Scientia Iranica E (2021) 28(5), 2789–2811

Sharif University of Technology
Scientia Iranica
Transactions E: Industrial Engineering
http://scientiairanica.sharif.edu

Metaheuristics for a new MINLP model with reduced response time for on-line order batching

L. Hojaghani[a], J. Nematian[b,*], A.A. Shojaie[a], and M. Javadi[c]

a. *Department of Industrial Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran.*
b. *Department of Industrial Engineering, University of Tabriz, Tabriz, P.O. Box 51666-14766, Iran.*
c. *Department of Mechanical Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran.*

Received 21 July 2018; received in revised form 17 August 2019; accepted 28 December 2019

KEYWORDS
Warehousing;
On-line order batching;
Warehouse blocking;
Zoning system;
Idle time;
ABC meta-heuristic algorithm;
ACO meta-heuristic algorithm.

Abstract. Companies are looking for effective strategies to improve warehouse performance quality due to customers dissatisfaction of service. The order picking process is one of the main warehouse management strategies. As the inventory of stored items and the number of orders increased, the picking process and response time became more important. Effective coordination between order batching and order picking process is essential to improve the efficiency of the warehouse management system. In this paper, a novel Mixed-Integer Nonlinear Programming (MINLP) model for on-line order batching is proposed for improving the warehouse performance, which in turn results in the reduction of the response and idle times. The proposed method takes aim at the investigation of order classification for the first time in the picker-to-part system as a manual picking system and an online order batching system, with the intent of minimizing the turnover time and idle time. Besides, an order batching model in a blocked warehouse using a zoning system is proposed which is called Online Order Batching in Blocked Warehouse with One Picker for each Block (OOBBWOPB). The mentioned model is solved by two algorithms: Artificial Bee-Colony (ABC) algorithm and Ant-Colony (ACO) algorithm. Two numerical case studies are defined and analyzed using MATLAB software. According to the results compared with the results of Zhang et al. (2017) the proposed model shows better performance and the average customer order response time is significantly reduced (2017) and the ACO yields better results than ABC.

1. Introduction

In a supply chain, efforts, cooperation, and coordination of all facilities are the key factors to achieve successful supply chain management. In order to deliver orders and services to customers at the lowest cost and specified time, each facility must optimize its operations [1,2]. A distribution warehouse includes many operations and order picking is considered as one of the most important parts of such operations. According to Bartholdi and Hackman [3], order picking has about 55% of total activities costs in a warehouse. Tompkins et al. [4] indicated that the travel time includes approximately 50% of all order picking operations. Order picking is a time-consuming and energy-consuming process, which could include approximately 60% of the human activities in the warehouse. When promoting order picking operations, decreasing human interference in the process is a crucial fac-

*. *Corresponding author. Fax: +98 41 33354153*
E-mail address: jnematian@tabrizu.ac.ir (J. Nematian)

doi: 10.24200/sci.2019.51452.2185

tor. To achieve this goal, various methods have been developed. Zoning is another method, which could effectively reduce the travel distance of the pickers. In this case, each picker only picks up and picks in his own predefined area [5].

The zoning technique classifies pick locations into some subclasses (i.e. zones) and each zone has its exclusive one or more pickers. Reducing travel time and reducing traffic congestion is one of the most important advantages of zoning technology [6]. The progressive zoning system is called the picking and delivery system, which divides the picking line into picking zones. Then, each zone is dedicated to an exclusive picker, and all of them are often linked together with the help of a conveyor in the warehouse [7]. The algorithm is mainly suitable for small and medium-sized commodities that can be stored and accessed in relatively small and available pick locations. In this regard, the commodities such as home appliances, health and beauty, office, and food can be mentioned [8].

Therefore, warehousing experts realize that the order picking process is the most important factor in increasing productivity. The process of order picking is a warehouse action related to data restoration (article units) from their storage place in line with meeting customer orders' satisfaction, both internally and externally [9].

The problem arises because a large number of purchased goods and goods enter the warehouse, but they are gradually sold to customers. The issue of order picking is critical to every supply chain, because problematic activity in the field bears heavy costs, resulting in unsatisfactory customer services. Although extensive attempts have been made to mechanize the order picking process, a large number of systems with manual intervention have still been discovered [10,11]. This kind of handy order picking process falls into two classes [12]: First, Picker to parts system; in this system, the order picker arrives to the warehouse and gathers the ordered goods. Second, the parts to picker system; in this case, the mechanized restoration system delivers the ordered goods to the picker. In the first system, which is investigated in this study, three operative activities are identified [5]: Determining the location of the items, transferring the ordered items to the picking order, and determining the path for the pickers in the warehouse. The purpose of this research is to investigate the second activity of the system, in which various orders placed by the customers are combined in batches and released together for pickup. This has been proven to be essential for improving the efficiency of operations in the warehouse [13].

Order batching as one of the most prominent approaches in the order picking process, could greatly enhance this process performance by minimizing the processing time [14–16]. Regarding the accessibility of the orders, two manners are assumed [17]: The offline (static) batching orders, in which all orders are known from starting time of the shift, and the online (dynamic) batching orders, in which the orders are accessed dynamically. To solve the problems raised in the field of the online batching, time window is the most popular strategy, which falls into two fundamental categories: Fixed Time Window Batching (FTWB) and Variable Time Window Batching (VTWB). Nevertheless, these solutions are not respondents to an environment, where type and quantity of the orders are changing permanently. Considerable amounts of undelivered orders will be prepared to be delivered in a fixed period of time order in the FTWB section along with those scenarios having a relatively high arrival rate [10]. On the contrary, a long time is required to make the orders ready to be delivered in the VTWB section and relatively low arrival rate scenarios. The FTWB is known as a time-based batching process [18] and the VTWB is known as a quantity-based batching role [5]. According to recent researches [18], a hybrid rule-based algorithm has better performance than the time-based and quantity-based algorithms, in terms of saving time. Thus, a hybrid rule-based process would be able to substantially improve time-saving and wage costs in terms of arrival rates, picking devices, and time intervals. Furthermore, according to the findings of the study it can be claimed that the hybrid method is the best method of categorization in on-line order systems [19].

Due to the mentioned observations, in this research, for the first time the order batching model in a blocked warehouse is introduced named Online Order Batching in Blocked Warehouse with One Picker for each Block (OOBBWOPB). This is examined by an online order entry system using a nonlinear programming model based on the Artificial Bee Colony (ABC) algorithm and the Ant Colony (ACO) algorithm metaheuristic method.

The ABC is one of the most recent algorithms defined by Dervis Karaboga in 2005 and inspired by the intelligent behavior of honey bees [20]. Due to its simplicity and effectiveness, it has attracted much attention in recent years [21]. In this algorithm, a set of honey bees, called swarm, can successfully accomplish tasks through social cooperation. In the ABC algorithm, there are three types of bees: employed bees, onlooker bees, and scout bees. The employed bees search for food around the food source in their memory; meanwhile they share the information of these food sources to the onlooker bees. The onlooker bees tend to select good food sources from those found by the employed bees. The sources with higher quality (fitness) will have a greater chance of being selected by onlooker bees than food sources with lower quality.

The scout bees are translated from a few employed bees, which abandon their food sources and search for new ones. In the algorithm, the first half of the swarm consists of employed bees, and the second half constitutes the onlooker bees. The number of employed bees or the onlooker bees is equal to the number of solutions in the swarm. The ABC generates a randomly distributed initial population of SN solutions (food sources), where SN denotes the swarm size [22].

The ACO is one of the most recent techniques for approximate optimization that was introduced by Dorigo in 1990. The source of inspiration for these algorithms are real ant colonies. More specifically, ACO is inspired by the ants foraging behavior [23]. The core of this behavior is the indirect communication between the ants through chemical pheromone traces, which enables them to find short paths between their nest and food sources. This characteristic of real ant colonies is used in the ACO algorithm to solve, for example, discrete optimization problems [24,25]. these algorithms solve this model to form the initial order batches and to improve the batches to improve the response time and reduce the unemployment time of the operators.

The OOBBWOPB model includes the customer order assignment to minimize the processing time, which is closely associated with the Order Batching Problem (OBP). In this model, the entire warehouse is divided into several blocks, and each block is assigned a picker. The ultimate goal in this method is tuning the batches to optimize the turnover and idle time.

The follow-up content of this study is arranged as follows: In Section 2 a review of the OBP literature is presented. Section 3 develops a new optimization approach for minimizing customers orders delivery time and the best picking time. In Section 4, two meta-heuristic algorithms are used to solve the model and the results are analyzed.

2. Review of the literature

In today competitive world, companies are constantly trying to reduce transportation costs, while the bulk of warehousing activities are the transportation of goods. Given that the order response process is very costly for companies and determines the level of customer service, the role of warehouse management in a company to store and distribute customer orders is very important. In a Competitive Supply Chain (CSC) system, the speed of response to orders indicates the level of product and service quality, and the goal of Supply Chain Management (SCM) is to increase the value of the organization by increasing the level of service to the customer. Also, competition between companies in the transportation sector is basically based on time, cost and level of customer service. Effective supply chain management is one of the main business survival, meanwhile, the use of information technology in supply chain activities has increased the potential for value creation in the chain. Generally speaking, supply chain management emphasizes the continuous improvement of adaptability and flexibility of enterprises, and the ability to respond to market changes quickly and effectively. In recent years, extensive research has been done on the supply chain e.g., Nia et al. [26], Marandi and Zegordi [27], Teimoury and Kazemi [28], and Chaharsooghi and Sajedinejad [29].

In large warehouses with the online ordering system and a large number of orders, we do not know the exact time of arrival of orders and the content of orders, therefore, to reduce response time and costs companies have turned to the order batching system. Elsayed et al. [30] investigated the penalty function of batching operation in a warehouse with the AS/RS system. According to Gu et al. [31], there is no much research on order batching, and the issue needs to be addressed in different systems and conditions. As previously mentioned, according to Henn's research, warehouses with dynamic ordering systems (online systems) in which there is no information about the time of order entry and its items in the order list, it is better to study time window batching problem. In this view, in order to reduce response time, we specify the ordered items to be placed in batches. Richards [32] have studied an online ordering system in a single-block warehouse with the purpose of reducing the response time of orders and determining the best route for operators.

Combining on-line batch programming patterns with findings of off-line order batching, Henn et al. [14,16,33] developed the underlying on-line algorithm. Henn et al. [15,16,34] developed a two-competitive on-line algorithm using competitive analysis for the on-line order batching system. Henn et al. [34] have developed a model for batching orders and examined how to combine customer orders into one batch to reduce the length of the collection route. To solve the model, two approaches based on Tabu search have been proposed. the first is a (classic) Tabu Search (TS), and the second is the attribute-based hill climber (ABHC). They have shown that the methods, enabling the warehouse to respond to orders more effectively.

Zhang et al. [18] studied the problem of online order batching with Several operators in a single-block warehouse. FTWB is known as time-based batching process and VTWB is known as quantity-based batching rule. According to researchers [18], A hybrid rule-based algorithm provides better performance than time and quantity in terms of time-saving. Bahrami et al. [36] simulated different ordering policies in the Picker-to-part systems and then statistically analyzed the simulation results by designing experiments.

