International Journal of Engineering

Journal Homepage: www.ije.ir

Modeling the Time Windows Vehicle Routing Problem in Cross-docking Strategy Using Two Meta-heuristic Algorithms

M. B. Fakhrzad[a*], A. Sadri Esfahani[b]

Department of Industrial Engineering, Faculty of Engineering, Yazd University, Yazd, Iran

PAPER INFO

Paper history:
Received 19 May 2013
Received in revised form 22 October 2013
Accepted in 21 November 2013

Keywords:
Cross-docking Strategy
Vehicle Routing Problem
Time Windows
Tabu Search
Variable Neighborhood Search

ABSTRACT

In cross docking strategy, arrived products are immediately classified, sorted and organized with respect to their destination. Among all the problems related to this strategy, the vehicle routing problem (VRP) is very important and of special attention in modern technology. This paper addresses the particular type of VRP, called VRPCDTW, considering a time limitation for each customer/retailer. This problem is known as NP-hard problem. Two meta-heuristic algorithms based on the Tabu search (TS) algorithm and variable neighborhood search (VNS) are proposed for its solution. These algorithms are designed for real-world cases and can be generalized to the more complex models such as those which deliveries can be specified in a split form. The proposed TS algorithm also offers a candidate list strategy which has no limitation for the number of nodes and vehicles. A computational experiment is performed to verify our presented algorithms. Through computational experiments, it is indicated that the proposed TS algorithm performs better than VNS algorithm in both aspects of the total cost and computation time.

doi: *10.5829/idosi.ije.2014.27.07a.13*

1. INTRODUCTION

Over the recent years, many companies confronted with more complicated and various customer demands. Thus, many companies are trying to achieve high level of agility, flexibility and reliability for various demands [1]. In the real world, the production procedure consists of purchasing raw material from suppliers, producing or manufacturing, storing and delivering the final product to customers. These systems that start by suppliers and end by customers are called supply chain systems [2]. In such systems, operations of a single company necessarily make no improvement in customer's satisfaction, because its operations may have interacting effects or even adverse effects on other companies in the supply chain system [1]. For this reason, nowadays, supply chain management is one of the most attractive issues in operational management. Apte and Viswanathan [3] express that over 30% of goods price is incurred in distribution process. Thus, the efficient solution on inventory control and distribution management is a vital success factor for companies [4]. In addition, distribution and control of the flow of inventory is one of the major concepts in supply chain management [2]. Typically, five distinct distribution strategies are utilized in the supply chain management. First, strategy is direct shipment in which items are directly shipped from suppliers to the retail stores without going through distribution centers. Milk run is the second distribution strategy. A milk run is a route in which a vehicle/truck delivers product from a single supplier to multiple retailers [5]. Third strategy is known as hub and spoke (H&S). H&S network involves a series of nodes (hubs), connected by arcs (spokes) that represent viable transportation alternatives between two nodes [5]. Pool distribution is the forth strategy. Pool distribution is the distribution of orders to numerous destination points within a particular geographic region. In this strategy, instead of shipping direct from origin supplier to retailer, orders are directly shipped to the regional terminals and then shipped to the retail stores [5]. Cross-docking is the fifth distribution strategy that recently has been regarded [6].

*Corresponding Author Email: mfakhrzad@yazduni.ac.ir (M. B. Fakhrzad)

Please cite this article as: M. B. Fakhrzad, A. Sadri Esfahani, Modeling the Time Windows Vehicle Routing Problem in Cross-docking Strategy Using Two Meta-heuristic Algorithms. International Journal of Engineering (IJE) Transactions A: Basics, Vol. 27, No. 7, (2014): 1113-1126

1. 1. Cross-docking Apte and Viswanathan [7] introduced cross-docking as one of the most strategic and technologic innovations in the supply chain management. In this system, cross docks/distribution centers function as inventory coordination points rather than as inventory storage points [6]. In the cross-docking, all deliverable products arrive to the cross dock and are immediately categorized, sorted and organized according to their destinations and customers' demand. Then, these products are moved to respective destinations, without storing in the cross dock. In other words, in the cross-docking systems, a few stocks are handled in temporary storage [8]. In this strategy, products usually are stored less than 12 hours at the cross dock ([3, 9]), sometime less than an hour [10]. Figure 1 illustrates the flow of material in the cross docking.

Cross-docking includes two key processes that are depicted in Figure 1. The pickup process is the product/material flow from supplier to the cross dock and the delivery process is the product/material flow from cross dock to customer. The key issues in the pickup and delivery processes are the simultaneous arrival at the cross dock and consolidation, respectively. The purpose of consolidation is to categorize and to sort the products at the cross dock with respect to their destinations. Regarding to the consolidation process, some of vehicles have to unload their entire burden completely and reload another product(s). Also, some of the vehicles have to unload partial of their products and reload another product(s) and some of the vehicles which have extra capacity, are only loaded with the new product(s). The purpose of simultaneous arrival to the cross dock is to reduce the waiting time of vehicles. If vehicles of the pickup fleet could not arrive at the cross-dock simultaneously, then the consolidation process has been postponed. Thereby, it might increase the waiting time and the inventory level at the cross dock [2].

1. 2. Literature Review and Research Motivation Recently, many studies treated various issues of cross-docking from different viewpoints. These investigations can be divided into two categories: studies that focus on (1) physical aspects of cross docking and (2) operational aspects of cross-docking.

Most of the first-category studies describe the cross-docking concept and its advantages [11-13], physical design of cross dock in cross-docking [3, 14-18], and optimal location of cross dock [19-21]. On the other hand, most of the investigations related to the second-category focused on the trucks scheduling in cross-docking system [8, 22-31]. In addition, over the last 30 years or so, the classical VRP has attended strongly in the literature. The classical VRP involves the service of a set of customers with known demands by a fleet of vehicles from a single distribution center [2].

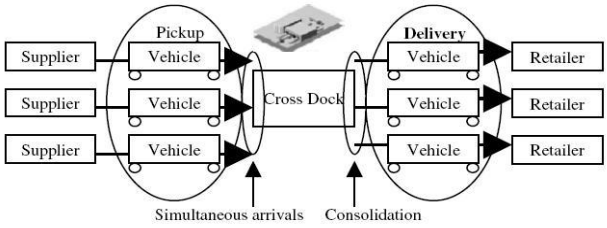

Figure 1. The concept of cross-docking [1].

The main objective of the problem is to design a set of routes starting and ending at the distribution center/depot such that all customers are serviced and the total cost of the set of routes is minimized [32]. Some of the most recent VRP papers, such as those have been published by Lin et al. [33], Cheng and Wang [34], Catay [32], Mirabi et al. [35] and Zachariadis and Kiranoudis [36]. Mosheiov [37] considered the pickup and delivery problems as a VRP problem and proposed the two heuristic algorithms to find a good solution to minimize transportation cost and maximize the efficiency of vehicles. VRP with time windows (VRPTW) can be very helpful encountering with cross-docking problems [1], because one the core issue in such problems is the simultaneous arrival at the cross dock. In general, time window models can be divided to three categories:

VRP with hard time window (VRPHTW): In such models, vehicles have to service customers in the specific time interval and any violation from the service time window HWV is not admissible for customer i, whatsoever.

VRP with soft time window (VRPSTW): In such models, vehicles are allowed to service customers before and after the earliest and latest time window bounds, respectively. If any violation occurs from the service time window $[a_i, b_i]$, by introducing appropriate penalties, measure of customer's non-satisfaction is reflected [38].

VRP with hard and soft time window (VRPHSTW): Such models are combination of the two mentioned models. Time window in these models include a hard and a soft interval. Hard interval cannot be violated and soft interval can be violated [38].

