

Research Article

Order Batching in Warehouses by Minimizing Total Tardiness: A Hybrid Approach of Weighted Association Rule Mining and Genetic Algorithms

Amir Hossein Azadnia,¹ Shahrooz Taheri,² Pezhman Ghadimi,³ Muhamad Zameri Mat Saman,¹ and Kuan Yew Wong¹

¹ Department of Manufacturing and Industrial Engineering, Faculty of Mechanical Engineering, Universiti Teknologi Malaysia, Johor Bahru, 81310 UTM Skudai, Malaysia

² Department of Computer Science, Faculty of Computer Science, Universiti Teknologi Malaysia, Johor Bahru, 81310 UTM Skudai, Malaysia

³ Enterprise Research Centre, University of Limerick, Limerick, Ireland

Correspondence should be addressed to Amir Hossein Azadnia; azadnia.ie@gmail.com

Received 17 April 2013; Accepted 23 May 2013

Academic Editors: Y.-P. Huang and P. Melin

Copyright © 2013 Amir Hossein Azadnia et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

One of the cost-intensive issues in managing warehouses is the order picking problem which deals with the retrieval of items from their storage locations in order to meet customer requests. Many solution approaches have been proposed in order to minimize traveling distance in the process of order picking. However, in practice, customer orders have to be completed by certain due dates in order to avoid tardiness which is neglected in most of the related scientific papers. Consequently, we proposed a novel solution approach in order to minimize tardiness which consists of four phases. First of all, weighted association rule mining has been used to calculate associations between orders with respect to their due date. Next, a batching model based on binary integer programming has been formulated to maximize the associations between orders within each batch. Subsequently, the order picking phase will come up which used a Genetic Algorithm integrated with the Traveling Salesman Problem in order to identify the most suitable travel path. Finally, the Genetic Algorithm has been applied for sequencing the constructed batches in order to minimize tardiness. Illustrative examples and comparisons are presented to demonstrate the proficiency and solution quality of the proposed approach.

1. Introduction

Based on ELA/AT Kearney [1], about twenty percent of the logistic costs of the surveyed companies were incurred due to warehousing in 2003. A vital part of the logistics system of a company is involved with its warehouses. Optimization of operations within every facility must be considered as an important part of its policies in order to promptly deliver goods or services to its customers at the least cost [2]. Responsiveness is a critical success factor in a warehousing system. Product movement within a warehouse can be facilitated effectively by consolidating orders into batches which can be done before picking customer orders [3].

In order picking systems, certain due dates are assigned to customer orders which should not be violated. In order to avoid production delays, item retrieval from the warehouse needs to be done at its appropriate time. In these cases, the tardiness of customer orders should be involved with order batching instead of using the total processing time as a measure for the solution quality [4]. Henn et al. [5] defined the tardiness of a customer order as the positive value between the completion time of a customer order with its due date. The time that an order picker finishes her/his tour of gathering all required items and comes back to the starting point is called completion time. Obviously, one of the factors that can influence the completion time is

the processing time of the orders. According to the above explanation, reducing the processing time and travel time for constructing a batch is an important fact in reducing the costs and minimizing delays of the customer responses. On the other hand, not considering the order's due date in the process of order batching can cause a huge dissatisfaction in customer expectations regarding the efficient responsiveness of the company. For explanation purposes, assume that we have some orders with different due dates, considering just processing time or traveling time in the process of order batching regardless of their due dates may consolidate orders with different due dates in a same batch. So, this issue may cause a problem for meeting customers' demand in a certain due date. Consequently, it is highlighted that order due dates should be considered as an important factor for order batching.

Based on the above illuminated problem, this research paper attempts to develop a novel model to overcome the aforementioned issue in the multiple parallel aisles with manual order picking system. In our proposed model, it was assumed that orders with similar items are grouped together. Therefore, batching orders with higher resemblance can reduce the order pickers' traveled distance. Also, for considering the due dates in our model, the due date of each order is defined as a weight parameter in calculating the association between orders in terms of support. The association information integrates the bathing process with various data regarding each order. For calculating the association between orders in terms of support, mining association rules with weighted items (MINWAL) were utilized. Then, a clustering approach based on a binary integer programming model has been used in order to maximize the customer demand association which is called support. Next, the Genetic Algorithm (GA) has been utilized to solve the Traveling Salesman Problem (TSP) to minimize the travelling time for collecting all items in a constructed batch. This process was followed by a batch sequencing process which utilized a GA in order to minimize the average tardiness of all orders. This paper copes with integrating batch ordering, picker routing, and the batch sequencing problem regarding orders' due date in order to minimize the average tardiness of all orders. We believe that the proposed approach, namely, ATGH is not considered in any existing published paper. In the next section, a comprehensive literature survey is presented in order to review the existing research activities related to the current research paper.

2. Literature Review

2.1. Order Batching Problem. Order picking is an action which takes place to fulfill a customer demand (internal/external) that is normally done by retrieving items from buffer locations inside the warehouse [6–8]. In practice, there are two types of systems for order picking: systems which are totally manual and involved with human order pickers and the systems that are entirely computerized in case of retrieving items from the warehouse. Picker-to-parts and parts-to-picker systems can be categorized as the systems that belong to the first group of warehousing systems [5, 9].

Picker-to-parts systems are the most commonly used systems in warehousing in which picking the items is done by an operator that drives or walks along the aisles [10, 11]. This scenario is different in parts-to-picker systems. Order pickers do not move in this system and are located in the depot. Pallets or bins (unit load) from the warehouse are retrieved by automated storage and retrieval systems (AS/RS) and delivered to them at the depot. After that, the requested items are detached by the operators stationed in the depot and the pallets or bins are returned to its spot in the warehouse by the AS/RS [5].

In a large amount of orders, a single order picking policy might be applied in which one order can be picked in each picking tour. However, in a small amount of orders, picking loads in a single picking tour can lead to a decrease in travel times [12]. Order batching is classified as a NP-hard problem which can be done to improve warehouse efficiency by reducing operational costs [12, 13]. Thus, many heuristic algorithms are proposed in the literature which help to solve this problem. Savings and seed algorithms are the most commonly used order batching heuristics in the case of manual picking systems. Seed algorithms were introduced by [14] in which a seed order would be selected for each batch. With respect to the picking device capacity, other orders would be assigned to the seed orders. Clarke and Wright [15] first developed the savings algorithms inspired by a vehicle routing algorithm. This algorithm works based on the saving that would be achieved in travel distance or travel time. After that, a pair of batches would be selected iteratively by the algorithm and combined with each other until a capacity constraint threshold is reached that stops combining of the batches.

In the current literature review, it was tried to review all existing papers which are involved in order batching problems considering the minimization of due dates, total travel distance, and processing time. First, Vinod [16] suggested that integer programming can be used as a flexible method for grouping any kind of objects. Armstrong et al. [17] also presented an integer programming model based on predetermined batch sizes considering proximity batching.

They tried to minimize the total processing time of all batches. Ratliff and Rosenthal [18] tried to minimize travel distance based on a procedure which helps to find the best picking tour traveled. By means of a distinguished TSP, a tour was formed in a 50 aisles warehouse. Kusiak et al. [19] used a quadratic integer programming model to obtain batches from eight orders. In their proposed model, it was tried to minimize the total sum of distances.