Recently, Van Gill et al. [37] have investigated the issue of increasing order picking efficiency by considering all storage, classification, zoning, and routing issues. The main purpose of this research was to identify the impact of these elements on response time reduction. It is worth noting that this research clearly analyzes the relationship between storage, classification, zoning, and routing for the first time.

One of the key and fundamental issues in every company is warehousing management which can significantly affect chain costs [38]. Zoning and batching are two major factors for improving the performance of the warehouse which in turn results in reducing the response time and increasing the customers satisfaction level [17]. Zoning and batching issues are often applied simultaneously in warehouses. However, in most literature, they are studied separately.

Choe [39] developed a single-block framework for picking locations and modeled the sorting function as a single automated server. This researcher has studied both FTWB and VTWB. In fact, the time of arrival of orders has been the main factor of their research. Tang and Chew [40] and Le-Duc and de Koster [5] studied VTWB. In their study, they investigated the exponential and/or processing times, and single-server station. In these studies, the sorting function is not directly modeled, and since the relationship between sorting and picking has not been determined, their results are limited when the knowledge about the sorting function is limited. Parikh and Meller [41] studied the applicability of batch picking and zone picking to existing distribution centers and proposed a cost model for choosing between picking strategies.

An analytical approach was developed by Van Nieuwenhuyse and de Koster [6] in order to estimate the throughput time of the expected system in an online ordering system for both VTWB and FTWB. All of these studies were conducted in randomized models.

Arranging a route for the means of transportation in the warehouse is the most important activity in the warehouse because it takes a long time to travel on this route compared to other activities. Therefore, it is a very important and inevitable problem to find the shortest route for goods in the warehouse for the picking and delivery of goods [42].

Matusiak et al. [43] proposed a simulated annealing algorithm for order batching in a warehouse with three cross-aisles. Considering on-line shopping, Valle et al. [44] investigated the problem of order batching and picker routing. Menéndez et al. [45] introduced a new routing method based on the ideas of the combined strategy for the OBP. In the context of the OBP, Menéndez et al. [46] proposed a heuristic approach based on the Variable Neighborhood Search method for picker routing a specific order due date.In the following, Jiang et al. [47] have also reviewed the ordering of orders and sequencing issues with regard to the pick-and-sort strategy in the online supermarket.

In recent years, extensive research has been conducted on the supply chain of which Menéndez et al. [46] and Lin et al. [48] and Hong and Kim [49] can be mentioned. Hong introduced a route-selecting model with an s-shape method in parallel-aisles in order to reduce the response time of the OBP. Considering all sub-problems of order batching, batch assignment and sequencing, as well as picker routing problems, Scholz et al. [13] proposed an approach to improve the operational efficiency of the distribution warehouse for the first time. In this paper, we have studied the effects of batching and zoning on the average throughput time of customers orders. Moreover, in this research, operator routing within blocks is based on s-shaped routing.

This paper extends the work of Tang and Chew [40], Le-Duc and de Koster [5], Parikh and Meller [41], and Yu and De Koster [17] into a multiple zones level. Recently, due to the emergence of e-commerce and e-business, global supply chain management pays special attention to small and frequent order delivery with lower total costs [50].

Therefore, to achieve efficient supply chain management, the picking and delivery system plays a major and critical role. Gibson et al. [51] asserted that most researches on collecting orders, have used the classic picker-to-parts system. In this traditional system, the benchmark for performance measurement is the vehicle or operator travel time. Therefore, by reducing the travel time, we can significantly reduce the system costs. Now considering that in pick-and-pass systems, the route and the length of the route are fixed for one operator or vehicle, to reduce the cost of travel, we should reduce the number of order batches and use the maximum capacity of each batch. This process can reduce the number of fixed route, thereby reducing response time [8].

In Pan investigation, based on group genetic algorithm a batching method is proposed which decreases the processing time of order collection. In this method, the issues of line balancing and reduction of the formed batches are considered.

In order batching, each order can have different numbers and items. The order picker must go through the entire warehouse with a vehicle of a certain volume and collect the ordered items. With the increase of order items as well as the number of orders received in a period of time, this process is very time-consuming and as a result, the response of orders becomes difficult. This problem is called NP-hard. If there are at most two orders per batch, it can be solved in polynomial time [52]. So far different models for order batching are proposed, some of which include: Elsayed et al. [30], Chen et al. [50], Gademann and Velde [52], Henn and Schmid [10], Kulak et al. [53], and Jiang et al. [47].

The previous researches did not consider the issue of warehouse blocking, and the lack of research regarding the classification of the online order batching system is very evident. However, according to some studies [8], the impact of warehouse blocking on the reduction of the order batch time and the improvement of customer satisfaction is obvious and significant. Hence, in this paper a new nonlinear model is proposed for online order batching by considering the blocked warehouse.

3. On-line Order Batching in Blocked Warehouse with One Picker for each Block (OOBBWOPB)

3.1. Problem description

According to Choe [39], there are two different viewpoints for order batching: Time window batching and proximity of pick location. Also, according to Zhang, the function of order batching is based on the time index. The ordering system basically conforms to the queuing process, and it operates in batch generation and batch operations.

This paper investigates the effect of batching on the reduction of the orders response time and the idle time of operators. As you know, the ordering system is based on queuing systems, which operate on batch generation and batch operation. The batches formation take place through different methods. According to Zhang, the hybrid method, which combines two methods of FTWB and VTWB, is one of the best approaches that applies both volumetric capacity constraints of the batches and the time duration of the batch formation. In fact, the purpose of batching is to specify which orders are best placed in a batch and when it is better for the batches to be dealt with. The standard response process of orders batch generation system is shown in Figure 1. It is similar to the Zhang arrangement, except that the latter considered the warehouse as an integrated unit. Each operator collects the ordered items from the relevant racks according to the inventory of the entire warehouse. In this study, the warehouse is divided into k blocks, and an operator is assigned to each block. The formed batches are segregated based on the items of each block during operation, and the list of all blocks is sent to the operators simultaneously. The completion time of an order batch is the time period during which orders collection in all blocks is terminated.

In the illustration, the arrive time is the moment when the customer order is entered into the system, and enter time is the moment at which the prepared batch is entering the batch operation. Between these two points, there is a batch time, which is the time interval between the arrive and enter time; This time interval extends from the first order batch to the batch which is ready for the next stage. The prepared batches are operated according to First In-First Out (FIFO) system, and the duration of batch queuing is called the wait time. Order batch items are segregated according to which block they belong to and are sent to the picker of the corresponding block through a separate list for each block. It is worth mentioning that the sum of all lists items related to the k block is equal to the sum of the batch items. In addition, when the picking system is used to collect items, the goods associated with each order can be identified. In this way, the picker is labeled during the collection. All blocks are similarly equipped with parallel aisles, and there is a depot in front of the leftmost side of the aisle inside each block. It should be noted that the warehouse layout is based on the random storage system. The service time includes the time interval for the picker to receive the list and deliver the collected items to the warehouse. At this time, it

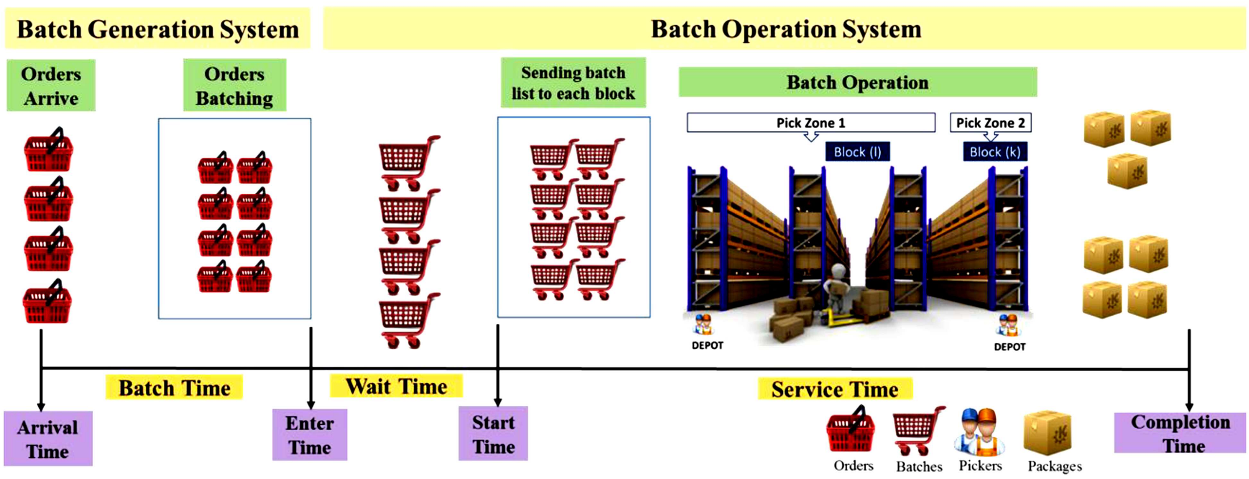

Figure 1. The Online Order Batching in Blocked Warehouse with one Picker for each Block (OOBBWOPB) system: Batch generation system and batch operation system.

is not possible to pause the process and edit the list. When all the batch items are collected and delivered from different blocks, a specific batch is completed.

According to the aforementioned statement, in a warehousing system, the improvement of the picker-to-parts response system has been examined. In this system, the warehouse has benefited from the Automatic Storage and Retrieval System (AS/RS). In this regard, new influential parameters involved in the order picking process, have been identified. By analyzing these parameters, not only can the existing problems such as wasting operator time and increasing the length of the route be solved, but the response time has also been improved, thereby increasing the overall efficiency of the system. Regarding the fixed route connected length of different sections of ASRS warehouse, we decided to decrease the number of batches of this fixed route to minimize the response time (turnover time) and total picker idle time in each block which in turn could lead to the optimization of the cost operation.

3.2. Current batching rules

For batching rules, shown in Figure 2, each step is described below:

- First step: Customer orders enter the batch generation system at the start time t^{start}. The maximum capacity of the order picking a vehicle for block k is Q_k;
- Second step: In FTWB $[t^{\text{start}}, t^{\text{end}}]$ ($t^{\text{end}} = t^{\text{start}} + t^b$, where t^b is the batching time), all arriving customer orders are assigned to one batch, if the quantity of the items $S \leq Q_k, t^b \leq T$, and $t^b \leq t^{TQ}$, where S is sum of the volume of items related to each block in the batch, T is upper bound for batching time and t^{TQ} is the time needed for entering orders to a batch to reach the specified volume of TQ; otherwise we can generate new batch;
- Third step: One should back to the second step until the ending time of the system;
- Fourth step: The generated batches are entered into the batch operating system.

Table 1 shows the algorithm for batch generation and batch operation according to the program written by the MATLAB R2013b software.