Comparatively, a few number of research projects considered both cross-docking and VRP simultaneously. Earlier, Lee et al. [1] proposed a tabu search algorithm (TS) to determine the number of vehicles and the optimal vehicle routing schedule at a cross-dock to minimize the sum of transportation cost and fixed cost of vehicles. Liao et al. [2] developed new TS algorithm and compared its performance with Lee et al.'s TS [1]. Dondo et al. [39] 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. Hasani-Goodarzi and Tavakkoli-Moghaddam [40] considered a split vehicle routing problem (SVRP) with capacity constraint for multi-product cross-docks. Mousavi and Tavakkoli-Moghaddam [41] presented a two-stage mixed-integer programming (MIP) model for the location of cross-docking centers and vehicle routing scheduling problems with multiple cross-docking centers due to potential applications in the distribution networks. To the best of our knowledge, there have been a few papers that take both cross-docking and time scheduling with time windows into consideration simultaneously. Ma et al. [42] proposed a new shipment consolidation and transportation problem in cross-docking distribution networks by considering setup cost and time window constraint. Ma et al. [42] considered direct rout (supplier-to-retailer) as well as indirect rout (supplier-to-cross dock and cross dock-to-retailer). Ma et al. [42] also assumed that products can be stored at the cross-dock for relatively long times. Dondo and Cerdá [43] introduced a sweep heuristic for the VRPCD that determines pickup/delivery routes and schedules simultaneously with the truck scheduling at the terminal. Dondo and Cerdá [43] assumed that the pickup/delivery nodes should be visited within specific hard time windows. Unlike Ma et al. [42] and Dondo and Cerdá [43], this paper considers the multi-commodity consolidation and soft time windows for each delivery node. In this paper, it is assumed that all deliverable products are moved to respective destination without storing in the cross-dock. It is also assumed that direct shipping from the suppliers to retailers is not allowed. In this paper, the primary motivation is to present two algorithms for the solution of the above-described complex model of cross docking strategy.

This paper is organized as follows: section 2 presents the model assumptions and formulation. Section 3 describes the tabu search and variable neighborhood search algorithms for VRPCDTW and presents the steps of the algorithms. Section 4 compares the performance of the proposed algorithm. Finally, section 5 concludes the paper and presents the future research.

2. MODEL ASSUMPTIONS AND FORMULATION

The problem considered in this study, namely VRPCDTW, is VRP cross-docking with time window. The problem formulation is based on the following assumptions:

- This problem is goods transportation from a set of suppliers to a set of corresponding customers/retailers through a cross-dock. Direct shipping from the suppliers to retailers is not allowed.
- A soft time window is considered for each customer.
- A set of identical vehicles is used to transport goods from supplies to retailers.
- Consolidation process must be accomplished at the cross dock.
- The whole process must be completed in the planning horizon.
- Each supplier or retailer can only picked up or delivered once. Each location (pick up/delivery node) cannot be visited by the same vehicle more than once.
- No intermediate storage in the cross-dock is allowed.
- There are no pre-defined vehicles for some suppliers and retailers.

The main objectives are to determine the number of vehicles and the best routes and schedules to minimize the sum of transportation cost, vehicle operation cost and time window violation cost. The notations (sets, parameters and decision variables) and mathematical formulation are as follows. It should be noted that some of the used notations in the proposed model, are similar to that of Wen et al. [10]:

P : set of nodes in the pickup process

D : set of nodes in the delivery process

0 : cross dock index

n : number of suppliers/retailers

NV : number of available vehicles

Q : capacity of the vehicle(pallet)

A : the fixed time for loading, unloading and reloading at the cross-dock and each node

B : the time for loading, unloading and reloading a pallet at the cross-dock and each node

tc_{ij} : transportation cost from node i to node j

o_v : operation cost of vehicle v . It should be noted that all of the vehicles have an identical operation cost.

t_{ij} : travel time between node i and j

p_i : number of pallets loading in pickup node i

d_i : number of pallets unloading in delivery node i

DT_i^v : departure time of vehicle v from node i

AT^v : arrival time of vehicle v at the cross dock

S_i^v : service start time of vehicle v at node i

ye_{vi} : amount of start time earliness of vehicle v at node i

yl_{vi} : amount of start time lateness of vehicle v at node i

P_e : unit penalty cost for earliness

P_l : unit penalty cost for lateness

e_i : lower bound of hard time window at node i

l_i : upper bound of hard time window at node i

LB_i : lower bound of soft time window at node i

UB_i : upper bound of soft time window at node i

M : very big number

T : time horizon

Decision variables:

x_{ij}^v **:** if the vehicle v move from the node i to the node j , 1, otherwise,0. ($i, j \in P$ or D)

u_i^v : if the vehicle v unload the goods i at the cross dock,1, otherwise,0. ($i \in P$)

r_i^v : if the vehicle v load the goods i at the cross dock,1, otherwise,0. ($i \in P$)

tu^v : the time at which vehicle v finishes unloading at the cross dock

tr^v : the time at which vehicle v starts reloading at the cross dock

v_i : the time at which goods i is unloaded at the cross dock

Mathematical model:

$$Min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{v=1}^{NV} tc_{ij}\, x_{ij}^v + 2\sum_{v=1}^{NV}\sum_{j=1}^{n} o_v\, x_{0j}^v + \sum_{\substack{i=1 \\ i\in D}}^{n}\sum_{v=1}^{NV}(P_e.ye_{vi} + P_l.yl_{vi}) \quad (1)$$

s.t.

$$\sum_{i=1}^{n}\sum_{v=1}^{NV} x_{ij}^v = 1, \forall j \quad (2)$$

$$\sum_{j=1}^{n}\sum_{v=1}^{NV} x_{ij}^v = 1, \forall i \quad (3)$$

$$\sum_{j=1}^{n} x_{0j}^v \le 1, \forall v \quad (4)$$

$$\sum_{i=1}^{n} x_{i0}^v \le 1, \forall v \quad (5)$$

$$\sum_{i=1}^{n} x_{ip}^v - \sum_{j=1}^{n} x_{pj}^v = 0, \forall p, v \quad (6)$$

$$\sum_{v=1}^{NV}\sum_{j=1}^{n} x_{0j}^v \le NV, \forall v \quad (7)$$

$$\sum_{\substack{i=1 \\ i,j\in P}}^{n} p_i x_{ij}^v \le Q, \forall v \quad (8)$$

$$\sum_{\substack{i=1 \\ i,j\in D}}^{n} d_i x_{ij}^v \le Q, \forall v \quad (9)$$

$$DT_j^v = (t_{ij} + DT_i^v + A + p_i.B).x_{ij}^v, \forall v, j \in P \quad (10)$$

$$DT_j^v = (t_{ij} + DT_i^v + A + d_i.B).x_{ij}^v, \forall v, j \in D \quad (11)$$

$$\sum_{\substack{i=1 \\ i\in P}}^{n} p_i = \sum_{\substack{i=1 \\ i\in D}}^{n} d_i \quad (12)$$

$$\sum_{\substack{i=1 \\ i\in P}}^{n}\sum_{\substack{j=1 \\ j\in P}}^{n}(A + p_i.B).x_{ij}^v + \sum_{\substack{i=1 \\ i\in D}}^{n}\sum_{\substack{j=1 \\ j\in D}}^{n}(A + d_i.B).x_{ij}^v + \sum_{\substack{i=1 \\ i\in P}}^{n}\sum_{\substack{j=1 \\ j\in P}}^{n} t_{ij}.x_{ij}^v + \sum_{\substack{i=1 \\ i\in D}}^{n}\sum_{\substack{j=1 \\ j\in D}}^{n} t_{ij}.x_{ij}^v \le T, \forall v \quad (13)$$