One of their limitations was related to their model range of applicability which is not well suited for large numbers of orders. Gibson and Sharp [20] presented an order batching procedure integrated with a computer simulation which resulted in minimization of travel distance for each tour. Rosenwein [21] utilized Centre of Gravity (COG) and Minimum Additional Aisle (MAA) for order batching in order to measure travel distance. It was shown that the MAA metric performed better than the COG metric regarding travel tour generation. Gademann et al. [22] targeted to minimize the total travel time for all batches using a branch-and-bound

algorithm. They tested their model by several test sets with a maximum order number of 32 which were compared together regarding their CPU times. Hsu et al. [3] applied Genetic Algorithms (GAs) to deal with order batching problems with any kind of configuration. In their proposed approach, they endeavored to minimize the total travel distance. In another research activity done by Chen and Wu [23], data mining and integer programming were combined to form a solution approach regarding the order batching problem. They have tried to maximize the similarity between orders in a batch in terms of support without considering orders' due dates. Based on the presented results which was attempted to compare first-come first-served (FCFS), MAA, COG, and Gibson and Sharp's method (GSM), the authors stated that the proposed approach is effective in solving the order batching problems in terms of reducing traveling distances. Based on the analysis done by Gademann and van de Velde [24], the authors stated that the order batching problem is NP-hard to a great extent provided that no batch has more than two orders. They developed a branch-and-price algorithm to formulate the problem in order to minimize the total travel distance. Three years later, a mixed-integer programming model was developed by Bozer and Kile [25] in order to minimize the length of trips or travel distance. They stated that acquiring minimum travel distance for a large number of orders requires substantial computational time. Kulak et al. [26] proposed a solution approach to reduce the mathematical burden of order batching and picker routing problems. They proved the efficiency of their cluster-based tabu search approaches by running several test examples. They have focused on minimizing the travel distance between locations.

In practice, certain completion due dates are assigned to customer orders. For instance, in distribution warehouses, the scheduled departure of trucks has to be done by certain due dates to ensure the on time delivery of the requested items to the customers [22]. In material warehouses, avoiding production delays can be guaranteed, if retrieving the items from the warehouse is done based on their schedules. In these cases, assessing the tardiness of the customer orders needs to be focused in the process of batching customer orders into picking orders [4]. The tardiness is the difference between the order finish time and the due date, if this difference is positive. So, it is very important to consider the orders' due dates when batching processes are being done. To our knowledge, there are few papers that consider due dates or minimize tardiness in their model. Elsayed et al. [4] and Elsayed and Lee [27] focused on developing minimization models of order tardiness and their incurred costs. The solution approaches proposed by these two research papers are of limited applicability to manual picker-to-parts systems. Single processing times are used in both solution approaches in order to determine a sequence of batches. Therefore, they will not yield competitive results in the situation described in our paper. Won and Olafsson [28] endeavored to integrate order batching and picker routing problems together in order to minimize the travel time. They compared their proposed heuristics with other existing algorithms such as FCFS. It can be perceived that their simulation results improvement is

reduced when the numbers of items for each order increase. Tsai et al. [29] attempted to solve a batch picking model considering earliness and tardiness penalty using a multiple-GA method for obtaining the best possible batch picking plans. They tried to develop a flexible model to cope with the current dynamic environment in terms of responsiveness. Upon studying their proposed method, it was determined that the orders were split which causes items in an order to be collected in different tours. Also, their proposed model allocates penalty to batches overweight (when the batch volume exceeds the picker capacity) which is not applicable in practice.

2.2. Picker Routing Problem. The picker routing problem is defined as the process of identifying the minimum distance which would be traveled by the order picker in a warehouse upon identifying which order should be picked first [30, 31]. There are some existing heuristics for picker routing such as return, s-shape, largest gap, and combined routing. De Koster et al. [32] tried to solve order batching problems by evaluating two groups of heuristic methods such as seed algorithms and time savings algorithms by means of s-shape and largest gap strategies as two different routing strategies. According to the obtained results, integrating the s-shape strategy and an oversized capacity pick device together with seed algorithms can yield better performance. In contrast, the largest gap strategy using a small capacity pick device gives best performance when it integrated with time savings algorithms. Ho and Tseng [33] considered the largest gap routing strategy together with a simulated annealing optimization method not only to optimize the total travel distance but also to find alternative routes that are better than ones found by just the largest gap strategy. In a research activity done by Theys et al. [34], s-shape, largest gap and some other routing heuristics were compared with the Lin-Kernighan-Helsgaun TSP heuristic. Based on their results, this TSP heuristic helps to reduce travel distance up to the rate of 48% on average. In the current research activity, a GA_{TSP} model has been used in order to solve the picker routing problem.

2.3. Batch Sequencing Problem. The problem of batch sequencing can be defined as finding the orders of constructed batches to be processed further. Henn and Schmid [35] used metaheuristics to solve their proposed model of order batching and sequencing to minimize the orders' tardiness. Iterated Local Search and Attribute-Based Hill Climber are the two employed metaheuristics in their model. According to different test problems, the efficiency of their proposed methods was illustrated. Provided solutions can be improved by 46% on average in comparison with the ones obtained by standard constructive heuristics such as an application of the Earliest Due Date rule. They conducted their proposed method for a maximum of 80 items.

Other sections of this paper are as follows. First, the proposed model is described in Section 3 which covers all the aspects that are involved with it. After that, Section 4 presents the research methodology of this paper. This is followed by Section 5 which encompasses the numerical examples, results, and discussion of this research activity.

Finally, Section 6 gives an end to the paper with a brief conclusion.

3. Model Description

The assumptions of this research are described as follows.

- (1) A manual picker to part system in a parallel aisle warehouse is considered in this research.
- (2) There is only one location for storing each type of item and vice versa.
- (3) Multiple picker devices are not allowed. Therefore the maximum number of pickers is only one.
- (4) The volume for each order should not exceed the capacity of the picker.
- (5) Batch size does not exceed the capacity of picker.
- (6) The storage size in each location is identical.
- (7) All customer orders are known in advance.

3.1. Weighted Association Rule Mining Binary Integer Clustering Model. In this section, an association based clustering model developed by Chen and Wu [23] was modified in order to create the batch structure regarding orders' due date. First, the association between customers' demand will be calculated based on the similarity of items in orders with respect to their due date using MINWAL. Orders' due dates are considered as the weight of association rule mining. Although batching orders with higher associations generated by similarity between their items can decrease the distance travelled by order pickers, not considering the due dates would cause the orders to be batched inappropriately with different due dates. Some problems may occur by the aforementioned issue such as shortage or excess inventory costs. In order to solve this problem, the order due date has been considered as a weight of each order for calculating the support between orders. Relationship maximization of customer orders based on similar items within each order and due dates of each order can be done with the weighted association rules. Travel distance would be reduced by taking into account batching of orders with higher relationships. Other than that, higher associations can batch orders with similar due dates in order to minimize tardiness. This is followed by utilizing a binary integer programming model integrated with a clustering algorithm which assembles the orders into their respective batches.

