there are many other possibilities to choose from in order to have solutions with less overlap while maintaining travel time efficiency. Further comparison of the Pareto and company's solutions reveals two intriguing trends. First, when picking capacity is low, there are more solutions with the company's level of total time efficiency and lower overlap to select from. For example, in the problem with $N = 50$, $Q = 50$, and $\Delta = 30$, one can sustain the company's total travel time and improve the overlap by approximately 8.7%. Additionally, it seems smaller waves have a higher chance of obtaining a better overlap value for the same total travel time level. The second observation pertains to the relationship between overlap and total travel time. When the picking capacity increases, total travel time and overlap's correlation increases. Thus, low travel time solutions yield a lower overlap. This effect will be examined in more detail in the following sections.

6.4. Case of no pick overlap

In some circumstances, one may be interested in no overlap scenarios where the pickers avoid picking overlap by waiting until the other pickers have cleared the shelf. In this case, no pick overlap will translate into an increase in the total travel time, and the overlap objective will be eliminated. Appendix A presents a no pick overlap model $\boldsymbol{P}'$ along with the results of NSGAII, SPEA2, and NSGAIII with no overlap considerations and S-shape routing procedure. To implement the no overlap policy, a first-in-first-out (FIFO) pick procedure is applied where the picker who has arrived at a pick location earlier starts picking first. At the same time, other pickers wait for the pick area to clear off. As can be observed from Appendix A, the proposed metaheuristics are quite competent for dealing with high overlap instances compared to the company's current practice. In fact, when no overlap is allowed, the proposed metaheuristics tend to significantly improve the company's solutions compared to the cases where the overlap is permitted. Notably, as the minimum physical distance increases (and consequently waiting times increases), the percentage of the company's total travel time improvement obtained by the metaheuristics also increases. This observation shows the superior performance of the proposed multi-objective methods when dealing with no-overlap instances.

While the metaheuristics perform well on the no overlap instances, it is noteworthy that accepting a certain level of pick overlap may be preferred to zero overlap due to two reasons. First, reducing overlap is one infection spread mitigation strategy that must be considered in conjunction with the utilization of appropriate personal protective equipment (PPE) and testing procedures. In the presence of medical tests and PPE usage, mandating zero overlaps may result in unnecessarily conservative solutions with excessive total picking times. Second, a model that requires zero overlaps may not be suitable for less severe diseases. In model $\boldsymbol{P}$, pick overlap is considered as an objective

Algorithm 1 Company's order batching algorithm

```
$\mathcal{N}^- = \mathcal{N}$
$v_{i'k} = 0 \quad \forall k \in \mathcal{K}$
$v'_k = 0 \quad \forall k \in \mathcal{K}$
$v_{i'k_{max}} = 1$
$\gamma = \frac{H}{L/n}$
for $k \in \mathcal{K}$ do
    if $k \neq k_{max}$ then
        $\Psi_{\mathcal{K}}[k] = -(n + \gamma|k - k_{max}|)$
    else
        $\Psi_{\mathcal{K}}[k] = 0$
    end if
end for
while batch capacity allows do
    for $i \in \mathcal{N}^-$ do
        $s_i = 0$
        for $k \in \mathcal{K}$ do
            $s_i = s_i + \Psi_{\mathcal{K}}[k]$
        end for
    end for
    $i^{best} = \text{argmin}_i \, s_i$
    $\mathcal{N}^- = \mathcal{N}^- \setminus i^{best}$
    for $k \in \mathcal{K}$ do
        if $v_{i^{best}k} = 1$ and $v'_k = 0$ then
            $v'_k = 1$
        end if
    end for
    determine $k_{left}$ and $k_{right}$
    for $k \in \{1, 2, \ldots, k_{left}\}$ do
        $\Psi_{\mathcal{K}}[k] = -(n + \gamma|k - k_{left}|)$
    end for
    for $k \in \{k_{left}, k_{left} + 1, \ldots, k_{right}\}$ do
        if $v'_k = 0$ then
            $\Psi_{\mathcal{K}}[k] = -n$
        end if
    end for
    for $k \in \{k_{right}, k_{right} + 1, \ldots, K\}$ do
        $\Psi_{\mathcal{K}}[k] = -(n + \gamma|k - k_{right}|)$
    end for
end while
```

(soft constraint) so that depending on the severity of the disease, the solutions can be adjusted. For example, during a less severe flu season, it may be reasonable to emphasize total picking time compared to overlap.

6.5. *Effect of routing policy*

Routing policy can impact the quality of solutions. Specifically, when an infection spread is of concern, one may prefer to use a routing policy that decreases the risk of spread. The model and main results of this study are derived using the S-shape policy as it is the current routing procedure of the study's warehouse. The S-shape policy is hypothesized to have anti-congestion properties [8], and since picker congestion can act as an infection spread facilitator, it may be a suitable procedure for the infection risk mitigation. On the other hand, there is some evidence that the S-shape policy may lead to excessively long travel times, which could be the result of both long travel distances and frequent blocking and congestion [81]. To evaluate the S-shape policy's effectiveness for the problem of this study, a Midpoint policy was implemented. Conducting statistical tests and comparisons between the policies revealed that the S-shape policy does not hold any advantage over the Midpoint policy in terms of overlap. In fact, the Hypervolumes comparison showed that the Midpoint policy offers a slightly better tradeoff between the objectives. Finding the best routing policy that minimizes the infection spread requires a comprehensive routing policy comparison that falls beyond the current study's scope. The details of the Midpoint policy results are tabulated in Appendix B. Note that computational times are not reported in Appendix B as no significant difference was observed between S-shape and Midpoint policies in terms of CPU time. Moreover, since model ***P*** is formulated based on the S-shape policy, no exact approach result is reported in Appendix B.

Table B.8
Company's method using Midpoint policy.

