International Journal of Engineering

Journal Homepage: www.ije.ir

Integrated Order Batching and Distribution Scheduling in a Single-block Order Picking Warehouse Considering S-Shape Routing Policy

Y. Zare Mehrjerdi*, M. Alipour, A. Mostafaeipour

Department of Industrial Engineering, Yazd university, Yazd, Iran

PAPER INFO

Paper history:
Received 21 November 2017
Received in revised form 12 May 2017
Accepted 17 August 2018

Keywords:
Warehouse
Order Picking System
Order Batching
Picker-to-Part Systems
Distribution Scheduling

ABSTRACT

In this paper, a mixed-integer linear programming model is proposed to integrate batch picking and distribution scheduling problems in order to optimize them simultaneously in an order picking warehouse. A tow-phase heuristic algorithm is presented to solve it in reasonable time. The first phase uses a genetic algorithm to evaluate and select permutations of the given set of customers. The second phase uses the route first-cluster method to obtain an effective schedule for a given permutation of customers. Computational experiments represent that integrated approach can lead to significant reduction in the makespan. Moreover, Empirical observations on the performance of the heuristic algorithm are reported.

doi*: 10.5829/ije.2018.31.10a.15*

1. INTRODUCTION

The process of retrieving items from storage shelves in a warehouse in order to satisfy customer demands, is named order picking. Since incoming items in a warehouse are received and stored in large-volume and customers order small volumes, order picking problem arises. The negative impact of unsatisfactory customer service as a result of long processing and delivery times on the one hand and high labor and delivery costs on the other hand, may reduce the competitiveness of the warehouse [1]. In this paper, we consider a picker-to-parts order picking system in which pickers walk or ride through the warehouse and collect the required items. In a picker-to-part order picking system, customer orders are generally combined to form picking batches (order batching) which can increase the efficiency of warehouse operations considerably by reducing the picking time and cost [2]. The main concentration of this paper is on the order batching problem.

Customers prefer to receive their requested products at the same day or the day after ordering at minimal cost [3, 4]. Low cost delivery imposes higher costs on companies. To meet what customers expect, companies need to optimize their distribution processes along with optimizing internal processes of their warehouses [5]. Since there are dependencies between order picking and order distribution, we can integrated these problems and optimize them jointly to obtain a global optimum solution rather than considering them uncordinated and optimizing them separately. According to the uncoordinated approach, order picking processes are separated from delivery processes by a fixed departure time [6-9]. In the integrated approach unlike the uncoordinated approach, the order picking and the distribution processes can overlap in time. As a result, the warehouse has more flexibility in executing the distribution process [5].

In this paper, decisions related to picking and distribution of customer orders are integrated. It is supposed that customer orders are batched before being picked and there are one order picker and one vehicle in the warehouse to pick and distribute customer orders, respectively. A mathematical model will be proposed to: 1) construct batches from customer orders, 2) schedule the picking process of the constructed batches, 3)

*Corresponding Author Email: *yzare@yazd.ac.ir* (Y. Zare Mehrjerdi)

Please cite this article as: Y. Zare Mehrjerdi, M. Alipour, A. Mostafaeipour, Integrated Order Batching and Distribution Scheduling in a Single-block Order Picking Warehouse Considering S-Shape Routing Policy, International Journal of Engineering (IJE), IJE TRANSACTIONS A: Basics Vol. 31, No. 10, (October 2018) 1723-1733

schedule the distribution process of the completed batches. The objective is minimizing the makespan, the time it takes to deliver the last batch to its corresponding customers. This objective was chosen because it requires close cooperation by the order picking and the distribution phases.

2. LITERATURE SURVEY

The integrated order picking and distribution scheduling (IOPDS) and the integrated production and distribution scheduling problems (IPDS) both have production and distribution phases. Therefore, they share some similar characteristics. The IPDS has been widely studied in recent years. IPDS models in the literature can be classified according to five major characteristics [10]: machine configuration in the production plant(s), order parameters, objective function, delivery process and the number of customers. In the model of this paper, the single machine configuration is considered since one order picker exists in the warehouse (order picker is considered as a machine), the objective function is minimizing the makespan and the delivery method is the routing-based batch delivery method. The difference between IPDS and IOPDS is that the processing time in IPDS in generally constant, whereas in IOPDS, an NP-hard optimization problem (batching problem) needs to be solved to obtain the processing time of a batch.

Similar studies which are the most relevance to our study, considering the production terminology, are conducted by Low et al. [11-13] which tried to integrate practical scheduling and VRP problems in a distribution center. These studies are different from our work since they do not consider the order batching problem. Therefore, we need to discuss the review of order batching models.

Order batching has an important role in optimizing order picking processes [14]. A mixed integer programming approach is proposed by Bozer and Kile [15] to obtain near-optimal solutions for the order batching problem. The proposed approach is only applicable for instances with small number of orders (up to 25). Gademann and Velde [16] presented a branch-and-price method to solve small-sized order batching problem in reasonable computational time. Since the order batching problem in NP-had [16], heuristic approaches should be used to solve the large-sized order batching problems in reasonable computational time. There are four types of heuristic approaches in the literature that are utilized to solve the order batching problem [14]: seed algorithms [17], saving algorithms [2], priority rule based algorithms [18], and meta-heuristic algorithms [19-23].

Few researches have tried to model the IOPDS. Zhang et al. [9] integrated order picking and distribution operations considering a B2C e-commerce context. They assumed that distribution operations are implemented by a 3PL service provider. As a result, fixed departure times are considered at which the 3PL service provider starts the delivery process of customer orders. Thus, the decisions related to vehicle routing are not taken explicitly into account by the study. Moon et al. [5] are the only authors who integrated order picking and vehicle routing decisions. They proposed a mathematical model for the integrated order picking and vehicle routing problem considering time windows but they did not consider batching of customer orders in order picking. In this paper, however, the order picking and distribution problems are integrated by taking into account decisions related to both batching of orders and routing the vehicle. To the authors' best of knowledge, this study is the first one which integrates batch picking and distribution operations explicitly. Our model solves these decision issues: (1) which customer orders should be in the same batch; (2) when should the picking process of each batch be started; (3) at which time should be started delivery process of each batch. The objective function is minimizing the makespan, the time needed to deliver the last batch to its corresponding customers.

