

A hybrid meta-heuristic algorithm for vehicle routing and packing problem with cross-docking

İlker Küçükkoğlu¹ · Nursel Öztürk¹

Received: 20 July 2015 / Accepted: 7 October 2015 / Published online: 19 October 2015
© Springer Science+Business Media New York 2015

Abstract Cross-docking is a relatively new logistics strategy that has a great potential to eliminate storage cost and speed up the product flows. This paper considers the vehicle routing and packing problem with cross-docking and presents a mixed integer linear mathematical model. In the model, a set of trucks are used to transport products from suppliers to customers through cross-docking centers. Each supplier and customer node can be visited only once and directly shipping is not allowed from suppliers to customers. Moreover, truck capacities are identified with physical dimensional limits on the contrary of weight or amount of load. The objective of the study is to determine the vehicle routes that minimize the total distance. Due to the complexity of the mathematical model, a hybrid meta-heuristic algorithm (HMA), which integrates tabu search (TS) algorithm within simulated annealing (SA) algorithm, is proposed to solve the problem. Proposed HMA is tested on a well-known benchmark problem data set and compared with the SA and TS solutions. Results show that proposed HMA can produce effective solutions and outperforms the SA and TS especially for the large-sized problems.

Keywords Hybrid meta-heuristic algorithm · Cross-docking · Vehicle routing · 2-Dimensional vehicle loading

Introduction

The efficiency of transportation is one of the most important factors for supply chain management. For this reason, many

companies develop various strategies to boost customer satisfaction and bring down the total costs. Cross-docking which almost eliminates the storage costs and speeds up the product flows is one of these strategies. It can be described as the process of moving products from suppliers to customers through cross-docking centers without storing products for a long time in these centers (Ladier and Alpan 2014). At each center, incoming trucks loaded from different suppliers arrives to incoming doors and unloaded to incoming area quickly. These products are sorted and consolidated according to their destinations and then reloaded to outgoing trucks for distribution (Zarandi et al. 2014). This strategy provides different advantages compared with traditional distribution centers: the consolidation of shipments, a shorter delivery lead time, the reduction of costs, improved customer service, fewer overstocks, etc. (Van Belle et al. 2012). As a result of these advantages, cross-docking has become an interesting logistics strategy that can give companies important competitive benefits.

Until now, considerable researches on cross-docking have been studied in the literature and they categorized in many ways. Van Belle et al. (2012) present a comprehensive literature review about cross-docking and classified them based on the problem type: location of cross-docks, cross-docking layout design, cross-docking networks, vehicle routing, dock door assignment, truck scheduling, storage and other issues. Some of these problems are more concerned about long term decisions (strategic or tactical), while others deal with short-term decisions (operational).

The vehicle routing problem for cross-docking (VRPCD) involves the fulfilment of a set of customer orders which are picked up at various supplier locations and delivered to their destinations by a fleet of vehicle after consolidation at a cross-docking center. The objective is to find the best pickup and delivery routes to minimize total transportation cost. As in

✉ İlker Küçükkoğlu
ikucukoglu@uludag.edu.tr

¹ Industrial Engineering Department, Faculty of Engineering, Uludag University, Bursa, Turkey

the extensive variants of the vehicle routing problem, the time windows constraint is widely considered in the literature so that all locations must be visited within their time windows and all pickup and delivery operations must be completed before the end of the planning horizon. [Lee et al. \(2006\)](#) consider the vehicle routing problem for cross-docking. The aim of their study is to find an optimal routing schedule for pickup and delivery operations that minimizes the total transportation cost and fixed costs of the vehicles. The split deliveries are not allowed in their study and it is assumed that all incoming trucks should arrive at the cross-dock simultaneously to prevent waiting for outgoing trucks. An integer mathematical model is presented for the problem and a tabu search (TS) algorithm is proposed as a solution approach. The same problem is taken into account by [Liao et al. \(2010\)](#). The authors propose a new TS algorithm which can achieve better results than the existing TS algorithm in short computational times. [Wen et al. \(2009\)](#) study the VRPCD, which differs from the existing studies by considering time windows. The authors formulate the proposed problem by using mixed integer model. A TS algorithm is used to solve the problem and tested on large scale realistic data set. Their results show that proposed algorithm finds close results to optimum solutions in less than five minutes. [Tarantilis \(2013\)](#) improves the TS proposed by [Wen et al. \(2009\)](#) with adaptive multi-restart strategy, in which the algorithm restarts the search from a new solution once a region has been extensively explored. Computational results demonstrate the efficiency of the algorithm with new improved upper bounds. Another study on VRPCD with time windows constraints is carried out by [Vahdani et al. \(2012\)](#). The authors introduce a hybrid algorithm which incorporates the elements of particle swarm optimization (PSO), simulated annealing (SA) and variable neighborhood search (VNS). The proposed algorithm is compared with TS presented by [Lee et al. \(2006\)](#) and results show that the hybrid algorithm capable to find better solution. [Morais et al. \(2014\)](#) develop a iterated local search heuristic for the VRPCD that includes a constructive heuristics and six local search procedures. This approach is tested on different data sets and compared with the TS proposed by [Wen et al. \(2009\)](#). Computational studies show that iterated local search method finds better solution than the best known solutions for many problems and outperforms the TS. [Dondo et al. \(2011\)](#) study the multi-echelon vehicle routing problem for cross-docking. In their paper, a hybrid strategy combining direct shipping, warehousing and cross-docking is allowed. A monolithic optimization framework based on a mixed integer linear mathematical model is presented for the problem. [Dondo and Cerdá \(2013, 2014\)](#) improve this approach by sweep-based heuristic which is incorporated into the model in order to derive more efficient formulation. Computational results show that sweep heuristic embedded model finds near optimal solution to large scale problems. [Hasani-Goodarzi](#)

[and Tavakkoli-Moghaddam \(2012\)](#) consider the VRPCD for multi-product with split delivery and pickups. The authors also take into account the traveling times of the vehicles in order to complete all picking up operations in a specified time horizon. The problem is formulated as a mixed integer mathematical model which is tested on small-sized instances. Similarly, [Hu et al. \(2013\)](#) introduce a mathematical model that combines two sub models to optimize the overall travelling time, distance and waiting time at the cross-docking center. [Vahdani and Sadigh \(2014\)](#) investigate the multi-product VRPCD under uncertainty and present a hybrid solution methodologies by combining fuzzy probabilistic programming and stochastic programming. Another study that considers the split delivery condition is carried out by [Moghadam et al. \(2014\)](#). In order to solve the problem SA and a hybrid algorithm combining ant colony optimization (ACO) and SA are proposed. According to the computational studies on different data sets, hybrid algorithm exhibits better performance than SA. A different variant of the VRPCD is studied by [Agustina et al. \(2014\)](#). Their research focuses on the integration of the VRPCD with cross-docking scheduling problem which include product consolidation decisions at the cross-docking center. The authors propose a mixed integer linear program for the problem which obtains the solutions for the medium sized problems in a reasonable time.

Although the vehicle routing problem provides practical decisions on operational processes for distribution systems, logistics managers usually have to deal with routing and packing problems simultaneously because of the placement of freight, which cannot be stacked on each other, generally affects the route plans in real life applications ([Wei et al. 2015](#)). Therefore, there exist several works in literature, which consider the vehicle routing problem with 2-dimensional truck loading constraints and introduce different solution methodology in order to solve large scale problems, such as: [Gendreau et al. \(2006, 2008\)](#), [Zachariadis et al. \(2009\)](#), [Leung et al. \(2011\)](#) and [Wei et al. \(2011\)](#) propose TS based heuristic methods, [Leung et al. \(2013\)](#) propose a SA based heuristic method, [Fuellerer et al. \(2009\)](#) propose an ACO method, [Khebbache-Hadji et al. \(2013\)](#) propose memetic algorithm and recently [Wei et al. \(2015\)](#) propose a VNS algorithm. Motivated by the importance of the 2-dimensional packing operations on routing plans, this study considers the VRPCD with 2-dimensional truck loading constraints, which is called the vehicle routing problem for cross-docking with 2-dimensional truck loading (VRPCD-2D). The VRPCD-2D differs from the existing studies on VRPCD in that truck capacities are identified with the physical dimensional limits on the contrary of weight or amount of load which provides more practical and realistic plan for pickup and delivery operations. To the best of our knowledge, no work has been taken into account the

2-dimensional truck loading constraints for VRPCD though it is a practical problem in real-world transportation and logistics industries.