N	Q	Δ	F	C_{max}	Φ
10	50	0	1280	790	0
10	50	10	1280	790	0
10	50	30	1280	790	0
10	100	0	1060	1060	0
10	100	10	1060	1060	0
10	100	30	1060	1060	0
25	50	0	2350	810	0
25	50	10	2350	810	520
25	50	30	2350	810	560
25	100	0	2150	1340	0
25	100	10	2150	1340	10
25	100	30	2150	1340	70
50	50	0	6130	910	0
50	50	10	6130	910	3090
50	50	30	6130	910	5740
50	100	0	5010	1370	0
50	100	10	5010	1370	2510
50	100	30	5010	1370	3310
100	50	0	12 230	900	0
100	50	10	12 230	900	14 860
100	50	30	12 230	900	21 840
100	100	0	10 160	1380	0
100	100	10	10 160	1380	8770
100	100	30	10 160	1380	12 160
200	50	0	13 580	860	0
200	50	10	13 580	860	19 450
200	50	30	13 580	860	26 650
200	100	0	11 800	1410	0
200	100	10	11 800	1410	8890
200	100	30	11 800	1410	10 680
500	50	0	36 580	880	0
500	50	10	36 580	880	138 240
500	50	30	36 580	880	235 030
500	100	0	31 550	1390	0
500	100	10	31 550	1390	66 960
500	100	30	31 550	1390	114 240

6.6. *Seeding and convergence*

In this section, the effectiveness of the proposed seeding procedure is examined. Fig. 7 illustrates the convergence of NSGAII, SPEA2, and NSGAIII in terms of log hypervolume over the first 200 generations. As observed, when seeded, all algorithms significantly outperform their not seeded counterparts. The same effect is observed for different values of Q and Δ. However, the gap between the seeded and not seeded solutions seem to be more prevalent for more constrained instances of the problem (i.e., smaller Q). This effect may be attributed to the difficulty of finding high-quality feasible solutions in tightly constrained search spaces, and the fact that seeding can bypass this difficulty. Furthermore, the seeding procedure produces highly efficient solutions in terms of total travel time, which have little room for improvement. As a result, the proposed algorithms mostly concentrate on enhancing makespan and overlap. In other words, seeding with a superior solution in terms of one of the objectives, functions as if the objective is relaxed and thus, the algorithm has a more manageable feasible space to search.

Table B.9
Best objective values obtained by NSGAII, SPEA2, and NSGAIII using Midpoint routing policy.

Problem			NSGAII			SPEA2			NSGAIII		
N	Q	Δ	F	C_{max}	Φ	F	C_{max}	Φ	F	C_{max}	Φ
10	50	0	**1210**	**490**	**0**	**1210**	**490**	**0**	**1210**	**490**	**0**
10	50	10	**1210**	**490**	**0**	**1210**	**490**	**0**	**1210**	**490**	**0**
10	50	30	**1210**	**490**	**0**	**1210**	**490**	**0**	**1210**	**490**	**0**
10	100	0	**1060**	**490**	**0**	**1060**	**490**	**0**	**1060**	**490**	**0**
10	100	10	**1060**	**490**	**0**	**1060**	**490**	**0**	**1060**	**490**	**0**
10	100	30	**1060**	**490**	**0**	**1060**	**490**	**0**	**1060**	**490**	**0**
25	50	0	**2330**	**810**	**0**	**2330**	**810**	**0**	**2330**	**810**	**0**
25	50	10	**2310**	**810**	230	2350	**810**	**100**	2330	**810**	120
25	50	30	**2330**	**810**	310	**2330**	**810**	**150**	**2330**	**810**	250
25	100	0	2150	**810**	**0**	2150	**810**	**0**	**2090**	**810**	**0**
25	100	10	2150	**810**	10	**2080**	**810**	10	**2080**	**810**	**0**
25	100	30	2110	**810**	70	**2080**	**810**	70	**2080**	**810**	**20**
50	50	0	6080	**700**	**0**	**6030**	**700**	**0**	6080	**700**	**0**
50	50	10	**5900**	**700**	2120	6060	**700**	**2070**	6030	**700**	2170
50	50	30	6030	**700**	4790	**5840**	**700**	**4640**	6020	**700**	4750
50	100	0	**4970**	**700**	**0**	**4970**	**700**	**0**	**4970**	**700**	**0**
50	100	10	4980	**700**	630	**4960**	**700**	**410**	4980	**700**	550
50	100	30	5010	**700**	1920	4980	**700**	1950	**4970**	**700**	**1770**
100	50	0	12 210	**820**	**0**	12 210	**820**	**0**	**12 140**	**820**	**0**
100	50	10	12 210	**820**	12 350	**12 140**	**820**	**11 610**	12 200	**820**	11 880
100	50	30	**12 140**	**820**	19 810	12 210	**820**	19 710	12 170	**820**	**19 430**
100	100	0	**10 150**	**820**	**0**	10 160	**820**	**0**	10 160	**820**	**0**
100	100	10	10 160	**820**	**4450**	**10 060**	**820**	5500	10 160	**820**	6250
100	100	30	**10 070**	**820**	8990	10 160	**820**	**8950**	10 160	**820**	9150
200	50	0	**13 520**	**730**	**0**	13 580	**730**	**0**	13 580	**730**	**0**
200	50	10	13 580	**730**	**16 740**	**13 510**	**730**	17 650	13 570	**730**	16 850
200	50	30	13 580	**730**	**24 990**	**13 560**	**730**	25 880	13 580	**730**	25 790
200	100	0	**11 790**	**730**	**0**	**11 790**	**730**	**0**	11 800	**730**	**0**
200	100	10	11 800	**730**	7980	**11 780**	**730**	8020	11 800	**730**	**7710**
200	100	30	**11 800**	**730**	**10 250**	**11 800**	**730**	10 280	**11 800**	**730**	10 280
500	50	0	**36 580**	**780**	**0**	**36 580**	**780**	**0**	**36 580**	**780**	**0**
500	50	10	**36 510**	**780**	**133 800**	36 580	**780**	135 190	36 580	**780**	135 260
500	50	30	**36 580**	**780**	**229 160**	**36 580**	**780**	231 690	**36 580**	**780**	231 320
500	100	0	31 550	**780**	**0**	31 550	**780**	**0**	**31 540**	**780**	**0**
500	100	10	**31 550**	**780**	63 080	**31 550**	**780**	63 370	**31 550**	**780**	**62 620**
500	100	30	**31 550**	**780**	**112 540**	**31 550**	**780**	113 240	**31 550**	**780**	112 550