This paper is structured as follows. In section 3, we proposed a mathematical model for the integrated order batching and distribution scheduling problem. The heuristic solution approach is presented in section 4. The genetic algorithm is explained in section 5. In section 6, we compared the presented integrated model with the uncoordinated approach in terms of the achieved makespan and computational times. Moreover, results of the heuristic algorithm is compared to results which obtained from MILP solver to assess performance of the heuristic algorithm. Finally in section 7, conclusions and future research directions are discussed.

3. INTEGRATED MODEL DESCRIPTION AND FORMULATION

3. 1. Integrated Model Description

In this section, an integrated model will be proposed to define a global solution for both the order picking and the order delivery problems. In the following, we will explain the order picking and order delivery processes.

In the order picking system, orders which are received from a set of customers with definite geographic locations, are picked by an order picker. The picking process in an order picking warehouse can be described briefly as follows: the order picker begins to move from the depot with a pick list which represents the storage shelves of the requested articles and the number of items requested for each article. He/she walks or ride by a vehicle through the aisles of the warehouse and picks the requested items from different shelves according to the pick list. Then he/she comes back to the depot and delivers the collected items. The route through which the order picker walks or rides to pick the requested items is determined by S-Shape routing policy [14].

In the order delivery system, a vehicle with limited capacity is used to deliver the orders for which the picking process is finished. Since the vehicle has a limited capacity, it will need to deliver the orders to customers by using multiple trips, i.e. the vehicle will need to come back to the warehouse several times during the planning period. In each trip, the vehicle starts from the depot, visits customer locations for a batch in sequence, and at last returns to the depot. It is supposed that each customer is demanded one order. Therefore, each customer is visited exactly once during the planning period.

3. 2. Integrated Model Formulation The following assumptions are made to construct the mathematical model for the integrated order batching and distribution scheduling problem:
(1) The layout of picking area is a single-block warehouse which have two cross-aisles (see in [14]). This layout which is a common layout in the literature, is shown in Figure 1; (2) S-shape routing policy is utilized to define picking routes for batches since this method provides near-optimal straightforward routes [14]. An example for S-shape routing is illustrated in Figure 1; (3) The order picker can handle a batch only when the picking process of the previous batch which is assigned to him/her is finished; (4) Each batch is distributed among its corresponding customers using a separate vehicle route; (5) The vehicle trip to deliver a batch can start only when the picking process of the batch is already finished and the vehicle is also available; (6) A truck is used by the order picker to pick customer orders in the warehouse which it's capacity is nearly equal to the capacity of the distribution vehicle; (7) The time needed to rearrange batches for delivery is considerable. Therefore, the orders are delivered to customers as they batched for picking; (8) Splitting the orders among more than one batch is prohibited, since it would result in additional sorting effort.

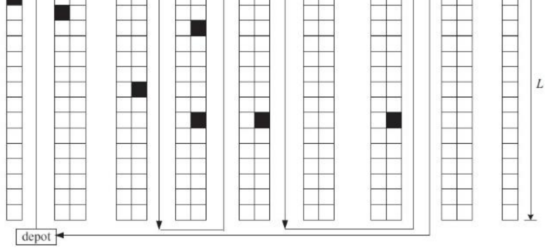


Figure 1. The layout of picking area

The following sets, indices and parameters are used in the mathematical model :

$M = \{0,1,2,\ldots,m\}$: set of customers and the depot, indices i and j , where $i = 0$ indicates the depot

$N = \{1,2,\ldots,n\}$: set of batches (vehicle trips), index k (an upper bound for m can be $m = n$)

$A = \{1,2,\ldots,a\}$: set of aisles in the warehouse layout, index g

$L = \{1,2,\ldots,l\}$: set of pick locations in each side of an aisle in the warehouse, index h

v_{travel} : travel velocity of the order picker, distance that can be covered by the order picker per unit of time

v_{pick} : pick velocity, number of items that can be searched and picked by the order picker per time unit

v_{dist} : vehicle velocity, distance that can be covered by the vehicle per unit of time

b_i : number of items requested in the order requested by customer i

C : batch capacity, i.e. maximum number of items in a batch

$d_{i,j}$: distance between locations of customers i and j

t^{b}_{setup} : batch setup time, i.e. time needed for administrative tasks for each batch

t^{v}_{setup} : vehicle setup time, i.e. time needed for packing customer orders and loading vehicle

γ_{igh} : 1 if the order requested by customer i has at least one item to be picked in pick location h of aisle g , 0 otherwise

μ_{ig} : 1 if the order requested by customer i has at least one item to be picked in aisle g , 0 otherwise

W : distance between centers of two adjacent aisles in the warehouse

L : length of each aisle

r : number of pick locations (rows) of the warehouse

The following decision variables are used in the mathematical model:

x_{ik} : 1 if the order requested by customer i is assigned to batch k , 0 otherwise

y_{kij} : 1 if customer j is visited after customer i in k th trip of the vehicle

s^{pick}_{k} : start time of picking batch k

f^{pick}_{k} : finish time of picking batch k

s^{dist}_{k} : start time of the k th trip of the vehicle

f^{dist}_{ki} : delivery moment for the order requested by customer i in k th trip of the vehicle (batch k)

α_k : number of aisles in which there are pick locations which must be visited for picking batch k

α^{l}_{k} : number of the leftmost aisle in which there is at least one pick location which must be visited for picking batch k

α_k^f : number of the rightmost aisle in which there is at least one pick location which must be visited for picking batch k

θ_k^f : distance from front aisle to the farthest item which must be picked in batch k

β_{kg} : 1 if there is at least one pick location for batch k in aisle g , 0 otherwise

π_{kg}^f : number of the last row to visit in aisle g for batch k

dis_k : distance the order picker needs to travel in warehouse for picking batch k

δ_k : 1 if an even number of aisles must be visited for picking batch k , 0 otherwise