3.3. OOBBWOPB optimization model

The heuristic algorithms of ABC and ACO are used to formulate and solve the optimized Mixed Integer Nonlinear Programming (MINLP) model for online batching of orders. In order to analyze the problem, a model that requires complete information of all incoming orders is proposed. The constants and variables used in this model are listed in Tables 2 and 3, respectively.

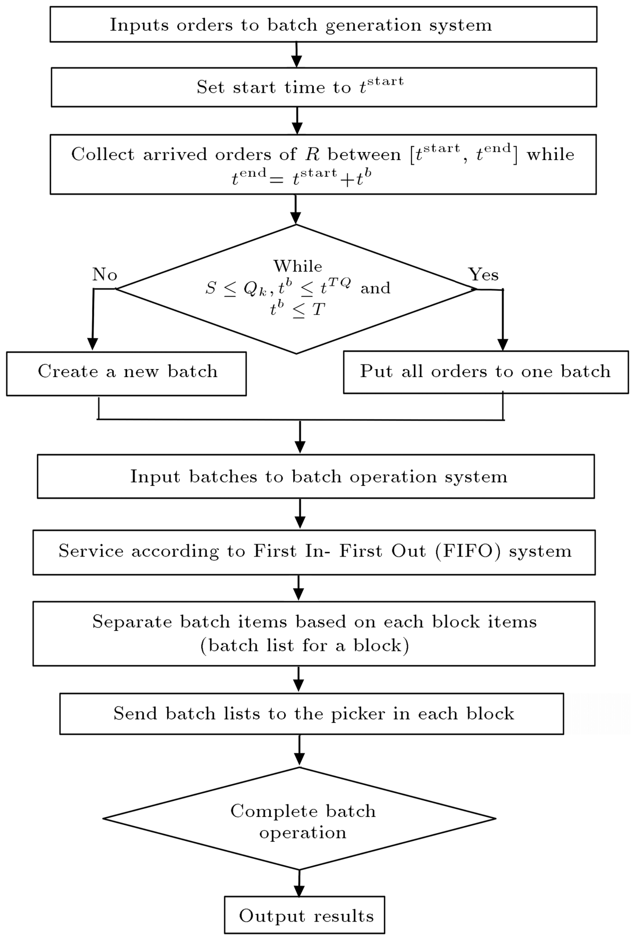

Figure 2. The flowchart of batch generation and batch operation system.

Therefore, due to the aforementioned constants and variables, according to the ABC and ACO algorithms the OOBBWOPB is modeled as follow.

The objective function (1) minimizes the sum of the maximum response time of the order and the total amount of idle time of the picking operator in each block:

$$\min \max_j \in \{1, 2, \ldots, m\} \left\{ T_j^{\text{response}} + \sum_{k=1}^{l} IT_{jk} \right\}. \tag{1}$$

Constraint (2) ensures that each order is limited to one batch:

$$\sum_{j=1}^{m} x_{ij} = 1, \qquad \forall_{i \in} \{1, \ldots, n\}. \tag{2}$$

Constraint (3) ensures that all items of batch j are assigned to the pickers:

$$\sum_{i=1}^{n} q_i . x_{ij} = \sum_{k=1}^{l} y_{jk}, \qquad \forall_{j \in} \{1, \ldots, m\}. \tag{3}$$

Constraint (4) ensures that the volume of goods allocated to each picker is not greater than the capacity of

Table 1. Batch generation and batch operation algorithm.

Customer orders start to enter the batch generation system at time t^{start}.
In FTWB $[t^{\text{start}}, t^{\text{end}}]$ ($t^{\text{end}} = t^{\text{start}} + t^{b}$) all arrival customer orders are assigned based on the following situations:
If $t^{b} \leq \min(t^{TQ}, T)$ and $S \leq Q_k$
Input the orders directly into the batch operation system;
(t^{b} is the batching time, t^{TQ} is the time needed for entering orders to a batch to reach the specified volume of TQ, T is the upper bound of the batching time and S is sum of the volume of items related to each block in the batch)
Else
Generate a new batch;
End
Input the orders into the batch operation system;
Service as First In-First Out (FIFO) system;
Orders of each block are isolated and sent to their respective operators at the same time; in each block, order pickers are routed by using an S-shaped strategy to pick orders;
If all items in a batch are collected and delivered in different blocks:
Batch operation is complete and new batch should be isolated;
End

Table 2. The model constants.

n	Number of customer orders
m	The upper limit of the number of batches (a trivial upper bound can be $n = m$)
T	Upper bound of the batching time
A	Total number of aisles
P	Number of pickers for a block
L	Number of storage locations
W_K	Center to center distance between two aisles
L_k	Length of aisles
x_k^{veh}	The distance where the transport vehicle of the block k will travel to delivery place
t_i^a	Arrival time of customer order i to the order batching system where $(0 < t^a{}_i \leq t^a{}_{i+1})$
t_{setup}	Setup time (time needed for each picker before starting to prepare each batch)
q_i	Number of items of custom order i
q_k	The capacity of the vehicle for the block k
Q_p	Quantity of goods
v_{travel}	Travel speed, the distance each picker can travel in the unit of time
v_{pick}	The picking speed, the number of items the picker searches and takes in the time unit
$v_{v(k)}$	The speed of the vehicle k
k	Number of the warehouse blocks (each block has been assigned by an operator)
I	Upper bound of the warehouse blocks number
TQ	Sum of the capacity of the order picking vehicles for all blocks

the transport vehicle associated with the picker:

$$y_{jk}.Q_p \leq Q_k, \qquad \forall_{j\in}\{1,\ldots,m\}, \forall_{k\in}\{1,\ldots,l\}. \quad (4)$$

Constraint (5) calculates the entrance time to the service queue:

$$t_j^{\text{enter}} = t_j^{\text{batch}} + \min\{x_{ij}.t_i^a\}, \quad \forall_{j\in}\{1,\ldots,m\}. \quad (5)$$

Constraint (6) calculates the duration of the batching time:

$$t_j^{\text{batch}} = \min\left\{t_j^{TQ}, T\right\}, \qquad \forall_{j\in}\{1,\ldots,m\}. \quad (6)$$

Constraint (7) calculates the waiting time for the order batching.

$$t_j^{\text{wait}} = t_j^{\text{start}} - t_j^{\text{enter}}, \qquad \forall_{j\in}\{1,\ldots,m\}. \quad (7)$$

Inequality (8) shows the start time of the j batch and ensures that when the total goods related to the batch

Table 3. The model variables.

t_j^{TQ}	Duration of time needed for orders of batch j to reach the specified volume of TQ
t_j^{batch}	Duration of time to generate batch j
t_j^{enter}	Entry time for batch j
t_j^{wait}	Waiting time for batch j
t_j^{start}	Starting time for responding to batch j
t_j^{service}	Service time of batch j
x_{ij}	1 if order i is assigned to batch j, 0 otherwise
T	Upper bound for batching time
y_{jk}	The number of goods that corresponds to the picker k of the batch j
dis_{jk}	Distance function of the picking tour for batch j related to the block k, while the routing method is S-shape
A_{jk}	The number of aisles in block k having at least one picking place in batch j
A_{jk}^L	The number of the left side aisle of block k with at least one picking place in batch j
A_{jk}^F	Farthest aisle number from the block k that contains at least one picking place in batch j
D_{jk}^F	Distance of the farthest goods of block k that must be picked in the batch j, from the starting point of each aisle
T_j^{response}	The completion time for batch j
T_k^{veh}	The time required to carry the k section package to the delivery location
T_{jk}^{complete}	The completion time of the operator k activity for batch j
T_{jk}^{operator}	The time that the operator k serves for the order batch j
IT_{jk}	Idleness time of the operator k during the preparation time of the order batch j (Idleness of the operator k after completion of the activity related to the batch j until the start of order $j+1$)
T_j^{operator}	The completion time of all operators task in batch j

are identified, services can be started for each category:

$$t_j^{\text{start}} \geq \max\{x_{ij}.t_i^a\}, \qquad \forall_{j\in}\{1,\ldots,m\}. \tag{8}$$

Inequality (9) ensures that the system sends the list of each order batch to the picking operator. At this time, all operators have completed the work on the previous category.

$$t_{j+1}^{\text{start}} \geq \max_k \left\{T_{jk}^{\text{complete}}\right\}, \qquad \forall_{j\in}\{1,\ldots,m\}. \tag{9}$$

Eq. (10) calculates the duration of service time for each category and ensures that the termination of service to each order batch is the time when all the goods belonging to that category have arrived at the place of delivery. T_{jk}^{complete} and T_k^{veh} will be calculated in Eqs. (11) and (16).

$$T_j^{\text{service}} = \max_k \left\{T_{jk}^{\text{complete}} + T_k^{\text{veh}}\right\}, \qquad \forall_{j\in}\{1,\ldots,m\}. \tag{10}$$

Eq. (11) calculates the completion time of each operator work in each order batch:

$$T_{jk}^{\text{complete}} = T_j^{\text{start}} + T_{jk}^{\text{operator}}, \qquad \forall_{j\in}\{1,\ldots,m\}, \forall_{k\in}\{1,\ldots,l\}. \tag{11}$$

Eq. (12) calculates the completion time of each order batch:

$$T_j^{\text{response}} = \max_k \left\{T_{jk}^{\text{complete}}\right\}, \quad \forall_{j\in}\{1,\ldots,m\}. \tag{12}$$

Eq. (13) calculates the interval between service start points of two consecutive order batches.

$$T_j^{\text{operator}} = \max_k \left\{T_{jk}^{\text{operator}}\right\}, \quad \forall_{j\in}\{1,...,m\}. \tag{13}$$

Eq. (14) calculates the idling time for each operator during the service completion of each batch:

$$IT_{jk} = T_j^{\text{operator}} - T_{jk}^{\text{operator}}, \qquad \forall_{j\in}\{1,\ldots,m\}, \qquad \forall_{k\in}\{1,\ldots,l\}. \tag{14}$$

Eq. (15) calculates the duration of service time of the

$$dis_{jk}=\begin{cases}\left(A^L_{jk}-1\right)W_k+2D^F_{jk}+\left(A^F_{jk}-1\right)W_k & A^L_{jk}=A^F_{jk}, A_{jk}=1\\ \left(A^L_{jk}-1\right)W_k+2D^F_{jk}+\left(A^F_{jk}-1\right)W_k, & A_{jk}=\text{even}\\ \left(A^L_{jk}-1\right)W_k+(A_{jk}-1)L_k+2D^F_{jk}+(A_{jk}-1)W_k+\left(A^F_{jk}-1\right)W_k, & A_{jk}=odd.\end{cases} \tag{17}$$

Box I

operator to each order batch, which is the sum of the operator moving time at a specific speed, the time of picking up of the goods from the shelf with the specified picking speed, and the launching time.

$$T^{\text{operator}}_{jk}=\frac{dis_{jk}}{v_{\text{travel}}}+\frac{y_{jk}}{v_{\text{pick}}}+t_{\text{setup}},$$

$$\forall_{j\in}\{1,\ldots,m\},\forall_{k\in}\{1,\ldots,l\}. \tag{15}$$

Eq. (16) calculates the time of shipment of the vehicle:

$$T^{veh}_k=\frac{x^{veh}_k}{v_v(k)},\qquad \forall_{k\in}\{1,\ldots,l\}. \tag{16}$$

Eq. (17), shown in Box I, is the distance the operator of each block k travels to collect the items of each order batch j corresponding to that block.