7. Discussion

In this section, the managerial implications of the model and considering overlap objective are discussed. As was illustrated earlier, picking capacity Q is a determining factor in the relationship between the total travel time and overlap. To investigate this observation further, the correlation between the three objectives was measured throughout NSGAII's evolution process. Fig. 8 depicts the Pearson correlation ρ between total travel time F, makespan C_{max}, and overlap Φ for the best found solutions at each generation of NSGAII and for various picking capacities. As can be observed, correlations eventually converge for all instances. However, the limit to which the correlations converge is different for each picking capacity. When $Q = 50$, total travel time and overlap correlation $\rho_{F,\Phi}$ reaches converges to a negative value. In other words, for tight picking capacities, total travel time and overlap act as conflicting objectives. However, as Q increases, the convergent correlation between total travel time and overlap increases. A positive correlation between total travel time and overlap suggests that the two objectives are cooperating, and one can be improved by improving the other one. Another intriguing observation is the correlation between makespan and the other two objectives. As Q increases, makespan's correlation with other objectives points to a more conflicting state.

From a managerial standpoint, the aforementioned observations can be interpreted as follows. When a warehouse's objective is to minimize the total travel time, it may benefit from increasing the picking capacity as it allows for reducing the overlap. In other words, for warehouses to minimize total travel time, increasing picking capacity helps with mitigating the risk of infection spread. This dynamic can be explained by considering that an increased picking capacity means fewer batches and fewer walking pickers to interact. Thus, increasing picking capacity contributes to lower overlap by reducing the number of pickers that are required to traverse the warehouse at a time.[1]

A second managerial lesson of this study originates from the correlation between makespan and overlap. As observed in Fig. 8, makespan always correlates with the other objectives negatively. However, this correlation's absolute value is smaller for the lower values of the picking capacity Q. Thus, a lower picking capacity provides a more favorable tradeoff between overlap and makespan. Thus, warehouse environments where the makespan is of significance, such as a wave-picking warehouse, may benefit from reducing the picking capacity to control the overlap and spread of infection. Remember that in this study, each batch is picked by a separate picker. Therefore, lower picking capacity translates into smaller waves, which means reducing the wave size can help mitigate the risk of infection spread in wave picking warehouses. It is noteworthy that reducing the wave size can potentially eliminate the benefits of picking postponement. Consequently, wave picking warehouses may find themselves at

[1] A second explanation of this effect rises from the structure of optimum solutions. In general, it is easier to find degenerate solutions with the same total travel time but different picking tours when the picking capacity is large. While such solutions have the same total travel time, they may have different overlap values, and thus, there is a higher chance of finding low overlap solutions among the degenerate ones. It is noteworthy that in this context, the term degeneracy is used with a more general purpose in mind, and the solutions that have the same value for one objective are also categorized as degenerate.

Table B.10
Log hypervolume of NSGAII, SPEA2, and NSGAIII using Midpoint policy.

N	Q	Δ	NSGAII	SPEA2	NSGAIII
10	50	0	15.8292108	**15.8292109**	15.8292104
10	50	10	15.8292105	**15.8292110**	15.8292109
10	50	30	15.8292074	**15.8292110**	15.8292104
10	100	0	15.8293241	**15.8293259**	15.8293236
10	100	10	**15.8293256**	15.8293252	15.8293254
10	100	30	15.8293252	15.8293255	**15.8293258**
25	50	0	15.7962663	15.7962663	**15.7962698**
25	50	10	15.7962411	**15.7962428**	15.7962364
25	50	30	15.7962249	**15.7962300**	15.7962225
25	100	0	15.7964191	15.7964194	**15.7964249**
25	100	10	15.7964186	**15.7964316**	15.7964310
25	100	30	15.7964166	**15.7964330**	15.7964329
50	50	0	15.8042284	**15.8042312**	15.8042252
50	50	10	**15.8039280**	15.8039163	15.8039128
50	50	30	15.8035199	**15.8036067**	15.8035184
50	100	0	15.8050989	**15.8051151**	15.8050104
50	100	10	15.8049318	**15.8050142**	15.8049783
50	100	30	15.8047398	15.8047735	**15.8048077**
100	50	0	15.7865143	15.7865107	**15.7865229**
100	50	10	15.7846663	**15.7847199**	15.7846971
100	50	30	15.7836004	15.7835981	**15.7836278**
100	100	0	15.7862847	**15.7875388**	15.7867881
100	100	10	15.7867374	**15.7872334**	15.7862651
100	100	30	15.7864145	**15.7867042**	15.7866721
200	50	0	**15.7945433**	15.7945408	15.7944954
200	50	10	**15.7920349**	15.7919416	15.7920052
200	50	30	**15.7908413**	15.7907606	15.7907689
200	100	0	15.7954581	**15.7957586**	15.7951220
200	100	10	15.7942315	**15.7943641**	15.7943298
200	100	30	15.7937532	**15.7939827**	15.7939300
500	50	0	15.7683191	**15.7683776**	15.7682822
500	50	10	**15.7483181**	15.7482213	15.7481996
500	50	30	**15.7333490**	15.7332081	15.7332045
500	100	0	15.7705533	**15.7714877**	15.7700123
500	100	10	15.7601920	**15.7610085**	15.7609468
500	100	30	15.7518709	**15.7529691**	15.7528894

a more challenging state of the tradeoff between makespan and overlap.