$$AT^v = (DT_i^v + t_{i0}).x_{i0}^v, \forall v, i \in P \quad (14)$$

$$AT^v = AT^{v'}, \forall v \ne v' \quad (15)$$

$$S_j^v + M(1 - x_{ij}^v) - DT_i^v - t_{ij} \ge 0, \forall v, i, j \quad (16)$$

$$LB_i \le S_i^v \le UB_i, \forall v, i \in D \quad (17)$$

$$ye_{vi} \ge e_i - S_i^v, i \in D \quad (18)$$

$$yl_{vi} \ge S_i^v - l_i, \forall v, i \in D \quad (19)$$

$$u_i^v + r_i^v \le 1, \forall v, i \in P \quad (20)$$

$$u_i^v - r_i^v = \sum_{j\in P\cup 0} x_{ij}^v - \sum_{j\in D\cup 0} x_{i+n,j}^v, \forall v, i \in P \quad (21)$$

$$tr^v \ge tu^v, \forall v \quad (22)$$

$$tu^v \ge v_i - M(1 - u_i^v), \forall v, i \in P \quad (23)$$

$$tr^v \ge v_i - M(1 - r_i^v), \forall v, i \in P \quad (24)$$

$$x_{ij}^v, u_i^v, r_i^v \in \{0,1\}, \forall v, i, j \in P or D \quad (25)$$

$$tu^v, tr^v, v_i \ge 0 \quad (26)$$

The objective function of this problem is expressed by Equation (1). It is tried to minimize the sum of transportation cost, operation cost of vehicles and, earliness and lateness of vehicles cost. Equations (2) and (3) show that one vehicle has to arrive at and leave one node. Equations (4) and (5) show that one vehicle has to leave the cross dock from one node and arrive at one node at the cross dock. Equation (6) expresses the consecutive movement of vehicles. Equation (7) shows that the number of vehicles that leave the cross-dock must be less than the number of available vehicles, *NV*. Equations (8) and (9) express that the quantity of loaded products in a certain vehicle cannot exceed the maximum capacity of the vehicle in the pickup and delivery processes, respectively. Equations (10) and (11) express that the departure time of a vehicle from node, j , is determined by the sum of the departure time of a vehicle from previous node , i, the length of a visit (sum of the fixed time and variable time for loading, unloading and reloading), and the time to move.

Equilibrium equation is shown in Equation (12). In Equation (13), the sum of the total length of the visit to each node and total transportation time must be less than the planning horizon, T. The arrival time at a cross-dock is represented by Equation (14). The constraint for simultaneous arrival to a cross-dock is given in Equation (15). Equation (16) determines the service start time for each node. The constraint for soft time windows is represented in Equation (17). Equations (18) and (19) determine the amount of earliness and lateness of a start time. This fact that a vehicle, v should unload or reload product, i, depends on its pickup and delivery routes is expressed by Equation (20). Equation (21) shows the linkage between the pickup and delivery process in the consolidation decisions at the cross dock. Equation (22) indicates that a vehicle cannot start reloading until it finishes unloading. Equations (23) and (24) express the time at which vehicle, v finishes unloading and starts reloading at a cross dock.

3. META-HEURISTICS FOR THE VRPCDTW

Since VRPCDTW is considered as a NP-hard problem [1], an efficient meta-heuristic method is needed to achieve a good solution in a reasonable amount of time. TS was successfully applied to solve the various types of VRP [1, 2, 44, 45]. VNS was also successfully applied to solve the multi depot routing problem [46] and scheduling the trucks in cross-docking systems [26]. In this paper, two meta-heuristics algorithms based on TS and VNS are presented to solve the VRPCDTW. Sections 3.1 and 3.2 are devoted to the TS-based and VNS-based algorithms, respectively.

3. 1. A TS-based Meta-heuristic for the VRPCDTW

TS is an iterative local search algorithm, which cycling back to previously visited solution is prevented by the use of memories. TS was originally developed by Glover [47]. Some of the basic components of TS method are: initial solution, neighborhood structure, stopping criteria, tabu list and aspiration criteria. TS, at each iteration, explores the solution space and tries to make the best possible moves from the current solution x to the best solution x' in its neighborhood $N(x)$, even if the move may deteriorate the objective function value. A tabu mechanism is put in place to prevent the process from cycling over a sequence of solutions. TS exploits tabu list to prevent cycling and local optima. Some attributes of the past solutions are registered and any solution possessing these attributes may not be considered. Temporarily are also declared tabu for θ iterations (it is called tabu tenure). However, tabu moves can be overridden if the aspiration criterion is satisfied. Here, a TS-based heuristic to solve the VRPCDTW is developed.

3. 1. 1. Initial Solution Similar to the other local search algorithms, TS is an iterative procedure that starts from an initial solution. This solution is the starting point for subsequence exploration in the solution space. Here, an initial solution scheme for VRPCDTW is introduced.

Pickup process:

1) Calculate the minimum number of vehicles.

$$NV_{\min} = \sum p_i / Q$$

2) Sequence vehicles in descending order of the remaining space. In the stage of initialization, all of the vehicles are available at the cross dock. They are empty and sort according to their index. $S_1, S_2, \ldots, S_{NV_{\min}}$

3) Determine all possible routes that the first vehicle can be moved in the sequence of step 2. Calculate the ratio of transportation cost to the minimum transportation cost between the determined routes. Generate the candidate list of routes that their calculated ratios are less than α. Select a route and its related node randomly from the candidate list.

4) Replicate the steps 2 and 3 until all nodes are assigned. If the remaining capacity of a vehicle is less than the supply of node i, select next vehicle in the sequence of step 2. If no vehicle was found, add one additional vehicle to the set of vehicles.

Delivery process

1) Calculate the remaining time for delivery process,. $t_d = T - t_p$ where, t_d is the remaining time for delivery process, T is the time horizon and t_p is the completion time of the pickup process.

2) All vehicles travel to the corresponding customer without any consolidation. For example, if the first vehicle is made pickups in the nodes 1, 2 and 4, thus, shipment of this vehicle must be delivered to the customers 1, 2 and 4.

3. 1. 2. Objective Function Evaluation The proposed TS algorithm is based on that of Cordeau et al. [48], in which the infeasible solutions are allowed during the search.

According to the initial solution scheme, the time horizon and time windows may be violated. In addition, these violations can be occurred during the search process. Thus, a penalized objective function $f(s)$ is considered to evaluate each solution generated from the neighborhood of the current solution. The presented objective function evaluation procedure is based on that of Wen et al. [10]. It is defined as follows:

$$f(s)=c(s)+\alpha TV(s)+\beta HWV(s)+M.SWV(s) \quad (27)$$

where, $f(s)$ is the objective function in the algorithm iterations, $c(s)$ is the original objective function defined in (1), TV, HWV and SWV are the violation measures for delivery process remaining time, hard time windows and soft time windows, respectively. If the solution is feasible for these three constraints, TV, HWV and SWV are equal to 0. α and β are the penalty coefficients for the time horizon and hard time windows violation, respectively. In order to satisfy the soft time windows, a big penalty coefficient (M) is considered for SWV. TV, HWV and, SWV are defined as follows:

$$TV=\sum_{v=1}^{NV}(\sum_{j\in D}(S_j^v+A+d_jB+t_{j0}).x_{j0}^v-t_d) \quad (28)$$

$$HWV=\sum_{v=1}^{NV}[\sum_{i\in D}[\sum_{j\in D}((S_i^v+A+d_i.B).x_{ij}^v-l_i)^+]]+\sum_{v=1}^{NV}[\sum_{i\in D}[\sum_{j\in D}(e_i-S_i^v.x_{ij}^v)^+]] \quad (29)$$

$$SWV=\sum_{v=1}^{NV}[\sum_{i\in D}[\sum_{j\in D}((S_i^v+A+d_i.B).x_{ij}^v-UB_i)^+]]+\sum_{v=1}^{NV}[\sum_{i\in D}[\sum_{j\in D}(LB_i-S_i^v.x_{ij}^v)^+]] \quad (30)$$

where $(x)^+=\max(0,x)$.

