Journal of Manufacturing Systems 32 (2013) 335–347

Contents lists available at SciVerse ScienceDirect

Journal of Manufacturing Systems

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

Technical paper

A hybrid simulated annealing algorithm for location and routing scheduling problems with cross-docking in the supply chain

S. Meysam Mousavi*, Reza Tavakkoli-Moghaddam

Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran

ARTICLE INFO

Article history:
Received 24 April 2012
Received in revised form 1 August 2012
Accepted 7 December 2012
Available online 16 January 2013

Keywords:
Supply chain management
Distribution networks
Location of cross-docking centers
Vehicle routing scheduling
Simulated annealing
Tabu search

ABSTRACT

The location and routing scheduling problems with cross-docking can be regarded as new research directions for distribution networks in the supply chain. The aims of these problems are to concurrently design a cross-docking center location and a vehicle routing scheduling model, known as NP-hard problems. This paper presents a two-stage mixed-integer programming (MIP) model for the location of cross-docking centers and vehicle routing scheduling problems with cross-docking due to potential applications in the distribution networks. Then, a new algorithm based on a two-stage hybrid simulated annealing (HSA) with a tabu list taken from tabu search (TS) is proposed to solve the presented model. This proposed HSA not only prevents revisiting the solution but also maintains the stochastic nature. Finally, small and large-scale test problems are randomly generated and solved by the HSA algorithm. The computational results for different problems show that the proposed HSA performs well and converges fast to reasonable solutions.

© 2012 The Society of Manufacturing Engineers. Published by Elsevier Ltd. All rights reserved.

1. Introduction

The location and routing scheduling problems with cross-docking are recognized new areas of research, which take account of two main components of cross-docking distribution networks, namely cross-docking centers location and vehicle routing scheduling. The location and routing scheduling problem involves the strategic (i.e., location) and tactical/operational (i.e., routing scheduling) decision levels in supply chain management. In the earlier studies, they are regarded as interdependent components. Focusing separately on two components has limitations on the distribution networks design [1,2]; hence, it is important to tackle these components concurrently, aiming at efficiently servicing customers (i.e., pickup nodes) from a well-selected set of cross-docking centers location.

Generally speaking, the location and routing problems have been widely employed in a variety of problems in distribution networks for real-life applications, for instance, bill delivery [3], medical evacuation [4] and waste collection [5]. The location and routing scheduling with cross-docking are NP-hard problems [e.g., 6–9]. Heuristics and meta-heuristics are only recommended as viable solving approaches by increasing the size of these problems or considering real-world cases.

Cross-docking distribution networks have attracted strong interest among researchers in the last decade. Jayaraman [10] considered the conventional warehouse problem and provided a number and location of warehouses, and then allocated the customers at a minimum cost without violating the capacity restriction on warehouses. Donaldson et al. [11] concentrated on a schedule-driven transportation planning in the cross-docking distribution networks design. Jayaraman and Ross [6] presented a practical approach for solving a multi-product multi-echelon problem to distribution network design by the simulated annealing (SA) algorithm. Li et al. [12] addressed a cross-docking center operation in order to eliminate or minimize storage and order picking activity in the cross-docking by using just-in-time (JIT) scheduling. Their problem was then converted into a machine scheduling problem. Lim et al. [13] developed the traditional transshipment problem that consisted of a number of supply, transshipment and demand nodes. Lee et al. [8] were the first who presented an integration model of cross-docking centers problem with vehicle routing scheduling for the distribution network design. Reeves [14] presented two case studies by considering the supply chain governance and illustrated two contrasting approaches to the provision of cross-docking services in the automotive industry.

Ross and Jayaraman [7] addressed an evaluation of meta-heuristics for the location of cross-docking centers in cross-docking distribution networks in the supply chain. Bachlaus et al. [15] suggested an integrated multi-echelon agile supply chain network. The problem was formulated as a multi-objective mathematical

* Corresponding author. Tel.: +98 21 82084183.
E-mail address: sm.mousavi@ut.ac.ir (S.M. Mousavi).

0278-6125/$ – see front matter © 2012 The Society of Manufacturing Engineers. Published by Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jmsy.2012.12.002

programming model in order to minimize the fixed and variable costs and to maximize the plant flexibility and volume flexibility. Liao et al. [16] proposed a tabu search (TS) algorithm to solve a model that considered the vehicle routing scheduling problem with the single cross-docking center in order to transport goods from supplies to retailers. Musa et al. [17] presented an ant colony optimization (ACO) heuristic to solve the transportation problem in the cross-docking distribution networks. The study illustrated that the proposed heuristic provided appropriate results in the reasonable time. Yang et al. [18] investigated the decisions for transporting freight between inbound and outbound trailers in a cross-dock by using simulation. Liao et al. [19] proposed two hybrid differential evolution algorithms for optimal inbound and outbound truck sequencing in the operations of cross-docking center. Melo et al. [20] considered the problem of redesigning a supply chain network with multiple echelons and commodities, and modeled as a large-scale mixed-integer linear program. Then, they proposed a TS heuristic for solving the presented model. Ma et al. [21] focused on a new shipment consolidation and transportation problem in cross-docking distribution networks by considering setup cost and time window constraint. Alpan et al. [22] addressed transshipment problem in a multi-door cross-docking warehouse, and made an attempt to find the best schedule of transshipment operations in order to minimize the sum of inventory holding and truck replacement costs. Dondo et al. [23] presented a hybrid multi-echelon multi-item distribution network that contained multi-echelon vehicle routing problem with cross-docking in supply chain management by minimizing total transportation cost.

Going through the literature indicates that the location of cross-docking centers and vehicle routing scheduling problems simultaneously have not been taken into consideration for the distribution networks in the supply chain management. In fact, these problems can be interrelated components to address most concerns of logistic managers in numerous real-life applications. Also, an integration of location and routing scheduling can help the managers to achieve significant productivity gains by making the decisions in strategic and tactical/operational levels for the planning of cross-docking distribution networks.

This paper presents a two-stage mixed-integer programming (MIP) model for the location of cross-docking centers and vehicle routing scheduling problems for the cross-docking distribution networks in the supply chain. The objective functions are to minimize the fixed costs, total transportation costs in the pickup and delivery processes, operational costs of vehicles and penalty costs for total earliness and tardiness deliveries to customers. Then, this paper proposes a new two-stage hybrid simulated annealing (HSA) algorithm embedded with TS that characterizes a special solution representation for the location of cross-docking centers and vehicle routing scheduling in the distribution systems. Finally, the computational results indicate that the proposed HSA performs well on small and large-sized test problems in terms of objective function values and CPU times.

Unlike the previous studies in the literature of the cross-docking, this paper pays special attentions not only in presenting an effective framework by a combination of two location of cross-docking centers and vehicle routing scheduling problems via formulating a new MIP model, but also in solving jointly the location and routing scheduling in two decision levels of the cross-docking systems via an efficient hybrid algorithm. From the problem-solving viewpoint, this paper carefully designs a new hybrid meta-heuristic algorithm, aiming at benefiting from the main advantages of two well-known algorithms (i.e., SA and TS) concurrently to reach a near-optimal solution with minimal iterations. By using the proposed HSA, a number of solution revisits can be decreased by providing a short-term memory suggested by a tabu list while keeping the stochastic nature of the SA algorithm.

The structure of this paper is organized in six sections. In the next section, the location and routing scheduling problems with cross-docking are defined. Section 3 introduces the proposed two-stage MIP model formulation for the cross-docking distribution networks in the supply chain. The proposed HSA meta-heuristic algorithm as the problem-solving approach is presented in Section 4. Then, computational results are discussed in Section 5. Finally, the remarkable conclusion is provided in Section 6.

2. Problem definition

A cross-docking center is an intermediate node in distribution networks in order to decrease inventory while satisfying customers' requirements. Through cross-docking, different goods are delivered to the center by inbound vehicles. They are immediately consolidated out based on the destinations and then shipped to outbound vehicles for delivery to customers for a short time aiming at eliminating inventory storage [24]. Hence, the most costly component of conventional warehousing can be reduced. Indeed, the cross-docking is introduced as a new logistic strategy for companies involving in the retail, grocery, food and drink distributions industries in recent year [16,25,26]. Fig. 1 illustrates the concept of proposed cross-docking distribution network, in which two main nodes (i.e., pickup and delivery nodes) are simultaneous arrival and consolidation. The distribution network discussed in this paper is a single period, single product, multi-echelon logistic network including suppliers, cross-docking centers and customers.

The location of cross-docking centers and vehicle routing scheduling problems in the distribution network can be stated as follows: a set of customers with known demand and a set of potential cross-docking centers are provided. The location of the cross-docking centers is determined in the first stage. The product should be delivered to customers via cross-docking centers. The shipment of each customer demand is met by potential vehicles in the delivery process that are dispatched from cross-docking centers, and operate on routes involving multiple customers. Then, the vehicle routing scheduling from the cross-docking centers is obtained in the pickup and delivery processes in order to minimize the sum of the total costs through the cross-docking centers. In the first stage, fixed costs associated with opening cross-docking centers at potential sites are regarded along with transportation costs for the movement of the product from the suppliers to cross-docking centers, and from cross-docking centers to the customers. In addition, in the second stage distribution costs associated with the routing of vehicles containing operational costs of vehicles and transportation costs are taken into consideration.