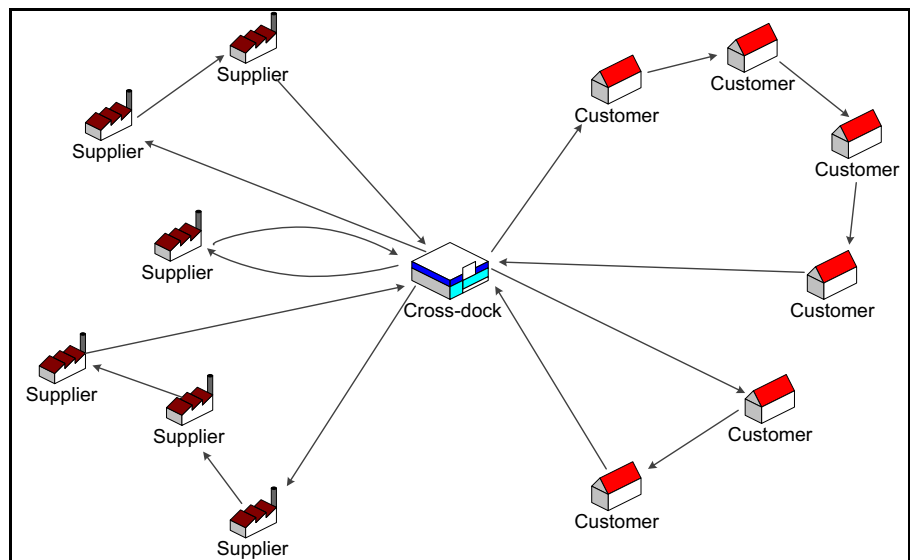
In this study, the VRPCD-2D is formulated as a mixed integer mathematical model and a hybrid meta-heuristic algorithm (HMA), which integrates TS within SA, is proposed in order to solve the problem. The hybridization of meta-heuristic algorithms are proposed by many researchers in order to solve various complex problems. Nearchou (2004) presents a HMA to solve flow-shop scheduling problem. Likewise, Dong and Wang (2012), Jolai et al. (2012), Azadeh et al. (2013) and Ramezani et al. (2015) consider the scheduling problems and solves their problems by using a HMA. Hamta et al. (2014) develop a hybrid framework of the VNS with TS to solve scheduling and line balancing problem. Poorzahedy and Rouhani (2007) compare the performance of different type hybrid algorithms for network design problem. Leno et al. (2015) propose an elitist strategy based hybrid genetic algorithm that uses SA as local search mechanism. Although, there exist numerous hybrid algorithm in literature, the hybrid structure of the SA and TS is one of the most popular HMA used to solve several problems: e.g. used by Osman and Christofides (1994) for capacitated clustering problems, by Lin and Ying (2009) for non-permutation flowshop scheduling problems, by Lin et al. (2012) for vehicle routing problem with time windows, by Katsigiannis et al. (2012) for optimal sizing of autonomous power systems with renewables, by Arıkan and Erol (2012) for part selection and machine loading problems in flexible manufacturing systems, by Kaviani et al. (2014) for quadratic assignment problem, etc. In this study, the proposed HMA of SA and TS is promoted with an efficient local search method and 2-dimensional packing heuristics in order to obtain efficient solutions. The performance of the HMA is tested on various

benchmark instances derived from vehicle routing problem test problems and the results are compared with the solutions of the SA and TS. A remainder of this paper is organized as following sections: “Problem definition and model formulation”, “Proposed algorithm”, “Computational results” and “Conclusions”.

Problem definition and model formulation

In this paper, VRPCD-2D is defined as the problem of transporting products from suppliers to customer locations through cross-docking center in order to satisfy a set of customer demand. The considered problem is described in Fig. 1, where the products from the suppliers are picked up by incoming trucks, consolidated at the cross-docking center, and immediately delivered to customers by outgoing trucks. The incoming trucks start their tour from cross-docking center, serve to the suppliers in a specified time windows and return to the cross-docking center. After all incoming trucks complete their routes products are loaded to outgoing trucks and delivered to their destinations. As in the incoming operations, customer nodes are served in a specified time windows. Each product of suppliers is loaded into the incoming and outgoing trucks by regarding the 2-dimensional truck loading constraints. The demand of a customer is defined by a set of rectangular items with given width and length. All the products belonging to one customer/supplier must be assigned to the same pickup/delivery route. The objective of this study is to find the best route plan for incoming and outgoing trucks in order to minimize total transportation cost. VRPCD-2D is considered with the following assumptions:

Fig. 1 The concept of the vehicle routing problem for cross-docking center



- Each vehicle starts and terminates its route at the cross-docking center.
- Split deliveries are not allowed and each supplier and customer node must be visited by only one vehicle.
- Directly shipping from suppliers to customers is not allowed.
- Vehicles can serve more than one supplier or customer node and each route starts and ends at the cross-docking center.
- Each vehicle can be used for pickup or delivery operations.
- The pickup and delivery operations must be carried out in supplier’s and customer’s time window interval. Also, the whole process must be completed within the planning horizon.
- Vehicles are considered homogeneous and capacities are identified with physical dimensional limits on the contrary of weight or amount of load.
- Products are considered as rectangular shapes with different sizes.
- Products can be loaded into the incoming or outgoing trucks with unrestricted order.

As mentioned in the literature review, the VRPCD is considered by several studies with different assumptions. However, the VRPCD with 2-dimensional truck loading plans have not been taken into account by any research. Therefore, the mathematical model of the VRPCD-2D is newly formed on the basis of assumptions described above by adapting the 2-dimensional pallet packing constraints proposed by [Chen et al. \(1991\)](#). The proposed mathematical model is formulated as follows.

Parameters and notations

- S Set of suppliers.
- D Set of destinations.
- N Set of all locations: $\{0\} \cup S \cup D$, where 0 denotes the cross-docking center.
- PR Set of products.
- K Set of vehicles.
- c_{ij} Transportation cost from node i to node j ; $\forall i, j \in N$.
- t_{ij} Travelling time from node i to node j ; $\forall i, j \in N$.
- e_i Time window lower bound for node i ; $\forall i \in N$.
- l_i Time window upper bound for node i ; $\forall i \in N$.
- s_i Service processing time for node i ; $\forall i \in N$.
- W Width of the trucks.
- L Length of the trucks.
- dem_m Supplier label of product m ; $\forall m \in PR$.
- dem'_m Destination label of product m ; $\forall m \in PR$.
- q_m Width of product m ; $\forall m \in PR$.
- p_m Length of product m ; $\forall m \in PR$.
- M Big number.

Decision variables

- z_{ijk} 1 if vehicle k travels from node i to node j , and 0 otherwise; $\forall i, j \in N, \forall k \in K$.
- w_i Service start time at node i ; $\forall i \in N$.
- w_{-c} Distribution starting time for cross-docking center which denotes the end of the picking up procedures.
- x_m x coordinate of product m in an incoming truck; $\forall m \in PR$.
- x'_m x coordinate of product m in an outgoing truck; $\forall m \in PR$.
- y_m y coordinate of product m in an incoming truck; $\forall m \in PR$.
- y'_m y coordinate of product m in an outgoing truck; $\forall m \in PR$.
- $\alpha_{mn}, \beta_{mn}, \gamma_{mn}, \delta_{mn}$ 1 if product m is on the left, right, bottom and top side of product n in a same incoming truck, respectively, and 0 otherwise; $\forall m, n \in PR$.
- $\alpha'_{mn}, \beta'_{mn}, \gamma'_{mn}, \delta'_{mn}$ 1 if product m is on the left, right, bottom and top side of product n in a same outgoing truck, respectively, and 0 otherwise; $\forall m, n \in PR$.