3.1.1. Weighted Association Mining for Determining Customer Order Relationships. Association rule mining can be defined as a data mining method which tries to distinguish interrelations of variables where large databases exist [36]. Some rules are involved with the association rule model where there is an association between some set of items with another set of items [37]. In order to calculate rule interestingness, support and confidence can be considered as the two most appropriate measures in association rule mining. The support for an itemset is the percentage of transactions that contain the itemset

TABLE 1: Order-item data.

Order	Items	Due date
O_1	A(3) B(4)	10
O_2	A(2) C(3)	30
O_3	A(1) B(3)	5
O_4	A(2) B(4) C(2)	50

TABLE 2: Item-order data.

Item	Order
A	O_1, O_2, O_3, O_4
B	O_1, O_3, O_4
C	O_2, O_4

in the database. A probability shows how frequently the rule head occurs among all the groups containing the rule body which can be defined as the confidence of an association rule ($\text{body} \Rightarrow \text{head}$) [38, 39].

Based on the definition by Agrawal et al. [36], the association rule mining problem is addressed as follows: items are described as $I = \{i_1, i_2, \dots, i_n\}$ which is a set of n binary attributes. The database is represented as $D = \{t_1, t_2, \dots, t_m\}$ which is a set of transactions. An exclusive transaction ID belongs to each transaction in D together with a subset of the items in I . A rule is defined as an implication of the form $X \Rightarrow Y$, where $X, Y \subseteq I$, and $X \cap Y = \emptyset$. The association rule, $X \Rightarrow Y$ [support = $s\%$, confidence = $c\%$], holds in the item-order data D with confidence c if $c\%$ of Order Identifiers in D contain orderset X and also contains orderset Y . The rule $X \Rightarrow Y$ has support s if $s\%$ of the item-order D contains both X and Y . For instance, upon batching the orders by taking into account the customer order patterns, association rule mining involves finding out the amount of support between demands of customers from the order database. In order batching, Table 1 which shows the order-item data is transposed to the item-order data form shown in Table 2 since the order associations are demanded and the product item associations is not necessary. In this example, association rule mining can perform its job to define the associations between customer orders that can be perceived from information presented in Table 2.

In the area of order batching, association rule mining was first used by Chen and Wu [23]. In their research paper, association rule mining was used to calculate the correlations between customers' demands in terms of support. They pointed out that orders with similar items should be consolidated together in order to minimize the total travel distance traversed by the picker. An Apriori algorithm has been used in their proposed method to calculate the association between orders. It should be highlighted that they only considered the relationships between orders in terms of similarities between items. However, in batch processing, each order has a due date which can be considered as the weight but this weight is not considered by any other existing method in the literature.

The classical Apriori algorithm [40] extracts binary association rules based on the downward closure property which

TABLE 3: Weight of each order.

Order	Weight
O_1	0.5827
O_2	0.1979
O_3	0.7634
O_4	0.0672

proves that subsets of a frequent itemset are also frequent [41]. However, the Apriori algorithm cannot be applied because it cannot handle the weighted case. Therefore, in this research activity, the MINWAL algorithm is applied in order to extract the weighted supports among the pair of orders to fulfill the task of considering both the support and the weights factors.

In order to use MINWAL [42] for mining weighted association rules, the weight should be in the range of the 0 to 1 interval. Based on the aforementioned fact, larger weights should be assigned to orders with smaller due dates in a comparison with other orders that have lots of time to be delivered in order to minimize the average tardiness of all orders. Therefore, in this study, (1) has been used to calculate the appropriate weights for each order as following:

$$\text{Weight}(O_i) = e^{(-0.1 * \beta * d_{O_i} / \text{Maximum Due Date})}, \quad (1)$$

where O_i denotes an order and d_{O_i} refers to the due date of order O_i . β is a constant value that is used to normalize the due dates which is considered as 30 (30 minute) in this study. Also, the maximum due date of the orders was used in order to provide an appropriate interval for the weights. For example, the weight of each order is tabulated in Table 3 which is calculated by (1).

Given a set of orders $O = \{o_1, o_2, \dots, o_m\}$, a weight w_j for each order o_j , within the values of 0 to 1 is assigned by (1) via its due date, where $j = \{1, 2, \dots, m\}$ shows the importance of it. The weighted support for the weighted association orders of O_1 and O_2 as $O_1 \Rightarrow O_2$ is defined, if the weighted support of such a pair of orders is no less than the minimum weighted support (wminsup) threshold. In this study, the minimum support threshold is considered as 0 in order to identify all the associations among orders. Equation (2) shows the support of an association for any pair of orders as following:

$$\left(\sum_{w_j \in (O_1 \cup O_2)} w_j \right) * \text{Support}(O_1 \cup O_2). \quad (2)$$

Algorithm 1 shows the MINWAL steps in order to mine weighted association rules. This algorithm consists of 6 subroutines such as, Search, Counting, Join, Prune, Checking and Rules. The search subroutine provides the size of the largest itemsets in database. Counting is responsible for counting every 1-itemset inside the database. The Join step is responsible for generating the k -itemsets (C_k) from $(k - 1)$ -itemsets (C_{k-1}). The Prune step removes inappropriate itemsets which do not exist. The Checking step updates the count of k -itemsets inside the transaction and prunes itemsets which do not meet the minimum support threshold. In the Rules step, every association rule will be extracted. Table 4 shows the notations of Algorithm 1.

```

(1) Main Algorithm (wminsup, mincon  $f$ ,  $D$ ,  $w$ )
(2)   size = Search ( $D$ )
(3)    $L = 0$ ;
(4)   for ( $i = 1; i \leq \text{size}; i++$ )
(5)      $C_i = L_i = 0$ ;
(6)   for each transaction do
(7)     ( $SC, C_1, \text{size}$ ) = Counting ( $D, w$ );
(8)      $k = 1$ ;
(9)     while ( $|C_k| \geq k$ )
(10)        $k++$ ;
(11)        $C_k = \text{Join} (C_{k-1})$ ;
(12)        $C_k = \text{Prune} (C_k)$ ;
(13)       ( $C_k, L_k$ ) = Checking ( $C_k, D$ );
(14)    $L = L \cup L_k$ ;
(15)   Rules ( $SC, L$ );
(16) ends;

```

ALGORITHM 1: MINWAL algorithm [42].

TABLE 4: Notations of Algorithm 1.

D	The database
w	The set of item weights
L_k	Set of large k -itemsets
C_k	Set of k -itemsets which may be k -subsets of large j -itemsets for $j \geq k$
$SC(X)$	No. of transactions containing itemset X
wminsup	Weighted support threshold
minconf	Confidence threshold
Size	Maximum possible large weighted itemsets

3.1.2. *Association Based Binary Integer Programming for Orders Batching.* In order to create the batch structure, a binary integer clustering model which has been developed by Chen and Wu [23] was applied. They developed a binary integer programming model for maximizing similarity between orders in terms of support in order to minimize the total traveling distance. In this research activity, support between orders was calculated based on weighted association mining using MINWAL which was described in Section 3.1.1. The binary integer model for order batching is described as follows.