It should be noted that the initial solution is designed in such a manner that the vehicles capacity have no violation. However, during the search algorithm, vehicle capacity may be violated. Therefore, a penalty coefficient for the capacity violation measure is considered. According to the above discussion, $f(s)$ is modified as follows:

$$f(s)=c(s)+\alpha TV(s)+\beta HWV(s)+M.SWV(s)+\gamma CV(s) \quad (31)$$

where, CV and γ are the capacity violation measure and its penalty coefficient, respectively.

3. 1. 3. Neighborhood Structure Neighborhood structure is the transformation mechanism that applies on the current solution to generate candidate solution. Insertion and exchange are the most simple and famous mechanisms to generate neighboring solution in the heuristic algorithms. Recently, $2-opt^*$ and CROSS mechanisms have generated good solutions. In the insertion mechanism, one node i removes from its original vehicle k and re-inserts to another vehicle k'. However, in the exchange mechanism, two nodes belonging to the two vehicles are swapped. $2-opt$ operator is applied in this paper. One vehicle is selected randomly and then two routes whose transportation cost between two nodes is more than the other nodes are found. These two selected routes are exchanged with the two corresponding routes in another randomly selected vehicle. If there are not any corresponding routes in the two randomly selected vehicles, then, the sequence of the two routes is changed in the first selected vehicle.

3. 1. 4. Candidate List Strategy For a given solution, x, it is computationally too expensive to explore its whole neighborhoods. Thus, in the proposed algorithm, instead of examining all neighborhoods, $N(x)$, a candidate list consisting of vehicles which have the most number of nodes is examined. For example, if the number of vehicles is more than a certain number (i.e. 5 vehicles), the known percent of vehicles (i.e. 50%) which have the most number of nodes are examined.

3. 1. 5. Tabu Status and Tabu List One of the TS's objectives is to encourage the exploration of parts of the solution space that have not been visited previously [45]. The complete solutions are not kept in the tabu list. The attributive memory is used for the tabu list, and the e-attributes of an accepted move are stored [49]. For $2-opt$ operator, the tabu status of the solution is defined by the vertex pair (i, j). These two vertices are stored in the tabu list, and any solution possessing these attributes may not be considered and temporarily declare the tabu for θ iterations. In this paper, two tabu lists are defined for both pickup and delivery processes. Each tabu list is an $n\times n$ matrix, where element $TABU_j(i,k)$ specifies the tabu status of arc (i,k) in $TABU_j$. If $TABU_j(i,k)\le 0$, arc (i,k) is not a tabu; otherwise tabu.

3. 1. 6. Aspiration Criterion In the TS algorithm, tabus may prohibit attractive moves, even when there is no danger of cycling. Hence, it is necessary to use algorithmic devices to allow one to revoke tabus. These are called aspiration criterion. In this paper, similar to the almost all of the TS implementation, most commonly aspiration criterion is used. In this aspiration criterion, a tabu move can be overridden if it has less objective value than the best solution found so far.

3. 1. 7. Stopping Criterion In this paper, when the maximum iteration bound determined by the user is reached, the search is stopped. The iteration number is calculated by the following formula:

$$E\times(1+\frac{NV-1}{F}) \quad (32)$$

where, E and F are parameters and NV is the number of vehicles. Hence, the number of iterations increases with NV.

3. 1. 8. Proposed Algorithm Steps Before describing the algorithm flow and presenting the flowchart of the algorithm, some parameters are introduced in the following:

n_{c_P}: current number of the generated candidate solutions, in the pickup side;

n_{c_D}: current number of the generated candidate solutions, in the delivery side;

$n_{\max}$: maximum number of the candidate moves;

A_P: set of candidate solutions in the pickup process;

A_D: set of candidate solutions in the delivery process;

$count$: counter of iterations;

NUI: current number of iterations without improvement;

$iteration_{\max}$: maximum number of algorithm iterations;

$NUI_{\max}$: maximum number of iterations allowed without improvement;

x^*: the best-found solution;

η: adjusting coefficient for the penalty coefficients;

Step 1 Initialization

1.1) Generate an initial solution (x) based on 3.1.1.

1.2) Set the algorithm parameters, α, β, γ, η, $NUI_{\max}$, tabu tenure(θ), $TABU(i,k)=\phi$, E, F, $count=1$, and $NUI=0$. Calculate $iteration_{\max}$ based on (32).

1.3) set: $x^* \leftarrow x$ and $f(x^*) \leftarrow f(x)$.

Step 2 Generate the admissible solution in the pickup side

Set $n_{c_P}=1$ and $A_P=\phi$.

2.1) If $n_{c_P} \geq n_{\max}$, go to step 3.

2.2) Based on section 3.1.4, determine the vehicle candidate list for generating the neighborhoods. Apply predetermined neighbor-generating method described in section 3.3. A solution $x_{n_{c_P}}$ is generated from the neighborhood $N(x)$ of x and added to the candidate solution. Set, $A_P \leftarrow A_P \cup x_{n_{c_P}}$, $n_{c_P} \leftarrow n_{c_P}+1$ and go to step 2.

Step 3 Select the best move in the pickup side amongst A_P

Evaluate all the solutions in A_P according to section 3.2. Put all non-tabu solution in $N'(x)$, set $i=1$, $x_{best_P}=x_{initial}$, $f(x_{best_P})=f(x_{initial})$. Put all tabu solutions in $N''(x)$.

3.1) If $i > n_{\max}$, go to step 4.

3.2) If $x_i \in N'(x) \cup N''(x)$ and $f(x_i) \leq f(x_{bestP})$, set: $x_{best\,P} \leftarrow x_i$, $i \leftarrow i+1$ and go to step 3.1.

Step 4 Generate the admissible solution in the delivery side

Set $n_{c_D}=1$ and $A_D=\phi$.

4.1) If $n_{c_D} \geq n_{\max}$, go to step 5.

4.2) Based on section 3.1.4, determine the vehicle candidate list from the solution obtained in step 3, for generating the neighborhoods. Apply exchange operator according to section *3.3.* A solution $x_{n_{c_D}}$ is generated from the neighborhood $N(x)$ of x and added to the candidate solution. Set: $A_D \leftarrow A_D \cup x_{n_{c_D}}$, $n_{c_D} \leftarrow n_{c_D}+1$ and go to step 4.1.

Step 5 Select the best move in the delivery side amongst A_D

Evaluate all solutions in A_D according to section 3.1.2. Put all non-tabu solution in $N'(x)$, set $i=1$, $x_{best}=x_{bestP}$, $f(x_{best})=f(x_{best_P})$. Put all tabu solutions in $N''(x)$.

5.1) If $i > n_{\max}$, go to step 6.

5.2) If $x_i \in N'(x) \cup N''(x)$ and $f(x_i) \leq f(x_{best})$, set: $x_{best} \leftarrow x_i$. $i \leftarrow i+1$ and go to step 5.1.

Step 6 Update the statistical information

If $f(x_i) \geq f(x_{best})$, set $NUI \leftarrow NUI+1$ and go to step 1. Otherwise: set $x^* \leftarrow x_{best}$ and $f(x^*) \leftarrow f(x_{best})$.

Step 7 Update the memory structure

Update the tabu list according to x^*. Set $TABU_j(i,k) \leftarrow TABU(i,k)+1$, $j=1,2$ where the indices 1 and 2 demonstrate the pickup and delivery processes, respectively.

Step 8 Update the penalty coefficients

If the current solution is feasible with respect to each items of the objective functions (TV, HWV, SWV and CV), the value of the corresponding penalty coefficient is divided by $1+\eta$; otherwise, it is multiplied by $1+\eta$. Set $count \leftarrow count+1$.