Another managerial implication of this study pertains to the tradeoff patterns between the objectives. Fig. 9 depicts the Pareto frontier of the instance with 50 orders ($N = 50$) and minimum physical distance's walking time of 30 ($\Delta = 30$) while picking capacity Q varies from 50 to 400.[2] As can be observed, the tradeoff between total travel time and makespan remains almost linear for different values of Q. However, the tradeoff between the makespan and overlap gradually shifts towards an exponential decay pattern with diminishing returns. From a managerial perspective, this observation shows that there is a threshold for larger picking capacities, after which reducing makespan requires a much higher level of overlap. Thus, as was previously demonstrated through the correlation analysis of the objectives, makespan-driven warehouses may benefit from a lower picking capacity and a subsequent more favorable tradeoff between makespan and overlap.

8. Conclusion

Being a significant source of expenses in a supply chain, order picking operations are not well-positioned to rise to pandemic challenges such as the ones posed by COVID-19. This inability is mainly due to a cost-driven mindset in the order picking operations, which seems reasonable in a normal situation. This study suggests that the risk of pickers contracting Coronavirus or any other infectious disease must be taken into account in the order picking problems. For this purpose, the well-known order batching problem was revisited and reformulated by considering the risk of infection spread due to workers picking orders in close vicinity of each other. A tri-objective model was proposed that aimed to simultaneously minimize the total travel time, makespan, and picking overlap among the pickers. Furthermore, three evolutionary methods, namely, NSGAII, SPEA2, and NSGAIII, were proposed to solve the problem. The model and the proposed methods were tested on the data of a major US-based logistics company and compared against the company's current practices. It was found that while the company performs at an efficient level of total travel time, their method is not well suited when the pickers' overlap is considered.

Through an extensive numerical experiment and comparison with the company's method, it was found that reducing the picking overlap, as a proxy of infection spread, and minimizing the total travel time are not necessarily conflicting objectives. However, makespan was observed at odds with the overlap in all experiments. This study has two key takeaways for practitioners and managers. First, it is possible to maintain the order batching solutions' travel time efficiency while minimizing the overlap. In fact, these two objectives seem to correlate positively. However, for such a correlation to be realized, it is necessary to have a sufficient picking capacity. This study showed that a low picking capacity could erode all the overlap cross-benefits obtained as a result of minimizing the total travel time. Second, it was found that minimizing the makespan has an adverse effect on the overlap, and thus, some wave-picking warehouses may find it more challenging to prepare for the pandemic situation. To surmount this hurdle, a wave size reduction was suggested for such warehouses.

The main limitation of the presented study pertains to the stochastic nature of pickers' travel times. When batching, one typically assumes a constant picker speed and reliably uniform picking times, which makes the calculation of the objectives feasible. However, a decision-maker needs to consider the impact of a non-uniform walking speed on calculations of overlap and possibly equip the batching method with simple heuristics that regulate how pickers should behave in case of overlaps occurrence. One possible strategy to mitigate the effect of travel time uncertainty on the overlap calculation is to increase the minimum physical distance to overemphasize the importance of overlap when constructing solutions. However, this may lead to sub-optimal solutions in terms of makespan or total travel time. Thus, a robust or stochastic oriented approach to reduce the pick overlap remains an important future research direction.

Incorporation of infection risk considerations in order picking problems poses a new set of challenges in the operational level of supply chains. In this context, two future research directions are suggested. First, virtually all order picking problems can be revisited with a pandemic lens of investigation to ensure a safe environment for the pickers. Second, there are tactical problems that are directly affected by the pandemic. For example, warehouses typically prefer to locate products that are ordered together frequently in the proximity of each other. However, this routine may not be entirely desirable as it may cause pickers' routes to overlap further. Thus, it is necessary to rethink the order association among the products to ensure a safer workplace.

[2] To better visualize the Pareto front, a curve of the form $C_{max} = c_1 F + c_2 \Phi + c_3 e^{c_4 F} + c_5 e^{c_6 F} + c_7$ is fit on the solutions of the front. This equation is adopted to capture both linear and a common nonlinear form of tradeoffs in multi-objective problems. For each curve fit, the value of R^2 is exhibited on the top right of the figure.

Table B.11
NPS, MID, SNS, and RAS obtained by NSGAII, SPEA2, and NSGAIII using Midpoint policy.