Eq. (18) shows that listed variables are natural numbers, i.e. positive and integer:

$$y_{jk},t^{\text{enter}}_j,t^{\text{batch}}_j,t^{TQ}_j,t^{\text{wait}}_j,t^{\text{start}}_j,T^{\text{service}}_j,T^{\text{complete}}_{jk},$$

$$T^{\text{operator}}_j,T^{veh}_k,IT_{jk},T^{\text{response}}_j\geq 0. \tag{18}$$

Eq. (19) shows that x_{ij} is binary:

$$x_{ij}=0 \text{ or } 1 \qquad \forall_{i\in}\{1,\ldots,n\},\forall_{j\in}\{1,\ldots,m\}. \tag{19}$$

The distance function (dis_{jk}) is derived from Zhang's research [18] and conforms to the s-shaped routing method [14,49]. In this study, the ABC and ACO algorithms are used to solve the nonlinear model. In the next section, the corresponding real numerical example is analyzed, and then the results are presented.

4. Experiment parameters

4.1. Purpose

A series of experiments were considered in order to identify warehouse blocking performance. The purpose of these tests and experiments is to discuss turnover time in a number of blocks, and batching time per block (t^{batch}_j) in upper bound time. Then results will be compared to those reported by Zhang et al. [18]. All tests were performed on an Intel Core M processor with 8.0GB RAM. For simulation study MATLAB R2013b software is used to run the algorithm. Two case studies are considered and parameters, routing strategies, and Design Of Experiment (DOE) of each case are described in detail in each section.

4.2. Parameters of Case Study 1

The experiments are developed by defining a set of parameters including warehouse layout A, P, L, L_k, W_k, v_{travel}, v_{pick}, t_{setup}, Q_k, and q_i, which were introduced in Subsection 3.3 in Table 2. In addition, t^a_i is used as the arrival time of the customer order, which is randomly valued according to the customer order quantity. The total number of ordered items are $n=120$, $n=240$, and $n=480$. Moreover, the warehouse values are given in Table 4.

The routing strategy of the system is s-shaped, and the random-based storage policy has been formulated for the whole system. According to the following assumption, there are also several other effective parameters in this field: The ordered goods arrive during

Table 4. Parameters value of the experiment of Case Study 1.

Row	Parameter	Symbol	Value
1	Total number of aisles	A	10
2	Number of pickers for a block	P	1
3	Number of storage locations	L	900 (90 storage locations have been placed on both sides of the aisle)
4	Length of an aisles	L_k	45 meters
5	Center-to-center distance between aisles	W_K	5 meters
6	Travelling speed of picker	v_{travel}	48 meter per minute
7	Pickup speed	v_{pick}	6 item per minute
8	Setup time	t_{setup}	3 minutes
9	Capacity of order picker vehicle	Q_k	30, 45, 60, 75
10	Constant distribution of number of goods for each order	q_i	$\mathrm{q_i}\sim\mathrm{U}\,(1,5)$

Table 5. Parameter's value for statistical analysis of Case Study 2.

Row	Parameter		Value
1	Upper bound of the batching time	T	10, 20, 60
2	Capacity of order picker vehicle	Q_k	30, 45, 60
3	Number of customer orders	n	120, 240, 360
4	Number of the warehouse blocks	k	2, 4, 6

4 hours and the inter-arrival times (the time in minutes between the arrival of customer order i with $i + 1$) are distributed exponentially and in proportion to the arrival rate λ, where $\lambda = 0.5$, $\lambda = 1$, or $\lambda = 2$. The samples are generated by choosing different values for λ, Q_k and T.

4.3. Parameters of Case Study 2

In Case Study 2, in order to better analyze the blocking effect, a small problem was developed and discussed in which the bottom warehouse involved 100 similar storage locations and 10 locations on both sides of the aisle. Each picker collects the orders from both the left and right sides of the aisle, simultaneously. The rest of the information used in this case, is exactly similar to the previous one. In Case Study 2, 23 scenarios are developed, in which four main parameters of the model i.e., T, Q_k, n, and k are identified for sensitivity investigation. Also, three-parameter levels of high, average, and low are considered in order to examine these parameter's effects on target function. The parameter values are shown in Table 5.

4.4. Experimental results analysis

4.4.1. Results of Case Study 1

In the first case study, the number of 36 problem categories as shown in Table 6, has been generated. 50 instance and finally, the total number of 1800 samples have been calculated for each problem. In this case study, response time extracted from the non-blocking model proposed by Zhang et al. [18] is compared with this model considering the blocking system. As can be seen from the table, T^{response}, l^{upper}, and K^{upper} represent the response time of orders, the upper bound for picker number, and the upper bound for block number respectively. Table 6 also calculates and shows the improvement percentage of the presented model. Obviously, after applying a blocking system in a storage system, the trend of response time was much higher than that of a non-blocking system. The results obtained from the studies are compared with those of Ho and Tseng [19] research. It can be seen that the expected response time is reduced after the application of the blocking system. In addition, scenarios numbers 10 and 11 have a block that indicates data storage status similar to those without blocking. There is also a minimal improvement that shows that the blocking system with ABC and ACO algorithms is more efficient than the non-blocking system proposed by Zhang, in which the genetic algorithm is used. Regarding response time with the upper bound block number, it should be taken into account that a larger number of pickers will lead to better performance and efficiency, although this will certainly lead to higher wage costs.

The ABC algorithm

The research results in Table 6 show that when $n = 120$, $n = 240$, and $n = 480$, the following minimum response times with ABC algorithm are obtained under T and N, respectively: 196.6649, 204.8746, 175.1411, 195.9874, 196.6079, 215.1282, 229.7651, 228.2962, 175.9501, 189.2771, 190.998, and 201.6838.

It can be seen that blocking leads to a reduction in response time. This type of decline is approximately 1.03%–26.54%. For example when $n = 120$, $Q_k = 30$, $T = 10$ (Scenario 1), and when there are two pickers in a warehouse that is in non-blocking mode, the order response time is expected to be 229.37. However, when the warehouse is divided into two blocks, the response time becomes 207.1348, indicating a reduction of approximately 9.69%. Consequently, in most cases, increasing the number of blocks leads to a decrease in response time.

Table 7 shows an example of the best results for Scenario 1, in which orders are divided into 10 batches, including entry, waiting, start, service, completion, response, and idle times. According to obtained result, the average response time for this repetition is 207.2487, although the average response time for 10 repetitions is reported in Table 6 as 207.1348. The idle time of each picker is another parameter that has been investigated. According to statistics in Table 7, when there are 2 pickers for scenario number 1, the sum of idle times of almost all batch numbers remains almost unchanged. In fact, each picker in a batch should have a specific time to run the order batching system. Batch processing will not terminate unless each picker completes its task in each block. For example in a system with 2 blocks and pickers, each picker needs a specific time to accomplish their duties. When the last picker completes the task, the batch is completed, and the next batch starts. Thus the sum of picker idle time

Table 6. The response time with upper bound block number, different capacities and various time intervals.

Series no.	n	Q_k	T	No blocking [19]		Blocking				
							ABC algorithm		ACO algorithm	
				T^{response}	l^{upper}	K^{upper}	T^{response}	Improvement percentage	T^{response}	Improvement percentage
1	120	30	10	229.37	2	2	207.1348	9.6940	196.8475	14.17
2	–	30	20	238.62	2	2	211.9492	11.1771	204.4650	14.3135
3	–	30	30	227.06	3	3	196.6649	13.3864	189.6013	16.4972
4	–	45	15	230.55	2	2	218.4684	5.24	214.6384	6.90
5	–	45	30	231.08	2	2	209.8094	9.20	198.9310	13.91
6	–	45	45	246.39	2	2	204.8746	16.85	197.9802	19.65
7	–	60	20	238.62	2	2	198.028	17.011	190.2251	20.28
8	–	60	40	224.12	2	2	175.1411	21.8539	171.4753	23.49
9	–	60	60	264.28	2	2	210.1725	20.47	204.2734	22.70
10	–	75	25	243.66	1	1	241.1398	1.03	240.1629	1.43
11	–	75	50	268.56	1	1	264.7907	1.40	262.5840	2.22
12	–	75	75	251.77	2	2	195.9874	22.1562	191.1731	24.07
13	240	30	5	258.74	4	4	234.8567	9.2306	229.6815	11.23
14	–	30	10	258.15	4	4	216.9057	15.9769	211.0133	18.25
15	–	30	15	259.15	5	5	196.6079	24.1336	190.2511	26.59
16	–	45	7.5	260.89	3	3	250.3443	4.0422	250.1242	4.12
17	–	45	15	261.75	3	3	215.1282	17.8116	218.4771	16.53
18	–	45	22.5	269.2	3	3	234.9026	12.7405	217.8330	19.08
19	–	60	10	260.67	2	2	250.8928	3.750	250.1285	4.044
20	–	60	20	246.36	2	2	229.8074	6.7188	210.0428	14.74
21	–	60	30	265.17	2	2	229.7651	13.3518	227.8410	14.07
22	–	75	12.5	260.43	2	2	253.3516	2.7179	247.2745	5.051
23	–	75	25	276.17	2	2	228.2962	24.5768	229.1821	17.01
24	–	75	37.5	288.64	2	2	241.7122	16.2583	237.0395	17.87
25	480	30	2.5	239.54	8	8	175.9501	26.54	190.9626	20.27
26	–	30	5	245.18	7	7	208.8257	14.82	205.9912	15.98
27	–	30	7.5	247.77	7	7	201.2619	18.77	199.8566	19.34
28	–	45	3.75	244.77	6	6	189.2771	22.6714	177.2863	27.57
29	–	45	7.5	247.7	5	5	201.3472	18.7133	200.8328	18.92
30	–	45	11.25	246.61	5	5	204.5003	17.075	200.9098	18.53
31	–	60	5	240.51	5	5	215.1638	10.5385	209.6521	12.83
32	–	60	10	244.46	5	5	190.998	21.8694	190.677	22.00
33	–	60	15	249.39	4	4	229.7128	7.8901	212.1236	14.94
34	–	75	6.25	242.01	4	4	236.8028	2.1516	225.8842	6.66
35	–	75	12.5	251.59	5	5	208.0244	17.3161	196.9139	21.73
36	–	75	18.75	251.53	4	4	201.6838	19.8171	200.2188	20.40

Table 7. The result of Artificial Bee-Colony (ABC) algorithm for scenario number 1.