Using the parameters, notations and decision variables described above, the VRPCD-2D is formulated as:

$$\text{Min } z = \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} c_{ij} z_{ijk} \tag{1}$$

Subject to

$$\sum_{\substack{j \in N \setminus D \\ i \neq j}} \sum_{k \in K} z_{ijk} = 1 \quad \forall i \in S \tag{2}$$

$$\sum_{\substack{i \in N \setminus D \\ i \neq j}} \sum_{k \in K} z_{ijk} = 1 \quad \forall j \in S \tag{3}$$

$$\sum_{\substack{j \in N \setminus S \\ i \neq j}} \sum_{k \in K} z_{ijk} = 1 \quad \forall i \in D \tag{4}$$

$$\sum_{\substack{i \in N \setminus S \\ i \neq j}} \sum_{k \in K} z_{ijk} = 1 \quad \forall j \in D \tag{5}$$

$$\sum_{j \in N \setminus \{0\}} z_{0jk} = 1 \quad \forall k \in K \tag{6}$$

$$\sum_{i \in N \setminus \{0\}} z_{i0k} = 1 \quad \forall k \in K \tag{7}$$

$$\sum_{\substack{i \in N \\ i \neq j}} z_{ijk} = \sum_{\substack{i \in N \\ i \neq j}} z_{jik} \quad \forall j \in N \setminus \{0\}, \quad \forall k \in K \tag{8}$$

$$w_i + s_i + t_{ij} \leq w_j + M \left(1 - \sum_{k \in K} z_{ijk} \right) \quad \forall i \in N \setminus D, \quad \forall j \in S, \quad i \neq j \tag{9}$$

$$w_i + s_i + t_{i0} \leq w_{-c} + M(1 - z_{i0k}) \quad \forall i \in S, \quad \forall k \in K \tag{10}$$

$$w_{-c} + s_0 + t_{0j} \leq w_j + M(1 - z_{0jk}) \quad \forall j \in D, \quad \forall k \in K \tag{11}$$

$$w_i + s_i + t_{ij} \leq w_j + M \left(1 - \sum_{k \in K} z_{ijk} \right) \quad \forall i, j \in D, \quad i \neq j \tag{12}$$

$$w_i + s_i + t_{i0} \leq l_0 + M \left(1 - \sum_{k \in K} z_{i0k} \right) \quad \forall i \in D \tag{13}$$

$$e_i \leq w_i \leq l_i \quad \forall i \in N \setminus \{0\} \tag{14}$$

$$x_m + q_m \leq x_n + M(1 - \alpha_{mn}) \quad \forall m, n \in PR, \quad m < n \tag{15}$$

$$x_n + q_n \leq x_m + M(1 - \beta_{mn}) \quad \forall m, n \in PR, \quad m < n \tag{16}$$

$$y_m + p_m \leq y_n + M(1 - \gamma_{mn}) \quad \forall m, n \in PR, \quad m < n \tag{17}$$

$$y_n + p_n \leq y_m + M(1 - \delta_{mn}) \quad \forall m, n \in PR, \quad m < n \tag{18}$$

$$\alpha_{mn} + \beta_{mn} + \gamma_{mn} + \delta_{mn} + 1 \geq \sum_{j \in N \setminus D} (z_{dem_mjk} + z_{dem_njk}) \quad \forall k \in K, \quad \forall m, n \in PR, \quad m < n \tag{19}$$

$$x_m + q_m \leq W \quad \forall m \in PR \tag{20}$$

$$y_m + p_m \leq L \quad \forall m \in PR \tag{21}$$

$$x'_m + q_m \leq x'_n + M(1 - \alpha'_{mn}) \quad \forall m, n \in PR, \quad m < n \tag{22}$$

$$x'_n + q_n \leq x'_m + M(1 - \beta'_{mn}) \quad \forall m, n \in PR, \quad m < n \tag{23}$$

$$y'_m + p_m \leq y'_n + M(1 - \gamma'_{mn}) \quad \forall m, n \in PR, \quad m < n \tag{24}$$

$$y'_n + p_n \leq y'_m + M(1 - \delta'_{mn}) \quad \forall m, n \in PR, \quad m < n \tag{25}$$

$$\alpha'_{mn} + \beta'_{mn} + \gamma'_{mn} + \delta'_{mn} + 1 \geq \sum_{j \in N \setminus S} (z_{dem'_mjk} + z_{dem'_njk}) \quad \forall k \in K, \quad \forall m, n \in PR, \quad m < n \tag{26}$$

$$x'_m + q_m \leq W \quad \forall m \in PR \tag{27}$$

$$y'_m + p_m \leq L \quad \forall m \in PR \tag{28}$$

The objective function (1) minimizes the total transportation costs of the incoming and outgoing trucks. Constraints (2)

and (3) ensures that each supplier node can be serviced by only one vehicle and similarly constraints (4) and (5) provide this condition for customer nodes. Constraints (6) and (7) ensure that all routes leave from the cross-docking center and return to the cross-docking center at the end of the service. Constraint (8) guarantees that a vehicle leaves from the same node it has entered. Constraints (9)–(13) provide the feasibility of the schedule in accordance with the time considerations. Constraint (14) guarantees that each supplier and customer node is serviced within the time window. Constraints (15)–(18) and (22)–(25) assure that products do not overlap each other and these constraints are considered only if a pair of products is in the same incoming or outgoing truck. This is taken care of in constraint (19) for incoming trucks and constraint (26) for outgoing trucks. Constraints (20), (21), (27) and (28) provide that a product loaded in an incoming or outgoing truck, cannot exceed the truck’s physical dimensions.

Proposed algorithm

This section introduces main components of the proposed HMA for VRPCD-2D, which are two well-known meta-heuristics, SA and TS, and also introduces the packing heuristics used for the 2-dimensional loading procedures.

Simulated annealing algorithm

SA is a stochastic method for solving combinatorial problems proposed by Kirkpatrick et al. (1983). The SA methodology draws its inspiration from annealing process in metallurgy. It works by emulating the physical process so that a solid is heated to a high temperature and step by step cooled to low it to crystallize.

SA uses a stochastic approach to guide the search. In addition to accepting better solutions, SA allows the search to proceed to a neighboring state even if the move causes the value of the objective function to become worse. SA processes the local search in the following way. If a move to neighbor X' in neighborhood ensures improvement in objective value, or leaves it unchanged, then the move is always accepted. More precisely, the solution X' is accepted as the new solution if $\Delta \leq 0$, where $\Delta = (f(X') - f(X))$ and $f(X)$ is the value of objective function. Moves, which increase the objective function means that $\Delta > 0$, are accepted according to a probability function $e^{(-\Delta/T)} > \theta$, where T is the parameter of temperature, and θ is a random number between $[0, 1]$. The value of T varies from a relatively large number to a small value close to zero, which is often controlled by linear equations for reducing temperature linearly. The stopping criterion of the SA is based on the probability that a move from a local minimum to a neighbor with the lowest score is

```

1:  $T$  = Initial temperature
2:  $c$  = Cooling parameter
3:  $X^{best} = \emptyset$ 
4: Generate  $X$ 
5:  $X^{best} = X$ 
6: Do
7:   Generate  $X'$  from  $X$  by using a Local Search Method
8:   If  $f(X') \leq f(X)$  Then
9:      $X = X'$ 
10:  Else
11:     $\Delta = f(X') - f(X)$ 
12:    If  $\exp\left(-\frac{\Delta}{T}\right) > \text{random}[0,1]$  Then
13:       $X = X'$ 
14:    End If
15:  End If
16:   $T = T \times c$ 
17:  If  $f(X) < f(X^{best})$  Then
18:     $X^{best} = X$ 
19:  End If
20: Loop Until (Stopping Criteria)

```

Fig. 2 Pseudo code of the SA algorithm

accepted. If that probability is low compared to the selection probability the algorithm is stopped in that approach (Otten and Van Ginneken 1988). However, there have been various stopping criteria, which are developed on the basis of considered problem. The pseudo code of the SA algorithm is shown in Fig. 2.

Tabu search algorithm

TS was introduced by Fred Glover in 1986 as an iterative meta-heuristic algorithm that guides a local search procedure to explore the solution space beyond local optimality (Glover 1989, 1990). TS algorithm is different from the SA algorithm in that the tabu search includes a memory mechanism that prevents the search from cycling back to previously visited solutions. The memory mechanism which is called as *tabu list* maintains the search history by keeping either some of the moves or just their attributes, and prevents the reversing these moves for a given number of iterations. Data structure of the tabu list is necessary to ensure the proper management of the tabu restriction (Landrieu et al. 2001). On the other hand, this restriction can be ignored if the attempted move leads to find a new best solution for the algorithm. This procedure is defined as *aspiration criterion*, which allows for exception from tabu list, if any move leads to promising solution (Armentano and Yamashita 2000; Geyik and Cedimoglu 2004). The pseudo code of the TS algorithm is shown in Fig. 3.