N	*Q*	*Δ*	NSGAII				SPEA2				NSGAIII			
			NPS	MID	SNS	RAS	NPS	MID	SNS	RAS	NPS	MID	SNS	RAS
10	50	0	5.0	1438.3	**67.6**	1888.2	**5.2**	**1428.5**	56.1	1882.4	4.4	1439.0	56.4	**1894.9**
10	50	10	6.0	1471.0	95.8	**1053.5**	6.2	1467.3	**103.7**	951.9	**7.6**	**1464.8**	96.2	881.3
10	50	30	4.4	1434.8	58.7	**1220.0**	5.0	**1427.1**	59.5	1151.9	**7.0**	1448.9	**60.2**	973.5
10	100	0	6.8	1467.3	56.3	1986.9	**7.2**	**1454.0**	58.1	1959.4	5.8	1473.3	**62.3**	**1997.4**
10	100	10	**12.8**	1515.5	84.7	**681.3**	12.4	**1506.8**	81.6	669.0	12.2	1518.4	**89.9**	680.6
10	100	30	**11.4**	1505.6	70.8	688.2	10.8	1498.0	**77.4**	751.1	10.4	**1489.3**	72.1	**889.2**
25	50	0	**1.0**	2478.1	**0.0**	**3149.0**	**1.0**	2478.1	**0.0**	**3149.0**	**1.0**	**2474.3**	**0.0**	3145.0
25	50	10	8.8	2756.2	**207.9**	8.1	**11.6**	2749.9	121.2	**13.1**	10.4	**2718.2**	162.9	10.0
25	50	30	7.4	**2718.3**	155.1	5.7	**9.4**	2785.6	183.7	**8.9**	8.6	2781.1	**204.6**	7.1
25	100	0	12.0	2556.5	108.6	3324.3	**12.6**	**2546.6**	96.8	3318.0	11.8	2589.7	**127.6**	**3368.0**
25	100	10	**82.6**	2655.9	**161.4**	66.0	76.2	**2642.6**	148.8	72.8	74.8	2644.1	150.9	**105.3**
25	100	30	**106.4**	2738.5	**228.8**	18.4	104.4	2689.9	178.7	20.6	90.0	**2665.2**	163.7	**22.2**
50	50	0	3.2	6252.2	106.0	6934.2	**4.4**	**6251.1**	98.8	6950.8	**4.4**	6297.0	**143.0**	**6999.0**
50	50	10	84.6	7150.1	224.8	10.1	**91.8**	7106.2	222.3	**10.4**	87.6	**7074.8**	**227.0**	10.3
50	50	30	65.6	8661.8	350.9	**13.7**	63.0	**8445.3**	**358.9**	13.7	**77.0**	8467.8	243.4	13.6
50	100	0	**24.6**	6088.1	791.6	6989.5	22.2	**5869.1**	489.7	6752.6	19.4	6377.1	**1070.3**	**7321.0**
50	100	10	154.4	6919.5	761.1	**7.9**	**231.0**	6518.4	730.3	7.4	192.8	**6505.3**	**810.3**	7.2
50	100	30	174.8	7579.0	**1234.2**	**8.8**	**212.2**	7305.9	1134.9	8.6	193.6	**7104.9**	1137.8	8.2
100	50	0	5.2	12 787.2	505.9	13603.3	5.6	**12732.2**	389.5	13550.9	**5.8**	12 926.2	**734.8**	**13 744.8**
100	50	10	73.0	19 026.7	344.5	**29.4**	**113.8**	18884.7	395.8	29.1	92.6	**18 832.0**	**399.2**	29.0
100	50	30	68.2	24 935.1	446.6	38.1	57.4	24742.9	436.4	**38.1**	**79.8**	**24 644.6**	**479.2**	37.6
100	100	0	21.0	13 212.0	2122.9	14287.2	**26.2**	**12073.3**	1080.7	13074.8	18.8	13 933.7	**2749.6**	**15 014.2**
100	100	10	208.4	16 568.5	**2347.9**	**19.6**	**273.4**	15884.4	2166.2	18.7	247.2	**15 569.3**	2089.3	18.1
100	100	30	151.6	19 219.6	**3706.5**	23.5	**221.8**	19353.0	3589.8	**24.0**	194.4	**18 419.6**	3428.9	22.1
200	50	0	8.2	14 984.3	1562.7	15744.9	**9.0**	**14390.6**	560.1	15149.0	8.6	16 325.2	**2367.5**	**17 085.8**
200	50	10	**90.8**	23 566.0	**633.3**	39.5	90.2	23746.9	573.7	**40.2**	80.2	**23 382.6**	614.5	39.2
200	50	30	**53.4**	**30 449.2**	**1177.0**	49.3	41.4	30499.4	829.6	50.1	47.4	30 584.4	893.7	**50.3**
200	100	0	30.8	16 963.6	3757.6	17 962.0	**37.0**	**15256.9**	2545.4	16222.1	20.4	18 274.5	**5106.8**	**19 319.4**
200	100	10	132.2	22 889.9	**6726.8**	31.5	**212.6**	23487.7	5724.3	**33.6**	184.6	**20 676.0**	6009.9	27.0
200	100	30	105.2	26 014.5	**9684.8**	34.7	**138.4**	27955.4	8191.6	**40.3**	121.4	**25 903.2**	8742.5	35.7
500	50	0	7.4	40 708.9	4454.5	41522.4	**7.8**	**37964.6**	1384.7	38777.4	6.8	41 838.0	**5330.4**	**42 654.4**
500	50	10	64.4	143 043.4	2026.6	207.0	**94.4**	**142919.5**	1733.9	206.3	90.2	143 081.5	**2220.2**	**207.5**
500	50	30	43.2	**240 265.9**	**6187.7**	324.0	**53.0**	240 297.6	4907.3	326.3	44.4	241 026.3	4395.3	**330.5**
500	100	0	30.4	47 758.7	12 415.5	48 852.5	**42.2**	**44 521.8**	8095.7	45561.4	22.4	50 680.4	**14 022.1**	**51 784.0**
500	100	10	103.2	124 851.1	**53 144.6**	**154.8**	**145.4**	121 961.3	39 226.1	154.2	114.2	**116 580.2**	41 211.7	145.7
500	100	30	80.2	**198 928.2**	**83 802.3**	230.2	**100.8**	199 729.3	69 931.3	239.6	93.8	202 496.0	70 480.6	**241.5**

$$t^{+}_{b_2v_2r_2} + \Delta(1 - \eta^{b_1v_1r_1}_{b_2v_2r_2}) \geq t^{-}_{b_1v_1r_1} \qquad \forall b_1, b_2 \in \mathcal{B}, b_1 \neq b_2; v_1, v_2 \in \mathcal{V}; r_1, r_2 \in \mathcal{R}, \delta_{\Delta}(v_1, v_2) = 1 \qquad \text{(A.1)}$$

$$t^{+}_{b_1v_1r_1} + \Delta(1 - \eta^{b_2v_2r_2}_{b_1v_1r_1}) \geq t^{-}_{b_2v_2r_2} \qquad \forall b_1, b_2 \in \mathcal{B}, b_1 \neq b_2; v_1, v_2 \in \mathcal{V}; r_1, r_2 \in \mathcal{R}, \delta_{\Delta}(v_1, v_2) = 1 \qquad \text{(A.2)}$$

$$\eta^{b_1v_1r_1}_{b_2v_2r_2} + \eta^{b_2v_2r_2}_{b_1v_1r_1} \geq 1 \qquad \forall b_1, b_2 \in \mathcal{B}, b_1 \neq b_2; v_1, v_2 \in \mathcal{V}; r_1, r_2 \in \mathcal{R}, \delta_{\Delta}(v_1, v_2) = 1 \qquad \text{(A.3)}$$

$$\eta^{b_1v_1r_1}_{b_2v_2r_2} \in \{0, 1\} \qquad \forall b_1, b_2 \in \mathcal{B}; v_1, v_2 \in \mathcal{V}; r_1, r_2 \in \mathcal{R} \qquad \text{(A.4)}$$

Box II.