λ_k : An auxiliary integer variable

The integrated order batching and distribution scheduling problem is formulated as follows:

$$\min \max_{i \in M, k \in N} \left\{ f_{ki}^{dist} \right\} \tag{1}$$

$$\sum_{k=1}^{n} x_{ik} = 1, \;\; i \in M, i \geq 1 \tag{2}$$

$$\sum_{i=1}^{m} x_{ik} \leq m \sum_{i=1}^{m} x_{i,k-1}, \;\; k \in N, k \geq 2 \tag{3}$$

$$\sum_{i=1}^{m} b_i x_{ik} \leq C, \;\; k \in N \tag{4}$$

$$s_k^{pick} \geq f_{k-1}^{pick}, \;\; k \in N, k \geq 2 \tag{5}$$

$$f_k^{pick} \geq s_k^{pick} + dis_k / v_{travel} + \sum_{i=1}^{m} b_i x_{ik} / v_{pick} + t_{setup}^{b}, \;\; k \in N \tag{6}$$

$$\sum_{j=1}^{m} y_{k0j} \leq \sum_{i=1}^{m} x_{ik}, \;\; k \in N \tag{7}$$

$$\sum_{i=1}^{m} x_{ik} \leq m \sum_{j=1}^{m} y_{k0j}, \;\; k \in N \tag{8}$$

$$\sum_{j=1}^{m} y_{k0j} \leq 1, \;\; k \in N \tag{9}$$

$$\sum_{j=0, j \neq i}^{m} y_{kij} = \sum_{j=0, j \neq i}^{m} y_{kji}, \;\; k \in N, i \in M \tag{10}$$

$$\sum_{j=0, j \neq i}^{m} y_{kji} \leq x_{ik}, \;\; k \in N, i \geq 1 \tag{11}$$

$$x_{ik} \leq \sum_{j=0, j \neq i}^{m} y_{kji}, \;\; k \in N, i \geq 1 \tag{12}$$

$$f_{k0}^{dist} \geq s_k^{dist} + t_{setup}^{v}, \;\; k \in N \tag{13}$$

$$s_k^{dist} \geq \max \left\{ f_k^{pick}, f_{(k-1)i}^{dist} + \left(\sum_{j=1}^{m} d_{j0} y_{(k-1)j0} / v_{dist} \right) \right\}, \;\; k \in N, i \in M, k \geq 2, i \geq 1 \tag{14}$$

$$f_{kj}^{dist} \geq f_{ki}^{dist} + d_{ij} / v_{dist} - M\left(1 - y_{kij}\right), \;\; k \in N, i \in M, j \geq 1 \tag{15}$$

$$\beta_{kg} \leq \sum_{i=1}^{m} \mu_{ig} x_{ik}, \;\; k \in N, g \in A \tag{16}$$

$$\sum_{i=1}^{m} \mu_{ig} x_{ik} \leq m.\beta_{kg}, \;\; k \in N, g \in A \tag{17}$$

$$\alpha_k = \sum_{g=1}^{a} \beta_{kg}, \;\; k \in N \tag{18}$$

$$\alpha_k^l = \min_{g \in A} \left\{ g.\beta_{kg} + a\left(1 - \beta_{kg}\right) \right\}, \;\; k \in N \tag{19}$$

$$\alpha_k^f = \max_{g \in A} \left\{ g.\beta_{kg} \right\}, \;\; k \in N \tag{20}$$

$$\pi_{kg}^f = \max_{i \in M, h \in L} \left\{ h \gamma_{igh} x_{ik} \right\}, \;\; k \in N, g \in A \tag{21}$$

$$\left| \theta_k^f - L / r\left(\pi_{kg}^f - 0.5\right) \right| \leq M \left| \alpha_k^f - g \right|, \;\; k \in N, g \in A \tag{22}$$

$$\left| \alpha_k - 2\lambda_k - 1 \right| \leq M \delta_k, \;\; k \in N \tag{23}$$

$$\left| \alpha_k - 2\lambda_k \right| \leq M\left(1 - \delta_k\right), \;\; k \in N \tag{24}$$

$$dis_k \geq \left(\left(\alpha_k^l - 1 \right) W + \alpha_k L + \left(\alpha_k - 1 \right) W + \left(\alpha_k^f - 1 \right) W \right) - M\left(1 - \delta_k\right), k \in N \tag{25}$$

$$dis_k \geq \left(\left(\alpha_k^l - 1\right)W + (\alpha_k - 1)L + 2\theta_k^f + (\alpha_k - 1)W + \left(\alpha_k^f - 1\right)W\right) - M\delta_k, k \in N \quad (26)$$

$$x_{ik}, y_{ijk}, \pi_{kg}, \beta_{kg}, \{0,1\}, \quad i, j \in M, k \in N, g \in A \quad (27)$$

$$s_k^{pick}, f_k^{pick}, s_k^{dist}, f_{ki}^{pick}, \theta_k^f \geq 0, \theta_k^f \leq L, \pi_{kg}^f \leq l, \quad i \in M, k \in N \quad (28)$$

$$\alpha_k, \alpha_k^l, \alpha_k^f \{1, \ldots a\}, \lambda_k \leq \frac{a}{2}, k \in N \quad (29)$$

Objective function (1) tries to minimize the makespan. Equation (2) indicates that each customer order must be assigned to only one batch. Inequality (3) guaranties that batches are opened sequentially. Inequality (4) is the capacity constraint of a batch. Inequality (5) indicates that the order picker can start picking of a batch only when picking of the previous batch is finished. Equation (6) computes the finish time of picking a batch. The finish time of picking a batch is the moment at which picking process of the batch finishes, which is obtained by adding the service time of the batch in the order picking sub-system to the start time of picking the batch. The service time of a batch is composed of setup time, travel time and picking time . Inequalities (7), (8) and (9) simultaneously ensure that the vehicle leaves the depot exactly once only for opened batches (batches that at least one customer order is assigned to them). According to inequalities (10), (11) and (12) in a trip of the vehicle, only those customers whose orders are assigned to that trip are visited exactly once and other customers are not visited. Inequality (13) specifies that the vehicle needs a setup before it can leave the depot for a trip. From inequality (14) it can be inferred that the vehicle can start the delivery process of a batch only when picking of the batch is finished and the vehicle is already returned to the warehouse from the previous trip. Inequality (15) is to calculate the delivery moments for customers in each trip of the vehicle based on distances between customer locations and average speed of the vehicle. This inequality is also a sub-tour elimination constraint.

We used the formula presented in literature [24] to calculate S-shape picking distance for a batch k. The formula is as follows:

$$dis_k = \begin{cases} (\alpha_k^l - 1)W + \alpha_k L + (\alpha_k - 1)W + (\alpha_k^f - 1)W, \alpha_k \text{is even} \\ (\alpha_k^l - 1)W + (\alpha_k - 1)L + 2\theta_k^f + (\alpha_k - 1)W + (\alpha_k^f - 1)W, \alpha_k \text{is odd} \end{cases} \quad (30)$$

To calculate dis_k, first we need to calculate α_k, α_k^l, α_k^f and θ_k^f. The mathematical relations (17) to (24) are to calculate α_k, α_k^l, α_k^f and θ_k^f and the relations (25) to (26) help to calculate dis_k based on the obtained values for α_k, α_k^l, α_k^f and θ_k^f. The Inequalities (16) and (17) is to calculate β_{kg} for each $k \in N$ and each $g \in A$. Considering batch k and aisle g for example, if at least one item of batch k is stored in aisle g, β_{kg} will be equal to 1, otherwise β_{kg} will be equal to zero. Equation (18) computes number of aisles that must be visited for picking a batch. Equations (19) and (20) help to obtain the first (leftmost) aisle number and the last (rightmost) aisle number which must be visited for picking a batch, respectively.