Hybrid meta-heuristic algorithm

The proposed HMA for the VRPCD-2D is based on the integration of TS within the SA. The HMA takes the advantages

```

1: Empty Tabu List
2:  $X^{best} = \emptyset$ 
3: Generate  $X$ 
4:  $X^{best} = X$ 
5: Do
6:   Do
7:     Find Best Neighbor  $X'$  According to a Local Search Method
8:     If  $f(X') < f(X^{best})$  Then
9:       End Loop
10:    End If
11:   Loop Until ( $X'$  is not tabu)
12:   Add Move to Tabu List
13:    $X = X'$ 
14:   If  $f(X) < f(X^{best})$  Then
15:      $X^{best} = X$ 
16:   End If
17: Loop Until (Stopping Criteria)

```

Fig. 3 Pseudo code of the TS algorithm

of the SA and TS including the stochastic feature to escape local optima and tabu list to avoid cycling. The detail of the HMA is presented in the following subsections.

Presentation and initial solution

An integer string of length $(N + K)$, which provides an incentive to return to the original route information after decoding, is used to present a solution for the algorithm (Tan et al. 2001). An example for presentation of three routes is as follows, where the zeros represent the bounds of the routes:

String Code $0 \rightarrow 7 \rightarrow 3 \rightarrow 8 \rightarrow 10 \rightarrow 1 \rightarrow 0 \rightarrow 2 \rightarrow$
 $9 \rightarrow 0 \rightarrow 5 \rightarrow 4 \rightarrow 6 \rightarrow 0$
Route No. 1 $0 \rightarrow 7 \rightarrow 3 \rightarrow 8 \rightarrow 10 \rightarrow 1 \rightarrow 0$
Route No. 2 $0 \rightarrow 2 \rightarrow 9 \rightarrow 0$
Route No. 3 $0 \rightarrow 5 \rightarrow 4 \rightarrow 6 \rightarrow 0$

In order to obtain a feasible initial solution for the VRPCD-2D, an insertion based procedure, which primarily takes into account the free vehicle loading area, is applied for the HMA. At first $|K|$ empty routes are generated and non-occupied surface area of the vehicles $F_k (k \in K)$ is set to the $W \times L$. Locations are sorted by the value of $A_i (\forall i \in S \cup D)$, where A_i is the total area of products which have to pick up or drop off at location i . Then, each location i from the sorted sequence is inserted into the routes one by one by checking the 2-dimensional loading and time windows constraints. In this case, the location type is another factor to choose appropriate vehicle, such that a supplier location can be inserted into the route if the route is empty or the route contains another supplier location. Likewise, a customer location can be inserted into the route if the route is empty or the route contains another customer location. As a result of feasibility checking process, the initial solution procedure progress with respect to following three conditions:

- If the location i can be inserted feasibly into the only one route, then it is assigned to appropriate route.
- If location i can be inserted feasibly into the more than one route, then it is assigned to route that minimize $F_k - A_i$.
- If the location i cannot be inserted feasibly into the any route, then the sequence of locations is updated by exchanging two randomly selected locations and the construction procedure restarts with the new location sequence.

This above process is performed until all the locations that can be feasibly inserted into any route.

Neighborhood generation

In order to explore the search space, HMA employs the λ -interchange local search, which was first introduced by Osman and Christofides (1994). This methods is based on the λ number, where the λ is the maximum number of locations that can be interchanged between routes, and includes both shift and exchange procedures. The locations that are interchanged can be selected randomly or systematically. For example, consider $\lambda = 2$, which means that up to two locations may be interchanged between routes with the following operators: (0, 1), (1, 0), (1, 1), (0, 2), (2, 0), (1, 2), (2, 1) and (2, 2).

In this study, λ -interchange method is adapted with TS algorithm by using the tabu list property, in which the memory remembers the tabu moves for a certain number of iterations and does not allow using of any move until it leaves the tabu list. In addition to this prohibition, λ -interchange method ignores the any move that leads to infeasible solution for a route, which avoids the redundant computations for the HMA. As a consequence, any move of a location j can be inserted between location i and location $i + 1$ into the route k if the following conditions are met respectively;

- If the route is empty or both of the location i and location j are supplier or customer locations.
- $e_i + s_i + t_{ij} \leq l_j$.
- $e_j + s_j + t_{j(i+1)} \leq l_{(i+1)}$.
- If the total area of products at location i is less than the non-occupied surface area of the vehicle k .
- If the all products of location i can be loaded into the vehicle k successfully.

The sequence of the checking list is arranged according to the computational complexity of the examination and a move is rejected when it failures for a condition. At the end of the checking feasibility, the fitness function value of a new route is determined by using the distance between the locations.

For the computational studies, the proposed λ -interchange method is applied for $\lambda = 1$ and $\lambda = 2$ and best improvement strategy is implemented in selecting a new neighbor.

Packing heuristic

In order to determine whether all the products required by the locations can be feasibly loaded into the vehicles, five packing heuristics $Heur_h$ ($h = 1, \dots, 5$) developed by Zachariadis et al. (2009) are used in HMA. Each of the heuristics, which have different position selection criteria, loads a product into the vehicle from the feasible position list (pos_list) as follows:

Heur₁ Bottom Left Fill (W-axis) selects the position from the pos_list , which has the minimum *W-axis* coordinate, breaking ties by minimum *L-axis* coordinate.

Heur₂ Bottom Left Fill (L-axis) selects the position from the pos_list , which has the minimum *L-axis* coordinate, breaking ties by minimum *W-axis* coordinate.

Heur₃ Max Touching Perimeter Heuristic selects the position from the pos_list , which has the maximum touching perimeter value, which is evaluated as the sum of common edges of the inserted product with the edges of the already inserted products, and the edges of the loading surface of the vehicle.

Heur₄ Max Touching Perimeter No Walls Heuristic selects the position from the pos_list , which has the maximum touching perimeter value, which is evaluated as the sum of common edges of the inserted product with the edges of the already inserted products. Distinctly from the *Max Touching Perimeter Heuristic*, the common edges of the inserted product with the loading surface are not taken into account for this heuristic.

Heur₅ Min Area Heuristic selects the position from the pos_list , which has the minimum rectangle surface corresponding to position.

As mentioned in the problem definition part, it is assumed that the products can be loaded into the vehicles in unrestricted order. Thus, the sequence of the locations in a route does not affect the loading feasibility. When the packing procedure is carried out for a route, first the products belongs to the locations are sorted by the area in decreasing order. Then the five packing heuristics are called in sequence. If a feasible loading plan is found by a heuristic method, the packing procedure stops and returns the result as *success*. On the other hand, if the current heuristic fails to find a feasible solution, then the next heuristic is called. At the end of the fifth heuristic if a feasible solution cannot be found then the packing procedure returns the result as *failure*.

```

1: Function loading_check(k)
2: If Memory_Structure Contains the Route k Then
3:   If Route Information is success Then
4:     Return success
5:   Else If Route Information is fail Then
6:     Return fail
7:   End If
8: Else
9:   For h = 1 to 5
10:    If Heurh = success Then
11:      Add Route Information into the Memory_Structure as success
12:      Return success
13:    Else If h = 5 and Heurh = fail Then
14:      Add Route Information into the Memory_Structure as fail
15:      Return fail
16:    End If
17:  Next
18: End If

```

Fig. 4 Pseudo code of the loading check procedure

Because of the packing procedure is very time consuming, a memory structure is used to speed up the loading check computations, which avoids duplicate examinations. This structure is employed to store the loading feasibility information of a route, and when a route is examined, this information can be easily obtained if the route has been generated previously. Otherwise, the packing procedure is applied for the route and the new information is recorded to memory in a string form. The string form of the route includes the locations excepting the cross-docking center and loading information which are split with the use of comma: e.g. the route 0–1–2–3–4–5–6–7–8–9–10–0 and its loading information is *success* is coded as “1, 2, 3, 4, 5, 6, 7, 8, 9, 10, success”. To reduce the searching process time for retrieving the stored information of a route, this memory structure is designed in the form of a three dimensional array. Each dimension of the array indicates a key factor of a route: the first location of the route, the number of locations in the route and the occupied area of the vehicle in integer form. Instead of a single array, the proposed key factors prevent the unnecessary searches. Figure 4 shows the pseudo-code of the memory-structure-adapted loading check procedure.