CRediT authorship contribution statement

Ehsan Ardjmand: Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing - original draft, Writing - review & editing, Visualization, Supervision, Project administration. **Manjeet Singh:** Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing - original draft. **Heman Shakeri:** Software, Validation, Formal analysis, Writing - original draft, Writing - review & editing. **Ali Tavasoli:** Software, Validation, Formal analysis, Writing - original draft, Writing - review & editing. **William A. Young II:** Conceptualization, Supervision, Writing - original draft, Writing - review & editing, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. No overlap model

No overlap model $\boldsymbol{P}'$, is formulated as follows:

$$\min \quad (1) \text{ and } (2)$$

subject to constraints (4) to (24), (28) to (32), (34) and (35), Eqs. (A.1)–(A.4) which are given in Box II.

In model $\boldsymbol{P}'$, variable $\eta^{b_1v_1r_1}_{b_2v_2r_2}$ equals 1 if r_1th visit of node v_1 in batch b_1 is prior to the r_2th visit of node v_2 in batch b_2. Additionally, constraints (A.1) to (A.3) ensure no picking overlap among locations that are positioned closer than the minimum physical distance (see Table A.7).

Appendix B. Midpoint policy results

See Tables B.8–B.11.

References

[1] Center for Systems Science and Engineering, Coronavirus covid-19 global cases, 2020, URL: https://coronavirus.jhu.edu/map.html.

[2] University of Washington, Covid-19 projections, 2020, URL: https://covid19.healthdata.org/united-states-of-america.

[3] D. Ivanov, Predicting the impacts of epidemic outbreaks on global supply chains: A simulation-based analysis on the coronavirus outbreak (covid-19/sars-cov-2) case, Transp. Res. E 136 (2020) 101922.

[4] R.J. Glass, L.M. Glass, W.E. Beyeler, H.J. Min, Targeted social distancing designs for pandemic influenza, Emerg. Infect. Dis. 12 (11) (2006) 1671.

[5] T.C. Reluga, Game theory of social distancing in response to an epidemic, PLoS Comput. Biol. 6 (5) (2010).

[6] J.K. Kelso, G.J. Milne, H. Kelly, Simulation suggests that rapid activation of social distancing can arrest epidemic development due to a novel strain of influenza, BMC Public Health 9 (1) (2009) 117.

[7] T. Öncan, Milp formulations and an iterated local search algorithm with tabu thresholding for the order batching problem, European J. Oper. Res. 243 (1) (2015) 142–155.

[8] R. De Koster, T. Le-Duc, K.J. Roodbergen, Design and control of warehouse order picking: A literature review, European J. Oper. Res. 182 (2) (2007) 481–501.

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

[10] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[11] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm, TIK-report 103, Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische …, 2001.

[12] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints, IEEE Trans. Evol. Comput. 18 (4) (2013) 577–601.

[13] E. Ardjmand, E.M. Youssef, A. Moyer, W.A.Y. Ii, G.R. Weckman, H. Shakeri, A multi-objective model for minimising makespan and total travel time in put wall-based picking systems, Int. J. Logist. Syst. Manage. 36 (1) (2020) 138–176.

[14] E. Ardjmand, D.W. Huh, Coordinated warehouse order picking and production scheduling: A nsga-ii approach, in: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2017, pp. 1–8.

[15] A. Scholz, D. Schubert, G. Wäscher, Order picking with multiple pickers and due dates–simultaneous solution of order batching, batch assignment and sequencing, and picker routing problems, European J. Oper. Res. 263 (2) (2017) 461–478.

[16] E. Ardjmand, H. Shakeri, M. Singh, O.S. Bajgiran, Minimizing order picking makespan with multiple pickers in a wave picking warehouse, Int. J. Prod. Econ. 206 (2018) 169–183.

[17] E. Ardjmand, I. Ghalehkhondabi, W.A. Young II, A. Sadeghi, G.R. Weckman, H. Shakeri, A hybrid artificial neural network, genetic algorithm and column generation heuristic for minimizing makespan in manual order picking operations, Expert Syst. Appl. (2020) 113566.

[18] I. Žulj, S. Kramer, M. Schneider, A hybrid of adaptive large neighborhood search and tabu search for the order-batching problem, European J. Oper. Res. 264 (2) (2018) 653–664.

[19] B. Menéndez, E.G. Pardo, A. Alonso-Ayuso, E. Molina, A. Duarte, Variable neighborhood search strategies for the order batching problem, Comput. Oper. Res. 78 (2017) 500–512.

[20] S. Hong, Y. Kim, A route-selecting order batching model with the S-shape routes in a parallel-aisle order picking system, European J. Oper. Res. 257 (1) (2017) 185–196.

[21] Y.-C. Ho, T.-S. Su, Z.-B. Shi, Order-batching methods for an order-picking warehouse with two cross aisles, Comput. Ind. Eng. 55 (2) (2008) 321–347.

[22] Y.-C. Ho, Y.-Y. Tseng, A study on order-batching methods of order-picking in a distribution centre with two cross-aisles, Int. J. Prod. Res. 44 (17) (2006) 3391–3417.