In sum, the distribution and routing scheduling plan with multiple cross-docking centers can be designed so that the demand of each customer can be satisfied. Each customer is served by only one vehicle. The number of the vehicles in the pickup and delivery processes is limited. Moreover, it is assumed that all vehicles are located in the multiple cross-docking centers, and split pickup and delivery are not allowed. The total demand on each route is less than or equal to the capacity of each vehicle assigned to that route. Each route starts and ends at the same cross-docking centers. Also, the total quantity of pickup should be equal to the quantity to be delivered. Finally, the aim of proposed two-stage MIP model is to obtain the minimum number of cross-docking centers among a discrete set of location sites in the first stage. Then, the aim of the second stage is to obtain the number of vehicles and the best route as well as the arrival time of each vehicle in the distribution network with multiple cross-docking centers.

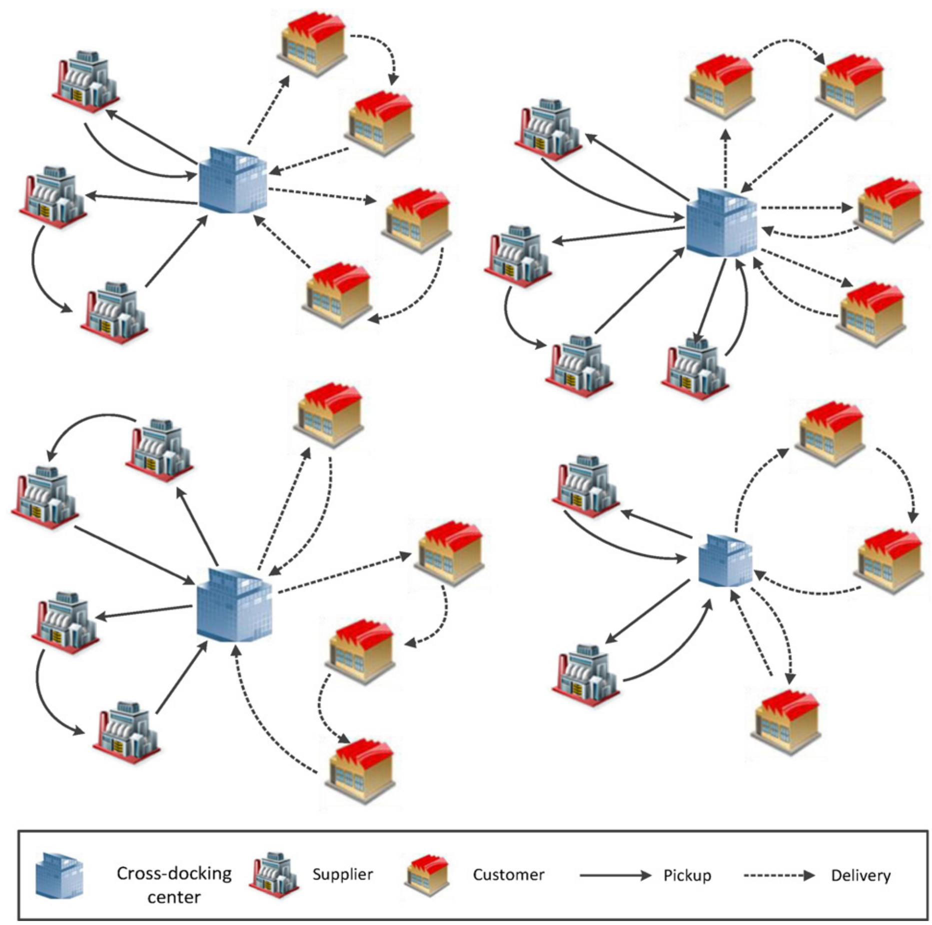

Fig. 1. Proposed cross-docking distribution network in the supply chain.

3. Proposed two-stage MIP model formulation

To describe two-stage MIP model in this section, the notations, input parameters and decision variables are presented as follows:

Sets and input parameters:

P: set of suppliers in the pickup process
O: set of cross-docking centers
D: set of customers in the delivery process
$D_{i'}$: demand of customer i'.
CA_p: capacity of cross-docking center p to handle product
S_i: quantity of product from supplier i
F_p: fixed operating cost to open cross-docking center p
c_{ip}: cost to transport product from supplier i to cross-docking center p
$c_{pi'}$: cost to transport product from cross-docking center p to customer i'.
TC: maximum total cost that could pay for opening cross-docking centers
K: number of available vehicles in the pickup process
K': number of available vehicles in the delivery process
Q: maximum capacity of each vehicle
p_i: loaded amount of product in node i in the pickup process
$d_{i'}$: unloaded amount of product in node i' in the delivery process
c_{ij}: transportation cost from node i to node j in the pickup process
$c_{i'j'}$: transportation cost from node i' to node j' in the delivery process
c_k: operational cost of the vehicle k
$c_{k'}$: operational cost of the vehicle k'.
d_{ij}: distance from node i to node j in the pickup process
$d_{i'j'}$: distance from node i' to node j' in the delivery process
t_i^k: length of a visit for the vehicle k in node i in the pickup process
$t_{i'}^{k'}$: length of a visit for the vehicle k' to node i' in the delivery process
et_{ij}: time for the vehicle to move from node i to node j in the pickup process
$et_{i'j'}$: time for the vehicle to move from node i' to node j' in the delivery process
$E_{i'}$: total earliness delivery penalty demanded by customer in node i' in delivery process
$L_{i'}$: total tardiness delivery penalty demanded by customer in node i' in delivery process
$\propto_{i'}$: penalty unit of early delivery from customer in node i' in delivery process
$\beta_{i'}$: penalty unit of tardy delivery from customer in node i' in delivery process
$du_{i'}$: due-date demanded from customer in node i' in delivery process

Decision variables:

$$x_{ip} = \begin{cases} 1 & \text{if supplier } i \text{ is assigned to cross-docking center } p \text{ for product,} \\ 0 & \text{otherwise,} \end{cases}$$

$$y_{pi'} = \begin{cases} 1 & \text{if cross-docking center } p \text{ is assigned to customer } i' \text{ for product,} \\ 0 & \text{otherwise,} \end{cases}$$

$$z_p = \begin{cases} 1 & \text{if cross-docking center } p \text{ is open,} \\ 0 & \text{otherwise,} \end{cases}$$

$$x_{ij}^{k} = \begin{cases} 1 & \text{if vehicle } k \text{ transports product from node } i \text{ to node } j \text{ in the pickup process,} \\ 0 & \text{otherwise,} \end{cases}$$

$$x_{i'j'}^{k'} = \begin{cases} 1 & \text{if vehicle } k' \text{ transports product from node } i' \text{ to node } j' \text{ in the delivery process,} \\ 0 & \text{otherwise,} \end{cases}$$

y_{ij}: transported amount of product from node i to node j in the pickup process

$z_{i'j'}$: transported amount of product from node i' to node j' in the delivery process

DT_i^k: departure time of vehicle k from node i in the pickup process

$DT_{i'}^{k'}$: departure time of vehicle k' from node i' in the delivery process

DT_j^k: departure time of vehicle k from node j in the pickup process

$DT_{j'}^{k'}$: departure time of vehicle k' from node j' in the delivery process

AT_j^k: arrival time of vehicle k at node j in the pickup process

$AT_{j'}^{k'}$: arrival time of vehicle k' at node j' in delivery process

AT_p^k: arrival time of vehicle k at cross-docking center p in the pickup process

$AT_{p'}^{k'}$: arrival time of vehicle k' at cross-docking center p' in the delivery process

3.1. Cross-docking centers location (stage 1)

The above notations are used in the formulation of the proposed MIP model for the cross-docking centers location problem in the first stage. The location problem can be formulated as below:

$$Min Z_1 = \sum_{p=1}^{O} F_p z_p + \sum_{i=1}^{n}\sum_{p=1}^{O} c_{ip} x_{ip} + \sum_{p=1}^{O}\sum_{i'=1}^{m} c_{pi'} y_{pi'} \tag{1}$$

Subject to:

$$\sum_{p=1}^{O} y_{pi'} = 1, \quad \forall i' \tag{2}$$

$$\sum_{p=1}^{O} x_{ip} = 1, \quad \forall i \tag{3}$$

$$\sum_{i=1}^{n} S_i x_{ip} \le CA_p, \quad \forall p \tag{4}$$

$$\sum_{i'=1}^{m} D_{i'} y_{pi'} \le CA_p, \quad \forall p \tag{5}$$

$$x_{ip} \le z_p, \quad \forall i,\ p \tag{6}$$

$$y_{pi'} \le z_p, \quad \forall i',\ p \tag{7}$$

$$\sum_{p=1}^{O} F_p z_p \le TC \tag{8}$$

$$x_{ip},\ y_{pi'},\ z_p \in \{0, 1\}, \quad \forall i, i', p \tag{9}$$

The objective function (1) minimizes fixed costs to open cross-docking centers and costs to the movement of the product from suppliers to cross-docking centers in the pickup process as well as costs to supply the product from cross-docking centers to meet the demand of customers in the delivery process. Constraint (2) assures that the demand of each customer in the delivery process is only met by one of the open cross-docking centers. Constraint (3) assures that the quantity of the product from each supplier is only satisfied by one of the open cross-docking centers. Constraint (4) assures that the quantity of product from each supplier should be equal/less than capacities of open cross-docking centers in the pickup process. Constraint (5) represents that the demand of each customer should be equal/less than capacities of open cross-docking centers in the delivery process. Constraints (6) and (7) assure that the movement of the product from suppliers to cross-docking center, and from cross-docking center to customers in the pickup and delivery processes can be conducted only when the corresponding cross-docking center is open. Constraint (8) considers a limitation on total cost that can be paid for opening cross-docking centers. Constraint (9) defines corresponding decision variables of the model.