Parameters

S_{ij} Support between order i and j determined by MINWAL

C_v Capacity of picking vehicle

V_i Weight of order i

K Number of batches

X_{ij} $\begin{cases} 1 & \text{if order } i \text{ is assigned to batch } j; \\ 0 & \text{otherwise;} \end{cases}$

$$Y_i \begin{cases} 1 & \text{if order } j \text{ is chosen as a batch median;} \\ 0 & \text{otherwise;} \end{cases}$$

$$\text{Maximize} \quad \sum_{i=1}^N \sum_{j=1}^N S_{ij} * X_{ij} \quad (3)$$

$$\text{Subject to} \quad \sum_{j=1}^N X_{ij} = 1 \quad \text{for } i, j = 1, 2, \dots, N \quad (4)$$

$$X_{ij} \leq Y_j \quad \text{for } i, j = 1, 2, \dots, N \quad (5)$$

$$\sum_{j=1}^N Y_j = K \quad (6)$$

$$\sum_{i=1}^N V_i X_{ij} \leq C_v \quad \text{for } j = 1, 2, \dots, N \quad (7)$$

$$X_{ij} = 0, 1 \quad \text{for } i, j = 1, 2, \dots, N \quad (8)$$

$$Y_j = 0, 1 \quad \text{for } j = 1, 2, \dots, N. \quad (9)$$

In the current model, the objective function (3) maximizes the sum of support from all orders to their relevant batch medians as K orders are selected as batch medians. This model will maximize the support between orders in the batches. Membership of each order in just one batch would be guaranteed by constraint (4). The number of batches with K is limited using constraints (5) and (6). The total quantity in a tour is also represented as a limitation in constraint (7). It cannot exceed the capacity of order picker C_v . For the order batching model to have a binary solution would be ensured by constraints (8) and (9). For calculating the initial numbers of batches (10) will be used where $\sum V_i$ represents total volume of all orders as following:

$$K_{\text{initial}} = \left\lceil \frac{\sum V_i}{C_v} \right\rceil. \quad (10)$$

3.2. Picker Routing

3.2.1. Genetic Algorithms. In this research, GAs which were introduced by Holland [43] have been used in order to solve the picker routing and batch sequencing problem. The biological evolution process is used as a base fact in a GA for problem solving to a great extent. The main steps of a GA can be described as follows [44, 45]:

- (i) populating a series of solutions,
- (ii) solutions assessment,
- (iii) picking the most appropriate solutions,
- (iv) generating new solutions based on some genetic operation.

In step (iii), selection forms the main part of a GA. The bad solutions are removed by selection and the good ones remain. Random generation over the feasible or infeasible

solution space of a problem is used as a basis for creating the population. Each solution is evaluated using the fitness function. Selection would be done according to certain selections of fitness together with the probability that is assigned to it. Generating a fresh population and a better solution are involved with recombination of the individual solution. Crossover and mutation are the most used GA operators for generating new offspring from the old generation of parents. Parameters of the GA are originated from the probabilities of mutation and crossover. Exploration of the sample space is diversified by these operators. In this research, the GA has been utilized in two steps. First, the GA has been used in order to solve the picker routing problem. Consider that in the warehouse a set of items should be picked by a picker in one tour. Therefore, the picker routing problem in the warehouse could be categorized as a TSP. The goal in the picker routing problem as a TSP in the warehouse is to find an optimal path for minimizing the travel distance traversed by the picking machine where each node is visited at least once. In this research, for a given number of itemset in a batch, GA_{TSP} has been used to find the minimum travel distance which is traversed by the picking machine. Second, a GA has been utilized in order to sequence all batches, namely, $\text{GA}_{\text{sequencing}}$ which would be further processed. The batches are sequenced based on a fitness function which minimizes the tardiness of all orders. The detailed procedures of GA_{TSP} and $\text{GA}_{\text{sequencing}}$ are illustrated in Sections 3.2.3 and 3.2.4.

3.2.2. Warehouse Layout. The warehouse discussed throughout this research is assumed to have a 3D multi-parallel-aisle warehouse layout. The picker will start the tour from the depot which is located in front of the leftmost aisle and returns back to it. A picker leaves the depot. After that, it travels to a certain buffer spot where it picks the requested order(s) and puts the order(s) into a certain picker. Then, it continues on its way to another spot. These procedures would be repeated again and again so that all the requested orders are picked. Then, it is time for the picker to return to the depot.

The graphic layout of the warehouse zone involved in this research activity is pictorially displayed in Figure 1. It shows the x and y dimensions of the 3D warehouse layout. Each item in the warehouse has a specific location. For example, the location of item i can be addressed by $L_i = (x_i, y_i, z_i)$. Consider two locations as $L_a = (x_a, y_a, z_a)$ and $L_b = (x_b, y_b, z_b)$. If two locations are located in the same sub-aisles of the same block, the distance between these two items locations, $(D_{a,b})$, will be calculated using

$$D_{a,b} = |x_a - x_b| + |y_a - y_b| + |z_a - z_b|. \quad (11)$$

Equation (11) is not applicable if the two storage locations belong to different subaisles that are from the same block [34]. In order to solve this problem, distances of all possible paths between two locations have been calculated and the shortest one was considered.

3.2.3. GA_{TSP} for the Picker Routing Problem. In order to determine the optimal route of a picker related to a given

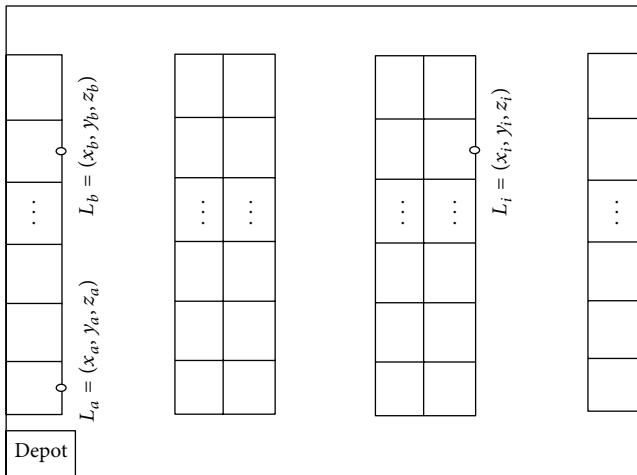


FIGURE 1: Warehouse layout.

1	2	3	4	5	6	7	8	9
L_0	L_1	L_8	L_{11}	L_9	L_{13}	L_{12}	L_{10}	L_0

FIGURE 2: A feasible chromosome encoding in GA_{TSP}.

set of items in a batch, a GA has been used to solve the TSP problem to minimize travelling time. In a GA, each solution is considered as a chromosome. The steps of GA_{TSP} are described as follows:

- (1) for each batch establishing the structure of a chromosomes,
- (2) generate initial feasible chromosomes,
- (3) evaluate each chromosome based on the GA_{TSP} fitness function,
- (4) apply the Selection operation,
- (5) apply the Crossover operation,
- (6) apply the Mutation operation,
- (7) evaluate each offspring based on the fitness function,
- (8) go to step 4, if the termination criterion is not satisfied, otherwise terminate and return the best solution as the optimal path.