Stopping criteria

The maximum iteration number is used as primary termination criteria for the proposed algorithm and the temperature is reduced progressively by the cooling parameter at each iteration. Additionally, the algorithm stops if there is no improvement in the objective function during the specified number of iterations. Based on the described statements above, the pseudo code of the proposed HMA is shown in Fig. 5.

Computational results

The proposed HMA developed in the Visual Basic programming language, and numerical experiments are performed

```

1: T = Initial temperature
2: c = Cooling parameter
3: Empty Tabu List
4: Xbest =  $\emptyset$ 
5: Generate Initial Solution X
6: Xbest = X
7: Do
8:   Randomly Specify the Value of the  $\lambda$ 
9:   Generate All Potential Neighbors by Using  $\lambda$ -Interchange Method
10:  Find Best Neighbor and Specify as X'
11:  Add Move to Tabu List and Update the List via FIFO Rule
12:  If  $f(X') \leq f(X)$  Then
13:    X = X'
14:  Else
15:     $\Delta = f(X') - f(X)$ 
16:    If  $\exp\left(-\frac{\Delta}{T}\right) > \text{random}[0,1]$  Then
17:      X = X'
18:    End If
19:  End If
20:  T = T  $\times$  c
21:  If  $f(X) < f(X^{\text{best}})$  Then
22:    Xbest = X
23:  End If
24: Loop Until (Stopping Criteria)

```

Fig. 5 Pseudo code of the HMA

on 2.20-GHz Intel Core i7 processor with 8-GB memory. To evaluate the performance of the HMA, a well-known benchmark problem data set, which was developed by Solomon (1987) to test the vehicle routing problem with time windows, is used and adapted to VRPCD-2D. Solomon’s benchmark problems consist of totally 100 randomly located demand nodes and a depot node. Also, smaller problems are regarded by just considering the first 25 and 50 nodes. For the VRPCD-2D, the locations of the instances are divided into two groups: supplier locations and customer locations, in which the time windows of the customer locations are adjusted to further time period while the time windows of supplier locations remained as the same. Moreover, because of the customers should request the products from more than one supplier location and adapt the instances to the packing problem, the demand amount of the customers are regenerated randomly with 2-dimensional physical data. The problems formed for the VRPCD-2D are described with the supplier and customer locations: e.g. the problem “R101_2D_5_20” considers the first 25 demand node of the R101 problem of the Solomon’s data set where the first 5 demand nodes are represented as supplier locations while the other 20 demand nodes are customer locations.

The computational studies consist of two parts. First, HMA, SA and TS are performed on 24 different small-sized instances and the results of the algorithms are compared with the CPLEX. Then, each algorithm is practiced on more complex problems include 25, 50 and 100 locations, respectively. In this part, the performance of the HMA is demonstrated by comparing the results with SA and TS solutions. Table 1 represents the principle data of the problems used to generate

Table 1 Characteristics of the generated problems

Experimental part	Number of locations	Number of supplier locations	Number of customer locations	Demand range ^a	Size of the products		Size of the vehicles	
					W	L	W	L
Part I	[14, 40]	[4, 12]	[7, 30]	[0, 2]	[1, 2]	[2, 4]	5	28
Part II	25	[5, 12]	[13, 20]	[0, 1]	[1, 3]	[3, 8]	20	40
Part II	50	[10, 25]	[25, 40]	[0, 1]	[1, 3]	[3, 8]	20	40
Part II	100	[15, 25]	[75, 85]	[0, 1]	[1, 3]	[3, 8]	20	40

^a Maximum number of demand value of a customer location requests from a supplier location

instances with their intervals and Table 2 reports the some details of the problems used in the second part of the computational studies which identify the problem size. Table 2 also presents the best integer solution of the first 12 problems with their optimality gaps obtained by CPLEX 12.5 with 5 h time limitation. For the other problems which include 50 or 100 locations, the CPLEX could not reach any integer solution in specified time horizon. For each part of the computational studies for heuristic algorithms, the maximum iteration number is set to the 1000 iterations and each run is terminated if the algorithm does not provide a new best solution during the 100 iteration. Moreover, the algorithms are executed 10 times for each instance and the results are identified with best and average result of the runs.

For the first part of the computational studies, Table 3 shows the characteristics of the problems and the result of CPLEX solutions with best bound solution, best integer solution, number of vehicles, optimality gap of the solution and CPU time of the computations. On the other hand, the results of the HMA, SA and TS are presented in Table 4 with best solutions and average solution. Also, the table presents the gap between the result of heuristic methods and results of best integer CPLEX solution by using the following formulation;

$$\%Gap_C = \frac{CPLEX\ Solution - Heuristic\ Solution}{CPLEX\ Solution} \times 100\%$$

It can be seen from the Table 3 that, an optimal solution can be found for nine problems within the specified time limitation. For the other problems, CPLEX could find the results with the optimality gap range 0.03 and 24.20 %. When the best results of the heuristic methods are compared with the best integer solution of CPLEX, HMA finds 1.25 % and SA finds 1.24 % better solution on average. However, the average %Gap_C of the TS for best results is less than zero. Similarly, the average solutions of the HMA and SA are better than the best integer solution of the CPLEX while the average %Gap_C of the TS is less than zero for most of the instances.

For the second part of the computational studies, Tables 5, 6 and 7, which are classified according to the number of locations, show the details of the results obtained by HMA, SA and TS. Each table compares the HMA solutions

with SA and TS solutions with respect to best and average solution by using the following formulation;

$$\%Gap_{Heuristic\ Method} = \frac{Heuristic\ Method\ Solution - HMA\ Solution}{Heuristic\ Method\ Solution} \times 100\%$$

where the positive variables pointed out with bold characters indicate that better solution is obtained by the HMA.

The results for 25 locations, which are reported in Table 5, show that proposed HMA could obtain the result with lower total distance for several problems with respect to best and average solutions. When the gaps are compared on the basis of best results, HMA provides 0.45 and 2.42 % better solution according to the SA and TS, respectively. Similarly, when the average solutions are considered the average gaps between the HMA and SA is 1.16 %, and between the HMA and TS is 3.17. In terms of the computational times, TS could reach the solutions in shorter times in general with respect to the HMA and SA.

For the 50 locations, the results are presented in Table 6. Regarding to the best solutions, HMA could obtain the best result for eight problems. For this case, there is only one problem that TS could obtain better result than the HMA. Considering the average results, HMA exhibits superior performance and finds better results for all problems except one problem. Additionally, when the results are compared according to the gaps of the best results, the average gap between the HMA and SA is 0.33 % and the average gap between the HMA and TS is 0.44 %. Moreover, the gaps rises up when the average solutions are considered, where the average gap between the HMA and SA is 1.00 % and the average gap between the HMA and TS is 1.45 %.

Finally, for the large sized problems, which consist of 100 locations, the results are shown in Table 7. The gaps between HMA and other two heuristics are significantly different, where the average gap between HMA and SA is 2.45 % and the average gap between HMA and TS is 3.24 %, insomuch that this value exceeds 5.00 % for some of the instances. Additionally, HMA could obtain best result for nine problems and obtain better average solution for all problems. On the other hand, SA exhibits better performance for 3 prob-

Table 2 Details of the problems

Number of locations	Problem description	Number of suppliers	Number of customers	Number of products	CPLEX solution	
					Best integer solution	Optimality gap (%)
25	C104_2D_6_19	6	19	61	222.0	35.27
	C107_2D_11_14	11	14	74	220.5	14.02
	C201_2D_11_14	11	14	90	303.3	8.44
	C208_2D_11_14	11	14	72	328.2	36.08
	R104_2D_5_20	5	20	52	538.7	14.05
	R112_2D_9_16	9	16	63	507.6	20.67
	R203_2D_12_13	12	13	97	559.8	16.40
	R205_2D_7_18	7	18	58	506.0	0.00
	RC103_2D_6_19	6	19	54	411.8	53.79
	RC105_2D_8_17	8	17	63	411.3	14.41
	RC207_2D_9_16	9	16	75	418.3	51.59
	RC208_2D_11_14	11	14	74	421.0	58.81
50	C107_2D_10_40	10	40	260	–	–
	C103_2D_14_36	14	36	206	–	–
	C204_2D_17_33	17	33	295	–	–
	C206_2D_15_35	15	35	267	–	–
	R106_2D_18_32	18	32	282	–	–
	R110_2D_14_36	14	36	271	–	–
	R205_2D_20_30	20	30	293	–	–
	R211_2D_15_35	15	35	277	–	–
	RC102_2D_10_40	10	40	218	–	–
	RC108_2D_22_28	22	28	304	–	–
	RC201_2D_19_31	19	31	296	–	–
	RC205_2D_13_37	13	37	251	–	–
100	C103_2D_25_75	25	75	734	–	–
	C106_2D_18_82	18	82	591	–	–
	C202_2D_24_76	24	76	740	–	–
	C203_2D_17_83	17	83	597	–	–
	R102_2D_16_84	16	84	538	–	–
	R105_2D_15_85	15	85	513	–	–
	R201_2D_20_80	20	80	633	–	–
	R210_2D_16_84	16	84	539	–	–
	RC102_2D_23_77	23	77	714	–	–
	RC107_2D_16_84	16	84	532	–	–
	RC201_2D_21_79	21	79	701	–	–
	RC205_2D_19_81	19	81	583	–	–

lems according to the best solution. On the basis of CPU times, each algorithm represents similar performance.