[23] J. Li, R. Huang, J.B. Dai, Joint optimisation of order batching and picker routing in the online retailer's warehouse in China, Int. J. Prod. Res. 55 (2) (2017) 447–461.

[24] H.D. Ratliff, A.S. Rosenthal, Order-picking in a rectangular warehouse: a solvable case of the traveling salesman problem, Oper. Res. 31 (3) (1983) 507–521.

[25] A. Scholz, S. Henn, M. Stuhlmann, G. Wäscher, A new mathematical programming formulation for the single-picker routing problem, European J. Oper. Res. 253 (1) (2016) 68–84.

[26] L. Pansart, N. Catusse, H. Cambazard, Exact algorithms for the order picking problem, Comput. Oper. Res. 100 (2018) 117–127.

[27] M. Masae, C.H. Glock, P. Vichitkunakorn, Optimal order picker routing in the chevron warehouse, IISE Trans. 52 (6) (2020) 665–687.

[28] J.A. Cano, R.A. Gomez, F. Salazar, Routing policies in multi-parallel warehouses: an analysis of computing times, Espacios 38 (51) (2017) 23.

[29] F. Chen, H. Wang, Y. Xie, C. Qi, An aco-based online routing method for multiple order pickers with congestion consideration in warehouse, J. Intell. Manuf. 27 (2) (2016) 389–408.

[30] C.-Y. Cheng, Y.-Y. Chen, T.-L. Chen, J.J.-W. 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, Int. J. Prod. Econ. 170 (2015) 805–814.

[31] T.-L. Chen, C.-Y. Cheng, Y.-Y. Chen, L.-K. Chan, An efficient hybrid algorithm for integrated order batching, sequencing and routing problem, Int. J. Prod. Econ. 159 (2015) 158–167.

[32] J.P. Van Den Berg, H.H. Van Der Hoff, et al., An order batching algorithm for wave picking in a parallel-aisle warehouse, IIE Trans. 33 (5) (2001) 385–398.

[33] K.J. Roodbergen, R. De Koster, Routing order pickers in a warehouse with a middle aisle, European J. Oper. Res. 133 (1) (2001) 32–43.

[34] H. Hwang*, Y. Oh, Y. Lee, An evaluation of routing policies for order-picking operations in low-level picker-to-part system, Int. J. Prod. Res. 42 (18) (2004) 3873–3889.

[35] H. Hwang*, D. Kim, Order-batching heuristics based on cluster analysis in a low-level picker-to-part warehousing system, Int. J. Prod. Res. 43 (17) (2005) 3657–3670.

[36] M.-C. Chen, C.-L. Huang, K.-Y. Chen, H.-P. Wu, Aggregation of orders in distribution centers using data mining, Expert Syst. Appl. 28 (3) (2005) 453–460.

[37] J. Won, S. Olafsson*, Joint order batching and order picking in warehouse operations, Int. J. Prod. Res. 43 (7) (2005) 1427–1442.

[38] Y.A. Bozer, J.W. Kile, Order batching in walk-and-pick order picking systems, Int. J. Prod. Res. 46 (7) (2008) 1887–1909.

[39] C.-Y. Tsai, J.J. Liou, T.-M. Huang, Using a multiple-ga method to solve the batch picking problem: considering travel distance and order due time, Int. J. Prod. Res. 46 (22) (2008) 6533–6555.

[40] M. Albareda-Sambola, A. Alonso-Ayuso, E. Molina, C.S. De Blas, Variable neighborhood search for order batching in a warehouse, Asia-Pac. J. Oper. Res. 26 (05) (2009) 655–683.

[41] L.-F. Hsieh, Y.-C. Huang, New batch construction heuristics to optimise the performance of order picking systems, Int. J. Prod. Econ. 131 (2) (2011) 618–630.

[42] S. Henn, Algorithms for on-line order batching in an order picking warehouse, Comput. Oper. Res. 39 (11) (2012) 2549–2563.

[43] S. Henn, G. Wäscher, Tabu search heuristics for the order batching problem in manual order picking systems, European J. Oper. Res. 222 (3) (2012) 484–494.

[44] S. Hong, A.L. Johnson, B.A. Peters, Large-scale order batching in parallel-aisle picking systems, IIE Trans. 44 (2) (2012) 88–106.

[45] O. Kulak, Y. Sahin, M.E. Taner, Joint order batching and picker routing in single and multiple-cross-aisle warehouses using cluster-based tabu search algorithms, Flexible Serv. Manuf. J. 24 (1) (2012) 52–80.

[46] A.H. Azadnia, S. Taheri, P. Ghadimi, M.Z. Mat Saman, K.Y. Wong, Order batching in warehouses by minimizing total tardiness: a hybrid approach of weighted association rule mining and genetic algorithms, Sci. World J. 2013 (2013).

[47] S. Henn, V. Schmid, Metaheuristics for order batching and sequencing in manual order picking systems, Comput. Ind. Eng. 66 (2) (2013) 338–351.

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

[49] M. Matusiak, R. de Koster, L. Kroon, J. Saarinen, A fast simulated annealing method for batching precedence-constrained customer orders in a warehouse, European J. Oper. Res. 236 (3) (2014) 968–977.

[50] S. Henn, Order batching and sequencing for the minimization of the total tardiness in picker-to-part warehouses, Flexible Serv. Manuf. J. 27 (1) (2015) 86–114.

[51] İ. Muter, T. Öncan, An exact solution approach for the order batching problem, IIE Trans. 47 (7) (2015) 728–738.

[52] J.C.-H. Pan, P.-H. Shih, M.-H. Wu, Order batching in a pick-and-pass warehousing system with group genetic algorithm, Omega 57 (2015) 238–248.

[53] C. Isler, G. Righetto, R. Morabito, Optimizing the order picking of a scholar and office supplies warehouse, Int. J. Adv. Manuf. Technol. 87 (5–8) (2016) 2327–2336.