Step 9 Stopping

If $count < iteration_{\max}$ and $NUI < NUI_{\max}$, go to step 2; otherwise show the best x solution.

The flowchart of the proposed algorithm is given in Figure 2. In addition, the pseudo code of the TS algorithm is given below.

1. **Begin**
2. Generate an initial solution.
3. Set the initial solution as the current and also the best solution.
4. Generate an empty list for keeping the frequency of the solution changes in the pickup side.
5. Generate an empty list for keeping the frequency of the solution changes in the delivery side.
6. **While** stopping criterion is not met **do**
7. **For** $n_{c_p}=1$ to $n_{\max}=1$

Search the neighborhood of the current solution in the pickup side.

$current\ solution \leftarrow neighborhood\ of\ the\ current\ solution$.

Then update the tabu list.

8. **If**

$current\ solution\ objective\ function < best\ solution\ objective\ function$

then

9. $best\ solution \leftarrow current\ solution$
10. **Endif**
11. **Endfor**
12. $current\ solution \leftarrow best\ solution$
13. **For** $n_{c_d}=1$ to $n_{\max}=1$

Search the neighborhood of the current solution in the delivery side.

$current\ solution \leftarrow neighborhood\ of\ the\ current\ solution$.

Then, update the tabu list.

14. **If**

$current\ solution\ objective\ function < best\ solution\ objective\ function$

$best\ solution \leftarrow current\ solution$

15. **Endif**
16. **Endfor**
17. **Endwhile**
18. **End**

3. 2. A VNS-Based Meta-heuristic for the VRPCDTW VNS is a new meta-heuristic for solving the combinatorial and global optimization problems that proposes systematic changes of the neighborhood structure during the search. VNS originally proposed by Hansen and Mladenovic [50]. Basic idea of the VNS is a systematic changing of neighborhood both within a descent phase to find a local optimum and within a perturbation phase to get out of the corresponding valley [51]. VNS explores close and then increasingly far neighborhoods of the best known solution in a probabilistic way. In other word, VNS applies a local search procedure repeatedly to get from neighboring solutions to local optima [51]. Because of relying on very few parameters, such as stopping criterion and number of neighborhoods, VNS is very easy to implement. A very comprehensive study about VNS can be found in Hansen and Mladenovic [50, 52]. Here, a VNS-based heuristic to solve the VRPCDTW is developed.

3. 2. 1. Initial Solution Similar to the other meta-heuristic algorithm, an initial feasible solution is necessary to start VNS procedure. In this paper, initial solution scheme described in TS algorithm is applied, as initial solution.

3. 2. 2. Neighborhood Structures In almost all of the meta-heuristic algorithms, one or more neighborhood structures is utilized as a means of defining admissible moves to transition from one solution to another solution. Because of the good performance, $2-opt$ operator is applied in this paper. One vehicle is randomly selected and then two routes whose transportation cost between two nodes is more than the other nodes are found. These two selected routes are exchanged with the two corresponding routes in another randomly selected vehicle. If there are not any corresponding routes in the two randomly selected vehicles, then, the sequence of the two routes is changed in the first selected vehicle.

3. 2. 3. Shake Procedure The shake procedure generates a x' at random from neighborhood of x, i.e. $x' \in N(x)$.

3. 2. 4. Stopping Criterion In this paper, the stopping criterion is determined by the maximum number of iterations between two improvements. In this paper, the iteration number is calculated by formula (32).

3. 2. 5. Proposed Algorithm Procedure

Step 1 Initialization

1.1) Generate an initial solution (x) based on 3.1.1.

1.2) Determine the neighborhood structure. In this paper, $N_{2-opt}(.)$.

1.3) Set the algorithm parameters, α, β, γ, E, F, $count=1$. Calculate $iteration_{\max}$ based on formula (32).

1.4) set: $x^* \leftarrow x$ and $f(x^*) \leftarrow f(x)$.

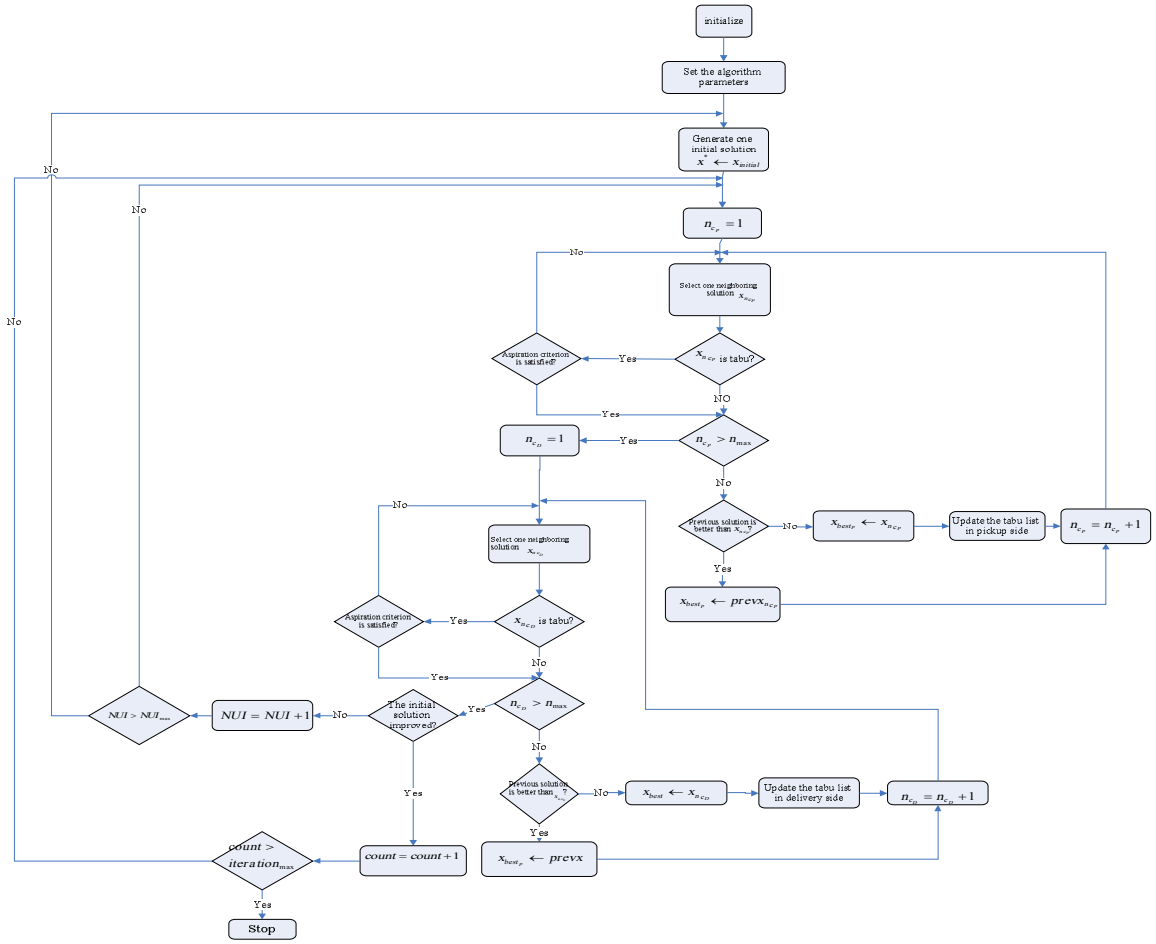

Figure 2. Flowchart of the proposed algorithm.

Step 2 shake and local search in pickup and delivery sides

2.1) Shake routine in the pickup side. Find a random solution $x' \in N_{2-opt}(x)$.

2.2) Local search in the pickup side. Apply neighborhood structure on x' to find a solution x_{pickup}.