3.2. Vehicle routing scheduling (stage 2)

The above notations are used in the formulation of the proposed MIP model for the vehicle routing scheduling problem with multiple cross-docking centers in the second stage. The route scheduling problem can be formulated as below:

$$\begin{aligned} Min Z_2 = {} & \sum_{i\in P}\sum_{j\in (P\cup O)}\sum_{k\in K}(c_{ij}d_{ij})x_{ij}^{k} + \sum_{i\in O}\sum_{j\in P}\sum_{k\in K}[(c_{ij}d_{ij}) + c_k]x_{ij}^{k} + \\ & \sum_{i'\in D}\sum_{j'\in (D\cup O)}\sum_{k'\in K'}(c_{i'j'}d_{i'j'})x_{i'j'}^{k'} + \sum_{i'\in O}\sum_{j'\in D}\sum_{k'\in K'}\left[(c_{i'j'}d_{i'j'}) + c_{k'}\right]x_{i'j'}^{k'} \\ & + \sum_{i'\in D}\sum_{j'\in D}\sum_{k'\in K'}(E_{i'} + L_{i'})x_{i'j'}^{k'} \end{aligned} \tag{10}$$

Subject to:

$$\sum_{i\in (P\cup O)}\sum_{k\in K} x_{ij}^{k} = 1, \quad \forall j \tag{11}$$

$$\sum_{j\in (P\cup O)}\sum_{k\in K} x_{ij}^{k} = 1, \quad \forall i \tag{12}$$

$$\sum_{i'\in (D\cup O)}\sum_{k'\in K'} x_{i'j'}^{k'} = 1, \quad \forall j' \tag{13}$$

$$\sum_{j'\in (D\cup O)}\sum_{k'\in K'} x_{i'j'}^{k'} = 1, \quad \forall i' \tag{14}$$

$$\sum_{i\in (P\cup O)}\sum_{k\in K} x_{ij}^{k} \ge 1, \quad \forall j \in O \tag{15}$$

$$\sum_{i'\in (D\cup O)}\sum_{k'\in K'} x_{i'j'}^{k'} \ge 1, \quad \forall j' \in O \tag{16}$$

$$\sum_{i\in (P\cup O)} x_{ir}^{k} = \sum_{j\in (P\cup O)} x_{rj}^{k}, \quad \forall k, r \in (P\cup O) \tag{17}$$

$$\sum_{i'\in (D\cup O)} x_{i'r'}^{k'} = \sum_{j'\in (D\cup O)} x_{r'j'}^{k'}, \quad \forall k',\ r \in (D\cup O) \tag{18}$$

$$\sum_{i\in (P\cup O)}\sum_{j\in (P\cup O)} x_{ij}^{k} \le 1, \quad \forall k \tag{19}$$

$$\sum_{i' \in (D \cup O)} \sum_{j' \in (D \cup O)} x_{i'j'}^{k'} \leq 1, \quad \forall k' \tag{20}$$

$$\sum_{i \in (P \cup O)} \sum_{j \in (P \cup O)} \sum_{k \in K} x_{ij}^{k} \leq K \tag{21}$$

$$\sum_{i' \in (D \cup O)} \sum_{j' \in (D \cup O)} \sum_{k' \in K'} x_{i'j'}^{k'} \leq K' \tag{22}$$

$$y_{ij} \leq Q \quad \forall i, j \in (P \cup O) \tag{23}$$

$$z_{i'j'} \leq Q \quad \forall i', j' \in (D \cup O) \tag{24}$$

$$\sum_{i \in P} p_i = \sum_{i' \in D} d_{i'} \tag{25}$$

$$y_{jr} = y_{ij} - p_j \qquad if\ j \in P, \forall i, r \tag{26}$$

$$y_{jr} = y_{ij} - \sum_{i \in P} p_i \qquad if\ j \in O, \forall j, r \tag{27}$$

$$z_{i'j'} = z_{j'r'} + d_{j'} \qquad if\ j' \in D, \forall i', r' \tag{28}$$

$$z_{i'j'} = z_{j'r'} + \sum_{i' \in D} d_{i'} \qquad if\ j' \in O, \forall i', r' \tag{29}$$

$$u_i - u_j + n \sum_{k \in K} x_{ij}^{k} \leq (n-1), \quad \forall i, j \tag{30}$$

$$u_{i'} - u_{j'} + m \sum_{k' \in K'} x_{i'j'}^{k'} \leq (m-1), \quad \forall i', j' \tag{31}$$

$$DT_j^k \geq (et_{ij} + DT_i^k + t_i^k) x_{ij}^k, \quad \forall k, i, j \tag{32}$$

$$DT_{j'}^{k'} \geq (et_{i'j'} + DT_{i'}^{k'} + t_{i'}^{k'}) x_{i'j'}^{k'}, \quad \forall k', i', j' \tag{33}$$

$$AT_j^k \geq (DT_i^k + et_{ij}) x_{ij}^k, \quad \forall k, i, \forall j \in O \tag{34}$$

$$AT_{j'}^{k'} \geq (DT_{i'}^{k'} + et_{i'j'}) x_{i'j'}^{k'}, \forall k', i', \forall j' \in O \tag{35}$$

$$AT_p^k = AT_{p'}^{k''}, \quad \forall k \neq k'', \quad \forall p \neq p' \tag{36}$$

$$AT_p^{k'} = AT_{p'}^{k'''}, \qquad \forall k' \neq k''', \forall p \neq p' \tag{37}$$

$$E_{i'} = \max(0; du_{i'} - AT_{i'}^{k'}) \propto_{i'}, \quad \forall i' \in D \text{ and } \forall k' \in K' \tag{38}$$

$$L_{i'} = \max(0; AT_{i'}^{k'} - du_{i'}) \beta_{i'}, \qquad \forall i' \in D \text{ and } \forall k' \in K' \tag{39}$$

$$u_i, u_j \leq n, \qquad \forall i, j \tag{40}$$

$$u_{i'}, u_{j'} \leq m, \qquad \forall i', j' \tag{41}$$

$$x_{ij}^k, x_{i'j'}^{k'}, x_{pj}^{k'}, x_{pj'}^{k'}, x_{ip}^k, x_{i'p}^{k'} \in \{0, 1\}, \qquad \forall i, j, i', j', p, k, k' \tag{42}$$

$$y_{ij}, z_{i'j'}, DT_i^k, DT_{i'}^{k'}, DT_j^k, DT_{j'}^{k'}, AT_p^k, AT_{p'}^{k'}, u_i, u_j, u_{i'}, u_{j'} \geq 0, \quad \forall i, j, i', j', p, p', k, k' \tag{43}$$

The objective function (10) minimizes total transportation costs associated with moving product in the pickup and delivery processes, operational cost of each vehicle in these processes separately, and penalty costs for the total earliness and tardiness deliveries to customers in the delivery process. Constraints (11) and (12) represent that one vehicle has to arrive and leave one node in the pickup process. Constraints (13) and (14) represent that one vehicle has to arrive and leave one node in the delivery process. Constraints (15) and (16) consider that every supplier or customer belongs to one and only one route, but cross-docking centers may belong to more than one route. Constraints (17) and (18) represent the consecutive movement of vehicles. Constraints (19) and (20) consider whether or not a vehicle arrives and leaves a cross-docking center in the pickup and delivery processes. Constraints (21) and (22) consider that the numbers of vehicles that arrive or leave a cross-docking center in the pickup or delivery processes should be less than the number of available vehicles. Constraints (23) and (24) enforce that the quantity of loaded product in a vehicle cannot exceed the maximum capacity of the vehicle. The flow conservation for product is represented in constraint (25). The quantity of products between nodes is taken into consideration in the pickup and delivery processes in constraints (26)–(29). Constraints (30) and (31) assure that every customer is on a route connected to the set of cross-docking centers. Constraints (32) and (33) illustrate that the departure time of a vehicle from a node is determined by the sum of the arrival time at a node, the length of a visit, and time to move in the pickup and delivery processes. The arrival time at a cross-docking center is represented in constraints (34) and (35) for the pickup and delivery processes. The constraints for simultaneous arrival to a cross-docking center are provided in Eqs. (36) and (37). Constraints (38) and (39) represent the calculations for the earliness and tardiness penalties in the delivery process. Constraints (40)–(43) enforce the integrality restrictions on the corresponding decision variables of the model.