In addition to comparisons of the algorithms with respect to best and average solution, a statistical analysis for solutions are conducted by applying paired-*t* test at significance level $\alpha = 0.05$ to reveal whether there exists significant differences between HMA and other two heuristics in terms of the solution quality. Therefore, the null hypothesis is formed

as $H_0 : \mu_{HMA} - \mu_{CHA} = 0$ and two sided alternative hypothesis as $H_1 : \mu_{HMA} - \mu_{CHA} \neq 0$, where the symbols μ_{HMA} and μ_{CHA} represent the population mean for HMA and the other compared meta-heuristic algorithm, respectively. Table 8 presents the result of the paired-*t* test for the algorithms which are separated according to the location number of the problems. It can be seen from table that proposed HMA is significantly different from the TS for each problem

Table 3 CPLEX solution for the small-sized instances

Problem	Number of suppliers	Number of customers	Number of products	Best bound	Best integer solution	Number of vehicles	Optimality gap (%)	CPU time (s)
R101_2D_5_20	5	20	66	726.8	727.0	11	0.03	18,000
R101_2D_8_15	8	15	36	668.2	668.2	11	0.00	350
R101_2D_10_30	10	30	87	1035.8	1035.8	14	0.00	417
R102_2D_5_12	5	12	38	459.6	470.7	6	2.36	18,000
R102_2D_6_14	6	14	29	531.4	531.4	8	0.00	157
R102_2D_8_18	8	18	41	635.6	650.3	8	2.26	18,000
R103_2D_5_9	5	9	19	373.5	373.5	4	0.00	336
R103_2D_7_13	7	13	61	423.3	510.4	6	17.06	18,000
R103_2D_9_8	9	8	24	411.4	411.4	5	0.00	1503
R104_2D_5_13	5	13	23	367.3	409.0	4	10.20	18,000
R104_2D_6_12	6	12	24	354.8	408.1	4	11.30	18,000
R104_2D_8_10	8	10	51	364.4	432.8	6	15.80	18,000
R105_2D_7_15	7	15	32	571.4	571.4	8	0.00	220
R105_2D_9_26	9	26	68	797.9	797.9	10	0.00	1405
R105_2D_12_20	12	20	69	730.5	730.5	8	0.00	4284
R106_2D_6_17	6	17	32	477.2	517.1	6	9.57	18,000
R106_2D_7_12	7	12	54	458.6	501.0	7	8.46	18,000
R106_2D_6_27	6	27	46	607.4	671.2	7	9.50	18,000
R107_2D_5_10	5	10	61	380.5	428.2	5	11.20	18,000
R107_2D_4_18	4	18	45	362.7	482.4	5	24.20	18,000
R107_2D_8_11	8	11	58	393.5	465.7	5	15.50	18,000
R108_2D_5_17	5	17	29	348.7	421.9	4	17.40	18,000
R108_2D_9_12	9	12	32	364.3	408.0	4	10.70	18,000
R108_2D_11_7	11	7	26	346.0	346.0	3	0.00	2433
Average	7.12	15.25	43.79	507.95	540.41	6.63	6.90	11,712.71

set and also it can be concluded that HMA is significantly different from the SA except one case which is formed for the problems consist of 25 locations.

Conclusions

In this paper, the vehicle routing problem with the 2-dimensional truck loading constraints is studied for the cross-docking center in order to minimize total transportation cost. The problem is formulated as a mixed integer mathematical model and due to the complexity of the problem, a HMA is proposed as a solution approach for the problem. The HMA combine the SA and TS algorithm, which are the two well-known powerful meta-heuristic algorithms to solve combinatorial problems. The HMA takes the advantages of SA and TS including the stochastics feature to escape from local optima and tabu list to avoid cycling. Moreover, five packing heuristic methods are integrated with the algorithm to check 2-dimensional loading feasibility. The proposed algorithm is tested with several instances which are obtained

from Solomon’s benchmark data sets and compared with the SA and TS methods. First, the algorithms are performed for small sized instances and compared based on the best integer solution of CPLEX. Then, the performance of the algorithms are analyzed on more complex problems. In addition to the comparisons of the algorithms with respect to best and average solution, a statistical analysis for solutions are conducted by applying paired-*t* test whether there exists significant differences between HMA and other two heuristics in terms of the solution quality. The computational results show that the proposed HMA exhibits superior performance and outperforms the SA and TS method with respect to the solution quality. On the other hand, the average computational times of the algorithms are significantly different. As a consequence, this study contributes to the literature by considering the 2-dimensional vehicle loading operations for VRPCD. Distinctly from the existing studies on VRPCD, the truck capacities are identified with the physical dimensional limits on the contrary of weight or amount of load in VRPCD-2D, which provides more practical and realistic plan for pickup and delivery operations. Also an efficient solution method-

Table 5 Computational results for 25 locations

Problem	SA			TS			HMA								
	Best solution			Average solution			Best solution			Average solution					
	Distance	CPU time (s)	%Gap _{SA}	Distance	CPU time (s)	%Gap _{SA}	Distance	CPU time (s)	%Gap _{SA}	Distance	CPU time (s)	%Gap _{SA}			
C104_2D_6_19	221.5	78.08	222.3	222.3	96.26	220.8	220.8	78.16	222.3	62.48	220.8	221.0	108.09	0.56	0.57
C107_2D_11_14	221.2	14.14	221.2	221.2	13.06	221.2	221.2	10.64	221.2	10.91	221.2	221.2	9.44	0.00	0.00
C201_2D_11_14	302.3	27.65	302.3	302.3	33.70	302.3	302.3	27.53	302.3	27.45	302.3	302.3	17.82	0.00	0.00
C208_2D_11_14	243.8	1.83	249.7	249.7	1.47	249.2	242.3	1.37	250.4	1.75	242.3	247.2	6.56	1.02	1.28
R104_2D_5_20	529.3	10.83	529.3	529.3	11.37	529.3	529.3	9.44	552.3	7.41	529.3	529.3	17.09	0.00	4.17
R112_2D_9_16	494.9	8.63	495.6	495.6	10.56	494.9	494.9	7.41	503.0	8.93	494.9	494.9	14.38	0.15	1.62
R203_2D_12_13	533.0	43.07	536.5	536.5	43.49	533.0	533.0	26.31	538.2	32.28	533.0	533.0	24.80	0.65	0.97
R205_2D_7_18	506.0	33.13	520.5	520.5	63.50	506.0	506.0	27.84	524.2	47.97	506.0	506.0	39.10	2.78	3.46
RC103_2D_6_19	407.4	19.26	408.0	408.0	15.56	404.5	401.0	17.82	407.6	14.12	401.0	403.5	26.75	1.10	1.02
RC105_2D_8_17	424.7	2.94	449.3	449.3	3.29	465.7	412.5	3.46	465.7	2.90	412.5	416.2	8.89	7.38	10.64
RC207_2D_9_16	338.2	48.34	339.3	339.3	48.93	392.6	338.2	7.08	392.6	7.50	338.2	338.3	38.98	0.30	13.84
RC208_2D_11_14	301.9	5.64	302.2	302.2	5.52	302.2	301.9	5.57	303.4	4.06	301.9	302.1	4.39	0.01	0.42
Average	377.02	24.46	381.34	381.34	28.89	385.14	375.28	18.55	390.26	18.98	375.28	376.24	26.36	1.16	3.17