Batch no.	Idle time	Batch time	Enter time	Wait time	Start time	Service time	Response time
1	$[0; 1.1249]$	19.7679	32.9849	31.1303	64.1152	95.99001	160.1052
2	$[0; 3.170]$	17.4446	18.0085	1.8918	19.9003	77.41685	97.3172
3	$[0; 6.1203]$	18.1321	96.8961	51.243	148.139	101.60701	251.7461
4	$[0; 0.1249]$	19.0508	94.2265	63.4094	137.636	21.90114	179.5371
5	$[0; 0.853]$	18.5165	198.79	27.7285	226.518	12.16182	238.6802
6	$[3.389; 0]$	19.0095	90.6824	20.2561	70.9386	61.2938	172.2324
7	$[0.373; 0]$	11.9295	126.2408	90.0659	176.306	26.16701	242.4738
8	$[0.360; 0]$	13.1276	135.654	64.4594	200.113	59.12892	259.2423
9	$[0; 1.9583]$	0	151.3443	29.6288	180.973	45.54527	226.5184
10	$[0; 0.9584]$	0	91.18621	148.4941	239.680	35.99001	244.6344

Table 8. The result of Artificial Bee-Colony (ABC) algorithm for scenario number 3.

Batch no.	Idle time	Batch time	Enter time	Wait time	Start time	Service time	Response time
1	[1.0416; 0; 3.45833]	29.127334	39.4037873	20.7524452	60.1562325	89.2494818	149.4057143
2	[1.0416; 0; 3.45853]	29.72972	75.5786389	57.1891349	132.767774	68.3577506	201.125524
3	[0.0416; 0; 3.45833]	29.78532	44.4982384	54.9074759	99.4057143	33.3620595	132.767774
4	[1.0416; 0; 3.45833]	13.8452	13.9803715	0	13.9803715	86.1758609	100.1562325
5	[0.70839; 0; 2.9583]	17.12197	136.894045	48.3355677	185.229612	89.9218081	275.151421
6	[1.04166; 0; 3.4583]	24.53316	104.214899	56.9106253	161.125524	84.104088	245.229612
7	[0.541664; 0; 3.458]	6.360553	179.579375	25.5720456	205.151421	16.0835369	221.234957
8	[0.541664; 0; 3.125]	0	219.157743	2.07721447	221.234957	27.01267354	248.247631

Table 9. The result of Ant-Colony (ACO) algorithm for scenario number 1.

Batch no.	Idle time	Batch time	Enter time	Wait time	Start time	Service time	Response time
1	[0.374099; 0]	19.7980	31.38320	30.9341	62.3173	103.792	169.1093
2	[0.374999; 0]	17.65466	17.85603	0	17.8561	44.4612	62.3173
3	[0.374999; 0]	19.4208	96.07582	63.4615	159.5372	73.2089	232.74615
4	[0.374993; 0]	19.1071	73.45186	58.78055	132.2325	77.3047	209.5372
5	[0.541667; 0]	19.3368	179.2573	47.2611	196.5186	53.1617	249.6803
6	[0.374993; 0]	19.1397	50.25339	49.8559	100.1093	62.1232	162.2325
7	[0.364293; 0]	12.8575	80.6898	72.0563	152.74615	29.7323	182.4739
8	[0.376665; 0]	13.0755	134.1049	68.3689	202.4739	56.7685	259.2424
9	[0; 1.958443]	0	141.8242	67.4182	209.2424	17.2761	226.5185
10	[2.3750; 0]	0	90.41424	19.2661	109.6803	104.9542	214.6345

Table 10. The result of Ant-Colony (ACO) algorithm for scenario number 3.

Batch no.	Idle time	Batch time	Enter time	Wait time	Start time	Service time	Response time
1	[0.374999; 0; 0.098]	19.79794	31.3832	30.9340	62.3172	77.9810	140.2982
2	[1.374999; 0; 0.115]	17.65465	17.8560	0	17.8560	120.9611	128.8172
3	[0.374993; 0; 0.987]	19.42086	96.0758	63.4613	159.5371	56.2090	215.7461
4	[1.374993; 0; 0.005]	19.10708	73.4518	58.7805	132.2324	60.6047	192.8371
5	[0.541667; 0; 0.0067]	19.33680	179.2573	47.2611	226.5184	43.1618	269.6803

should be changed equally in each batch to obtain a reasonable result. Thus, in the best sample result, it is logical to adjust the average idle time to approximately 1.84 minutes. In the following Table 8 shows one of the best sample results for scenario number 3 and batch number 8.

The ACO algorithm

Regarding the ACO algorithm, when $n = 120$, $n = 240$, and $n = 480$, the minimum response times under various T and N conditions are as following, respectively (Table 7): 189.6013, 197.9802, 171.4753, 191.1731, 190.2511, 217.8330, 210.0428, 229.1821, 190.9626, 177.2863, 190.677, and 196.9139. It can be seen from Table 7 that similarly, in the ABC algorithm, blocking will result in a reduction in response time. This decline is approximately 1.43% - 27.57%. For example, in a non-blocking warehouse with two operators, when $n = 120$, $Q_k = 30$, orders response time will be 229.37. When the warehouse is divided into 2 blocks, the order response time becomes 196.8475, which shows a decrease of approximately 14.17%. According to the results, scenario number 10 represents the lowest improvement in reducing response time, and scenario number 28 represents the highest improvement. Table 9 shows one of the best sample results for scenario number 1, batch number 10, and Table 10 shows one of the best sample results for scenario number 3, and batch number 5.

4.4.2. Results of Case Study 2

In the second case study, each scenario is repeated 10 times and the average results of the repetition in each scenario are shown in Tables 11–13. The result clearly

Table 11. The result for scenarios numbers 1–27.

Se. no	Number of blocks	Number of orders in 4 hour	Q_K	T	ABC algorithm Response time	ABC algorithm Mean of idle times (for per operator)	ACO algorithm Response time	ACO algorithm Mean of idle times (for per operator)
1	2	120	30	10	31.46586	0.625	26.46421	0.26
2				20	32.59738	0.717	28.90588	0.42
3				60	34.97678	1.038	32.02254	0.368
4			45	10	26.88803	0.671	23.13921	0.48
5				20	34.8953	0.6415	30.58088	0.2056
6				60	28.65056	0.482	31.34754	0.217
7			60	10	34.9871	1.2515	29.3562	0.3417
8				20	21.8313	0.4865	20.90588	0.4750
9				60	24.6595	0.625	32.02254	0.483
10		240	30	10	34.40607	3.9585	43.13921	0.3417
11				20	35.22805	2.7085	37.58088	0.3283
12				60	32.1056	5.625	35.34754	0.4483
13			45	10	32.31897	5.607	26.46421	1.2883
14				20	36.87066	1.0415	30.2547	0.1017
15				60	33.4589	5.2705	34.1235	0.1283
16			60	10	30.7990	1.0415	28.31	0.4217
17				20	36.787	2.2915	29.0234	0.21
18				60	50.86018	0.628	42.145	0.187
19		360	30	10	67.6630	1.0415	59.328	0.0903
20				20	52.9769	2.2915	52.1236	0.235
21				60	43.803	2.7085	40.18	0.1384
22			45	10	52.7657	3.9585	48.045	0.1263
23				20	46.71353	2.7095	43.2736	0.3957
24				60	41.98127	1.0415	38.3307	0.2312
25			60	10	46.71353	2.7085	43.0087	0.3417
26				20	41.98127	1.0415	40.18	0.3683
27				60	44.3746	2.7085	39.468	0.7350

shows the effects of blocking on reducing response time. For example, in the three scenarios including numbers 1, 28, and 55, the response time of both ABC and ACO algorithms is decreased by increasing the number of blocks with the same three-parameter values of T, n, Q_k equal to 10, 120, and 30 respectively. It is worth noting that in these three scenarios, the response time and average idle times for the ACO algorithm were lower than the ABC algorithm. Compared to other scenarios, the ACO algorithm showed a better performance. Conversely, the average idle time of each picker increased. However, the average increase in idle time was insignificant compared to the reduction in the response time level shown in Figure 3.

4.5. Design Of Experiment (DOE) for analyzing the research results

4.5.1. DOE of Case Study 1

The ABC algorithm

The two assumptions of H0 and H1, when comparing

Table 12. The result for scenarios no. 28–54.

Se. no	Number of blocks	Number of orders in 4 hour	Q_K	T	ABC algorithm: Response time	ABC algorithm: Mean of idle times (for per operator)	ACO algorithm: Response time	ACO algorithm: Mean of idle times (for per operator)
28	4	120	30	10	28.6257	4.0275	24.6651	1.03808
29				20	26.78054	5.972	22.13	1.2204
30				60	26.5773	4.0275	19.28	1.28
31			45	10	23.12815	5.1385	20.1269	1.3675
32				20	22.1472	4.861	21.7419	1.1592
33				60	23.42084	2.6385	17.8095	1.1806
34			60	10	23.17927	3.3335	18.4867	1.300
35				20	20.2773	3.9585	19.8810	1.2105
36				60	26.7017	3.5415	18.4992	1.4075
37		240	30	10	37.3493	3.9585	32.0070	1.4206
38				20	38.9842	4.0285	31.9859	1.3944
39				60	38.3076	5.2085	30.3012	1.2792
40			45	10	35.48914	4.935	25.7882	1.3275
41				20	30.1120	3.3335	21.6634	1.2342
42				60	28.9145	4.375	25.7635	1.3058
43			60	10	28.0672	6.0415	23.9582	1.2614
44				20	24.1576	3.3335	23.9980	1.2075
45				60	31.2005	4.375	24.1346	1.4057
46		360	30	10	52.4328	6.0415	43.5150	1.2743
47				20	61.9593	6.875	39.8497	1.2319
48				60	53.8346	4.375	34.2031	1.3140
49			45	10	43.83187	4.1665	35.7983	1.3927
50				20	34.4335	2.875	33.0984	1.2794
51				60	39.76728	4.2085	31.3467	1.1683
52			60	10	36.4128	1.875	29.6005	1.2118
53				20	34.8676	4.375	26.3298	1.1675
54				60	27.3302	4.4415	23.2455	0.9967

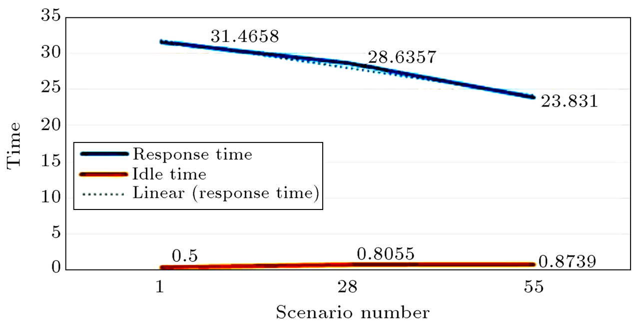

Figure 3. Average increased level of idle time and decrease level of order response time for the three scenario.

the results of the ABC algorithm with the non-blocking results in the first case study, are considered as follows:

H0: There is no significant difference between the two samples;

H1: There is a significant difference between the two samples.