4. A two-stage HSA meta-heuristic algorithm for solving the location and routing scheduling

The proposed HSA meta-heuristic is introduced for the location and routing scheduling problems based on the hybridization of two famous algorithms, namely SA and TS. The proposed HSA has a number of advantages, including stochastic feature avoiding cycling and tabu list to escape from local optima. These characteristics limit the search from a previously visited solution and improve the performance of the conventional SA remarkably.

The SA is regarded as a random search optimization algorithm. The SA was first introduced by Metropolis et al. [27] and popularized by Kirkpatrick et al. [28]. The algorithm works based on the annealing process that is applied to the metallurgical industry. The SA utilizes a stochastic approach to direct the search. The algorithm escapes from local optima via receiving non-improver solutions with a certain probability in each temperature. This algorithm has been widely applied to numerous complicated combinatorial optimization problems in real-life situations [e.g., 2,29–31].

The TS is regarded as a local search algorithm that is applied to combinatorial optimization. The TS was first introduced by Glover [32]. The algorithm is able to escape the local optima occurred during the search via the list of prohibited neighboring solutions, called tabu list. The TS has been used in a wide variety of conventional and practical optimization problems in real-life applications [e.g., 8,16,33].

In the following sections, the proposed HSA meta-heuristic algorithms supplementary with the tabu lists are described for the two-stage MIP model in detail for the location of cross-docking centers and vehicle routing scheduling in the cross-docking distribution network. The core of the HSA algorithm is originally based on the meta-heuristics presented in [6,33,34]. Readers for more details may refer to [e.g., 6,7,34–36] in the literature of supply chain.

4.1. Proposed meta-heuristic for the location (stage 1)

In this section, the steps of the proposed HSA algorithm with a tabu list taken from TS are described for the location problem. The search is conducted for least-cost solutions by a control parameter, called temperature, and the cooling schedule that determines the number of iterations (epochs) for the algorithm. Randomly generated initial configuration is first regarded in the proposed hybrid algorithm that denotes the cross-docking centers to be opened, the suppliers and customers assigned to the cross-docking centers. Then, the total cost is calculated by the objective function in the first stage of proposed mathematical model.

Step 1: Initialization. Initial and final values are taken into account for the control parameter temperature, known as T_0 and T_f respectively; i is the number of a particular iteration and N is the total number of iterations. An initial cross-docking center solution is randomly obtained by allocating adequate supply of suppliers and demand flows of customers between cross-docking centers and delivery nodes in the cross-docking distribution network. It leads to an initial feasible solution that involves the product flows. The value of objective function for the solution can be regarded as the value of objective function for the best configuration obtained (BS), current configuration $OBF(x_c)$ and the newest configuration $OBF(x_a)$. All counters are set to 1.

Step 2: Check feasibilities. The algorithm investigates product flow assignments for cross-docking centers to assure the capacity of cross-docking, fixed costs and number of potential cross-docking centers. Furthermore, the quantity of product and demand of customer should be considered to be met. If the configuration is not feasible, we return to step 1 [6].

Step 3: Provide a feasible neighboring solution. Once the network design problem has been initialized, a value of objective function is calculated and feasibility is considered. Then, the current feasible configuration of cross-docking distribution network is updated by choosing a supplier and reassigning the amount of product between a cross-docking center and supplier. Also, this method can be utilized for customers. It is executed by randomly choosing a supplier and a customer to perturb. Its flow is randomly allocated to another combination of pickup/cross-docking center/delivery nodes. All feasibilities must be investigated once again. Finally, the value of objective function is obtained for the neighboring solution $OBF(x_a)$.

Step 4: Assess current solution with neighboring solution. If the value of objective function for the neighboring solution is higher than the current solution ($OBF(x_a) > OBF(x_c)$), proceed to step 5. Otherwise, if the value of objective function for the newest configuration enhances the current solution ($OBF(x_a) < OBF(x_c)$), the neighboring solution can be regarded as the current solution. Then, this solution is compared to the best solution obtained (BS). If the value of objective function for the newest configuration is lower than the best one determined so far ($OBF(x_a) < BS$). Then replace the best solution with this neighboring solution. Proceed to step 8.

Step 5: Investigate Metropolis condition. The difference between the neighboring solution and the current solution is calculated, $\Delta cost = OBF(x_a) - OBF(x_c)$. Then, the Metropolis criterion is employed to obtain the probability, in which the relatively inferior neighboring solution can be accepted, $P(A)$. This probability is calculated by [6]:

$$P(A) = \exp\left(\frac{\Delta \cos t}{T_i}\right), \tag{44}$$

where T_i is the present temperature. Then, a random number is determined from the interval (0, 1). If the random number is lower than $P(A)$, then the neighboring solution is substituted for the current solution. Proceed to step 8.

Step 6: Tabu list. Tabu list can investigate for each step of the algorithm whether the obtained solution is latterly visited or not. Hence, this leads to the restriction of the algorithm regarding revisiting the pre-visited solutions. This characteristic of the HSA decreases the CPU time of algorithm to achieve reasonable solutions.

Step 7: Aspiration. Aspiration is linked to TS. It makes an attempt to restrict the search of the algorithm from being trapped at a solution which is surrounded by tabu neighbors. If an obtained solution has a neighborhood of the tabu solutions, the solution via the value of objective function higher than the aspiration is selected for further exploring [33].

Step 8: Increase counters. Memory and variables are updated. The counters can be incremented by one. If the iteration counter value is lower than or equal to the maximum iterations for the temperature level, then return to step 3. Otherwise, go to step 9.

Step 9: Adjust temperature. Temperature is adapted in iteration i using the cooling schedule:

$$T_i = \frac{1}{2}(T_0 - T_f)\left(1 - \tan h\left(\frac{10i}{N} - 5\right)\right) + T_f. \tag{45}$$

If the new value of T_i is higher than or equal to the stopping value (T_f), then iteration counters are restarted from one and return to step 3. Otherwise, the procedure stops.

4.2. Proposed meta-heuristic for the routing scheduling (stage 2)

In this section, the steps of the proposed HSA algorithm supplementary with tabu list are explained for the routing scheduling problem.

Step 1: Initialization. The path representation is applied to encode the solution of the vehicle routing scheduling problem with cross-docking centers in the distribution network. The idea of the path representation is that the suppliers and customers are listed in the order, in which through the cross-docking they are visited in the pickup and delivery processes. For instance, suppose that there are eight suppliers numbered 1–8. If the path representation is [082730410650], then three routes are needed to visit all these eight suppliers in the pickup process. In the first route, a vehicle starts from the cross-docking center, indicated as 0, travels to suppliers 8, 2, 7 and finally supplier 3. After that, the vehicle returns to the cross-docking center. In the second route, the vehicle begins with supplier 4 and then supplier 1. Similarly, the vehicle travels back to the cross-docking center after serving the suppliers. In the third route, the vehicle begins with supplier 6 and then supplier 5.

In the same way, the procedure can be utilized for the delivery process. For instance, suppose that there are seven customers numbered 1–7. If the path representation is [0713604520], then two routes are needed to serve all these seven customers in the delivery process. In the first route, a vehicle begins with the cross-docking center, indicated as 0, then travels to customers 7, 1, 3 and finally customer 6. After that, the vehicle returns to the cross-docking center. In the second route, in the delivery process, the vehicle starts with customer 4, then customer 5 and finally customer 2. Similarly, the vehicle travels back to the cross-docking center after serving three customers. It is worth to note that each solution contains O links if there exist O cross-docking centers in the vehicle routing scheduling problem. For this problem in the step of initialization, there are three sub-steps to provide a feasible initial solution. The first sub-step is to assign suppliers/customers to each of the O cross-docking centers or links, that is, the grouping problem. There are a number of cross-docking centers, suppliers and customers in the distribution network. Each supplier/customer should be allocated to one cross-docking center or link. Suppliers and customers are assigned to the cross-docking center by considering minimum opening and distribution costs due to the objective function of the proposed model with minimizing total opening and distribution costs. The second sub-step is to assign suppliers/customers in the same link to different routes by the saving method presented in [34,37]. This method builds a saving matrix for each of two suppliers/customers in the same link. Then, the suppliers/customers with large saving value can be grouped in the same route in which the vehicle capacity restriction and arrival time restriction are not violated. The third sub-step is to solve the scheduling problem by the NNH method presented in [34,35]. The principle of the NNH is to randomly begin with the first supplier and customer. Then, the

