Journal of Industrial and Systems Engineering
Vol. 11, No. 1, pp. 1-23
Winter (January) 2018

A multi-objective imperialist competitive algorithm for vehicle routing problem in cross-docking networks with time windows

Asefeh Hasani Goodarzi[1], Nasim Nahavandi[1*], Seyed Hessameddin Zegordi[1]

[1]*Faculty of Industrial & Systems Engineering, Tarbiat Modares University,Tehran, Iran*

asefeh.hasani@modares.ac.ir, n_nahavandi@modares.ac.ir, zegordi@modares.ac.ir

Abstract

This study addresses the pickup and delivery problem for cross-docking strategy, in which shipments from suppliers to retailers are directly as well as through cross-docks. Usual models that investigate vehicle routing in cross-docking networks force all vehicles to stop at the cross-dock even if a shipment is about to a full truckload or the vehicle collects and delivers the same set of products. In order to eliminate unnecessary stops at the dock, and thus reduce transportation costs, the designed model tries to decide about the best approach to deliver orders to retailers in a tailored network. In such a system, we have two objectives: minimization of the total transportation cost and minimization of the total earliness and tardiness of visiting retailers. In order to deal with this problem, we develop three multi-objective algorithms. An evolutionary algorithm based on Multi Objective Imperialist Competitive Algorithm (MOICA) is proposed, and the associated results are compared with the results obtained by Non-Dominated Sorting Genetic Algorithm (NSGA-II) and Pareto Archived Evolution Strategy (PAES) in terms of some metrics. The computational results show the superiority of the proposed algorithm compared to other algorithms in some metrics.

Keywords: Cross-docking, Vehicle Routing Problem, bi-objective mathematical model, time windows.

1- Introduction

Cross- docking is a logistic strategy implemented to consolidate shipments from different suppliers to distribute and deliver parts and products to various destinations. Inbound vehicles generally arrive from various origins, carrying shipments for different destinations. The incoming shipments are then unloaded at the inbound doors, sorted, consolidated and reloaded into outgoing vehicles according to delivery points, within less than 24 hours. Other handling operations (i.e., weighing, sizing, and packaging, pricing and labeling products) can also be done on different shipments.

In this strategy, consolidation of small orders into one big shipment is done to utilize the entire truck capacity (Truck-Load), instead of shipping small orders by individual vehicles, which in this case the entire truck capacity is not occupied (Less-Than-Truckload). Thus, cross-docking helps to make more frequent and economical collections and deliveries by trying to meet the entire truck capacity (Vasiljevic et al., 2013).

*Corresponding author
ISSN: 1735-8272,

In addition to consolidation of shipments, as discussed in the literature, other most important advantages of cross-docking are reducing transportation cost, as well as needs for warehouse space and labor cost, improving service level, and balancing of demand and supply.

Some goods and products are more suited to cross-docking implementation than others, such as fast-moving products with constant demand (Ross and Jayaraman, 2008) or low demand variance, perishable items that require immediate shipment, high quality products that do not need quality inspections during the receiving process, products that are pre-tagged (bar coded, RFID), and ready for sale at the customer.

According to Shaolong (2007), the basic concept of cross-docking can be considered in one of two following cases: I) one supplier and multiple retailers and II) multiple suppliers and a retailer in the network. In the first case, products from an overseas supplier are loaded to inbound vehicles (which usually have more capacitate in compare with outbound vehicles). Products are conveyed to the cross-dock, which is located in a region close to retailers, then sorted and loaded to outbound vehicles and finally delivered to retailers. In the second case, one customer orders to multiple suppliers located in a long distance. It is more economical to consolidate products from various suppliers at a cross-dock located in their nearby, instead of direct transportation from each supplier to customer (Shaolong, 2007).This study considered a network including multiple suppliers, multiple retailers and a cross-docking center.

A cross-docking system must deal appropriately with important issues. For example, an important question is that how one must schedule vehicle loading and unloading operations at the docks and how to route available vehicles to collect and deliver products (Santos et al., 2013). Recently, integrating vehicle routing problem with cross-docking method (VRPCD) has attracted attention from industrial and academic areas.

The goal of VRPCD is to find routes such that all products are collected from suppliers and delivered to their final destinations and the total transportation cost is minimized. The major characteristics of general VRPCD are as follows: i) each node must be met by a single vehicle only once, ii) vehicles can service more than one supplier or customer, iii) pickup and delivery routes start and end at the cross-dock, iv) the amount of load/unload at each pickup/delivery node is known previously, v) total quantity loaded at pickup nodes should be equal to total quantity that is unloaded at delivery nodes, which means that there is no inventory at the cross-dock. Figure 1 shows the flow of materials in a typical cross-docking terminal.