Table 6 Computational results for 50 locations

Problem	SA			TS			HMA							
	Best solution			Best solution			Best solution							
	Distance	CPU time (s)	Average solution	Distance	CPU time (s)	Average solution	Distance	CPU time (s)	Average solution					
		Distance	CPU time (s)	Distance	CPU time (s)	Distance	CPU time (s)	Distance	CPU time (s)					
C107_2D_10_40	511.8	1151.99	517.9	1175.13	512.5	735.70	518.7	590.56	510.6	650.83	512.4	646.95	1.06	1.22
C103_2D_14_36	466.0	664.20	469.3	625.76	466.0	250.01	472.5	295.50	466.0	374.14	466.0	510.22	0.69	1.38
C204_2D_17_33	693.2	1351.98	704.1	1289.48	689.4	760.40	702.8	628.90	680.8	1438.38	689.0	1305.67	2.14	1.97
C206_2D_15_35	657.6	2013.53	664.4	2684.80	666.1	1537.11	674.7	1191.35	657.2	2499.66	661.2	2106.59	0.48	1.99
R106_2D_18_32	998.0	106.62	1003.2	122.97	997.5	63.03	1005.1	66.41	997.5	110.06	998.2	109.70	0.50	0.69
R110_2D_14_36	897.1	142.99	914.3	241.38	897.1	124.47	916.3	124.42	897.1	347.59	914.8	183.81	-0.06	0.16
R205_2D_20_30	924.4	413.42	938.4	697.29	927.3	302.57	943.1	312.51	922.9	389.50	933.3	484.77	0.53	1.04
R211_2D_15_35	799.4	2189.93	819.4	2931.85	794.5	1709.52	827.4	1083.83	796.5	2401.84	813.4	1747.67	0.73	1.69
RC102_2D_10_40	1093.1	55.38	1122.7	60.49	1093.1	55.95	1128.9	56.00	1091.1	58.49	1110.3	53.16	1.11	1.65
RC108_2D_22_28	1007.3	235.14	1014.7	182.13	1007.3	129.54	1017.8	179.01	1003.3	255.52	1009.1	180.11	0.55	0.85
RC201_2D_19_31	1077.6	414.53	1097.6	553.30	1077.6	355.08	1105.4	396.00	1074.3	533.69	1077.7	374.95	1.81	2.51
RC205_2D_13_37	935.4	504.60	971.4	1111.14	943.4	970.97	969.8	888.22	931.9	995.98	947.7	790.08	2.43	2.28
Average	838.4	770.36	853.1	972.98	839.3	582.86	856.9	484.39	835.8	837.98	844.4	707.81	1.00	1.45

Table 7 Computational results for 100 locations

Problem	SA			TS			HMA						
	Best solution			Best solution			Best solution						
	Distance	CPU time (s)	Average solution	Distance	CPU time (s)	Average solution	Distance	CPU time (s)	Average solution				
		Distance	CPU time (s)	Distance	CPU time (s)	Distance	CPU time (s)	Distance	CPU time (s)				
C103_2D_25_75	1420.1	2301.11	1436.6	1956.46	1432.2	1806.61	1459.7	1894.44	1403.0	1756.69	1615.64	1.82	3.37
C106_2D_18_82	1272.6	2124.81	1306.1	2341.98	1316.7	2306.07	1334.9	2319.24	1281.4	2750.13	2389.74	1.25	3.38
C202_2D_24_76	1597.8	2560.54	1620.5	2521.83	1601.8	2654.24	1625.5	2236.39	1596.9	2352.28	2231.91	0.68	0.99
C203_2D_17_83	1395.5	1723.09	1420.0	605.57	1415.7	1683.82	1447.8	871.01	1357.0	1022.07	881.19	1.27	3.16
R102_2D_16_84	1825.9	1085.12	1850.3	1779.90	1836.6	627.12	1856.3	1262.92	1808.5	2303.04	1233.98	0.77	1.09
R105_2D_15_85	1733.8	694.50	1759.1	512.33	1736.9	211.86	1764.6	490.56	1702.5	775.54	533.37	1.90	2.20
R201_2D_20_80	1686.0	2654.06	1738.6	2231.33	1753.5	2301.40	1775.8	2069.13	1758.6	2759.36	2434.66	2.15	4.20
R210_2D_16_84	1415.8	2304.21	1469.5	2398.24	1415.8	2107.43	1481.3	1914.86	1432.4	2534.05	2774.73	1.97	2.75
RC102_2D_23_77	2193.2	1954.60	2238.0	1891.89	2184.0	1358.45	2224.0	1713.38	2170.1	1875.08	1401.09	2.19	1.58
RC107_2D_16_84	1843.0	1457.04	1852.3	2442.66	1843.0	1181.20	1851.4	2340.63	1731.0	2981.79	2000.15	5.97	5.93
RC201_2D_21_79	2218.8	2616.13	2301.1	2111.81	2224.7	2356.46	2318.8	2252.82	2158.7	2348.47	1819.88	4.48	5.21
RC205_2D_19_81	1914.7	1744.11	1949.6	1749.14	1924.9	1103.23	1951.8	1447.56	1819.7	1077.55	1474.40	4.91	5.01
Average	1709.8	1934.94	1745.1	1878.60	1757.7	1641.49	1757.7	1734.41	1685.0	2044.67	1732.56	2.45	3.24

Table 8 Results of the paired-*t* test

Number of locations	HMA–SA			HMA–TS		
	Mean of the difference	Standard deviation of the difference	<i>p</i> value	Mean of the difference	Standard deviation of the difference	<i>p</i> value
25	−5.1	9.7146	0.0692	−14.0	19.1428	0.0230
50	−8.7	7.3907	0.0025	−12.5	7.4692	0.0001
100	−34.5	29.2113	0.0024	−47.5	27.1320	0.0001

ology is introduced to solve VRPCD-2D, which is expressly better than the SA and TS. For the future work on this study, the HMA can be improved by considering more than one cross-docking center, which allows splitting the customer request in network. Also, the HMA can be built up to solve VRPCD with 3-dimensional loading plans or VRPCD-2D with heterogeneous fleet. Another research can be done by extending the VRPCD-2D with truck-door assignment plans in cross-docking center, which directly affects the distribution starting time of the vehicles.