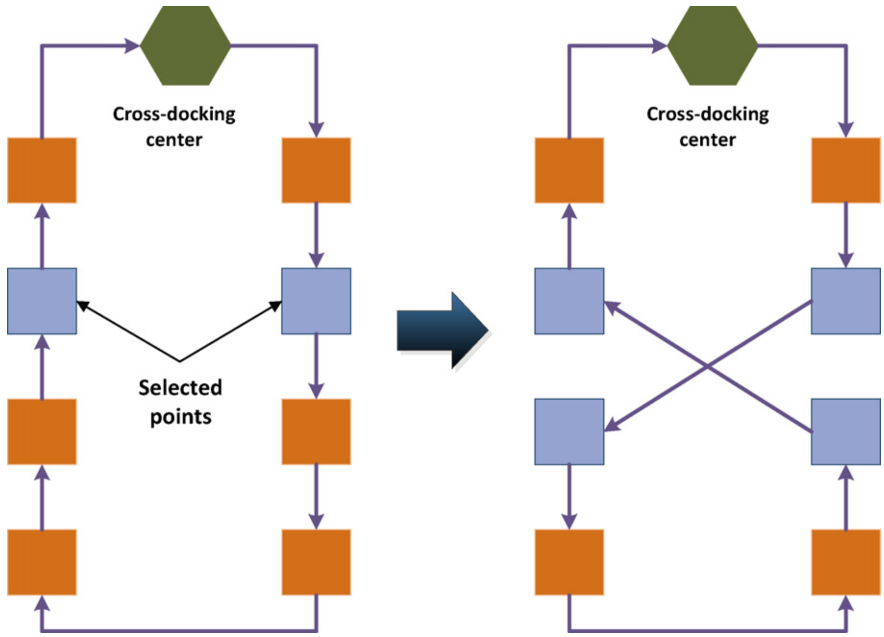

Fig. 2. 2-opt exchange operation for the second stage.

next supplier/customer is chosen and regarded as minimum cost to the previous one from those unselected suppliers/customers in order to build the pickup and delivery sequence until all suppliers and customers are taken into consideration.

Step 2: Improvement. This procedure is based on the combination of SA and TS to enhance the best solution determined at each step of the algorithm. The hybrid algorithm is presented as follows:

For $i = 1 - n$ do
For $i' = 1 - m$ do

(a) Initialize max-iterations, initial temperature. Set count = 1, T_0 = temp-start.
(b) Let the best solution determined in the initialization step be called the current solution, x_c.

Compute objective function value for current solution, $OBF(x_c)$. After obtaining initial solutions, one of the following steps is repeated for improving the initial solution.

- 2-opt exchange operator

The 2-opt operator can be employed to enhance a single route. This can exchange the route direction between two sequential pickup or delivery nodes. If the cost function associated with the route is enhanced, then the modified route is preserved; otherwise, the route returns to the last condition. For instance, the 2-opt exchange operation in pickup node is depicted in Fig. 2 [30]. This procedure can be conducted similarly for delivery nodes.

- Insertion method

Suppose a route at random, then select as the max $[0.1 \times$ (*length of the route*), 2] nodes in each route, and alter the integer number of selected pickup and delivery nodes randomly in the bound of [0, n] and [0, m] in order to alter the vehicle that services the chosen node. In other words, a node can be departed from one route and it can be added to another route by the insertion method. All feasibilities can be investigated once again. Finally, let neighboring solution be called the adjacent solution, x_a compute objective function for adjacent solution, $OBF(x_a)$.

(c) If $OBF(x_a) < OBF(x_c)$
Then set $x_c = x_a$
Else
Set $\Delta = OBF(x_a) - OBF(x_c)$;.
Set T = temp-start/log(1 + count);
With probability $e^{-\Delta/T}$ set $x_c = x_a$.
Increment count by 1.
(d) If count < max-iteration, go to step (b).

The annealing schedule employed in step (c) of the above algorithm is based on [34,38].

Step 3: Tabu list. Tabu list can investigate for each step of algorithm whether the obtained solution is latterly visited or not. Hence, this leads to the restriction of the algorithm regarding revisiting the pre-visited solutions. This characteristic of the HSA decreases the CPU time of algorithm to achieve reasonable solutions.

Step 4: Aspiration. Aspiration is linked to TS. It makes an attempt to restrict the search of the algorithm from being trapped at a solution which is surrounded by tabu neighbors. If an obtained solution has a neighborhood of the tabu solutions, the solution via the value of objective function higher than aspiration is selected for further exploring [33].

Step 5: Stopping criterion. The stopping condition is investigated in this step that can be regarded as the maximum number of iterations of the algorithm. If the number of iteration is higher than the predefined maximum number, the search process stops; otherwise, the procedure restarts from step 2.

Figs. 3 and 4 depict solution representations of an example with two cross-docking centers and visual illustrations by the presented HSA meta-heuristic algorithm for the second stage of the proposed mathematical model.

5. Computational results

Computational tests in this section are generated to verify and evaluate the performance of proposed HSA meta-heuristic algorithm for solving the proposed two-stage MIP model for the location of cross-docking centers and vehicle routing scheduling problems in the distribution network. For this purpose, fourteen test problems in the supply chain environment with varying sizes generated at random in small and large-scale cases. Hence, seven test

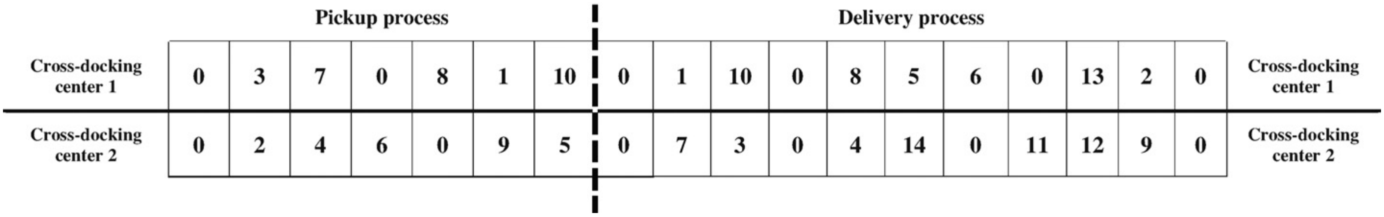

Fig. 3. An example of solution representations with two cross-docking centers.

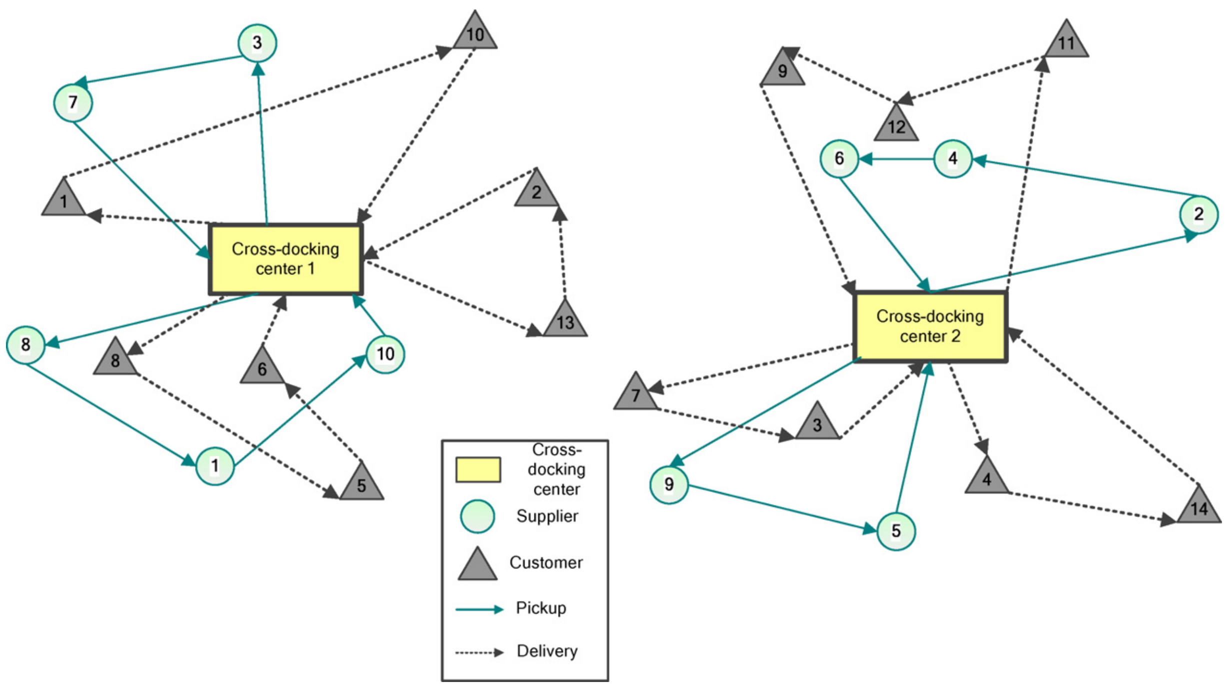

Fig. 4. Visual illustrations of the example solution with two cross-docking centers.