To test this hypothesis, paired T-test is used. This test is used when the same sample is tested in two different circumstances. Compared with other tests, one of the advantages of the T-test is that it allows us to easily identify the differences. As seen in Table 14, the average response time in the non-blocking mode was 250.2906, while in blocking mode using ABC method, it dropped to 216.1189. Table 15, shows the correlation between two groups with the value of 0.57. It is indicated that there is significant correlation between the two models. Table 16, shows the mean test results of the difference between the blocking and non-blocking methods and the average response time in terms of the average response time. This table shows that the reduction in the average response time

Table 13. The result for scenarios no. 55–81.

Se. no	Number of blocks	Number of orders in 4 hour	Q_K	T	ABC algorithm Response time	ABC algorithm Mean of idle times (for per operator)	ACO algorithm Response time	ACO algorithm Mean of idle times (for per operator)
55	6	120	30	10	23.8310	4.8695	17.1011	3.8038
56				20	19.7756	5.46875	19.4604	5.2153
57				60	26.9153	5.625	14.9024	4.3217
58			45	10	21.4276	3.3335	20.2223	3.0062
59				20	17.3798	3.0415	21.9404	2.3657
60				60	22.8120	2.625	22.0712	2.1850
61			60	10	18.4095	3.0805	16.0344	2.9654
62				20	14.6472	4.375	14.1159	4.1091
63				60	28.8574	3.368	25.7489	3.2074
64		240	30	10	36.6023	2.625	28.254	3.3114
65				20	33.97472	3.79	34.064	3.009
66				60	37.98916	3.374	33.1256	2.9804
67			45	10	28.4896	3.3625	25.0765	2.8037
68				20	28.02358	3.624	26.6843	3.2742
69				60	30.06142	5.06	29.1872	4.9040
70			60	10	27.5935	2.0305	25.554	1.9056
71				20	18.3930	2.625	22.1423	2.1083
72				60	29.7453	3.995	23.4217	2.4450
73		360	30	10	50.3123	4.6245	42.1438	3.8128
74				20	48.4042	2.375	40.1851	1.9453
75				60	51.5352	3.3395	50.0137	3.0927
76			45	10	34.7184	4.2185	30.1283	3.7124
77				20	36.7643	3.1	30.9873	2.583
78				60	37.5417	2.625	34.1760	2.1821
79			60	10	32.7923	4.1155	28.4354	3.0109
80				20	29.79935	4.507	27.0961	3.3426
81				60	34.3214	4.625	33.4357	3.836

Table 14. Paired samples statistics for Artificial Bee-Colony (ABC) algorithm and non-blocking results.

Paired samples statistics					
		Mean	N	Std. deviation	Std. error mean
Pair 1	Non-blocking	250.2906	36	14.22869	2.37145
	ABC	216.1189	36	21.7781	3.6296

Table 15. Paired samples correlations for Artificial Bee-Colony (ABC) algorithm and non-blocking results.

Paired samples correlations				
		N	Correlation	Sig.
Pair 1	Non-blocking & ABC	36	0.578	0.000

Table 16. Paired samples test for Artificial Bee-Colony (ABC) and non-blocking results.

		Paired differences: Mean	Paired differences: Std. deviation	Paired differences: Std. error mean	95% confidence interval of the difference: Lower	95% confidence interval of the difference: Upper	t	df	Sig. (2-tailed)
Pair 1	Non-blocking & ABC	34.1716	17.856	2.976	28.130	40.2132	11.482	35	0.000

Table 17. Paired samples statistics for Ant-Colony (ACO) algorithm and non-blocking results.

Paired samples correlations					
		Mean	***N***	**Std. deviation**	**Std. error mean**
Pair 1	Non-blocking	250.2906	36	14.22869	2.37145
	ACO	210.904	36	21.6268	3.6044

Table 18. Paired samples correlations for Ant-Colony (ACO) algorithm and non-blocking results.

Paired samples correlations.				
		N	**Correlation**	**Sig.**
Pair 1	Non-blocking & ACO	36	0.611	0.000

Table 19. Paired Samples test for Ant-Colony (ACO) algorithm and non-blocking results.

Paired samples test									
		Paired differences							
		Mean	**Std. deviation**	**Std. error mean**	**95% confidence interval of the difference**				**Sig. (2-tailed)**
					Lower	**Upper**	***t***	***df***	
Pair 1	Non-blocking - ACO	39.386	17.1544	2.8590	33.582	45.1904	13.776	35	0.000

Table 20. Paired samples statistics for Artificial Bee-Colony (ABC) and Ant-Colony (ACO) algorithms results.

		Mean	***N***	**Std. deviation**	**Std. error mean**
Pair 1	ABC	216.1188	36	21.7781	3.6296
	ACO	210.9042	36	21.6268	3.6044

Table 21. Paired samples correlations for Artificial Bee-Colony (ABC) and Ant-Colony (ACO) algorithms results.

Paired samples correlations				
		N	**Correlation**	**Sig.**
Pair 1	ACO & ABC	36	0.958	0.000

in the blocking method is significant due to obtained significance amount of zero which is less than 0.05. The correlation coefficient was 0.578, so there is a significant correlation between the two models.

The ACO algorithm

To examine the results obtained from the ACO algorithm with non-blocking results, we consider the following assumptions:

H0: There is no significant difference between the results of the two methods;

H1: There is a significant difference between the results of the two methods.

As is clear from Tables 17–19, the average score in non-blocking was 250.2906, which reduced to 210.904 using the ACO method. Because the significance level of 0.0 is less than 0.05, this value is statistically significant. The correlation coefficient was 0.611, indicating the significant correlation between the results of the two methods.

Comparison of the results of the ACO algorithm with the ABC algorithm

To examine the results of the ACO with the ABC solution method, we considered the following assumptions in Case Study 1:

H0: There is no significant difference between the results of the two methods;

H1: There is a significant difference between the results of the two methods.

As shown in Tables 20–22, the average ABC score was 216.1188, which dropped to 210.9042 in the ACO

Table 22. Paired samples test for Artificial Bee-Colony (ABC) and Ant-Colony (ACO) algorithms results.

Paired samples test									
		Paired differences							
		Mean	Std. deviation	Std. error mean	95% confidence interval of the difference				Sig. (2-tailed)
					Lower	Upper	***t***	***df***	
Pair 1	ACO & ABC	5.2145	6.2962	1.0493	3.0842	7.344	4.969	35	0.000

Table 23. The ANOVA analysis to the impact parameter recognition of k, n, Q_k, and T on response time for Artificial Bee-Colony (ABC) algorithm.

ANOVA					
	Sum of squares	***df***	**Mean square**		**Sum of squares**
Parameter k					
Between groups	1006.083	2	503.042	4.223	0.018
Within groups	9291.093	78	119.117	–	–
Total	10297.177	80	–	–	–
Parameter n					
Between groups	4751.698	2	2375.849	33.418	0.000
Within groups	5545.478	78	71.096	–	–
Total	10297.177	80	–	–	–
Parameter Q_k					
Between groups	1151.647	2	575.824	4.911	0.010
Within groups	9145.530	78	117.250	–	–
Total	10297.177	80	–	–	–
Parameter T					
Between groups	122.530	2	61.265	0.470	0.627
Within groups	10174.647	78	130.444	–	–
Total	10297.177	80	–	–	–

algorithm. Because the significance level of 0.0 is less than 0.05, this value is statistically significant. The correlation coefficient is calculated to be 0.958, so there is a significant correlation between the two groups. It is worth noting that by comparing the results of the average score of the ACO and ABC methods, it can be said that the improvement of the ACO method was more than that of the ABC method.

4.5.2. DOE of Case Study 2

By assuming H0 and H1 which represent the effect of three levels (Table 5) of 4 parameters of k, n, Q_k, and T in the ABC and ACO algorithms, the second case study will be obtained as below:

H0: There is no significant difference between the obtained results of three levels of k, n, Q_k, and T;

H1: There is a significant difference between the obtained results of three levels of k, n, Q_k, and T.

It is necessary to adjust the input parameters of the algorithms to improve the solution proposed by the algorithm function. The ANOVA test results for response times of Case Study 2 are presented in Tables 23 and 24. This is mainly used to identify the optimization level of each parameter. According to these results, parameters k, n, and Q_k are categorized as effective parameters. For example, for parameter k, the significance value of the ABC algorithm was 0.018, and the significance value of the ACO algorithm was 0.0, both of which are less than 0.05 and are categorized as the effective parameters. In addition, it is mentioned that the parameter T did not have a significant level of the obtained results. It is noteworthy that the parameter effectiveness could be investigated by changing the amount of these values.

In the following, the Tamhane test is used to determine the optimal values of the parameters defined in Table 5, and the results are presented in Tables 25 and 26. The values of the optimal parameters are

Table 24. The ANOVA analysis to the impact parameter recognition of k, n, Q_k, and T on response time for Ant-Colony (ACO) algorithm.

	Sum of squares	*df*	Mean square	*F*	Sig.
Parameter k					
Between groups	1401.696	2	700.848	10.588	0.000
Within groups	5163.229	78	66.195	–	–
Total	6564.925	80	–	–	–
Parameter n					
Between groups	3100.484	2	1550.242	34.903	0.000
Within groups	3464.441	78	44.416	–	–
Total	6564.925	80			
Parameter Q_k					
Between groups	665.923	2	332.961	4.403	0.015
Within groups	5899.002	78	75.628	–	–
Total	6564.925	80	–	–	–
Parameter T					
Between groups	14.878	2	7.439	0.089	0.915
Within groups	6550.047	78	83.975	–	–
Total	6564.925	80	–	–	–

Table 25. Tamhane test to determine the optimal values of the parameter k, n, Q_k and T time for Artificial Bee-Colony (ABC) algorithm. Dependent variable: response time.

$(I)K$	$(J)K$	Mean difference $(I-J)$	Std. error	Sig.	95% confidence interval	
					Lower bound	Upper bound
Parameter k						
2	4	5.35070556	2.97042932	.176	–1.7464247	12.4478358
	6	8.54229296*	2.97042932	.014	1.4451627	15.6394232
4	2	–5.35070556	2.97042932	.176	–12.4478358	1.7464247
	6	3.19158741	2.97042932	.533	–3.9055429	10.2887177
6	2	–8.54229296*	2.97042932	.014	–15.6394232	–1.4451627
	4	–3.19158741	2.97042932	.533	–10.2887177	3.9055429
Parameter n						
120	240	–8.16458667*	1.99260128	.001	–13.1150358	–3.2141375
	360	–18.71061815*	2.23687222	.000	–24.2893430	–13.1318933
240	120	8.16458667*	1.99260128	.001	3.2141375	13.1150358
	360	–10.54603148*	2.61247511	.001	–16.9961273	–4.0959357
360	120	18.71061815*	2.23687222	.000	13.1318933	24.2893430
	240	10.54603148*	2.61247511	.001	4.0959357	16.9961273
Parameter Q_k						
30	45	7.31140593	2.94706859	.052	–.0430417	14.6658535
	60	8.54322815**	2.94706859	.018	1.1887805	15.8976758
45	30	–7.31140593	2.94706859	.052	–14.6658535	.0430417
	60	1.23182222	2.94706859	.916	–6.1226254	8.5862698
60	30	–8.54322815*	2.94706859	.018	–15.8976758	–1.1887805
	45	–1.23182222	2.94706859	.916	–8.5862698	6.1226254

*The mean difference is significant at the 0.05 level.