At the first step of GA_{TSP}, the structure of the chromosome should be established. In the GA_{TSP}, the location of an item that should be visited is presented by the value of a gene. Accordingly, the sequence of visiting the item location is denoted by the order of a gene in a chromosome. Therefore, the total number of items locations in a specific batch is considered equal to the length of a chromosome. For example, if the length of a chromosome is nine it means nine places should be visited by the picker as shown in Figure 2. If L_i represents the item location in the warehouse, $i = \{1, \dots, n\}$, it shows that the picker leaves the depot (L_0 denotes the location of a depot), goes to $L_1, L_8, L_{11}, L_9, L_{13}, L_{12}, L_{10}$, and then comes back to the depot.

In the second step, the initial number of feasible chromosomes should be generated randomly for each batch.

The number of initial chromosomes could be different based on the complexity of the problem. It should be considered that the first and the last gene of each chromosome should be assigned individually to the location of a depot. In step three, each chromosome will be evaluated based on the GA_{TSP} fitness function which is formulated as follows:

$$\text{fitness} = \text{Min} \sum_{j=1}^{n-1} D_{j,j+1}, \quad (12)$$

where $D_{j,j+1}$ denotes the distance between two subsequent genes in a chromosome, $j = 1, \dots, n - 1$ represents the order of the genes, and n is number of genes.

Subsequently, step four is involved with the selection operation. In each iteration, the selection operator uses fitness values to select the parents of the next generation. In this research, the roulette wheel selection policy has been used to guarantee that the most appropriate pairs of chromosomes have been selected to generate offspring. It means that a higher probability of being selected to create a new population will be assigned to the chromosomes with higher fitness values.

Steps five and six are involved with conducting the crossover and mutation operation. The construction of the offspring is realized by the crossover and the diversity of the individuals is maintained by the mutation operator. Blocks of genes between chromosomes are traded by means of the crossover operation in which the exploitation of a particularly profitable portion of the parameter space is allowed to the block. Here, permutation order based crossover (POP) as a variation of the distinguished order crossover has been used. As shown in Figure 3, the original idea is about choosing two parents and a point for cutting. The first portion of offspring 2 contains the first portion of parent 1 up to the cut point. To form the second portion of offspring 2, consider all genes in parent 2 except those genes that already exist in the first portion of the offspring 2.

As shown in Figure 4, two positions are randomly selected in a chromosome and exchanged with each other; namely, the order based mutation approach, SWAP, is adopted for solving GA_{TSP}. Step six is followed by step seven which includes the evaluation of each offspring that has been generated in the previous step based on the GA_{TSP} fitness function. In step eight, termination criteria will be checked. If the termination condition is met, then the process will be terminated. After that, the best chromosome will be selected as the optimal solution. Otherwise, the selection process is to be conducted again to generate a new population.

3.2.4. GA for Sequencing the Batches. Sequencing the batches should be conducted in order to minimize the average tardiness of all orders. The problem of batch sequencing can be defined as finding the orders of batches which should be processed further. For this purpose, the GA_{sequencing} method has been utilized. In this research, all of the batches which were constructed using binary integer programming and GA_{TSP} will be sequenced by using the GA. The population generation, selection, mutation, and stopping threshold in the GA_{sequencing} are identical to those in the GA_{TSP}. However,

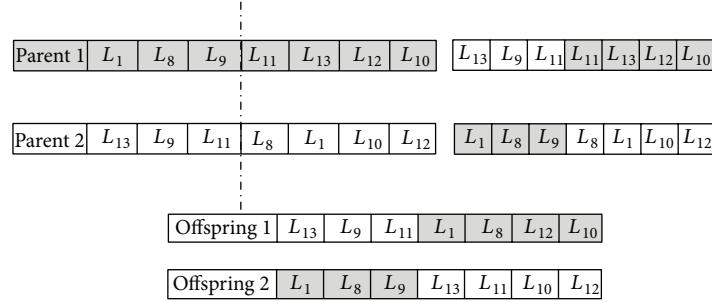


FIGURE 3: Crossover operation.

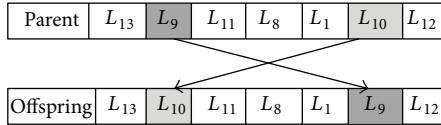


FIGURE 4: Mutation operation.

1	2	3	4	5
b_1	b_4	b_3	b_5	b_2

FIGURE 5: Chromosome structure.

the fitness function and chromosome structure are different than that of GA_{TSP}. In contrast with GA_{TSP} chromosomes structure, batch number is presented by the value of a gene. Accordingly, the sequence of batch is denoted by the order of a gene in a chromosome. For instance, Figure 5 shows the chromosome structure. If $b_i \in B$ denotes a set of batches, Figure 5 shows that b_1, b_4, b_3, b_5, b_2 should be processed, respectively. Therefore, the length of a chromosome is equal to the number of the batches. The GA_{sequencing} fitness function for chromosome y is described in

$$\text{fitness}_y = \frac{1}{N} \sum_{k=1}^n \sum_{i \in b_k} CT(b_k) - Du(o_i), \quad (13)$$

$$CT(b_k) = CT(b_{k-1}) + P_t(b_k), \quad (14)$$

$$P_t(b_k) = \frac{\text{Opt}(b_k)}{VS}, \quad (15)$$

where N denotes for the total number of orders. $k = \{1, \dots, n\}$ represents the batch sequence in which n is the total number of batches. o_i stands for the order i in which $i \in b_k$. b_k denotes the batch in k th position of a sequence for specific chromosome. $CT(b_k)$ stands for the completion time of a batch in the k th position of a sequence. $Du(o_i)$ is the symbol for due date of order i which belongs to b_k . $P_t(b_k)$ is the process time of b_k . $\text{Opt}(b_k)$ is the optimal path solution for b_k . Finally, VS is the moving speed of the picker.

4. Research Methodology

In this research, the proposed methodology consists of five steps. These steps are described as follows and shown pictorially in Figure 6.

Step 1. It is involved with utilizing the MINWAL algorithm in order to determine the associations between orders in terms of support considering their due dates. In this step, an order due date will be considered as the weight of an order. It means orders that are similar in items and due date should have more support for being batched together.

Step 2. It encompasses determining the initial number of batches. In this step, the initial number of batches will be calculated by (10).

Step 3. It includes the problem of modeling and solving; first, the binary integer programming model for order clustering and batching will be modeled. Next, this model will be solved in order to determine the structures of batches.

Step 4. It is about the picker routing problem. The optimal path for a picker within each batch will be determined by solving a TSP using GA.

Step 5. It is about sequencing the batches based on the fitness function which minimizes the average tardiness of all orders. In this step, GA algorithm will be utilized in order to solve the batch sequencing problem regarding the fitness function.

5. Computational Experiments and Results

5.1. Test Problems and Parameter Setting. In this section, five numerical test problems with different settings and parameters have been carried out in order to illustrate the proficiency of the proposed method with respect to average tardiness of all orders. Table 5 summarizes the description of these test problems. This table includes information with regard to number of items, number of orders, capacity of the picker, total weight of all the items in an order, and minimum number of batches.