2.3) Shake routine in the delivery side. Find a random solution $x'' \in N_{2-opt}(x)$.

2.4) Local search in the delivery side. Apply neighborhood structure on x'' to find a solution $x_{delivery}$.

2.5) If $f(x_{pickup}) < f(x^*)$ then $x^* \leftarrow x_{pickup}$. If $f(x_{delivery}) < f(x^*)$ then $x^* \leftarrow x_{delivery}$.

Set: $count \leftarrow count + 1$.

Step 3 stopping

If $count < iteration_{\max}$ then $x \leftarrow x^*$ and go to 2.1, else show x^*.

The pseudo code of the VNS algorithm is as follows:

1. **Begin**
2. Generate an initial solution (x).
3. Set the initial solution as the best solution (x^*).
4. **While** stopping criterion is not met **do**
5. **Repeat** the following steps
6. $x \leftarrow x^*$
7. Shake routine in the pickup side. Find a random solution $x' \in N_{2-opt}(x)$.
8. Local search in the pickup side. Apply neighborhood structure on x' to find a solution x_{pickup}.
9. Shake routine in the delivery side. Find a random solution $x'' \in N_{2-opt}(x)$.
10. Local search in the delivery side. Apply neighborhood structure on x'' to find a solution $x_{delivery}$.
11. **If** $f(x_{pickup}) < f(x^*)$ **then**
12. $x^* \leftarrow x_{pickup}$
13. **Break**
14. **If** $f(x_{delivery}) < f(x^*)$ **then**

15. $x^* \leftarrow x_{delivery}$
16. **Break**
17. **Endif**
18. **Endif**
19. **Endwhile**

End

4. COMPUTATIONAL EXPREMENTS

The aim of this section is to compare the performance of the two proposed algorithms described in section 3. It should be noted that, to verify and validate the mathematical model, several small size test problems were solved by the Lingo 8 software and optimal results were obtained. Optimal results of the four instances are presented in Table 2.

To evaluate the performance of the proposed algorithms, a computational study is carried out. For this purpose, the proposed algorithms are coded by using Visual Basic programming language. In the implemented programs, route of each vehicle is determined as well as respective costs, such as transportation cost, operation cost and earliness and lateness cost. The performances of the proposed algorithms are verified by comparing the obtained results from TS and VNS with the lower bound solution. The lower bound solution is developed by relaxing the constraints (17)-(19) that related to the time windows. To have a fair comparison both algorithms were executed on a special laptop. Due to lack of benchmark problems in the literature, randomly generated problems are considered. The approach of this paper to randomly design of the time windows is similar to Xiangyong et al. [45] approach as follows: The hard time window is determined by randomly generated cen_i and time width tw, ie., $[e_i, l_i]=\left[cen_i - \frac{1}{2}tw, cen_i + \frac{1}{2}tw\right]$. cen_i is an integer randomly generated from interval $[0 + t_{0i}, t_d - t_{i0} - (A + d_i B)]$ for each node $i \in D$, where $t_{0i} = t_{i0}$.Time width tw is an integer uniformly generated from the range $[10, 30]$. The soft time window is specified by $[LB_i, UB_i]=\left[e_i - \frac{1}{2}tw, l_i + \frac{1}{2}tw\right]$. Following Lee et al. [1] other parameter values are as follows: capacity of the vehicle (pallet) $Q = 70$, travel time between node i and j, $t_{ij} \equiv uniform(20, 200)$, transportation cost from node i to node j, $tc_{ij} \equiv uniform(48, 480)$, number of pallets loading in pickup node i and number of pallets unloading in delivery node i, $p_i, d_i \equiv uniform(5, 50)$. Also, operation cost of vehicle v, o_v, unit penalty cost for earliness, P_e, unit penalty cost for lateness, P_l, the fixed time for loading, unloading and reloading at the cross-dock and each node, A, and the time for loading, unloading and reloading a pallet at the cross-dock and each node, B, are specified by 100, 5, 5, 10 min and 1 min, respectively. In this paper, the approach of the parameters setting is similar to response surface methodology (RSM). For this purpose, critical factors of the TS algorithm that are statistically significant in aspects of performance and CPU time have been identified. Critical factors of the TS algorithm based on Vahdani and Zandieh [26] are maximum iteration ($iteration_{\max}$), maximum number of iterations allowed without improvement ($NUI_{\max}$), maximum number of the candidate moves ($n_{\max}$) and tabu tenure (θ). According to the determined neighborhood structure, the critical factor of the VNS algorithm is $iteration_{\max}$. By preliminary experiments, parameters of the proposed algorithms have been tuned, with which the algorithm had a relatively better performance and CPU time. Parameter setting is presented in Table 1.

Total cost comparison results of the two proposed algorithms versus Lingo results for the sample instances are provided in Table 2. The time horizon T is supposed to be 600 min.

Comparison results show that while using the Lingo software, the solution time is exponentially increased by increasing the number of nodes, whereas the proposed algorithms can solve the problem in much less time. Table 2 also shows that the presented algorithms can generate near optimal solution for the sample problems. Therefore, these algorithms can efficiently solve large size problems in the logical time. Comparative results of total costs and computation times of the TS and VNS algorithms as well as the lower bound solution for randomly generated large size instances are provided in Table 3 and Table 4, respectively. It should be noted that the value of each instance was reported from the average of 10 repetitions. Percentage gap between the total costs of the two presented algorithms and lower bound solution is calculated according to the following formula (I).

$$Gap = \frac{total\ cost\ of\ the\ proposed\ algorithm - total\ cost\ of\ the\ lower\ bound\ solution}{total\ cost\ of\ the\ lower\ bound\ solution} \times 100 \quad (I)$$

TABLE 1. Parameter setting

Parameter	Definition	Value for TS	Value for VNS
α	penalty coefficients for the time horizon violation	100	100
β	penalty coefficients for hard time windows violation	100	100
γ	penalty coefficient for capacity violation	50	50
η	adjusting coefficient for penalty coefficients	2	-
E	parameter for $iteration_{\max}$ based on formula (32)	60	80
F	parameter for $iteration_{\max}$ based on formula (32)	3	2
$NUI_{\max}$	maximum number of iterations allowed without improvement	$3n$	-
$n_{\max}$	maximum number of the candidate moves	50	-
θ	tabu tenure	$iteration_{\max}/2$	-

TABLE 2. Total cost comparison results of the proposed algorithms versus Lingo

Problem	No. of available vehicles	No. of nodes in each sides (pickup and delivery)	TS result		VNS result		Lingo	
			Total cost	cpu time (s)	Total cost	cpu time (s)	Total cost	cpu time (s)
VRPCDTW 1	5	5	2583.3	45	2590	48	2501	32
VRPCDTW 2	5	6	2936.75	61	3020	60	2899	48
VRPCDTW 3	5	7	3145.86	65	3266.9	69	3046.32	1180
VRPCDTW 4	5	8	4257.9	150	4320	151	3998.38	26000

TABLE 3. Total cost obtained by TS algorithm for the large size problems (T=600 min)

Problem	No. of available vehicles	No. of nodes in each sides (pickup and delivery)	Lower bound solution	TS result		Gap (%)
			Total cost	Total cost	cpu time (s)	
VRPCDTW5	10	15	7277	8364	100	14.94
VRPCDTW 6	10	20	8834	10036	210	13.61
VRPCDTW 7	20	25	12035	13299	275	10.50
VRPCDTW 8	20	30	12480	14437	486	15.68
VRPCDTW 9	30	40	15980	18907	840	18.32
VRPCDTW 10	30	50	19750	25003	1022	26.60
VRPCDTW 11	50	60	36551	40167	2306	9.89
VRPCDTW 12	50	70	49383	60225	3126	21.95
VRPCDTW 13	60	80	65511	78930	3840	20.48
VRPCDTW 14	60	90	89127	100128	4209	12.34
					Average gap (%)	**16.43**