θ_k^f is calculated using formula $\theta_k^f = L / r\left(\pi_{kg}^f - 0.5\right)$ from reference [24]. Equations (21) and (22) try to obtain θ_k^f using the abovementioned formula. First, π_{kg}^f is obtained for each k and g using equation (21). Then, θ_k^f is calculated for each k based on the obtained values for π_{kg}^f according to equation (22). In equation (22), if $\alpha_k^f = g$, θ_k^f will be equal to $L / r\left(\pi_{kg}^f - 0.5\right)$. Inequalities (23) and (24) aim to define the evenness or oddness of the number of aisles that must be visited for a batch. Considering batch k for example, if α_k is an odd number, δ_k will be equal to zero, otherwise δ_k will be equal to 1. Equations (25) and (26) calculate picking distance dis_k based on the evenness or oddness of α_k and considering equation (30). Considering batch k for example, if $\delta_k = 1$ then inequality (25) triggers, otherwise inequality (26) will be active. Inequalities (27), (28) and (29) define the domains of decision variables.

It should be clarified that the *max* function in objective function (1) and constraints (14), (20), (21), the *min* function in constraint (19) and the *absolute* function in constraints (22), (23), (24) are linearized using basic techniques of integer programming.

4. HEURISTIC APPROACH

4. 1. Finding a Feasible Set of Batches According to the route first-cluster method [25], a feasible set of batches can be obtained for a given permutation of customers by finding a path on a directed graph. Let us define graph G_ϕ for permutation ϕ of customers with nodes $V_{G_\phi} = \{0, \phi(1), \phi(2), ..., \phi(n)\}$ and edges E_{G_ϕ} as follows:

Definition 1. For each pair of nodes (customers), ϕ_i and $\phi_j \in V_{G_\phi}$, where $0 \leq i < j \leq n$ and $\phi(0) = 0$, there is a directed arc from ϕ_i to ϕ_j ($\phi_i \to \phi_j$) if and only if the total weight of orders which belong to customers $\phi(i+1)$, $\phi(i+2)$,..., $\phi(j)$ does not exceed the batch capacity. Note that arc $\phi_i \to \phi_j$ indicates a batch which contains orders of customers $\phi(i+1)$, $\phi(i+2)$,…, $\phi(j)$ and is delivered by the vehicle using route $0 \to \phi(i+1) \to \phi(i+2) \to ... \to \phi(j)$.

4. 2. Scheduling a Set Of Feasible Batches Consider that we have obtained a set of m feasible batches for permutation ϕ by finding a path on the corresponding graph G_ϕ. Let $\phi(i_k)$ represents the last customer visited in trip of batch k, where $k=1,...,m$, $i_0=0$ and $i_m=n$. The picking time of batch k can be calculated using the following equations:

$$T_k^{pick} = dis_k / v_{travel} + b_k / v_{pick} + t_{setup}^b \tag{31}$$

where, b_k represents number of items in batch k. Other parameters were previously explained in section 3.2. The distance of the picking tour for batch k, dis_k, can be obtained using equation (30).

Parameters α_k, α_k^l and α_k^f, θ_k^f, L and W were previously discussed in section 3.2. The time needed for the vehicle to deliver orders of batch k can be obtained from the following formula:

$$T_k^{dist} = (d_{0,\phi(i_{k-1}+1)} + \sum_{j=i_{k-1}+1}^{i_{k-1}} d_{\phi(k),\phi(k+1)} + d(\phi(i_k),0)) / v_{dist} \tag{32}$$

Parameters d_{ij} and V_{dist} were previously presented in section 3.2. Now we present a linear programming formulation to find a schedule with minimum makespan for a given set of feasible batches. The mathematical model for scheduling a given set of feasible batches is as follows:

$$\text{minimize } s_m^{dist} + T_m^{dist} \tag{33}$$

$$s_{k+1}^{pick} \geq s_k^{pick} + T_k^{pick}, \forall k \in \{1,...,m-1\} \tag{34}$$

$$s_{k+1}^{dist} \geq s_k^{dist} + T_k^{dist}, \forall k \in \{1,...,m-1\} \tag{35}$$

$$s_k^{dist} \geq s_k^{pick} + T_k^{pick}, \forall k \in \{1,...,m\} \tag{36}$$

$$s_k^{pick}, s_k^{dist} \geq 0, \forall k \in \{1,...,m\} \tag{37}$$

Objective function (33) tries to minimize the makespan, the moment at which the vehicle returns to the depot from the last delivery trip. Inequality (34) states that picking of batch $k+1$ cannot be started until picking of batch k has been completed. Inequality (35) indicates that distribution of batch $k+1$ cannot be started until distribution of batch k has been completed. Inequality (36) specifies that distribution of batch k cannot be started until picking of batch k has been completed. Inequality (37) defines the domains of decision variables. Since the presented model is of minimization type and sequence of batches is specified, the optimal solution of the mathematical model can be found using the following set of recursive equations:

$$s_{k+1}^{pick} = s_k^{pick} + T_k^{pick}, \forall k \in \{1,...,m-1\}, \ s_1^{pick} = 0 \tag{38}$$

$$s_{k+1}^{dist} = \max\{s_{k+1}^{pick} + T_{k+1}^{pick}, s_k^{dist} + T_k^{dist}\}, \forall k \in \{1,...,m-1\}, \quad s_1^{dist} = s_1^{pick} + T_1^{pick} \tag{39}$$

Therefore, the problem of scheduling a given set of feasible batches can be solved in $O(m)$.

4. 3. Finding Final Schedule for a Given Permutation In this section we will present the shortest path-based algorithm (SHPBA) which uses the elements of previous sections to choose a set of feasible batches with minimum total service time (picking and distribution times) for a permutation ϕ of customers. This algorithm assigns customer orders to batches by finding the shortest path on the graph of feasible batches which discussed in section 4.1. The algorithm is presented in Algorithm 1.