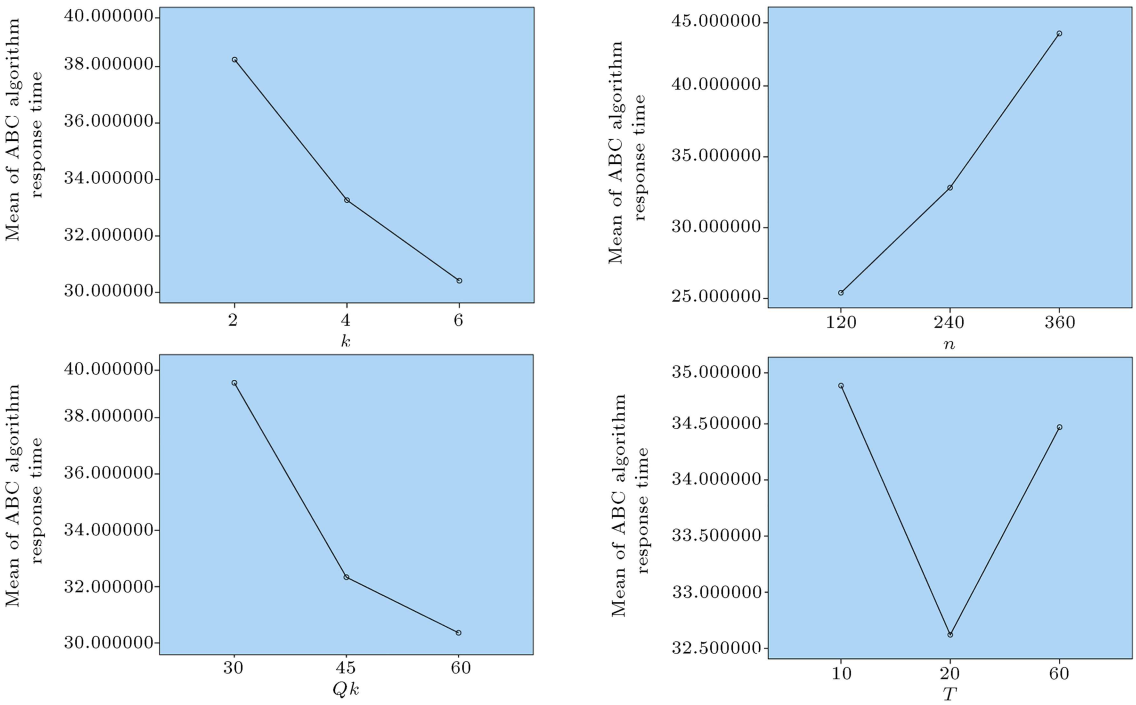

Figure 4. Change of response time by changing the value of parameters k, n, Q_k, and T for Artificial Bee-Colony (ABC) algorithm.

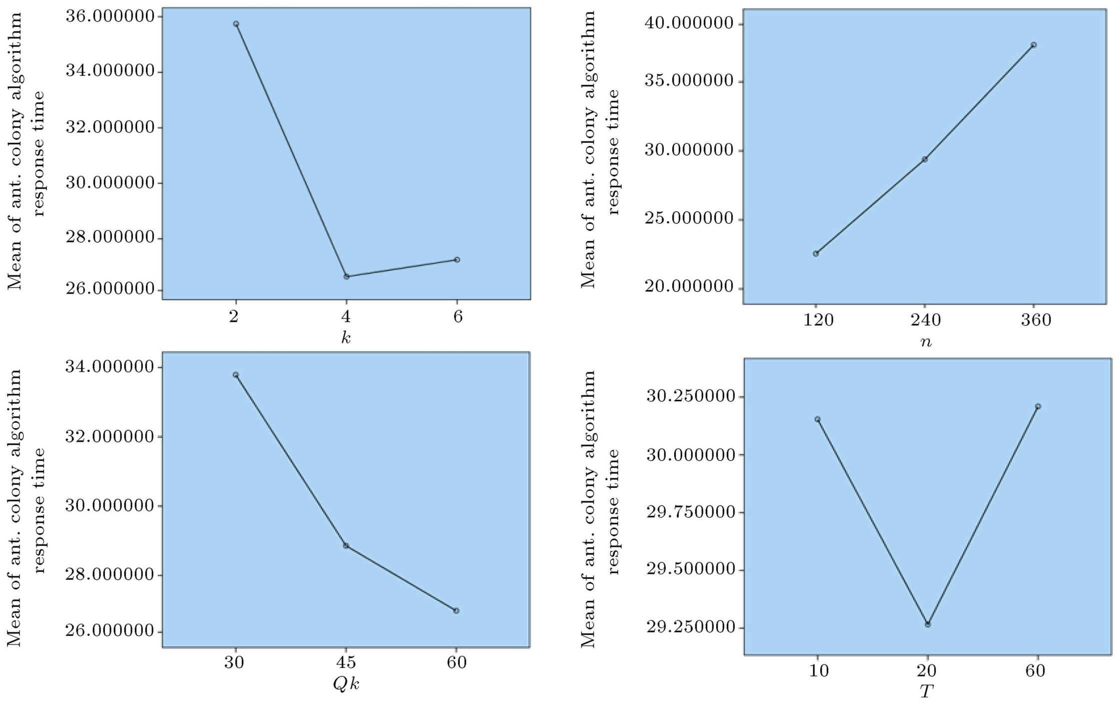

Figure 5. Change of response time by changing the value of parameters k, n, Q_k, T for Ant-Colony (ACO) algorithm.

specified based on the results shown in Figures 4 and 5. The values of the optimal parameters have been presented in Table 27.

5. Summary and future works

Throughput time of arbitrary order is called an order picking system. With the increase in order picking speed, the delivery time is shortened, and the quality of warehouse service is improved. Order picking area and order batching are the two most important factors to improve order picking efficiency. The current research aims to study online order batch processing using warehouse blocking systems in pick-to-parts and

Table 26. Tamhane test to determine the optimal values of the parameter k, n, Q_k and T for Ant-Colony (ACO) algorithm. Dependent variable: response time.

$(I)K$	$(J)K$	Mean difference $(I-J)$	Std. error	Sig.	95% confidence interval	
					Lower bound	Upper bound
Parameter k						
2	4	9.11419333*	2.21435118	.000	3.5882515	14.6401352
	6	8.50305630*	2.21435118	.001	2.9771144	14.0289982
4	2	–9.11419333*	2.21435118	.000	–14.6401352	–3.5882515
	6	–.61113704	2.21435118	.963	–6.1370789	4.9148048
6	2	–8.50305630*	2.21435118	.001	–14.0289982	–2.9771144
	4	.61113704	2.21435118	.963	–4.9148048	6.1370789
Parameter n						
120	240	–6.84212815*	1.49933021	.000	–10.5416043	–3.1426520
	360	–15.13166000*	1.93367942	.000	–19.9326440	–10.3306760
240	120	6.84212815*	1.49933021	.000	3.1426520	10.5416043
	360	–8.28953185*	1.97055692	.000	–13.1749446	–3.4041191
360	120	15.13166000*	1.93367942	.000	10.3306760	19.9326440
	240	8.28953185*	1.97055692	.000	3.4041191	13.1749446
Parameter Q_k						
30	45	4.92997481	2.36687369	.121	–.9765890	10.8365386
	60	6.79710148*	2.36687369	.020	.8905377	12.7036653
45	30	–4.92997481	2.36687369	.121	–10.8365386	.9765890
	60	1.86712667	2.36687369	.734	–4.0394371	7.7736904
60	30	–6.79710148*	2.36687369	.020	–12.7036653	–.8905377
	45	–1.86712667	2.36687369	.734	–7.7736904	4.0394371

* The mean difference is significant at the 0.05 level.

Table 27. The values of the optimal parameters.

	k	n	Q_k	T
ABC algorithm	6	120	60	20
ACO algorithm	4	120	60	20

wide-aisle warehouses. To decrease orders turnover (response time) and sum of the idle times of the operators, limitations of each block transportation device, order quantity, and time needed for each batch formation were investigated and finally a number of the formed batches and orders in each batch were studied subsequently.

The findings of the present study show that the warehouse blocking system affected the reduction of the completion time of warehouse orders. As is clear from the results of different scenarios, this effect is significant. The other important point extracted from this research is that the obtained results are positively correlated to the results of Zhang research which clearly confirms the validity of the presented model. According to the results, it can be said that the ACO algorithm improved more than the ABC algorithm.

In the past research, there are few and limited studies on order classification in warehouse blocking, and there is no research on online orders in the field of warehouse blocking. Therefore, in future researches, the effects of operator skill on the reduction of the search time should be investigated. Also, the study of the impacts of proper skill training on the goods searching and picking manner is necessary. Another issue that should be considered in the future is the inclusion of queuing systems to prioritize incoming orders in the order batching system. Finally, it is worth noting that adding more operators to each block or using operators as an auxiliary force in a block with a temporary increase in workload can help reduce the response time, thereby improving customer satisfaction.

References

1. Chen, T.-L., Cheng, C.-Y., Chen, Y.-Y., et al. "An efficient hybrid algorithm for integrated order batching, sequencing and routing problem", *International Journal of Production Economics*, **159**, pp. 158–167 (2015).

2. Rezaei, M., Shirazi, M.A., and Karimi, B. "A multi-

objective SCOR-based decision alignment for supply chain performance management", *Scientia Iranica., Transaction E, Industrial Engineering*, **25**(5), pp. 2807–2823 (2018).