problems are solved in small-sizes by the exact method with the aid of the GAMS software for two stages of the proposed model, including cross-docking center locating in stage one and vehicle route scheduling with multiple cross-docking centers in stage two. Sizes of the test problems are given in Table 1. All parameters are given in Tables 2 and 3 for the first and second stages of the proposed location and routing scheduling model with cross-docking. Some parameters are generated randomly in uniform distributions. It is noteworthy that the problem-solving approach by the presented HSA meta-heuristic algorithm is coded in the MATLAB®. All small and large-sized test problems are run by using the Intel Dual Core, 2.8 GHz compiler and 2 GB of RAM

For seven small-sized test problems, the reported results in these tables are calculated by Eq. (46) which denotes the gap between the optimal solutions and meta-heuristic solutions obtained by the proposed HSA.

$$\frac{obj_{meta\text{-}heuristic} - obj_{optimal\,solution}}{obj_{optimal\,solution}} \times 100. \tag{46}$$

Also, the objective function values, CPU times and the gaps of objective function values are reported in Tables 4 and 5 for the given seven small-sized problems that are solved by the proposed HSA meta-heuristic algorithm, conventional SA algorithm and exact method with the aid of the GAMS software. The comparison of exact method with the proposed hybrid algorithm illustrates that the HSA can approximately obtain a near-optimal solution in less time than the exact method. The average gaps between the optimal and meta-heuristic solutions for the first and second stages are 3.87% and 3.96% indicating the efficiency of the proposed HSA algorithm supplementary with a tabu list in the cross-docking distribution network. Moreover, increasing the size of the two-stage cross-docking distribution network problem increases the CPU time of the exact method exponentially while it does not significant impacts on the CPU time of the proposed hybrid algorithm.

The graphical representation for the CPU times of proposed HSA algorithm and exact method is shown in Figs. 5 and 6 for seven small-sized test problems in the first and second stages of the proposed MIP model. According to these figures, the CPU times increase by the increase in sizes of these test problems. It is worth to note that due to the NP-hard nature of the location and routing

Table 1
Sizes of small-sized test problems.

Problem no.	No. of suppliers (m)	No. of potential cross-docking centers (O)	No. of customers (n)
1	4	4	3
2	8	5	6
3	10	6	8
4	11	7	10
5	12	8	11
6	13	9	12
7	15	9	17

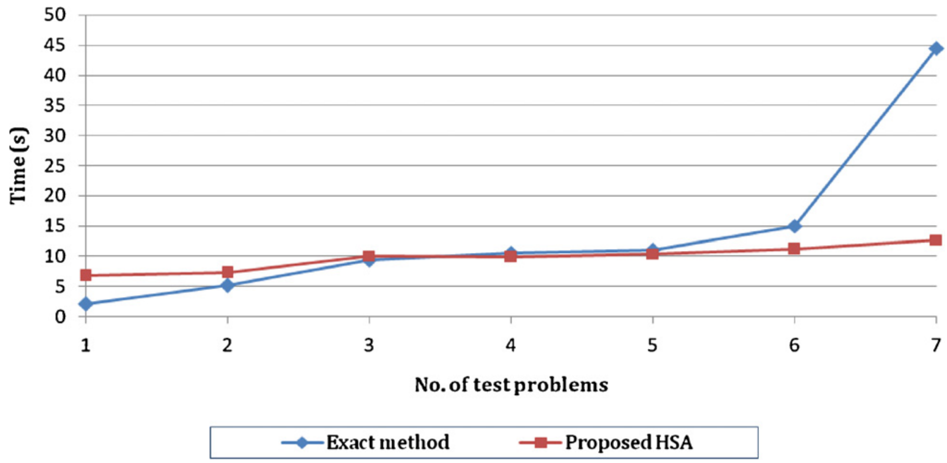

Fig. 5. Computational time for the proposed HSA and exact algorithms in the first stage.

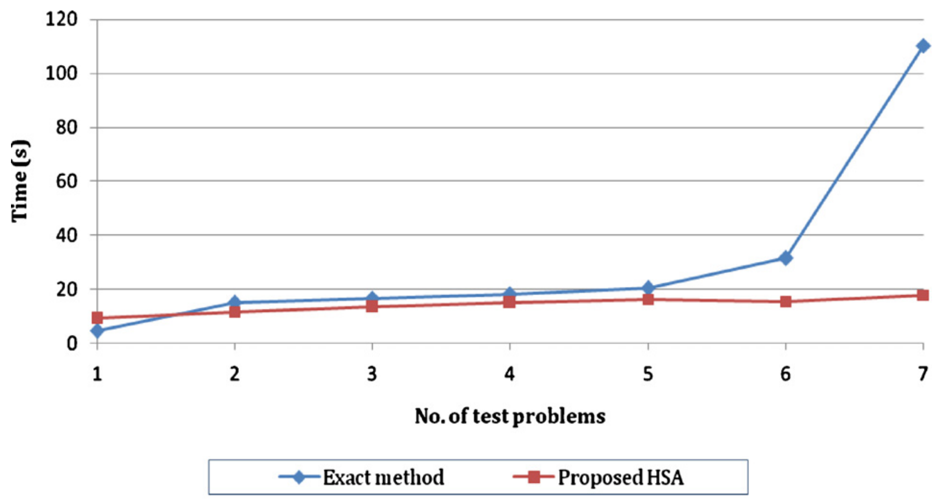

Fig. 6. Computational time for the proposed HSA and exact algorithms in the second stage.

scheduling models in large-sized test problems, the CPU times are not comparable with each other.

Some parameters are randomly generated in the uniform distributions for the seven large-sized test problems similar to small-sized test problems. In Tables 6 and 7, the computational results are given in large-sized test problems for the first and second stages of the proposed MIP model. The average times of the proposed hybrid algorithm based on the combination of SA and TS for seven large-sized test problems in 300 and 500 iterations are 281.3 (s) and 413.1 (s) for the first stage respectively. In addition, the average times of the proposed meta-heuristic in 300 and 500 iterations are 550.8 (s) and 763.1 (s) for the second stage respectively.

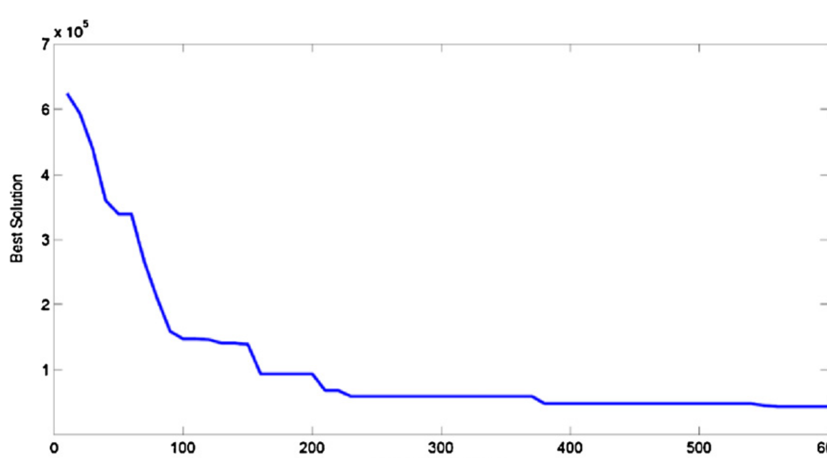

Fig. 7. Convergence rate for the seventh large-sized test problem in the first stage.

Table 2
Sources of random generations for the first stage of proposed location and routing scheduling model.

Parameters	Test problem 1	Test problem 2	Test problem 3	Test problem 4	Test problem 5	Test problem 6	Test problem 7
n	3	6	8	10	11	12	17
m	4	8	10	11	12	13	15
O	4	5	6	7	8	9	9
$D_{i'}$	∼Uniform (10,40)	∼Uniform (5,45)	∼Uniform (5,50)	∼Uniform (5,55)	∼Uniform (5,60)	∼Uniform (5,65)	∼Uniform (5,70)
CA_p	∼Uniform (600,1000)	∼Uniform (500,1100)	∼Uniform (400, 1200)	∼Uniform (400, 1400)	∼Uniform (500, 1500)	∼Uniform (600, 1500)	∼Uniform (700, 1500)
S_i	∼Uniform (10,30)	∼Uniform (5,40)	∼Uniform (5,45)	∼Uniform (5,50)	∼Uniform (5,55)	∼Uniform (5,60)	∼Uniform (5,65)
F_p	∼Uniform (300, 4000)	∼Uniform (200, 5000)	∼Uniform (200, 6000)	∼Uniform (100, 6500)	∼Uniform (50, 7000)	∼Uniform (50, 7500)	∼Uniform (50, 8000)
c_{ip}	∼Uniform (40,300)	∼Uniform (50,500)	∼Uniform (100,500)	∼Uniform (40,550)	∼Uniform (30,600)	∼Uniform (30,650)	∼Uniform (30,700)
$c_{pi'}$	∼Uniform (25,300)	∼Uniform (70,550)	∼Uniform (80,520)	∼Uniform (70,600)	∼Uniform (80,650)	∼Uniform (80,700)	∼Uniform (80,750)
TC	∼Uniform (5000, 30000)	∼Uniform (4000, 35000)	∼Uniform (3000, 40000)	∼Uniform (3000, 45000)	∼Uniform (3000, 50000)	∼Uniform (3000, 55000)	∼Uniform (3000, 60000)

Table 3
Sources of random generations for the second stage of proposed location and routing scheduling model.