TABLE 4. Total cost obtained by VNS algorithm for the large size problems (T=600 min)

Problem	No. of available vehicles	No. of nodes in each sides (pickup and delivery)	Lower bound solution	TS result		Gap (%)
			Total cost	Total cost	cpu time (s)	
VRPCDTW5	10	15	6327	9053	121	43.09
VRPCDTW 6	10	20	7474	10101	234	35.15
VRPCDTW 7	20	25	11229	14573	290	29.78
VRPCDTW 8	20	30	13590	15986	503	17.63
VRPCDTW 9	30	40	15963	19716	886	23.51
VRPCDTW 10	30	50	18768	25353	1120	35.09
VRPCDTW 11	50	60	39496	42931	2583	8.70
VRPCDTW 12	50	70	53282	61243	3220	14.94
VRPCDTW 13	60	80	60859	80074	3920	31.57
VRPCDTW 14	60	90	80800	101005	4602	25.01
					Average gap (%)	**26.45**

As depicted in Table 3, the average gap between the total costs of the TS solution and lower bound solution is 16.43% that is desirable. In addition, Table 4 shows that the average gap between the total costs of the VNS solution and lower bound solution is 26.45%. As a result, computational experiments indicate that the proposed TS algorithm performs better than VNS algorithm in aspect of the total cost.

5. CONCLUSION AND FUTURE RESEARCH

This paper presented a new model, namely VRPCDTW, integrating cross-docking with VRPTW. Since this problem is categorized as a NP-hard problem, two meta-heuristic algorithms based on the TS and VNS were proposed for its solution. The proposed algorithms were shown to have the adequate flexibility while encountering with the real-world cases. A computational experiment was carried out to compare the performance of the proposed algorithms. Experimental results showed that the proposed algorithms were capable of solving the large size problems in the logical time. Because of presenting a candidate list strategy, computational experiments indicated that the proposed TS algorithm performs better than VNS algorithm in both aspects of the total cost and computation time. Future research can be suggested in a few directions. It is interesting to solve the proposed model by new meta-heuristic algorithms and compare their results with the results of the proposed algorithms. Another extension is to consider the fuzzy time window in the both sides of the cross-dock (pickup and delivery sides). Future research can be considered the multiple cross-docks, capacity constraint at the cross-dock, direct shipment and non-identical vehicles.

6. REFERENCES

1. Lee, Y.H., Jung, J.W. and Lee, K.M., "Vehicle routing scheduling for cross-docking in the supply chain", ***Computers & Industrial Engineering***, Vol. 51, No. 2, (2006), 247-256.
2. Liao, C.-J., Lin, Y. and Shih, S.C., "Vehicle routing with cross-docking in the supply chain", ***Expert systems with applications***, Vol. 37, No. 10, (2010), 6868-6873.
3. Apte, U.M. and Viswanathan, S., "Effective cross docking for improving distribution efficiencies", ***International Journal of Logistics***, Vol. 3, No. 3, (2000), 291-302.
4. Wang, J.-L., "A supply chain application of fuzzy set theory to inventory control models–drp system analysis", ***Expert systems with applications***, Vol. 36, No. 5, (2009), 9229-9239.
5. Altekar, R.V., "Supply chain management: Concepts and cases, PHI Learning Pvt. Ltd., (2005).
6. Simchi-Levi, D., "Designing and managing the supply chain concepts strategies and case studies, Tata McGraw-Hill Education, (2009).
7. Apte, U.M. and Viswanathan, S., "Strategic and technological innovations in supply chain management", ***International Journal of Manufacturing Technology and Management***, Vol. 4, No. 3, (2002), 264-282.
8. Boloori Arabani, A., Fatemi Ghomi, S. and Zandieh, M., "Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage", ***Expert Systems with Applications***, Vol. 38, No. 3, (2011), 1964-1979.
9. Kreng, V.B. and Chen, F.-T., "The benefits of a cross-docking delivery strategy: A supply chain collaboration approach", ***Production Planning and Control***, Vol. 19, No. 3, (2008), 229-241.
10. Wen, M., Larsen, J., Clausen, J., Cordeau, J.-F. and Laporte, G., "Vehicle routing with cross-docking", ***Journal of the Operational Research Society***, Vol. 60, No. 12, (2009), 1708-1718.
11. Schaffer, B., "Cross docking can increase efficiency", ***Automatic I.D. News***, Vol. 14, No. 34-37.
12. Allen, J., "Cross docking: Is it right for you", ***Canadian Transportation Logistics***, Vol. 104, (2001), 22-25.
13. Luton, D., "Keep it moving: A cross-docking primer", ***Materials Management and Distribution***, Vol. 48, No. 5, (2003), 29.
14. Tsui, L.Y. and Chang, C.-H., "An optimal solution to a dock door assignment problem", ***Computers & Industrial Engineering***, Vol. 23, No. 1, (1992), 283-286.
15. Ratliff, H.D., Vate, J.V. and Zhang, M., "Network design for load-driven cross-docking systems", ***Georgia tech tli report, The Logistics Institute, Georgia Tech***, Vol., No., (1999).
16. Gue, G. and Bartholdi, J., "Reducing labor costs in an ltl cross-docking terminals", ***Operations Research***, Vol. 48, No. 6, (2000), 823-832.
17. Bartholdi, J.J. and Gue, K.R., "The best shape for a crossdock", ***Transportation Science***, Vol. 38, No. 2, (2004), 235-244.
18. Vis, I.F. and Roodbergen, K.J., "Positioning of goods in a cross-docking environment", ***Computers & Industrial Engineering***, Vol. 54, No. 3, (2008), 677-689.
19. Sung, C. and Song, S., "Integrated service network design for a cross-docking supply chain network", ***Journal of the Operational Research Society***, Vol. 54, No. 12, (2003), 1283-1295.
20. Gümüş, M. and Bookbinder, J.H., "Cross-docking and its implications in location-distribution systems", ***Journal of Business Logistics***, Vol. 25, No. 2, (2004), 199-228.
21. Ross, A. and Jayaraman, V., "An evaluation of new heuristics for the location of cross-docks distribution centers in supply chain network design", ***Computers & Industrial Engineering***, Vol. 55, No. 1, (2008), 64-79.
22. Donaldson, H., Johnson, E.L., Ratliff, H.D. and Zhang, M., "Schedule-driven cross-docking networks", ***Georgia tech tli report, The Logistics Institute, Georgia Tech***, (1998).
23. Yu, W. and Egbelu, P.J., "Scheduling of inbound and outbound trucks in cross docking systems with temporary storage", ***European Journal of Operational Research***, Vol. 184, No. 1, (2008), 377-396.
24. Soltani, R. and Sadjadi, S.J., "Scheduling trucks in cross-docking systems: A robust meta-heuristics approach", ***Transportation Research Part E: Logistics and Transportation Review***, Vol. 46, No. 5, (2010), 650-666.
25. Musa, R., Arnaout, J.-P. and Jung, H., "Ant colony optimization algorithm to solve for the transportation problem of cross-docking network", ***Computers & Industrial Engineering***, Vol. 59, No. 1, (2010), 85-92.
26. Vahdani, B. and Zandieh, M., "Scheduling trucks in cross-docking systems: Robust meta-heuristics", ***Computers & Industrial Engineering***, Vol. 58, No. 1, (2010), 12-24.