3. Bartholdi, J. and Hackman, S. "Warehouse & distribution science: Release 0.96. The supply chain and logistics institute, Georgia institute of technology", Atlanta, USA Google Scholar (2014).
4. Tompkins, J.A., White, J.A., Bozer, Y.A., et al. *Facilities Planning*, John Wiley & Sons (2010).
5. Le-Duc, T. and de Koster, R.M. "Travel time estimation and order batching in a 2-block warehouse", *European Journal of Operational Research*, **176**(1), pp. 374–388 (2007).
6. Van Nieuwenhuyse, I. and de Koster, R.B. "Evaluating order throughput time in 2-block warehouses with time window batching", *International Journal of Production Economics*, **121**(2), pp. 654–664 (2009).
7. Melacini, M., Perotti, V., and Tumino, V. "Development of a framework for pick-and-pass order picking system design", *The International Journal of Advanced Manufacturing Technology*, **53**(9–12), pp. 841–854 (2011).
8. Pan, J.C.-H., Shih, P.-H., and Wu, M.-H. "Order batching in a pick-and-pass warehousing system with group genetic algorithm", *Omega*, **57**, pp. 238–248 (2015).
9. Van Nieuwenhuyse, I., de Koster, R., and Colpaert, J. "Order batching in multi-server pick-and-sort warehouses", *DTEW-KBI_0731*, pp. 1–29 (2007).
10. Henn, S. and Schmid, V. "Metaheuristics for order batching and sequencing in manual order picking systems", *Computers & Industrial Engineering*, **66**(2), pp. 338–351 (2013).
11. Lee, J.A., Chang, Y.S., Shim, H.-J., et al. "A study on the picking process time", *Procedia Manufacturing*, **3**, pp. 731–738 (2015).
12. Pérez-Rodríguez, R., Hernández-Aguirre, A., and Jöns, S. "A continuous estimation of distribution algorithm for the online order-batching problem", *The International Journal of Advanced Manufacturing Technology*, **79**(1-4), pp. 569–588 (2015).
13. Scholz, A., Schubert, D., and Wäscher, G. "Order picking with multiple pickers and due dates-simultaneous solution of order batching, batch assignment and sequencing, and picker routing problems", *European Journal of Operational Research*, **263**(2), pp. 461–478 (2017).
14. Henn, S. "Algorithms for on-line order batching in an order picking warehouse", *Computers & Operations Research*, **39**(11), pp. 2549–2563 (2012).
15. Henn, S. "Order batching and sequencing for the minimization of the total tardiness in picker-to-part warehouses", *Flexible Services and Manufacturing Journal*, **27**(1), pp. 86–114 (2015).
16. Henn, S., Koch, S., and Wöscher, G. "Order batching in order picking warehouses: a survey of solution approaches", *Warehousing in the Global Supply Chain*, pp. 105–137, Springer (2012).
17. Yu, M. and De Koster, R.B. "The impact of order batching and picking area zoning on order picking system performance", *European Journal of Operational Research*, **198**(2), pp. 480–490 (2009.
18. Zhang, J., Wang, X., Chan, F.T.S., et al. "On-line order batching and sequencing problem with multiple pickers: A hybrid rule-based algorithm", *Applied Mathematical Modelling*, **45**, pp. 271–284 (2017).
19. Ho, Y-C. and Tseng, Y-Y. "A study on order batching methods of order-picking in a distribution centre with two cross-aisles", *International Journal of Production Research*, **44**(17), pp. 3391–3417 (2006).
20. Agarwal, V., Gupta, V., and Singh, P.K. "A systematic review on artificial bee colony optimization technique", *International Journal of Control Theory and Application*, **9**(11), pp. 5487–5500 (2016).
21. Saucedo, L.E.C., Sanchez-Solis, J.P., López-Ramos, F., et al. "Implementation of an Artificial Bee Colony to Solve an Order Picking Problem", *Handbook of Research on Metaheuristics for Order, Picking Optimization in Warehouses to Smart Cities*, pp. 144–160: IGI Global (2019).
22. Li, Z. and Zhou, V. "An effective batching method based on the artificial bee colony algorithm for order picking", In *Ninth International Conference on Natural Computation (ICNC)*, pp. 386–391 (2013).
23. Blum, C. "Ant colony optimization: Introduction and recent trends", *Physics of Life Reviews*, **2**(4), pp. 353–373 (2005).
24. Engin, O. and Güçlü, V. "A new hybrid ant colony optimization algorithm for solving the no-wait flow shop scheduling problems", *Applied Soft Computing*, **72**, pp. 166–176 (2018).
25. Jia, Z.-h., Wang, Y., Wu, C., et al. "Multi-objective energy-aware batch scheduling using ant colony optimization algorithm", *Computers & Industrial Engineering*, **131**, pp. 41–56 (2019).
26. Nia, A.R., Haleh, H., and Saghaei, A. "Energy-conscious dynamic sequencing method for dual command cycle unit-load multiple-rack automated storage and retrieval systems", *Scientia Iranica., Transaction E, Industrial Engineering*, **24**(6), pp. 3371–3393 (2017).
27. Marandi, F. and Zegordi, S. "Integrated production and distribution scheduling for perishable products", *Scientia Iranica*, **24**(4), pp. 2105–2118 (2017).
28. Teimoury, E. and Kazemi, S. "An integrated pricing and inventory model for deteriorating products in a two-stage supply chain under replacement and shortage", *Scientia Iranica., Transaction E, Industrial Engineering*, **24**(1), p. 342 (2017).

29. Chaharsooghi, S. and Sajedinejad, A. "Determination of the number of kanbans and batch sizes in a JIT supply chain system", *Scientia Iranica., Transaction E., Industrial Engineering*, **17**(2), p. 143 (2010).

30. Elsayed, E., Lee, M.-K., Kim, S., et al. "Sequencing and batching procedures for minimizing earliness and tardiness penalty of order retrievals", *The International Journal of Production Research*, **31**(3), pp. 727–738 (1993).

31. Gu, J., Goetschalckx, M., and McGinnis, L.F. "Research on warehouse operation: A comprehensive review", *European Journal of Operational Research*, **177**(1), pp. 1–21 (2007).

32. Richards, G., *Warehouse Management: A Complete Guide to Improving Efficiency and Minimizing Costs in the Modern Warehouse*, Kogan Page Publishers (2017).

33. Henn, S., Koch, S., Doerner, K.F., et al. "Metaheuristics for the order batching problem in manual order picking systems", *Business Research*, **3**(1), pp. 82–105 (2010).

34. Henn, S. and Wäscher, G. "Tabu search heuristics for the order batching problem in manual order picking systems", *European Journal of Operational Research*, **222**(3), pp. 484–494 (2012).

35. Li, W., Guo, C., Su, B., et al. "Photodegradation of four fluoroquinolone compounds by titanium dioxide under simulated solar light irradiation", *Journal of Chemical Technology & Biotechnology*, **87**(5), pp. 643–650 (2012).

36. Bahrami, B., Aghezzaf, E.H., and Limere, V. "Using simulation to analyze picker blocking in manual order picking systems", *Procedia Manufacturing*, **11**, pp. 1798–1808 (2017).

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

38. Muppani, V.R. and Adil, G.K. "Efficient formation of storage classes for warehouse storage location assignment: A simulated annealing approach", *Omega*, **36**(4), pp. 609–618 (2008).

39. Choe, K.-I., *Aisle-Based Order Pick Systems with Batching, Zoning, and Sorting*, Georgia Institute of Technology (1990).

40. Tang, L.C. and Chew, E.-P. "Order picking systems: batching and storage assignment strategies", *Computers & Industrial Engineering*, **33**(3–4), pp. 817–820 (1997).

41. Parikh, P.J. and Meller, R.D. "Selecting between batch and zone order picking strategies in a distribution center", *Transportation Research Part E: Logistics and Transportation Review*, **44**(5), pp. 696–719 (2008).

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

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

44. Valle, C.A., Beasley, J.E., and da Cunha, A.S. "Optimally solving the joint order batching and picker routing problem", *European Journal of Operational Research*, **262**(3), pp. 817–834 (2017).

45. Menéndez, B., Pardo, E.G., Alonso-Ayuso, A., et al. "Variable neighborhood search strategies for the order batching problem", Computers & Operations Research, **78**, pp. 500–512 (2017).

46. Menéndez, B., Bustillo, M., Pardo, E.G., et al. "General variable neighborhood search for the order batching and sequencing problem", *European Journal of Operational Research*, **263**(1), pp. 82–93 (2017).

47. Jiang, X., Zhou, Y., Zhang, Y., et al. "Order batching and sequencing problem under the pick-and-sort strategy in online supermarkets", *Procedia Computer Science*, **126**, pp. 1985–1993 (2018).

48. Lin, C.-C., Kang, J.-R., Hou, C.-C., et al. "Joint order batching and picker Manhattan routing problem", *Computers & Industrial Engineering*, **95**, pp. 164–174 (2016).

49. Hong, S. and Kim, Y. "A route-selecting order batching model with the S-shape routes in a parallel-aisle order picking system", *European Journal of Operational Research*, **257**(1), pp. 185–196 (2017).

50. Chen, M.-C. and Wu, H.-P. "An association-based clustering approach to order batching considering customer demand patterns", *Omega*, **33**(4), pp. 333–343 (2005).

51. Gibson, D.R. and Sharp, G.P. "Order batching procedures", *European Journal of Operational Research*, **58**(1), pp. 57–67 (1992).

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

53. Kulak, O., Sahin, Y., and Taner, M.E. "Joint order batching and picker routing in single and multiple-cross-aisle warehouses using cluster-based tabu search algorithms", *Flexible Services and Manufacturing Journal*, **24**(1), pp. 52–80 (2012).

54. Cergibozan, Ç. and Tasan, A.S. "Order batching operations: an overview of classification, solution techniques, and future research", *Journal of Intelligent Manufacturing*, **30**(1), pp. 1–15 (2016).

Biographies

Leily Hojaghani received a BS degree from the University of Science and Culture, Tehran, Iran, and an MS degree from Qazvin Islamic Azad University, Qazvin, Iran in Industrial Engineering, She is a PhD student of Industrial Engineering at Islamic Azad University

Javad Nematian received his Msc and PhD in Industrial Engineering from the Sharif University of Technology in 2004 and 2008. From 2007 to 2008, he was with SYSTeM Research Group at the University of Ghent as a visiting researcher. His research interests include integer programming and uncertain mathematical programming. From 2009 to 2014, he was with the Industrial Engineering Department at the University of Tabriz as an Assistant Professor. Since 2014, he has been an Associate Professor in the Industrial Engineering Department at the University of Tabriz. He is a member of the Iranian Operations Research Society. From 2013 to 2018, he was also the Head of the Industrial Engineering Department.

Amir Abbas Shojaie was born in Tehran, Iran, in 1970. He received his BS, MS, and PhD degrees in Industrial Engineering from the Tehran University, Science & Research, Tehran, Iran, in 2003. In 2007, he joined the School of Industrial Engineering, University of Azad, Tehran South Branch, as a Lecturer and after one year he became an Assistant Professor, where he was also dean of master of science students in 2009 till 2016. His current research interests include quality engineering, modeling, and project management optimization with a focus on customer satisfaction and transportation system, productivity, and modeling. Dr. Shojaie is a fellow of the Iranian National Association of Industrial Engineering; the National Quality Science Association, Iran; and the Institution of Engineers (Iran). He is also the quality lead assessor in AENOR, Spain. He is a board member of the Iran Social Responsibility-National Standard Technical Committee, the System Society of Iran. He has published more than 60 journal articles and 20 conference articles and 4 books in the field of Industrial Engineering. He was the General Chair of the International Conference on WTO & pioneer organization in Iran (2006-2009) held in New Tehran. He was awarded as the Best Researcher of the University in 2012 and the Best Professor of the Faculty. He has the membership of professional institutions of EFQM, IMCA, IRCA, and technical board of ISIRI. Dr. Shojaie is the member of Professional Committee on providing the Iranian National Standard such as ISO 29000, ISO 26000, and ISO 10006.

Mehrdad Javadi received the BSc and MS degrees in Mechanical Engineering from Sharif University, Tehran, Iran, and Tehran University, Tehran, Iran, respectively. He received his PhD degree from Islamic Azad University, Science and Research Branch, Tehran, Iran. He is also a member of the mechatronics department. His research interests are mechanical-electrical interfaces and advanced manufacturing systems.