[54] C.-C. Lin, J.-R. Kang, C.-C. Hou, C.-Y. Cheng, Joint order batching and picker manhattan routing problem, Comput. Ind. Eng. 95 (2016) 164–174.

[55] M. Çelik, H. Süral, Order picking in a parallel-aisle warehouse with turn penalties, Int. J. Prod. Res. 54 (14) (2016) 4340–4355.

[56] T. Chabot, L.C. Coelho, J. Renaud, J.-F. Côté, Mathematical model, heuristics and exact method for order picking in narrow aisles, J. Oper. Res. Soc. 69 (8) (2018) 1242–1253.

[57] P. Cortés, R.A. Gómez-Montoya, J. Muñuzuri, A. Correa-Espinal, A tabu search approach to solving the picking routing problem for large-and medium-size distribution centres considering the availability of inventory and k heterogeneous material handling equipment, Appl. Soft Comput. 53 (2017) 61–73.

[58] B. Menéndez, M. Bustillo, E.G. Pardo, A. Duarte, General variable neighborhood search for the order batching and sequencing problem, European J. Oper. Res. 263 (1) (2017) 82–93.

[59] B. Menendez, E.G. Pardo, J. Sánchez-Oro, A. Duarte, Parallel variable neighborhood search for the min–max order batching problem, Int. Trans. Oper. Res. 24 (3) (2017) 635–662.

[60] A.H. Schrotenboer, S. Wruck, K.J. Roodbergen, M. Veenstra, A.S. Dijkstra, Order picker routing with product returns and interaction delays, Int. J. Prod. Res. 55 (21) (2017) 6394–6406.

[61] C.A. Valle, J.E. Beasley, A.S. da Cunha, Optimally solving the joint order batching and picker routing problem, European J. Oper. Res. 262 (3) (2017) 817–834.

[62] T. Bódis, J. Botzheim, Bacterial memetic algorithms for order picking routing problem with loading constraints, Expert Syst. Appl. 105 (2018) 196–220.

[63] H.-Y. Lee, C.C. Murray, Robotics in order picking: evaluating warehouse layouts for pick, place, and transport vehicle routing systems, Int. J. Prod. Res. (2018) 1–21.

[64] D. Schubert, A. Scholz, G. Wäscher, Integrated order picking and vehicle routing with due dates, OR Spectrum 40 (4) (2018) 1109–1139.

[65] J. Zhang, X. Wang, K. Huang, On-line scheduling of order picking and delivery with multiple zones and limited vehicle capacity, Omega 79 (2018) 104–115.

[66] T. Van Gils, A. Caris, K. Ramaekers, K. Braekers, Formulating and solving the integrated batching, routing, and picker scheduling problem in a real-life spare parts warehouse, European J. Oper. Res. 277 (3) (2019) 814–830.

[67] Ç. Cergibozan, A.S. Tasan, Order batching operations: an overview of classification, solution techniques, and future research, J. Intell. Manuf. (2016) 1–15.

[68] J.A. Cano, A.A. Correa-Espinal, R.A. Gómez-Montoya, A review of research trends in order batching, sequencing and picker routing problems, Rev. Espac. 39 (04) (2018).

[69] E. Ardjmand, O.S. Bajgiran, E. Youssef, Using list-based simulated annealing and genetic algorithm for order batching and picker routing in put wall based picking systems, Appl. Soft Comput. 75 (2019) 106–119.

[70] I.G. Lee, S.H. Chung, S.W. Yoon, Two-stage storage assignment to minimize travel time and congestion for warehouse order picking operations, Comput. Ind. Eng. 139 (2020) 106129.

[71] D. Battini, C.H. Glock, E.H. Grosse, A. Persona, F. Sgarbossa, Human energy expenditure in order picking storage assignment: A bi-objective method, Comput. Ind. Eng. 94 (2016) 147–157.

[72] T. Djatna, M.Z. Hadi, Implementation of an ant colony approach to solve multi-objective order picking problem in beverage warehousing with drive-in rack system, in: 2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS), IEEE, 2017, pp. 137–142.

[73] S. Schwerdfeger, N. Boysen, Order picking along a crane-supplied pick face: The sku switching problem, European J. Oper. Res. 260 (2) (2017) 534–545.

[74] Ö. Özpeynirci, C. Kandemir, A pseudo-polynomial time algorithm for a special multiobjective order picking problem, Int. J. Inf. Technol. Decis. Mak. 14 (05) (2015) 1111–1128.

[75] Ç. Cergibozan, A.S. Tasan, Order batching operations: an overview of classification, solution techniques, and future research, J. Intell. Manuf. 30 (1) (2019) 335–349.

[76] M. Masae, C.H. Glock, E.H. Grosse, Order picker routing in warehouses: A systematic literature review, Int. J. Prod. Econ. 224 (2020) 107564.

[77] E. Ardjmand, W.A. Young II, G.R. Weckman, O.S. Bajgiran, B. Aminipour, N. Park, Applying genetic algorithm to a new bi-objective stochastic model for transportation, location, and allocation of hazardous materials, Expert Syst. Appl. 51 (2016) 49–58.

[78] E. Ardjmand, G. Weckman, N. Park, P. Taherkhani, M. Singh, Applying genetic algorithm to a new location and routing model of hazardous materials, Int. J. Prod. Res. 53 (3) (2015) 916–928.

[79] N. Karimi, M. Zandieh, H. Karamooz, Bi-objective group scheduling in hybrid flexible flowshop: a multi-phase approach, Expert Syst. Appl. 37 (6) (2010) 4024–4032.

[80] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V.G. Da Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, IEEE Trans. Evol. Comput. 7 (2) (2003) 117–132.

[81] T. Franzke, E.H. Grosse, C.H. Glock, R. Elbert, An investigation of the effects of storage assignment and picker routing on the occurrence of picker blocking in manual picker-to-parts warehouses, Int. J. Logist. Manage. (2017).