We can calculate complexity of the SHPBA by adding complexities of different elements of the algorithm. The complexity of creating graph G_ϕ for permutation ϕ of n customers equals $O(n \log n)$. The complexity of finding the shortest path on graph G_ϕ using the reaching algorithm is equal to $O(n)$ [26]. The complexity of scheduling a set of feasible batches is $O(m)$. Therefore, the complexity of the SHPBA is $O(n \log n)$.

Algorithm 1. Shortest path-based algorithm (SHPBA)
Input: A permutation ϕ of n customers;
Create graph G_ϕ for permutation ϕ according to section;
Calculate $T_k^{pick} + T_k^{dist}$ for each arc k (each batch) of graph G_ϕ according to section 4.2;
Find a shortest path p^* from node 0 to node $\phi(n)$;
Schedule batches which are identified by path p^* using the scheduling procedure in section 4.2.

5. GENETIC ALGORITHM

A genetic algorithm is used as the first phase of our algorithm to generate permutations according to which customers are served. Permutations which are generated by the genetic algorithm are entered the second phase (SHPBA) to obtain an effective schedule based on the route first-cluster method. The SHPBA is used to guide the searching process of the genetic algorithm.

Process of genetic algorithm starts by generating random permutation of customers as initial population. A schedule is found for each individual using the shortest path-based algorithm and fitness values are assigned to individuals based on the obtained makespans. In the next step, individuals are assigned to mating pool set stochastically based on their fitness values. Mating pool set is the same size as population set and it is possible for an individual to be assigned to mating pool multiple times. To assign individuals to mating pool, the binary tournament selection method is applied [27]. Then each two successive individuals in mating pool are paired together to form a couple. Each couple are combined using the order crossover [28] according to the crossover rate to form two new offspring. In the next step, each offspring is randomly mutated using the pairwise exchange mutation operator [28] according to mutation rate.

6. COMPUTATIONAL STUDY

6. 1. Purpose In this section, several experiments are implemented to assess performance of the integrated model. The integrated model is compared to the uncoordinated approaches in terms of the achieved makespan and computational times to obtain benefit of integration. Moreover, results of the proposed heuristic algorithm is compared to the results obtained using a MILP solver to evaluate performance of the heuristic algorithm.

6. 2. Data Generation We rely on the assumptions reported in reference [14] to generate data for test instances. A common single block warehouse with two cross aisles is considered. Cross aisles are located in front and back of the picking area. The picking area contains 900 storage locations where a different storage location is considered for each article. These storage locations are organized in 10 aisles such that 45 storage locations are included in each side of an aisle. The length of each storage location is one length unit (1 LU) and the center to center distance between two neighbour aisles is 5 LU. The depot is 1.5 LU away from the first storage location of the first aisle (the leftmost aisle) and the distance between the depot and the front cross-aisle is 0.5 LU. It is also assumed that the order picker can travel 48 LU per unit of time and 10 s is needed for the order picker to search and pick an item from a shelve, i.e. $v_{travel} = 48$ LU/min and v_{pick} =6 items/min. The setup time for each batch is 3 min. With regard to the distribution system, it is assumed that customers are located randomly in a square of 20 km by 20 km and the depot is located on the center of square. The vehicle can travel 50 km per hour. The setup time for each trip of the vehicle is 5 min.

It is supposed that items are stored in storage locations using a class-based storage policy. According to this policy, the articles are placed in three classes A, B and C, based on their demand frequencies. Articles with high demand frequency are included in class A and are stored in the first aisle. Articles with medium demand frequency are included in class B and are stored in the three subsequent aisles (aisles no. 2, 3 and no.4). Class C contains articles with low demand frequency which are stored in 6 remaining aisles. It is assumed that 52, 36 and 12% of the requested articles belong to the articles in A, B and C, respectively. It is also assumed that articles are stored randomly within a class.

Several classes of test instances are defined by considering 11 different values for number of customer orders ranging from 5 to 30 customer orders and two different values for batch capacity ($W = 45$ and $W = 75$). Therefore, 22 classes of test instances are used for experiments. The quantity of items for each batch is uniformly distributed in $\{5, 6, \ldots, 25\}$. The values for W and b are defined in a way that each batch averagely contains 3 to 5 customer orders. These assumptions are in line with those of reported in literature [14].

It should be noted that we use the Cplex solver of the NEOS optimization server [29-31] as a MILP solver to obtain the optimal solutions of the integrated and the uncoordinated approaches considering different test instances. We implemented the heuristic algorithm for all test instances using MATLAB R2014a on a Core i7 processor 1.6 GHz and 4.0 GB RAM.

6. 3. Computational Results The results of applying the integrated and the uncoordinated approaches are presented in Tables 1 and 2. The uncoordinated model is solved by means of the MILP solver and the integrated model is solved using both the MILP solver and the heuristic algorithm. Table 1 represents the computational times and Table 2 represents the makespan obtained using the aforementioned approaches. As we can see in Table 1, the computational times for all test problems on the MILP solver is less than or equal to 8 hours. That is because jobs can run at most 8 hours on NEOS optimization servers. It should be noted that, for the test instances for which the solver could not find the optimal solution of the integrated or the uncoordinated approaches, the best makespan obtained by the MILP solver within 8 hours is recorded in Table 2.

To obtain a value for benefit of integration, we compare the makespan values obtained using the integrated and the uncoordinated approaches and define the makespan Reduction Rate (MRR) as follows:

$$MRR = 100 \times [(M_{unc} - Mint) / M_{unc}] \quad (40)$$

where, M_{unc} and M_{int} represent the makespans obtained by solving the uncoordinated and the integrated models using the MILP solver respectively. As it can be seen in Table 1, with regard to the instances for which we could find the optimal solution (instances with 5 to 8 customer orders), the computational time needed to find the optimal solution for the integrated model increases exponentially as number of customer orders increases.