References

- Agustina, D., Lee, C. K. M., & Piplani, R. (2014). Vehicle scheduling and routing at a cross docking center for food supply chains. *International Journal of Production Economics*, *152*(1), 29–41.
- Arikan, M., & Erol, S. (2012). A hybrid simulated annealing-tabu search algorithm for the part selection and machine loading problems in flexible manufacturing systems. *The International Journal of Advanced Manufacturing Technology*, *59*(5–8), 669–679.
- Armentano, V. A., & Yamashita, D. S. (2000). Tabu search for scheduling on identical parallel machines to minimize mean tardiness. *Journal of Intelligent Manufacturing*, *11*(5), 453–460.
- Azadeh, A., Farahani, M. H., Eivazy, H., Nazari-Shirkouhi, S., & Asadipour, G. (2013). A hybrid meta-heuristic algorithm for optimization of crew scheduling. *Applied Soft Computing*, *13*(1), 158–164.
- Chen, C. S., Sarin, S., & Ram, B. (1991). The pallet packing problem for non-uniform box sizes. *International Journal of Production Research*, *29*(10), 1963–1968.
- Dondo, R., & Cerdá, J. (2013). A sweep-heuristic based formulation for the vehicle routing problem with cross-docking. *Computers and Chemical Engineering*, *48*(1), 293–311.
- Dondo, R., & Cerdá, J. (2014). A monolithic approach to vehicle routing and operations scheduling of a cross-dock system with multiple dock doors. *Computers and Chemical Engineering*, *63*(1), 184–205.
- Dondo, R., Méndez, C. A., & Cerdá, J. (2011). The multi-echelon vehicle routing problem with cross docking in supply chain management. *Computers and Chemical Engineering*, *35*(12), 3002–3024.
- Dong, M.-G., & Wang, N. (2012). A novel hybrid differential evolution approach to scheduling of large-scale zero-wait batch processes with setup times. *Computers and Chemical Engineering*, *45*(1), 72–83.
- Fuellerer, G., Doerner, K. F., Hartl, R. F., & Iori, M. (2009). Ant colony optimization for the two-dimensional loading vehicle routing problem. *Computers and Operations Research*, *36*(3), 655–673.
- Gendreau, M., Iori, M., Laporte, G., & Martello, S. (2006). A tabu search algorithm for a routing and container loading problem. *Transportation Science*, *40*(3), 342–350.
- Gendreau, M., Iori, M., Laporte, G., & Martello, S. (2008). A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks*, *51*(1), 4–18.
- Geyik, F., & Cedimoglu, I. H. (2004). The strategies and parameters of tabu search for job-shop scheduling. *Journal of Intelligent Manufacturing*, *15*(4), 439–448.
- Glover, F. (1989). Tabu search—Part I. *ORSA Journal on Computing*, *1*(3), 190–206.
- Glover, F. (1990). Tabu search—Part II. *ORSA Journal on Computing*, *2*(1), 4–32.
- Hamta, N., Ghomi, S. M. T. F., Tavakkoli-Moghaddam, R., & Jolai, F. (2014). A hybrid meta-heuristic for balancing and scheduling the assembly lines with sequence-independent setup times by considering deterioration tasks and learning effect. *Scientia Iranica, Transactions E: Industrial Engineering*, *21*(3), 963–979.
- Hasani-Goodarzi, A., & Tavakkoli-Moghaddam, R. (2012). Capacitated vehicle routing problem for multi-product cross-docking with split deliveries and pickups. *Procedia-Social and Behavioral Sciences*, *62*, 1360–1365.
- Hu, Z.-H., Zhao, Y., & Choi, T.-M. (2013). Vehicle routing problem for fashion supply chains with cross-docking. *Mathematical Problems in Engineering*, *2013*, 1–10.
- Jolai, F., Rabiee, M., & Asefi, H. (2012). A novel hybrid meta-heuristic algorithm for a no-wait flexible flow shop scheduling problem with sequence dependent setup times. *International Journal of Production Research*, *50*(24), 7447–7466.
- Katsigiannis, Y., Georgilakis, P. S., & Karapidakis, E. S. (2012). Hybrid simulated annealing-tabu search method for optimal sizing of autonomous power systems with renewables. *IEEE Transactions on Sustainable Energy*, *3*(3), 330–338.
- Kaviani, M. A., Abbasi, M., Rahpeyma, B., & Yusefi, M. M. (2014). A hybrid tabu search-simulated annealing method to solve quadratic assignment problem. *Decision Science Letters*, *3*(3), 391–396.
- Khebbache-Hadji, S., Prins, C., Yalaoui, A., & Reghioui, M. (2013). Heuristics and memetic algorithm for the two-dimensional loading capacitated vehicle routing problem with time windows. *Central European Journal of Operations Research*, *21*(2), 307–336.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671–680.
- Ladier, A.-L., & Alpan, G. (2014). Crossdock truck scheduling with time windows: Earliness, tardiness and storage policies. *Journal of Intelligent Manufacturing*, 1–15. doi:10.1007/s10845-014-1014-4.
- Landrieu, A., Mati, Y., & Binder, Z. (2001). A tabu search heuristic for the single vehicle pickup and delivery problem with time windows. *Journal of Intelligent Manufacturing*, *12*(5–6), 497–508.
- Lee, Y. H., Jung, J. W., & Lee, K. M. (2006). Vehicle routing scheduling for cross-docking in the supply chain. *Computers and Industrial Engineering*, *51*(2), 247–256.
- Leno, I. J., Sankar, S. S., & Ponnambalam, S. G. (2015). MIP model and elitist strategy hybrid GA–SA algorithm for layout design. *Journal of Intelligent Manufacturing*, 1–19. doi:10.1007/s10845-015-1113-x.

- Leung, S. C. H., Zhang, Z., Zhang, D., Hua, X., & Lim, M. K. (2013). A meta-heuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints. *European Journal of Operational Research*, 225(2), 199–210.
- Leung, S. C. H., Zhou, X., Zhang, D., & Zheng, J. (2011). Extended guided tabu search and a new packing algorithm for the two-dimensional loading vehicle routing problem. *Computers and Operations Research*, 38(1), 205–215.
- Liao, C.-J., Lin, Y., & Shih, S. C. (2010). Vehicle routing with cross-docking in the supply chain. *Expert Systems with Applications*, 37(10), 6868–6873.
- Lin, S.-W., Chou, S.-T., Lee, C.-Y., & Lee, Z.-J. (2012). A hybrid approach for vehicle routing problem with time windows. *Advances in Intelligent Transportation Systems*, 1(1), 11–18.
- Lin, S.-W., & Ying, K.-C. (2009). Applying a hybrid simulated annealing and tabu search approach to non-permutation flowshop scheduling problems. *International Journal of Production Research*, 47(5), 1411–1424.
- Moghadam, S. S., Ghomi, S. F., & Karimi, B. (2014). Vehicle routing scheduling problem with cross docking and split deliveries. *Computers and Chemical Engineering*, 69(1), 98–107.
- Morais, V. W. C., Mateus, G. R., & Noronha, T. F. (2014). Iterated local search heuristics for the vehicle routing problem with cross-docking. *Expert Systems with Applications*, 41(16), 7495–7506.
- Nearchou, A. C. (2004). Flow-shop sequencing using hybrid simulated annealing. *Journal of Intelligent Manufacturing*, 15(3), 317–328.
- Osman, I. H., & Christofides, N. (1994). Capacitated clustering problems by hybrid simulated annealing and tabu search. *International Transactions in Operational Research*, 1(3), 317–336.
- Otten, R. H. J. M., & Van Ginneken, L. P. P. P. (1988). Stop criteria in simulated annealing. *IEEE International Conference on Computer Design: VLSI in Computers and Processors*, 549–552. doi:10.1109/ICCD.1988.25760.
- Poorzahedy, H., & Rouhani, O. M. (2007). Hybrid meta-heuristic algorithms for solving network design problem. *European Journal of Operational Research*, 182(2), 578–596.
- Ramezani, P., Rabiee, M., & Jolai, F. (2015). No-wait flexible flowshop with uniform parallel machines and sequence-dependent setup time: A hybrid meta-heuristic approach. *Journal of Intelligent Manufacturing*, 26(4), 731–744.
- Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2), 254–265.
- Tan, K. C., Lee, L. H., Zhu, Q. L., & Ou, K. (2001). Heuristic methods for vehicle routing problem with time windows. *Artificial intelligence in Engineering*, 15(3), 281–295.
- Tarantilis, C. D. (2013). Adaptive multi-restart tabu search algorithm for the vehicle routing problem with cross-docking. *Optimization Letters*, 7(7), 1583–1596.
- Vahdani, B., & Sadigh, B. S. (2014). A multi-product vehicle routing scheduling model with time window constraints for cross docking system under uncertainty: A fuzzy possibilistic-stochastic programming. *International Journal of Industrial Mathematics*, 6(3), 215–228.
- Vahdani, B., Tavakkoli-Moghaddam, R., Zandieh, M., & Razmi, J. (2012). Vehicle routing scheduling using an enhanced hybrid optimization approach. *Journal of Intelligent Manufacturing*, 23(3), 759–774.
- Van Belle, J., Valckenaers, P., & Cattrysse, D. (2012). Cross-docking: State of the art. *Omega*, 40(6), 827–846.
- Wei, L., Oon, W.-C., Zhu, W., & Lim, A. (2011). A skyline heuristic for the 2D rectangular packing and strip packing problems. *European Journal of Operational Research*, 215(2), 337–346.
- Wei, L., Zhang, Z., Zhang, D., & Lim, A. (2015). A variable neighborhood search for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, 243(3), 798–814.
- Wen, M., Larsen, J., Clausen, J., Cordeau, J.-F., & Laporte, G. (2009). Vehicle routing with cross-docking. *Journal of the Operational Research Society*, 60(12), 1708–1718.
- Zachariadis, E. E., Tarantilis, C. D., & Kiranoudis, C. T. (2009). A guided tabu search for the vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, 195(3), 729–743.
- Zarandi, M. F., Khorshidian, H., & Shirazi, M. A. (2014). A constraint programming model for the scheduling of JIT cross-docking systems with preemption. *Journal of Intelligent Manufacturing*, 1–17. doi:10.1007/s10845-013-0860-9.