In these test problems, the quantity of items, quantity of orders, capacity of the picker device (C_v), and amount of each item are predetermined. Each order contains a certain

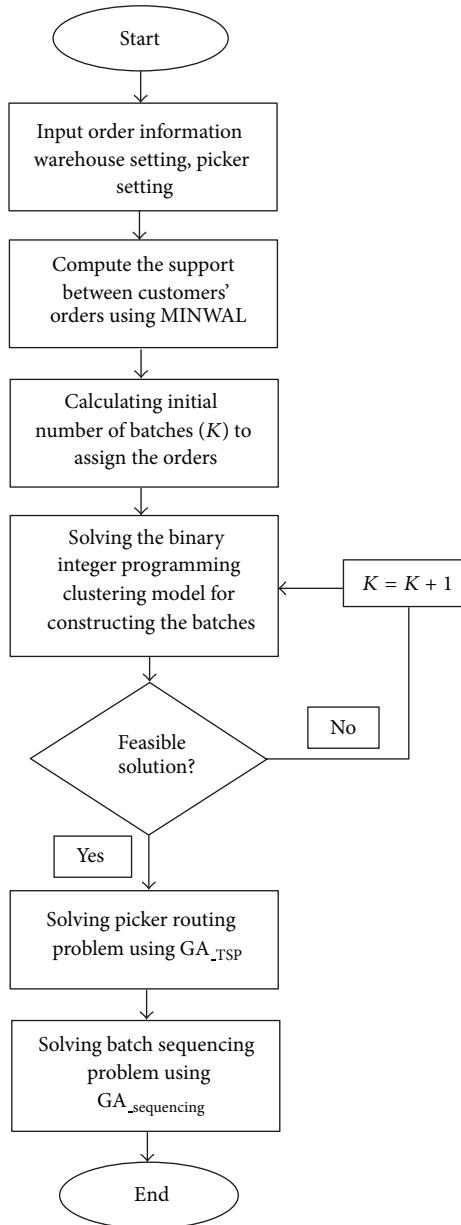


FIGURE 6: Methodology framework.

quantity of items and also each item has a certain quantity. These quantities are produced randomly. The distribution function type for the number of items in each order is Normal [5, 10]. Also, the distribution function type for each item quantity in an order is uniform [1, 10]. The picker moving speed is set as $VS = 2$ (m/s). The warehouse has 3000 item location. The GA control parameters for solving the TSP and sequencing problem are tabulated in Table 6.

All of the proposed approaches are coded using MATLAB version 7.9.0 and Java 1.5 programming language. Also, the IBM ILOG CPLEX optimization studio version 12.4 software was utilized jointly with MATLAB in order to facilitate the process of solving the binary integer programming problem. Examples are run by means of a computer featured by

4 Gigabytes random access memory (RAM) and a 2.20 Gigahertz Intel processor (T6600).

5.2. Computational Results and Discussion. In order to reveal the competency of the proposed solution approach, it was compared with the other existing methods in the literature. One of the latest published research activities in the literature which deals with minimization of tardiness has been conducted by Henn and schmid [35]. They utilized the Earliest Due Date (EDD) approach as their constructive algorithm for comparing with their proposed approaches. They proposed two heuristics. Iterated Local Search and Attribute-Based Hill Climber are the two employed metaheuristics in their model. According to different test problems, the efficiency of their

TABLE 5: Test problems description.

	Example 1	Example 2	Example 3	Example 4	Example 5
Number of orders	40	60	80	100	150
Number of items	80	120	160	200	200
Capacity of picker	200	200	300	400	550
Total weight	2212	3668	4393	6110	8369
Minimum number of batch	11	19	15	16	16
Due dates range	60	90	120	180	240

TABLE 6: GA control parameters.

	GA _{TSP}	GA _{sequencing}
Population size	500	200
Cross over strategy	Permutation order based crossover (POP)	Permutation order based crossover (POP)
Cross over probability	0.8	0.8
Mutation strategy	SWAP	SWAP
Mutation probability	0.2	0.2
Iteration number	10000	10000
Parent selection method	Roulette wheel	Roulette wheel

proposed methods was illustrated. Their obtained results from their proposed approaches showed a 46% improvement on tardiness compared to EDD. They used an s-shape and largest gap routing strategy for their proposed method. In order to make a reasonable comparison with their proposed method, EDD was integrated with the s-shape routing strategy and GA_{sequencing}. Then, based on improvement on EDD, we can compare our proposed method with their proposed method. In the EDD approach, due dates play an important role in sorting all orders. Based on an ascending sequence, assigning the orders to batches ensures that no capacity violation can happen for the picking device.

For the warehouse condition which is described in Section 3.2.3, our proposed approach, EDD integrated with GA_{TSP} and GA_{sequencing} and EDD integrated with s-shape algorithm and GA_{sequencing} have been evaluated in terms of average tardiness for all orders (T) which is calculated based on (13), number of batches (K), total process time (P_t), and average process time for each batch (A_{P_t}). All of the GA setting for the TSP and sequencing problems are considered the same for all methods. Also, the settings which have been mentioned in Section 5.1 were considered for all of the compared methods.

The results for all of the test problems for the three approaches are tabulated in Tables 7, 8, and 9. As it is illustrated in Table 10, EDD integrated with GA_{TSP} and GA_{sequencing} has a 31.76% improvement on average in terms of average tardiness of all orders than EDD integrated with the s-shape algorithm and GA_{sequencing}. It can be perceived that GA_{TSP} performs better than s-shape. Also, the ATGH method has a 68.11% improvement in average tardiness off all orders than EDD integrated with s-shape algorithm

and GA_{sequencing}. Consequently, the ATGH method achieves much improvement against Henn and Schmid's [35] proposed method. According to Tables 7 and 8, it is also worth to mention that GA_{TSP} performs better than the s-shape routing strategy in terms of process time. The improvements which have been achieved in this research could be justified based on the following reasons.

The problem of order batching deals with several significant drivers in which the number of batches can be considered as a key factor [23]. As a justification to this matter, it is worth to mention that total travel distance may not decrease directly due to a decline in the number of batches. But a decline in the number of batches will increase batch sizes which may lead to an increase in the distance travelled by order picker. However, an advanced exploitation of picker capacity to load efficiently may reduce the number of constructed batches together reducing the total travel distance. Improving the total travel distance traversed by the picker can minimize the average tardiness of all orders. In terms of number of the batches according to Tables 7, 8, and 9, it can be perceived that our proposed method assigns the orders to the batches optimally. The ATGH method assigns the orders to batches based on their item similarity and their due dates and it uses the picker capacity efficiently. But the EDD method allocates the orders to batches based on an ascending sequence of their due date and it should be ensured that no capacity violation will happen for the picking device which may lead to an inefficient order batching. To highlight the problem, an example can be made based on the results tabulated in Tables 7 and 8. The number of batches for example 1 in our method is 11 in comparison with 13 obtained from EDD method. 13 batches can be justified based on this scenario that two subsequent orders, say, the first order is 100 kg and the other is 150 kg, cannot be batched together due to violated batch capacity which is set to be 200 kg. The number of batches would increase due to the aforementioned issue which affects the total process time to have an increasing manner. This situation will rarely happen in the ATGH method because of the weighted association based binary integer programming utilized which batches the orders in order to maximize the association between them which is calculated based on the item similarity of orders and the weight of their corresponding due dates. The ATGH method uses the picker capacity efficiently.