TABLE 1. Computational times for all test instances (hours) implemented by means of the Cplex solver and the heuristic algorithm

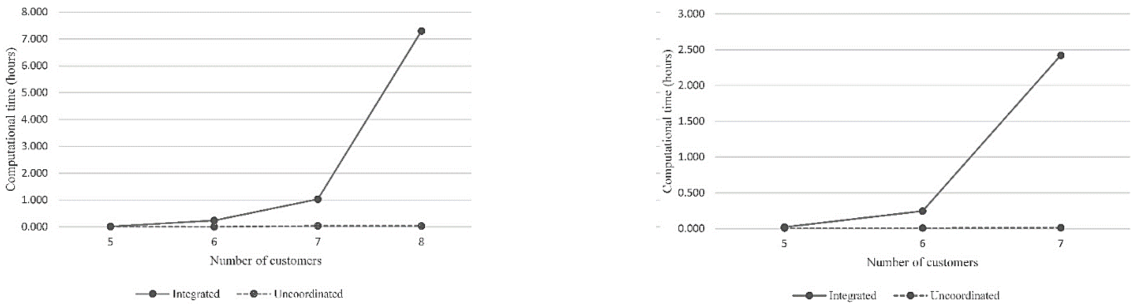
Number of customers	W	Integrated		Uncoordinated
		Cplex	Heuristic	Cplex
5	45	0.014	0.006	0.009
	75	0.021	0.005	0.006
6	45	0.237	0.007	0.004
	75	0.246	0.007	0.009
7	45	1.031	0.009	0.037
	75	2.421	0.009	0.013
8	45	7.295	0.012	0.035
	75	8.000	0.011	0.018
9	45	8.000	0.014	0.017
	75	8.000	0.014	0.013
10	45	8.000	0.017	0.032
	75	8.000	0.017	0.028
12	45	8.000	0.023	3.959
	75	8.000	0.023	0.474
15	45	8.000	0.048	4.909
	75	8.000	0.050	0.582
20	45	8.000	0.090	8.000
	75	8.000	0.069	8.000
25	45	8.000	0.119	8.000
	75	8.000	0.096	8.000
30	45	8.000	0.170	8.000
	75	8.000	0.190	8.000

TABLE 2. Comparison between the makespans obtained using the integrated and the uncoordinated approaches

Number of customers	W	Integrated		Uncoordinated	MRR (%)
		Cplex	Heuristic	Cplex	
5	45	69.39	69.39	81.29	14.6
	75	60.78	60.78	67.48	9.9
6	45	92.29	92.29	109.04	15.4
	75	80.45	80.47	86.85	7.4
7	45	87.25	87.26	121.38	28.1
	75	80.22	80.24	100.56	20.2
8	45	124.58	124.83	141.17	11.8
	75	102.02	102.70	123.70	17.5
9	45	134.97	135.30	164.99	18.2
	75	101.92	103.73	125.58	18.8
10	45	132.50	129.25	160.21	17.3
	75	109.41	110.29	132.82	17.6
12	45	167.55	159.28	192.85	13.1
	75	134.90	118.62	151.14	10.7
15	45	197.59	178.34	246.87	20.0
	75	156.43	142.21	182.88	14.5
20	45	268.09	247.82	358.00	25.1
	75	253.04	198.62	298.55	15.2
25	45	290.84	255.04	356.12	18.3
	75	242.63	195.45	285.14	14.9
30	45	512.11	382.98	576.37	11.1
	75	441.00	287.84	419.22	-5.2

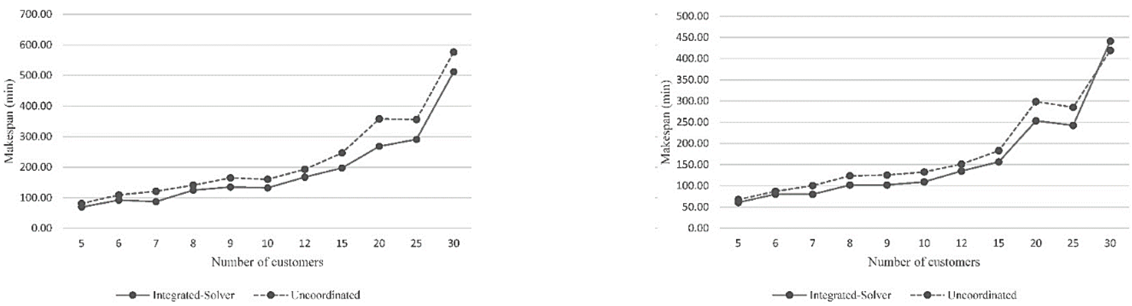
It indicates that heuristic solutions are required to solve realistic sizes of the integrated model in reasonable time. Figures 2a and 2b illustrate comparisons between the computational times of the integrated and the uncoordinated approaches considering instances for which we could obtain the optimal solution of the integrated model. As we can see in Figure 2, the uncoordinated approach needed much less computational time as compared to the integrated approach. But, we will illustrate that the integrated approach results in significant reduction in makespan in comparison to the uncoordinated approach. For this purpose, we compare the integrated and the uncoordinated approaches in terms of the achieved makespan by investigating the test instances in three different categories: 1) instances for which we could find the optimal solution for both the integrated and the uncoordinated approaches: instances with 5 to 8 customer orders and $W = 45$, and instances with 5 to 7 customer orders and $W = 75$, 2) instances for which we could find the optimal solution of the uncoordinated approach but we could not find the optimal solution of the integrated approach: instances with 9 to 15 customer orders and $W = 45$, and instances with 8 to 15 customer orders and $W = 75$,and 3) instances for which we could not find the optimal solution for any of the integrated and the uncoordinated approaches: instances with 20 to 30 customer orders and $W = 45$, and instances with 20 to 30 customer orders and $W = 75$. Considering the instances with $W = 45$ and $W = 75$ in the first category of instances, the makespan values achieved by the integrated approach was averagely 17.5% and 12.5% better than the makespan values achieved by the uncoordinated approach respectively. It can be seen in Table 2 that, for the first category of test instances, integrating the order batching and distribution scheduling problems resulted in time saving up to 28.1% and 20.2% for the $W = 45$ and $W = 75$ cases respectively. Considering the second category of instances with $W = 45$ and $W = 75$, the makespan obtained for the integrated model within 8 hours, although not optimal, is averagely 17.1% and 15.8% better than the optimal makespan obtained by applying the uncoordinated approach. In this case, as we can see in Table 2, integrating the order batching and distribution problems leaded in time saving up to 20 and 18.8% for the $W = 45$ and $W = 75$ cases respectively. Finally, with regard to the third category of instances with $W = 45$ and $W = 75$, the makespan obtained for the integrated model within 8 hours is averagely 18.2 and 8.3% better than the makespan obtained for the uncoordinated model within 8 hours. In this case, integrating the order batching and distribution problems resulted in time saving up to 25.1and 15.2% in comparison to the uncoordinated approach. Figure 3 illustrates a comparison between the integrated and the uncoordinated approaches in terms of the achieved makespans. As we can see in Figure 3, the integrated approach is superior to the uncoordinated approach in terms of the achieved makespan in almost all instances.