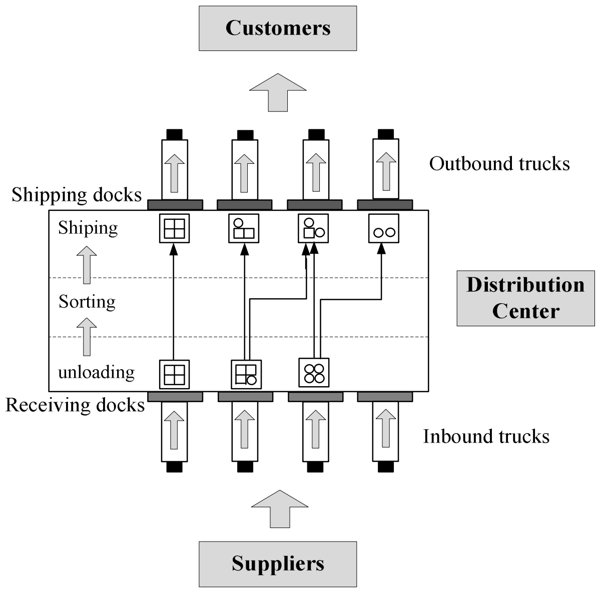

Fig. 1. Schematic representation of a cross-docking environment

The designed cross-docking system in this study can be applied in many industrial estates wherein there are multiple consumers ordering parts and products from multiple suppliers. In industrial estates, it is most possible to have routes between consumers, since they are located in the same cluster, as is assumed in this study.

In most academic works in cross-docking field, the researchers assume that demand of each consumer is less than vehicle capacity. In this article, we assume that according to real applications of cross-docks, the demand of each consumer may be more than vehicle capacity. So servicing nodes by multiple vehicles is allowed, therefore splitting shipments is also possible. In order to manage parts deliveries in an industrial estate, scheduling and time windows are critical factors that cannot be neglected. We assume that every consumer has a time window for receiving parts; therefore, minimizing earliness and tardiness of servicing consumers are another objective function in addition to minimizing transportation cost.

In this study, we try to design a tailored network based on real environment, in which in some cases direct shipment from suppliers to consumers is more economical than going through a cross-docking center. In the previous works, which considered both cross-docking and direct shipment as transportation strategies, the authors mostly did not consider the routing problem between nodes. In the cases that authors addressed vehicle routing and direct shipping strategy in the same study, scheduling problem is not considered and the problems have single objective formulations.

In the current study, a multi objective model is presented for transportation problem in the networks, in which both cross-docking and direct shipment are allowed to be utilized. Splitting services of nodes is possible in the cases that consignment amount is more that vehicle capacity. The word of “retailers” is also used for “consumers” in the article.

2- Literature

Recently cross-docking distribution networks have attracted strong attention among researchers. In the literature on cross-docking there exist problems in the strategic, tactical and operational level. The problems in strategic level often deal with decisions that are not taken frequently, for instance the design of cross-docking network and its layout. In the operational level, most researches are proposing models related to dock door assignment, vehicle routing and scheduling problems.

Gumus and Bookbinder (2004) studied a problem in which direct shipments are allowed and multiple product types are considered. Their model determines the cross-docking facility locations and direct shipments from suppliers to customers and shipments thorough cross-docking centers. They assumed that there is no route between different suppliers and routing within the cluster of retailers is outside of the location model.

Lee et al. (2006) were seemingly the first authors who presented vehicle routing scheduling in cross-docking problem. They considered single cross-docking center in order to transport goods from suppliers to retailers. They also proposed a Tabu Search (TS) to determine the number of vehicles and the optimal vehicle routing schedule at the cross-dock to minimize sum of transportation and operational cost of vehicles. Then, Liao et al. (2010) proposed another TS algorithm to solve the model proposed by Lee et al. (2006). They showed that their algorithm surpassed the previous one in the computational time and number of vehicles.

Wen et al. (2008) presented a mixed-integer programming (MIP) model for VRP in cross-docking networks with the objective of minimizing total traveled distance. They offered a new heuristic method based on TS to solve the problem.

Musa et al. (2010) addressed the transportation problem in cross-docking distribution networks and presented an Ant Colony Optimization (ACO) algorithm to solve it. They assumed that the freights could be transferred from suppliers to customers directly, as well as via cross-docks. The numerical experiments showed that the proposed method significantly reduces the shipping cost compared to branch and bound algorithm. Based on their model; Hosseini et al. (2014) proposed a new integer programming (IP) model for transportation problem in a consolidation system. The authors consider three transportation strategies to move goods from suppliers to customers including direct shipment, shipment through cross-dock and milk run. In this study, a hybrid algorithm based on harmony search (HS) and simulated annealing (SA) has been presented to solve the problem.

Yang et al. (2010) used simulation techniques to address decisions for transporting freight between inbound and outbound vehicles in a cross-dock. Ma et al. (2011) investigated a new consolidation and transportation problem in cross-docking, in which inventory also has been addressed to determine a trade-off between transportation cost, inventory and scheduling requirements. They also considered time windows and formulated the problem as an integer programming (IP) model and indicated that it is NP-complete.

Dondo et al. (2011) presented a mixed-integer linear programming (MILP) formulation for multi-echelon multi-item vehicle routing and scheduling problem in cross-docking with time windows. The objective of the problem is to satisfy customer demands at minimum total transportation cost.