Obviously, one of the factors that can influence the completion time and the average tardiness of all orders is the processing time of the orders. Reducing the processing time

TABLE 7: Computational results of ATGH.

	Example 1	Example 2	Example 3	Example 4	Example 5
T	5.6204	29.948	9.0202	1.63	5.8898
P_t	60.5833	121.7833	105.833	121.033	133.9
K	11	19	15	16	16
A_{P_t}	5.5076	6.409	7.055	7.564	8.368

TABLE 8: Computational results of EDD integrated with GA_{TSP} and GA_{sequencing}.

	Example 1	Example 2	Example 3	Example 4	Example 5
T	12.4550	42.976	18.222	5.981	15.47
P_t	66.7333	133.516	116.083	131.85	141.916
K	13	21	16	17	16
A_{P_t}	5.1333	6.357	7.255	7.775	8.869

TABLE 9: Computational results of EDD integrated with s-shape algorithm and GA_{sequencing}.

	Example 1	Example 2	Example 3	Example 4	Example 5
T	16.51	62.677	28.25	13.365	17.593
P_t	75.266	166.36	133.23	143.33	149.26
K	13	21	16	17	16
A_{P_t}	5.789	8.92	8.33	8.431	9.32

TABLE 10: Comparison results.

	EDD + s-shape + GA _{sequencing}	EDD + GA _{TSP} + GA _{sequencing}		ATGH	
		Tardiness (min)	Improvement (%)	Tardiness (min)	Improvement (%)
(1)	16.51	12.4550	24.56%	5.6204	65.96%
(2)	62.677	42.976	31.43%	29.948	52.22%
(3)	28.25	18.222	35.50%	9.0202	68.07%
(4)	13.365	5.981	55.25%	1.627	87.83%
(5)	17.593	15.47	12.06%	5.8898	66.52%
Average improvement			31.76%		68.11%

and travel time for constructing a batch is an outstanding fact in reducing the costs and minimizing delays of the customer responses. In our proposed model, orders with similar items are consolidated together. Therefore, batching orders with more similarity can reduce the distance traversed by the picker facility. Also, the due date of each order is defined as a weight parameter in calculating the association between orders in terms of support which was done for considering the due dates in our model. Thereupon, the model considers due date and the process time in an integrated manner which simultaneously affected the average tardiness of all orders. This spoken issue is different in the EDD approach in which the orders' due dates are just considered. Example 5 can be considered as evidence of this matter in which the total and average process time for ATGH is lower than EDD integrated with GA_{TSP} and GA_{sequencing} with respect to the fact that number of constructed batches for both approaches are happened to be equal after obtaining the results.

6. Conclusion

One of the most important concerns of warehouse managers is involved with finding the best optimal and cost efficient way to pick orders placed by customers in order to be known as a responsible company with satisfied customers in terms of on time delivery. In the current research activity, order batching, picker routing, and batch sequencing problems of warehouse processes were addressed which are to be solved jointly as they are mixed up with many manufacturing industries. Mathematical solutions of these NP-hard problems might be available for problems that are small in nature. But, these kinds of solutions may not be applicable to be considered by warehouse managers. Hence, we presented novel solution approach, namely, ATGH, to solve this issue in accordance with minimizing the average tardiness of all customers' orders which can be considered as a major objective of this research activity. A weighted association rule mining

algorithm, namely, MINWAL, was utilized to determine the customers' orders association regarding similarity between items and orders' due dates. Actually, in our proposed method the due date of each order is defined as a weight parameter in calculating the association between orders. So, orders with similar items and due dates have more chance to be in a same batch. This issue is not considered in most of the research activities in the literature related to our work and considering this issue in the order batching problem is the main contribution of our research activity. The previous step was followed by a binary integer clustering model in order to solve the order batching problem which maximizes the similarity between orders in terms of support. After that, a GA was applied to solve a TSP in order to find the optimal routing path.

Finally, a batch sequencing problem was addressed in conjunction with a GA.

The other research objective of this research addresses whether the proposed ATGH algorithm can further improve the solution quality for a multiple parallel aisle warehouse with manual picking system. Hence, comparisons were made considering ATGH, EDD integrated GA_{TSP} and GA_{sequencing}, and EDD integrated s-shape and GA_{sequencing} algorithms. As demonstrated by the obtained results, ATGH comes out to be the most striking solution approach in terms of tardiness. In spite of the fact that considering all customer orders, as discussed in this paper, would lead to have a static case, being involved with dynamic cases in which different points of time are assigned to customer's orders arrival would be beneficial. So, there might be some potential room of research to extend our discussed approach with respect to dynamic orders.

Conflict of Interests

The IBM ILOG CPLEX optimization studio version 12.4 software is not used for financial gain and there is no conflict of interests with any party as it is only a student version.