Parameters	Test problem 1	Test problem 2	Test problem 3	Test problem 4	Test problem 5	Test problem 6	Test problem 7
n	3	6	8	10	11	12	17
m	4	8	10	11	12	13	15
O	2	2	3	6	6	7	7
k	5	6	8	8	9	10	11
k'	4	5	6	7	8	9	10
Q	∼Uniform (500, 1000)	∼Uniform (400, 1100)	∼Uniform (300, 1200)	∼Uniform (300, 1400)	∼Uniform (200, 1400)	∼Uniform (200, 1500)	∼Uniform (200, 1600)
p_i	∼Uniform (20, 30)	∼Uniform (10, 40)	∼Uniform (5, 40)	∼Uniform (5, 45)	∼Uniform (5, 50)	∼Uniform (5, 55)	∼Uniform (5, 60)
d_i,	∼Uniform (15, 30)	∼Uniform (10, 35)	∼Uniform (5, 35)	∼Uniform (5, 40)	∼Uniform (20, 50)	∼Uniform (20, 55)	∼Uniform (20, 60)
c_{ij}, $c_{i'j'}$	∼Uniform (300, 500)	∼Uniform (200, 500)	∼Uniform (100,500)	∼Uniform (200, 500)	∼Uniform (150, 550)	∼Uniform (150, 600)	∼Uniform (150, 650)
d_{ij}, $d_{i'j'}$	∼Uniform (20, 30)	∼Uniform (20, 40)	∼Uniform (20, 50)	∼Uniform (15, 55)	∼Uniform (15, 60)	∼Uniform (15, 65)	∼Uniform (15, 70)
c_k, $c_{k'}$	∼Uniform (150, 250)	∼Uniform (250, 450)	∼Uniform (200,500)	∼Uniform (200,600)	∼Uniform (200,700)	∼Uniform (200,750)	∼Uniform (200,800)
t_i^k, $t_{i'}^{k'}$	∼Uniform (35,45)	∼Uniform (30,50)	∼Uniform (20,50)	∼Uniform (15,55)	∼Uniform (15,60)	∼Uniform (15,65)	∼Uniform (15,70)
et_{ij}, $et_{i'j'}$	∼Uniform (50,150)	∼Uniform (40, 200)	∼Uniform (40, 250)	∼Uniform (30, 250)	∼Uniform (20, 300)	∼Uniform (20, 350)	∼Uniform (20, 400)
$\propto_{i'}$, $\beta_{i'}$	∼Uniform (40,115)	∼Uniform (40,125)	∼Uniform (30,135)	∼Uniform (30,145)	∼Uniform (20,155)	∼Uniform (20,165)	∼Uniform (20,175)

Table 4
Results in small-sized test problems for the first stage.

No. of test problems	Exact method		SA (300 iterations)			Proposed HSA (300 iterations)		
	Best solution	Time (s)	Best solution	Time (s)	Gap (%)	Best solution	Time (s)	Gap (%)
1	4283.4	2.1	4442	7.2	3.70	4365.2	6.8	1.91
2	11,715	5.2	12,379	8.1	5.67	12,031	7.3	2.70
3	16,467	9.4	17,586.8	10.8	6.80	17,336.5	10.1	5.28
4	21,116	10.6	22,654.2	10.3	7.28	22,322	9.9	5.71
5	22,446	11.1	23,889.9	11.5	6.43	23,416	10.4	4.32
6	27,123.8	15.0	28,558	12.2	5.29	27,961.9	11.2	3.09
7	34,696	44.5	36,968.1	13.6	6.55	36,122.3	12.7	4.11
Average	19,692.5	13.9	20,925.4	10.5	5.96	20,507.8	9.8	3.87

Table 5
Results in small-sized test problems for the second stage.

No. of test problems	Exact method		SA (300 iterations)			Proposed HSA (300 iterations)		
	Best solution	Time (s)	Best solution	Time (s)	Gap (%)	Best solution	Time (s)	Gap (%)
1	56,122	4.5	57,794.4	10.9	2.98	56,694.4	9.1	1.02
2	111,542.2	15	116,751	13.2	4.67	114,677	11.6	2.81
3	151,448	16.7	159,171.8	15.8	5.10	158,354	13.4	4.56
4	187,723	18.0	200,506.6	17.1	6.81	198,103.8	14.9	5.53
5	217,932	20.2	231,073	19	6.03	230,790	16.3	5.90
6	256,535.4	31.4	273,979	18.2	6.80	265,232	15.5	3.39
7	306,856	110.3	325,881	19.3	6.20	320,664.5	17.8	4.50
Average	184,022.7	30.9	195,022.4	16.2	5.51	192,073.7	14.1	3.96

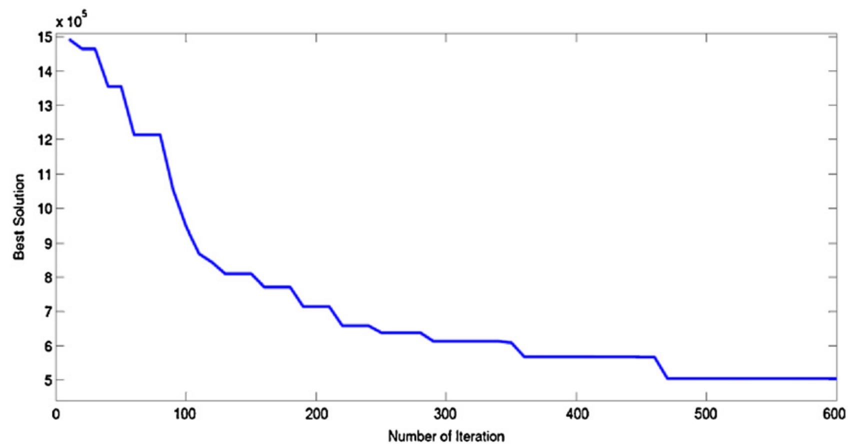

Fig. 8. Convergence rate for the seventh large-sized test problem in the second stage.

Table 6
Results in large-sized test problems for the first stage.

No. of test problems	No. of suppliers	No. of cross-docking centers	No. of customers	SA (500 iterations)		Proposed HSA (300 iterations)		Proposed HSA (500 iterations)	
				Best solution	Time (s)	Best solution	Time (s)	Best solution	Time (s)
1	25	10	30	31,045	294	30,938.4	226.3	30,413	284.9
2	30	12	35	36,106.1	412.2	37,028	249	34,694.4	402
3	35	15	40	33,639	469	33,261.3	277	32,007	414.4
4	40	18	45	42,902.2	411	40,996.8	305.2	40,666	400
5	45	20	50	44,957.4	438	45,285	242	41,464.8	392.7
6	50	24	55	51,429.3	498.2	50,201	333.1	48,771	482.9
7	55	28	60	56,569.6	539.4	64,273.5	336.6	53,696.8	514.8
Average	40	18	45	42,378.4	437.4	43,140.6	281.3	40,244.7	413.1

The best results are provided for 500 iterations in all large-sized test problems. The runtime of the presented HSA meta-heuristic algorithm is acceptable for solving these problems. For the first stage of the proposed model, the maximum runtimes in 300 and 500 iterations are 336.6 (s) and 514.8 (s), respectively, and for the second stage the maximum runtimes are 600 (s) and 809.3 (s) for the seventh large-sized test problem. Also, the convergence rates of the proposed hybrid meta-heuristic algorithm are depicted in Figs. 7 and 8 for this test problem. Finally, the results illustrate that the proposed HSA meta-heuristic algorithm for solving the location and routing scheduling problems with multiple cross-docking centers can perform well and converge fast to reasonable solutions.

Table 7
Results in large-sized test problems for the second stage.