Figures 4a and 4b illustrate performance of the presented heuristic algorithm in solving the integrated model in comparison to the MILP solver considering different test instances with $W = 45$ and $W = 75$, respectively. As we can see in Figures 4a and 4b and Table 2, considering the first category of instances for which we have the optimal solutions, the makespan obtained by the heuristic algorithm is only 0% and 0.04% worse than the makespan obtained by the heuristic algorithm for instances with $W = 45$ and $W = 75$, respectively. Considering the second and the third categories of instances together, for which the obtained solutions by the MILP solver are not necessarily the optimal solutions, the makespan achieved using the heuristic algorithm is averagely 9 and 12% and at its best performance 25 and 35% better than the makespan achieved using the MILP solver for instances with $W = 45$ and $W = 75$, respectively. It also should be noted that the heuristic algorithm found the aforementioned makespan values for each of test instances in less than 0.2 hours. Figure 5 represents the computational times of solving the integrated model using the heuristic algorithm for different test instances. Given the above, the heuristic algorithm can obtain good solutions for the integrated order batching and distribution scheduling problem in reasonable time.



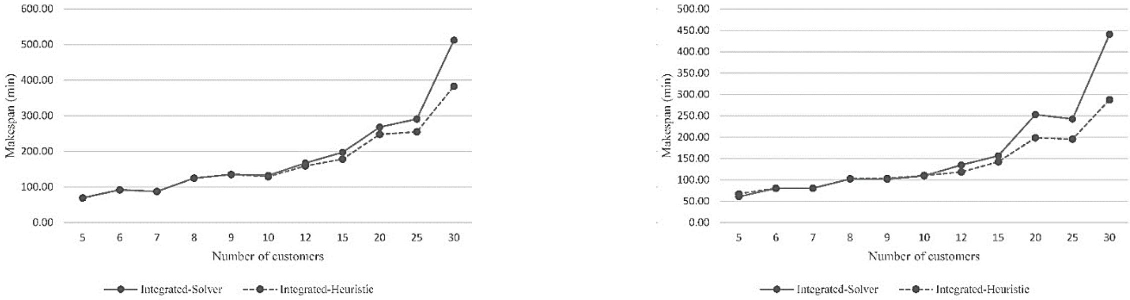
(a) Instances with $W = 45$ (b) Instances with $W = 75$

Figure 2. Comparison between the computational times of the integrated and the uncoordinated approaches



(a) Instances with $W = 45$ (b) Instances with $W = 75$

Figure 3. Comparison between the integrated and the uncoordinated approaches in terms of the achieved makespan



(a) Instances with $W = 45$ (b) Instances with $W = 75$

Figure 4. Comparison between the heuristic algorithm and the MILP solver in terms of the achieved makespan

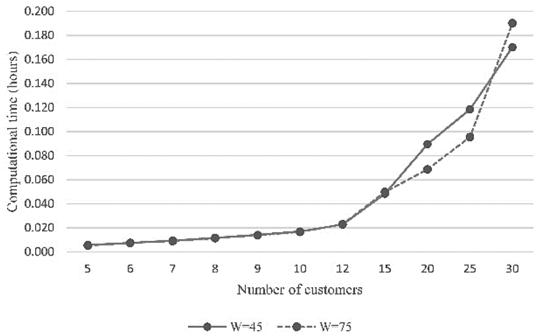


Figure 5. The computational times of solving the integrated model using the heuristic algorithm

7. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

In this paper, a mixed-integer linear programming model is proposed to model the integrated order batching and distribution scheduling problem by considering one order picker and one vehicle in the warehouse. The presented integrated model is compared to the uncoordinated approach in which the order batching and distribution scheduling problems are solved separately and in succession by means of several test instances and the benefit of integration is represented. Experiments represented that although the integrated approach needs much more computational time in comparison to the uncoordinated approach, it leads to significant time saving. A heuristic algorithm based on the route first-cluster method and the shortest path problem is proposed to solve the presented integrated model. Experimental results illustrated that the heuristic algorithm is able to find near-optimal solutions for the small-sized instances with 5 to 8 customer orders. Considering the instances with more than 8 customer orders, solutions obtained by the heuristic algorithm are significantly better than the solutions obtained using the MILP solver within 8 hours.

Extending the model to consider multiple order pickers and multiple vehicles and also Extending the proposed integrated model to on-line context could be directions for future researches.

8. REFERENCES

1. Experimental validations of the learnable evolution model Henn, S., Koch, S., Doerner, K.F., Strauss, C. and Wäscher, G.J.B.R., "Metaheuristics for the order batching problem in manual order picking systems", ***Business Research***, Vol. 3, No. 1, (2010), 82-105.
2. De Koster, M., Van der Poort, E.S. and Wolters, M.J.I.J.o.P.R., "Efficient orderbatching methods in warehouses", ***International Journal of Production Research***, Vol. 37, No. 7, (1999), 1479-1504.
3. De Koster, R.J.I.T.o.E.M., "Distribution strategies for online retailers", ***IEEE Transactions on Engineering Management***, Vol. 50, No. 4, (2003), 448-457.
4. Gong, Y. and De Koster, R.J.I.T., "A polling-based dynamic order picking system for online retailers", ***IIE transactions***, Vol. 40, No. 11, (2008), 1070-1082.
5. Moons, S., Ramaekers, K., Caris, A., Arda, Y.J.F.S. and Journal, M., "Integration of order picking and vehicle routing in a B2C e-commerce context", ***Flexible Services and Manufacturing Journal***, (2017), 1-31.
6. Hajiaghaei-Keshteli, M. and Aminnayeri, M.J.A.S.C., "Solving the integrated scheduling of production and rail transportation problem by keshtel algorithm", ***Applied Soft Computing***, Vol. 25, (2014), 184-203.
7. Nasiri, E., Afshari, A. and Hajiaghaei-Keshteli, M.J.I.J.o.E.-T.C.A., "Addressing the freight consolidation and containerization problem by recent and hybridized meta-heuristic algorithms", ***International Journal of Engineering-Transactions C: Aspects***, Vol. 30, No. 3, (2017), 403-410.
8. Yahyaei, M., Bashiri, M. and Garmeyi, Y.J.I.J.o.E.-T.B.A., "Multi-criteria logistic hub location by network segmentation under criteria weights uncertainty (research note)", ***International Journal of Engineering-Transactions B: Applications***, Vol. 27, No. 8, (2013), 1205-1214.