27. Alpan, G., Larbi, R. and Penz, B., "A bounded dynamic programming approach to schedule operations in a cross docking platform", ***Computers & Industrial Engineering***, Vol. 60, No. 3, (2011), 385-396.
28. Kuo, Y., "Optimizing truck sequencing and truck dock assignment in a cross docking system", ***Expert Systems with Applications***, Vol. 40, No. 14, (2013), 5532-5541.
29. Konur, D. and Golias, M.M., "Analysis of different approaches to cross-dock truck scheduling", ***Computers and Industrial Engineering***, Vol. 65, (2013), 663-672.
30. Konur, D. and Golias, M.M., "Cost-stable truck scheduling at a cross-dock facility with unknown truck arrivals: A meta-heuristic approach", ***Transportation Research Part E: Logistics and Transportation Review***, Vol. 49, No. 1, (2013), 71-91.
31. Liao, T., Egbelu, P. and Chang, P.-C., "Simultaneous dock assignment and sequencing of inbound trucks under a fixed outbound truck schedule in multi-door cross docking operations", ***International Journal of Production Economics***, Vol. 141, No. 1, (2013), 212-229.
32. Çatay, B., "A new saving-based ant algorithm for the vehicle routing problem with simultaneous pickup and delivery", ***Expert Systems with Applications***, Vol. 37, No. 10, (2010), 6809-6817.
33. Lin, S.-W., Lee, Z.-J., Ying, K.-C. and Lee, C.-Y., "Applying hybrid meta-heuristics for capacitated vehicle routing problem", ***Expert Systems with Applications***, Vol. 36, No. 2, (2009), 1505-1512.
34. Cheng, C.-B. and Wang, K.-P., "Solving a vehicle routing problem with time windows by a decomposition technique and a genetic algorithm", ***Expert Systems with Applications***, Vol. 36, No. 4, (2009), 7758-7763.
35. Mirabi, M., Fatemi Ghomi, S. and Jolai, F., "Efficient stochastic hybrid heuristics for the multi-depot vehicle routing problem", ***Robotics and Computer-Integrated Manufacturing***, Vol. 26, No. 6, (2010), 564-569.
36. Zachariadis, E.E. and Kiranoudis, C.T., "A local search metaheuristic algorithm for the vehicle routing problem with simultaneous pick-ups and deliveries", ***Expert Systems with Applications***, Vol. 38, No. 3, (2011), 2717-2726.
37. Mosheiov, G., "Vehicle routing with pick-up and delivery: Tour-partitioning heuristics", ***Computers & Industrial Engineering***, Vol. 34, No. 3, (1998), 669-684.
38. Ioannou, G., Kritikos, M. and Prastacos, G., "A problem generator-solver heuristic for vehicle routing with soft time windows", ***Omega***, Vol. 31, No. 1, (2003), 41-53.
39. Dondo, R., Méndez, C.A. and Cerdá, J., "The multi-echelon vehicle routing problem with cross docking in supply chain management", ***Computers & Chemical Engineering***, Vol. 35, No. 12, (2011), 3002-3024.
40. Hasani-Goodarzi, A. and Tavakkoli-Moghaddam, R., "Capacitated vehicle routing problem for multi-product cross-docking with split deliveries and pickups", ***Procedia-Social and Behavioral Sciences***, Vol. 62, (2012), 1360-1365.
41. Mousavi, S.M. and Tavakkoli-Moghaddam, R., "A hybrid simulated annealing algorithm for location and routing scheduling problems with cross-docking in the supply chain", ***Journal of Manufacturing Systems***, Vol. 32, No. 2, (2013), 335-347.
42. Ma, H., Miao, Z., Lim, A. and Rodrigues, B., "Crossdocking distribution networks with setup cost and time window constraint", ***Omega***, Vol. 39, No. 1, (2011), 64-72.
43. Dondo, R. and Cerdá, J., "A sweep-heuristic based formulation for the vehicle routing problem with cross-docking", ***Computers & Chemical Engineering***, Vol. 48, No., (2013), 293-311.
44. Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y. and Semet, F., "A guide to vehicle routing heuristics", ***Journal of the Operational Research Society***, Vol. 53, No. 5, (2002), 512-522.
45. Li, X., Tian, P. and Leung, S.C., "Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm", ***International Journal of Production Economics***, Vol. 125, No. 1, (2010), 137-145.
46. Polacek, M., Hartl, R.F., Doerner, K. and Reimann, M., "A variable neighborhood search for the multi depot vehicle routing problem with time windows", ***Journal of Heuristics***, Vol. 10, No. 6, (2004), 613-627.
47. Glover, F., "Tabu search-part i", ***ORSA Journal on computing***, Vol. 1, No. 3, (1989), 190-206.
48. Cordeau, J.-F., Laporte, G. and Mercier, A., "A unified tabu search heuristic for vehicle routing problems with time windows", ***Journal of the Operational Research Society***, Vol. 52, No. 8, (2001), 928-936.
49. Eshghi, K. and Pasalar, S., "Application of tabu search to a special class of multi-commodity distribution systems", ***Transl. Esteghlal***, Vol. 20, No., (2001).
50. Hansen, P. and Mladenović, N., "Variable neighborhood search: Principles and applications", ***European Journal of Operational Research***, Vol. 130, No. 3, (2001), 449-467.
51. Perron, S., Hansen, P., Le Digabel, S. and Mladenović, N., "Exact and heuristic solutions of the global supply chain problem with transfer pricing", ***European Journal of Operational Research***, Vol. 202, No. 3, (2010), 864-879.
52. Hansen, P. and Mladenović, N., "Variable neighborhood search, Springer, (2003).

Modeling the Time Windows Vehicle Routing Problem in Cross-docking Strategy Using Two Meta-heuristic Algorithms

M.B. Fakhrzad[a], A. Sadri Esfahani[b]

Department of Industrial Engineering, Faculty of Engineering, Yazd University, Yazd, Iran

P A P E R I N F O

Paper history:
Received 19 May 2013
Received in revised form 22 October 2013
Accepted in 21 November 2013

Keywords:
Cross-docking Strategy
Vehicle Routing Problem
Time Windows
Tabu Search
Variable Neighborhood Search

چکیده

در سالهای اخیر، بارگیری همزمان به عنوان یک استراتژی مهم توزیع در نظر گرفته می‌شود. در این راهبرد، محموله‌های ورودی بلافاصله بر اساس مقصد و تقاضای مشتریان طبقه‌بندی، مرتب‌سازی و سازماندهی می‌شوند. این راهبرد به عنوان یک رویکرد در زمینه مدیریت موجودی و مدیریت توزیع جهت کاهش موجودی و بهبود پاسخگویی به مشتری مورد توجه قرار گرفته است. از میان مسائل مرتبط با این راهبرد توزیع، مسأله مسیریابی وسیله نقلیه (VRP) از اهمیت خاصی برخوردار است. این مقاله به بررسی حالت خاصی از VRP به نام VRPCDTW می‌پردازد بطوریکه برای دریافت محموله توسط مشتری، محدودیت زمانی در نظر گرفته می‌شود. از آنجا که این مسأله جزء مسائل NP-hard است دو الگوریتم فراابتکاری بر پایه جستجوی ممنوع و جستجوی همسایگی متغیر برای حل آن پیشنهاد می‌شود. الگوریتمهای پیشنهادی برای مسائل دنیای واقعی طراحی شده است و به مسائل دیگر از جمله تفکیک محموله نیز قابل تعمیم می‌باشد. الگوریتم جستجوی ممنوع پیشنهادی هیچ محدودیتی بر روی تعداد گره‌ها و تعداد وسایل نقلیه در نظر نمی‌گیرد. در پایان یک آزمایش عددی برای اعتبار الگوریتمهای پیشنهادی ارائه می‌گردد. نتایج آزمایش عددی نشان داد که الگوریتم جستجوی ممنوع عملکرد بهتری را از نظر هزینه کل و زمان محاسبات از خود نشان می‌دهد.

doi: *10.5829/idosi.ije.2014.27.07a.13*