No. of test problems	No. of suppliers	No. of cross-docking centers	No. of retailers	SA (500 iterations)		Proposed HSA (300 iterations)		Proposed HSA (500 iterations)	
				Best solution	Time (s)	Best solution	Time (s)	Best solution	Time (s)
1	25	10	30	478,128	723.7	474,378	477.8	459,690	695
2	30	12	35	510,090	800.6	510,981.1	524.6	491,754.1	763.2
3	35	15	40	521,123	748.9	513,981	511	483,146.4	727.9
4	40	18	45	470,909.7	847	470,099.6	566	461,325.5	791.6
5	45	20	50	559,943	813.9	552,716.5	598.6	514,339.5	798
6	50	24	55	568,769.1	809	550,168	577.4	520,957	757
7	55	28	60	559,112.1	836	620,157	600	494,438	809.3
Average	40	18	45	524,010.7	797	527,497.3	550.8	489,378.6	763.1

6. Conclusion

In this paper, new location and routing scheduling problems have been addressed in cross-docking distribution networks. Single period multi-echelon logistic network, including suppliers, cross-docking centers and customers, has been presented where a single product can be shipped through multiple cross-docking centers to meet customers' demands. The nature of these problems is the NP-hard in the strong sense, and formulated as a mathematical programming. A two-stage mixed-integer programming (MIP) model has been proposed for the location of cross-docking centers and vehicle routing scheduling problems with multiple cross-docking centers for the distribution networks in the supply chain. To solve the presented two-stage MIP model, in this paper a new two-stage hybrid simulated annealing (HSA) algorithm with tabu list has been introduced. In the algorithm by the combination of simulated annealing (SA) and tabu search (TS), not only the number of solution revisits but also computational time to obtain a near-optimal solution has been remarkably decreased. The presented algorithm characterizes a special solution representation for the location and routing scheduling in the cross-docking distribution systems. To validate and verify the proposed HSA, fourteen randomly generated test problems with different-sizes have been solved by the HSA and exact algorithms. The presented solving approach has been illustrated to be efficient in finding near-optimal solutions for a variety of problem instances in terms of runtime and solution quality. The computational results have been very competitive in comparison with the optimization software. The HSA can be properly utilized in situations where popular commercial solvers are unavailable in large-sized location and routing scheduling problems in real-life applications for the supply chain management. Future research can be recommended in a few directions. It is interesting to consider the multi-commodity consolidation by considering time windows constraints in the proposed cross-docking distribution network. Another extension is to take account of uncertain parameters (e.g., fuzzy and stochastic values) due to the complex nature of the cross-docking systems in order to become a more realistic and practical mathematical model. Also, recent local search methods can be extended because of their potential to enhance the performance of the proposed solving approach to search the near-optimal solution in the reasonable time.

Acknowledgements

The authors thank the Editor-in-Chief Professor S.J. Hu and respected reviewers for valuable comments and suggestions. Their comments and recommendations have led to a substantial improvement of this research. The authors would like to acknowledge the partially financial support from the University of Tehran under the research grant no. 8106043/1/21.

References

[1] Min H, Jayaraman V, Srivastava R. Combined location-routing problems: a synthesis and future research directions. European Journal of Operational Research 1998;108(1):1–15.
[2] Yu VF, Lin S-W, Lee W, Ting C-J. A simulated annealing heuristic for the capacitated location routing problem. Computers and Industrial Engineering 2010;58:288–99.
[3] Lin CKY, Chow CK, Chen A. A location-routing-loading problem for bill delivery services. Computers and Industrial Engineering 2002;43(1–2):5–25.
[4] Chan Y, Carter WB, Burnes MD. A multiple-depot, multiple-vehicle, location-routing problem with stochastically processed demands. Computers and Operations Research 2001;28(8):803–26.
[5] Caballero R, Gonzalez M, Guerrero FM, Molina J, Paralera C. Solving a multiobjective location routing problem with a metaheuristic based on tabu search: application to a real case in Andalusia. European Journal of Operational Research 2007;177:1751–63.
[6] Jayaraman V, Ross A. A simulated annealing methodology to distribution network design and management. European Journal of Operational Research 2003;144:629–45.
[7] Ross A, Jayaraman V. An evaluation of new heuristics for the location of cross-docks distribution centers in supply chain network design. Computers and Industrial Engineering 2008;55:64–79.
[8] Lee YH, Jung WJ, Lee KM. Vehicle routing scheduling for cross docking in the supply chain. Computer and Industrial Engineering 2006;51:247–56.
[9] Tavakkoli-Moghaddam R, Makui A, Mazloomi Z. A new integrated mathematical model for a bi-objective multi-depot location-routing problem solved by a multi-objective scatter search algorithm. Journal of Manufacturing Systems 2010;29(2–3):111–9.
[10] Jayaraman V. An efficient heuristic procedure for practical sized capacitated warehouse design and management. Decision Sciences Journal 1998;29(3):729–45.
[11] Donaldson H, Johnson EL, Ratliff HD, Zhang M, Schedule-driven cross-docking networks. Technical report, Georgia Institute of Technology; 1999. http://www.isye.gatech.edu/apps/research-papers/papers/misc9904.pdf
[12] Li Y, Lim A, Rodrigues B. Crossdocking-JIT scheduling with time windows. Journal of the Operational Research Society 2004;55:1342–51.
[13] Lim A, Miao Z, Rodrigues B, Xu Z. Transshipment through crossdocks with inventory and time windows. Naval Research Logistics 2005;52(8):724–33.
[14] Reeves KA. Supply chain governance: a case of cross dock management in the automotive industry. IEEE Transactions on Engineering Management 2007;54(3):455–67.
[15] Bachlaus M, Kumar Pandey M, Mahajan C, Shankar R, Tiwari MK. Designing an integrated multi-echelon agile supply chain network: a hybrid taguchi-particle swarm optimization approach. Journal of Intelligent Manufacturing 2008;19:747–61.
[16] Liao C-J, Lin Y, Shih SC. Vehicle routing with cross-docking in the supply chain. Expert Systems with Applications 2010;37:6868–73.
[17] Musa R, Arnaout J-P, Jung H. Ant colony optimization algorithm to solve for the transportation problem of cross-docking network. Computers and Industrial Engineering 2010;59:85–92.
[18] Yang KK, Balakrishnan J, Cheng CH. An analysis of factors affecting cross docking operations. Journal of Business Logistics 2010;31(1):121–48.
[19] Liao TW, Egbelu PJ, Chang PC. Two hybrid differential evolution algorithms for optimal inbound and outbound truck sequencing in cross docking operations. Applied Soft Computing 2012;12:3683–97.
[20] Melo MT, Nickel S, Saldanha-da-Gama F. A tabu search heuristic for redesigning a multi-echelon supply chain network over a planning horizon. International Journal of Production Economics 2012;136:218–30.
[21] Ma H, Miao Z, Lim A, Rodrigues B. Cross docking distribution networks with setup cost and time window constraint. Omega 2011;39:64–72.
[22] Alpan G, Ladier A-L, Larbi R, Penz B. Heuristic solutions for transshipment problems in a multiple door cross docking warehouse. Computers and Industrial Engineering 2011;61:402–8.
[23] Dondo R, Méndez CA, Cerdá J. The multi-echelon vehicle routing problem with cross docking in supply chain management. Computers and Chemical Engineering 2011;35(12):3002–24.

[24] Joo CM, Kim BS, Scheduling compound trucks in multi-door cross-docking terminals, *International Journal of Advanced Manufacturing Technology*, in press

[25] Bartholdi III JJ, Gue KR. The best shape for a crossdock. Transportation Science 2004;38(2):235–44.

[26] Wen M, Larsen J, Clausen J, Cordeau J-F, Laporte G. Vehicle routing with cross-docking. Journal of the Operational Research Society 2009;60: 1708–18.

[27] Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equations of state calculations by fast computing machines. Journal of Chemical Physics 1953;21:1087–92.

[28] Kirkpatrick S, Gelatt Jr CD, Vecchi MP. Optimization by simulated annealing. Science 1983;220:671–80.

[29] Ghodratnama A, Rabbani M, Tavakkoli-Moghaddam R, Baboli A. Solving a single-machine scheduling problem with maintenance, job deterioration and learning effect by simulated annealing. Journal of Manufacturing Systems 2010;29:1–9.

[30] Tavakkoli-Moghaddam R, Gazanfari M, Alinaghian M, Salamatbakhsh A, Norouzi N. A new mathematical model for a competitive vehicle routing problem with time windows solved by simulated annealing. Journal of Manufacturing Systems 2011;30:83–92.

[31] Wu C-C, Hsu P-H, Lai K. Simulated-annealing heuristics for the single-machine scheduling problem with learning and unequal job release times. Journal of Manufacturing Systems 2011;30:54–62.

[32] Glover F. Tabu search: a tutorial. Interfaces 1990;20(4):74–94.

[33] Swarnkar R, Tiwari MK. Modeling machine loading problem of FMSs and its solution methodology using a hybrid tabu search and simulated annealing-based heuristic approach. Robotics and Computer-Integrated Manufacturing 2004;20:199–209.

[34] Mirabi M, Fatemi Ghomi SMT, Jolai F. Efficient stochastic hybrid heuristics for the multi-depot vehicle routing problem. Robotics and Computer-Integrated Manufacturing 2010;26:564–9.

[35] Reinelt G. The traveling salesman: computational solutions for TSP applications. Berlin, Heidelberg: Springer-Verlag; 1994.

[36] Gupta SR, Smith JS. Algorithms for single machine total tardiness scheduling with sequence dependent setups. European Journal of Operational Research 2006;175:722–39.

[37] Clarke G, Wright J. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 1964;12:568–81.

[38] Hajek B. A tutorial survey of theory and applications of simulated annealing. In: Proceedings of IEEE conference on decision and control. 1985. p. 755–60.