9. Zhang, J., Wang, X., Huang, K.J.C. and Engineering, I., "Integrated on-line scheduling of order batching and delivery under B2C e-commerce", ***Computers & Industrial Engineering***, Vol. 94, (2016), 280-289.
10. Chen, Z.-L.J.O.r., "Integrated production and outbound distribution scheduling: Review and extensions", ***Operations Research***, Vol. 58, No. 1, (2010), 130-148.
11. Low, C., Chang, C.-M., Gao, B.-Y.J.I.J.o.S.S.O. and Logistics, "Integration of production scheduling and delivery in two echelon supply chain", ***International Journal of Systems Science: Operations and Logistics***, Vol. 4, No. 2, (2017), 122-134.
12. Low, C., Chang, C.-M., Li, R.-K. and Huang, C.-L.J.I.J.o.P.E., "Coordination of production scheduling and delivery problems with heterogeneous fleet", ***International Journal of Production Economics***, Vol. 153, (2014), 139-148.
13. Low, C., Li, R.-K. and Chang, C.-M.J.I.J.o.P.R., "Integrated scheduling of production and delivery with time windows", ***International Journal of Production Research***, Vol. 51, No. 3, (2013), 897-909.
14. Henn, S.J.C. and Research, O., "Algorithms for on-line order batching in an order picking warehouse", ***Computers & Operations Research***, Vol. 39, No. 11, (2012), 2549-2563.
15. Bozer, Y.A. and Kile, J.W.J.I.J.o.P.R., "Order batching in walk-and-pick order picking systems", ***International Journal of Production Research***, Vol. 46, No. 7, (2008), 1887-1909.
16. Gademann, N. and Velde, S.J.I.t., "Order batching to minimize total travel time in a parallel-aisle warehouse", ***IIE Transactions***, Vol. 37, No. 1, (2005), 63-75.
17. Elsayed, E.A.J.T.I.J.o.P.R., "Algorithms for optimal material handling in automatic warehousing systems", ***The International Journal of Production Research***, Vol. 19, No. 5, (1981), 525-535.
18. Gibson, D.R. and Sharp, G.P.J.E.J.o.O.R., "Order batching procedures", ***European Journal of Operational Research***, Vol. 58, No. 1, (1992), 57-67.
19. Henn, S. and Wäscher, G.J.E.J.o.O.R., "Tabu search heuristics for the order batching problem in manual order picking systems", ***European Journal of Operational Research***, Vol. 222, No. 3, (2012), 484-494.
20. Henn, S.J.F.S. and Journal, M., "Order batching and sequencing for the minimization of the total tardiness in picker-to-part warehouses", ***Flexible Services and Manufacturing Journal***, Vol. 27, No. 1, (2015), 86-114.
21. Lin, C.-C., Kang, J.-R., Hou, C.-C., Cheng, C.-Y.J.C. and Engineering, I., "Joint order batching and picker manhattan routing problem", ***Computers & Industrial Engineering***, Vol. 95, (2016), 164-174.
22. Matusiak, M., de Koster, R., Kroon, L. and Saarinen, J.J.E.J.o.O.R., "A fast simulated annealing method for batching precedence-constrained customer orders in a warehouse", ***European Journal of Operational Research***, Vol. 236, No. 3, (2014), 968-977.
23. Menéndez, B., Pardo, E.G., Alonso-Ayuso, A., Molina, E., Duarte, A.J.C. and Research, O., "Variable neighborhood search strategies for the order batching problem", ***Computers & Operations Research***, Vol. 78, (2017), 500-512.
24. Zhang, J., Wang, X., Chan, F.T. and Ruan, J.J.A.M.M., "On-line order batching and sequencing problem with multiple pickers: A hybrid rule-based algorithm", ***Applied Mathematical Modelling***, Vol. 45, (2017), 271-284.
25. Beasley, J.E.J.O., "Route first—cluster second methods for vehicle routing", ***Omega***, Vol. 11, No. 4, (1983), 403-408.
26. Ahuja, R.K., "Network flows: Theory, algorithms, and applications, Pearson Education, (2017).
27. Cervone, G., Kaufman, K.K. and Michalski, R.S., "Experimental validations of the learnable evolution model", in Evolutionary Computation, 2000. Proceedings of the 2000 Congress on, IEEE. Vol. 2, (2000), 1064-1071.
28. Geismar, H.N., Laporte, G., Lei, L. and Sriskandarajah, C.J.I.J.o.C., "The integrated production and transportation scheduling problem for a product with a short lifespan", ***INFORMS Journal on Computing***, Vol. 20, No. 1, (2008), 21-33.
29. Czyzyk, J., Mesnier, M.P., Moré, J.J.J.I.C.S. and Engineering, "The neos server", ***IEEE Computational Science and Engineering***, Vol. 5, No. 3, (1998), 68-75.
30. Dolan, E. D. J.a.p.c., "Neos server 4.0 administrative guide", ***arXiv preprint cs/0107034***, Argonne National Laboratory (2001).
31. Gropp, W., More, J.J.A.t. and optimization, "Optimization environments and the neos server", ***Approximation Theory and Optimization***, (1997), 167-182.

Integrated Order Batching and Distribution Scheduling in a Single-block Order Picking Warehouse Considering S-Shape Routing Policy

Y. Zare Mehrjerdi, M. Alipour, A. Mostafaeipour

Department of Industrial Engineering, Yazd university, Yazd, Iran

P A P E R I N F O

Paper history:
Received 21 November 2017
Received in revised form 12 May 2017
Accepted 17 August 2018

Keywords:
Warehouse
Order Picking System
Order Batching
Picker-to-Part Systems
Distribution Scheduling

چکیده

در این مقاله، یک مدل برنامه ریزی خطی صحیح-مختلط به منظور یکپارچه سازی و بهینه سازی همزمان مسائل برداشت دسته ای سفارشات و توزیع سفارشات در یک انبار برداشت ارائه شده است. یک الگوریتم دو مرحله ای به منظور حل مدل در زمان قابل قبول طراحی شده است. در مرحله ی اول از الگوریتم ارائه شده، از یک الگوریتم ژنتیک به منظور ایجاد و ارزیابی توالی هایی از مشتریان استفاده می شود. در مرحله ی دوم از الگوریتم، از رویکرد مسیریابی اولین خوشه به منظور به دست آوردن یک برنامه ی زمان بندی مؤثر استفاده می شود. نتایج عددی نشان می دهد که یکپارچه در نظر گرفتن این دو مسأله می تواند باعث کاهش چشم گیری در زمان کل پاسخگویی به سفارشات مشتریان شود. همچنین، مشاهداتی که بر روی عملکرد الگوریتم ابتکاری انجام شده است، گزارش شده است

doi: *10.5829/ije.2018.31.10a.15*