References

- [1] ELA/AT Kearney, *Excellence in Logistics*, ELA, Brussels, Belgium, 2004.
- [2] Y.-C. Ho, T.-S. Su, and Z.-B. Shi, "Order-batching methods for an order-picking warehouse with two cross aisles," *Computers & Industrial Engineering*, vol. 55, no. 2, pp. 321–347, 2008.
- [3] C.-M. Hsu, K.-Y. Chen, and M.-C. Chen, "Batching orders in warehouses by minimizing travel distance with genetic algorithms," *Computers in Industry*, vol. 56, no. 2, pp. 169–178, 2005.
- [4] E. A. Elsayed, M.-K. Lee, S. Kim, and E. Scherer, "Sequencing and batching procedures for minimizing earliness and tardiness penalty of order retrievals," *The International Journal of Production Research*, vol. 31, no. 3, pp. 727–738, 1993.
- [5] S. Henn, S. Koch, and W. Gerhard, *Order Batching in Order Picking Warehouses: A Survey of Solution Approaches*, Springer, London, UK, 2012.
- [6] J. Drury, *Towards More Efficient Order Picking*, IMM Monograph no. 1, The Institute of Materials Management, Cranfield, UK, 1988.
- [7] M. Goetschalckx and J. Ashayeri, "Classification and design of order picking systems," *Logistics World*, pp. 99–106, 1989.
- [8] J. A. Tompkins, J. A. White, Y. A. Bozer, E. H. Frazelle, and J. M. A. Tancho, *Facilities Planning*, John Wiley & Sons, Hoboken, NJ, USA, 2003.
- [9] S. Henn and G. Wäscher, "Tabu search heuristics for the order batching problem in manual order picking systems," *European Journal of Operational Research*, vol. 222, no. 3, pp. 484–494, 2012.
- [10] S. Henn, "Algorithms for on-line order batching in an order picking warehouse," *Computers and Operations Research*, vol. 39, no. 11, pp. 2549–2563, 2012.
- [11] R. de Koster, "How to assess a warehouse operation in a single tour," Tech. Rep., RSM Erasmus University, Rotterdam, The Netherlands, 2004.
- [12] R. de Koster, T. Le-Duc, and K. J. Roodbergen, "Design and control of warehouse order picking: a literature review," *European Journal of Operational Research*, vol. 182, no. 2, pp. 481–501, 2007.
- [13] R. A. Ruben and F. R. Jacobs, "Batch construction heuristics and storage assignment strategies for walk/ride and pick systems," *Management Science*, vol. 45, no. 4, pp. 575–596, 1999.
- [14] E. A. Elsayed, "Algorithms for optimal material handling in automatic warehousing systems," *International Journal of Production Research*, vol. 19, no. 5, pp. 525–535, 1981.
- [15] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations Research*, vol. 12, no. 4, pp. 568–581, 1964.
- [16] H. D. Vinod, "Integer programming and the theory of grouping," *Journal of the American Statistical Association*, vol. 64, no. 326, pp. 506–519, 1969.
- [17] R. D. Armstrong, W. D. Cook, and A. L. Saipe, "Optimal batching in a semi-automated order picking system," *Journal of the Operational Research Society*, vol. 30, no. 8, pp. 711–720, 1979.
- [18] H. D. Ratliff and A. S. Rosenthal, "Order-picking in a rectangular warehouse: a solvable case of the traveling salesman problem," *Operations Research*, vol. 31, no. 3, pp. 507–521, 1983.
- [19] A. Kusiak, A. Vannelli, and R. K. Kumar, "Clustering analysis: models and algorithms," *Cybernetics*, vol. 15, pp. 139–154, 1986.
- [20] D. R. Gibson and G. P. Sharp, "Order batching procedures," *European Journal of Operational Research*, vol. 58, no. 1, pp. 57–67, 1992.
- [21] M. B. Rosenwein, "A comparison of heuristics for the problem of batching orders for warehouse selection," *International Journal of Production Research*, vol. 34, no. 3, pp. 657–664, 1996.
- [22] A. J. R. M. Gademann, J. P. van den Berg, and H. H. van der Hoff, "An order batching algorithm for wave picking in a parallel-aisle warehouse," *IIE Transactions*, vol. 33, no. 5, pp. 385–398, 2001.
- [23] M.-C. Chen and H.-P. Wu, "An association-based clustering approach to order batching considering customer demand patterns," *Omega*, vol. 33, no. 4, pp. 333–343, 2005.
- [24] N. Gademann and S. van de Velde, "Order batching to minimize total travel time in a parallel-aisle warehouse," *IIE Transactions*, vol. 37, no. 1, pp. 63–75, 2005.
- [25] Y. A. Bozer and J. W. Kile, "Order batching in walk-and-pick order picking systems," *International Journal of Production Research*, vol. 46, no. 7, pp. 1887–1909, 2008.
- [26] O. Kulak, Y. Sahin, and M. E. Taner, "Joint order batching and picker routing in single and multiple-cross-aisle warehouses using cluster-based tabu search algorithms," *Flexible Services and Manufacturing Journal*, vol. 24, no. 1, pp. 52–80, 2012.

- [27] E. A. Elsayed and M.-K. Lee, "Order processing in automated storage/retrieval systems with due dates," *IIE Transactions*, vol. 28, no. 7, pp. 567–577, 1996.
- [28] J. Won and S. Olafsson, "Joint order batching and order picking in warehouse operations," *International Journal of Production Research*, vol. 43, no. 7, pp. 1427–1442, 2005.
- [29] C.-Y. Tsai, J. J. H. Liou, and T.-M. Huang, "Using a multiple-GA method to solve the batch picking problem: considering travel distance and order due time," *International Journal of Production Research*, vol. 46, no. 22, pp. 6533–6555, 2008.
- [30] Y. Kao, M. H. Chen, and Y. T. Huang, "A hybrid algorithm based on ACO and PSO for capacitated vehicle routing problems," *Mathematical Problems in Engineering*, vol. 2012, Article ID 726564, 17 pages, 2012.
- [31] X. Gan, Y. Wang, S. Li, and B. Niu, "Vehicle routing problem with time windows and simultaneous delivery and pick-up service based on MCPSO," *Mathematical Problems in Engineering*, vol. 2012, Article ID 104279, 11 pages, 2012.
- [32] M. B. M. de Koster, E. S. van der Poort, and M. Wolters, "Efficient orderbatching methods in warehouses," *International Journal of Production Research*, vol. 37, no. 7, pp. 1479–1504, 1999.
- [33] Y.-C. Ho and Y.-Y. Tseng, "A study on order-batching methods of order-picking in a distribution centre with two cross-aisles," *International Journal of Production Research*, vol. 44, no. 17, pp. 3391–3417, 2006.
- [34] C. Theys, O. Bräysy, W. Dullaert, and B. Raa, "Using a TSP heuristic for routing order pickers in warehouses," *European Journal of Operational Research*, vol. 200, no. 3, pp. 755–763, 2010.
- [35] S. Henn and V. Schmid, *Metaheuristics for Order Batching and Sequencing in Manual Order Picking Systems*, Univ., Fak. für Wirtschaftswissenschaften, 2011.
- [36] M. Hahsler, B. Grün, and K. Hornik, "Arules—a computational environment for mining association rules and frequent item sets," *Journal of Statistical Software*, vol. 14, 2005.
- [37] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," *ACM SIGMOD Record*, vol. 22, no. 2, pp. 207–216, 1993.
- [38] S. Encheva, Y. Kondratenko, M. Z. Solesvik, and S. Tumin, "Decision support systems in logistics," in *Proceeding of International electronic conference on computer science*, pp. 254–256, 2007.
- [39] M.-C. Chen, "Configuration of cellular manufacturing systems using association rule induction," *International Journal of Production Research*, vol. 41, no. 2, pp. 381–395, 2003.
- [40] R. Srikant and R. Agrawal, "Mining generalized association rules," *Future Generation Computer Systems*, vol. 13, no. 2-3, pp. 161–180, 1997.
- [41] J. Wu and X. M. Li, "An effective mining algorithm for weighted association rules in communication networks," *Journal of Computers*, vol. 3, no. 10, pp. 20–27, 2008.
- [42] C. H. Cai, A. W. C. Fu, C. H. Cheng, and W. W. Kwong, "Mining association rules with weighted items," in *Proceedings of the International Symposium on Database Engineering & Applications*, pp. 68–77, Cardiff, UK, 1998.
- [43] J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.
- [44] A. A. Taleizadeh, F. Barzinpour, and H.-M. Wee, "Meta-heuristic algorithms for solving a fuzzy single-period problem," *Mathematical and Computer Modelling*, vol. 54, no. 5-6, pp. 1273–1285, 2011.
- [45] H. Yang, J. Yi, J. Zhao, and Zh. Dong, "Extreme learning machine based genetic algorithm and its application in power system economic dispatch," *Neurocomputing*, vol. 102, pp. 154–162, 2013.