Vahdani et al. (2012) suggested a hybrid metaheuristic algorithm to solve the model proposed by Lee et al. (2006). The algorithm combines particle swarm optimization (PSO), variable neighborhood search (VNS) and simulated annealing (SA) in a population-based context. Numerical results indicated that the hybrid metaheuristic algorithm outperforms the TS algorithm proposed by Lee et al. (2006).

Santos et al. (2011a, 2011b) considered a different VRPCD, in which a cost to be added in the objective function when a good is moved from a vehicle to another one at the cross-docking center. Then, Santos et al. (2013) extended their previous work (Santos et al., 2011a; 2011b) and consider a VRPCD where vehicles are allowed to avoid the stop at the cross-dock after pickup process. They presented an integer programming formulation for the problem and utilized a branch and price algorithm to solve it.

Konur and Golias (2013) tried to determine a cost-stable scheduling strategy at inbound doors of a cross-dock to minimize the average of total service costs. They formulated a bi-objective bi-level optimization problem and employed a genetic algorithm heuristic to find an efficient Pareto frontier. The proposed strategy is compared with first-come-first-served policy. Shi et al. (2013) addressed a multi-criteria robust design approach for cross-docking distribution center in logistic management of auto parts. They tried to design a robust configuration for cross-docking which is insensitive to the disturbances of supply uncertainty. Tarantilis (2013) investigated a multi-source vehicle routing problem with a single cross-docking and used TS approach to solve the problem.

Yin and Chuang (2016) proposed a new hybrid algorithm based on bee colony for green VRPCD. They added vehicle's fuel efficiency and carbon emissions to the problem. Yu et al. (2016) studied an open VRPCD and proposed a SA that incorporates several neighborhood structures to improve the performance of the algorithm. They acclaimed that the proposed SA outperforms existing methods for VRPCDs. Chen et al.(2016) addressed the VRP in a network with multiple cross-docks for processing multiple products. The proposed model is solved by PSO based algorithm.

Nikolopoulou et al. (2017) compared direct shipping with cross-docking strategies by performance analysis for them. They concluded that clustered distributions and large supplier-customer travel distances improve the cross-docking performance. They also stated that locating the depot in remote positions favors the direct-shipping strategy. The authors also examined a composite of direct-shipping and cross-docking transportation strategy.

Ladiera and Alpan (2016) analyzed the gaps between literature and industry practices to indicate future research directions. They visited eight cross-docking platforms and interviewed with their managers in order to capture the main issues and challenges in cross-docking operations in real cases.

Hasani Goodarzi and Zegordi (2016) also considered a location-routing problem in a distribution network. They developed a mixed integer non-linear programming model for the problem, in which the best location of cross-docks is determined and a fleet of vehicles is applied to transport goods from suppliers to customers (assembly plants) via two transportation strategies including direct shipment and indirect shipment (shipment through cross-dock). In the second strategy, it is possible to have routes between suppliers and routing problem is considered in the supply part. In the other word, there are many suppliers in the network with a limited number of customers. Indeed, splitting demand between vehicles is not allowed. In the study here, as mentioned before, we assume that there are multiple consumers and routing between them is permitted; thus, routing problem is assumed in delivery part. Moreover, splitting demand is also allowed that can be another extension to the work of Hasani Goodarzi and Zegordi (2016).In addition, consumers have time windows for receiving service; therefore, a multi objective model is considered which tries to minimize transportation cost and earliness and tardiness of servicing consumers.

Going through the literature indicates that the vehicle routing scheduling problems in cross-docking need some adaptations, so they can be applied in real cases. In distribution networks with cross-docking, especially for environments such as industrial states, moving shipments through cross-dock and consolidation process should be done in reasonable situations. In most studies that considered

VRPCD, the authors force all shipments to pass through a cross-docking center (direct shipment is not allowed), while in problems that considered both direct shipment and cross-docking, the authors assumed that there is no route between nodes. Hence, there are limited numbers of studies that consider both VRPCD and direct shipment as transportation strategies. To the best of our knowledge, in previous researches, the quantity of orders is assumed to be less than capacity of vehicles while in this study this limitation has been relaxed and demand quantity can be more than vehicle capacity, so picked up load can be split between multiple vehicles. Therefore, in this study a tailored distribution network has been addressed, in which direct shipment and cross-docking are allowed to transport products based on some considerations including the demand quantity, geographical distribution of nodes and so on. A bi- objective formulation is presented to model this tailored network.

3- Problem definition

The distribution network investigated in this study includes suppliers; cross-dock, and retailers. By using cross-docking, products in various locations are collected in the cross-dock prior to transportation to their final destination. The inbound vehicles are reloaded at cross-dock and after classifying products according to their destinations; products are shipped from cross-dock to their final locations. It is assumed that simultaneous arrival of inbound vehicles is not obligatory and time window exists in retailer locations. A customer served by direct shipment usually has a total demand near (or multiple of) TL (truckload) or direct distance from manufacturer is considerably less than possible indirect distances.

The objective is to find the best fleet routing/dispatching and consolidation plans such that the total transportation cost and the earliness and tardiness of servicing nodes are minimized. To do so, the model determines the loads to be sent directly from suppliers to retailers and amounts to be sent indirectly through cross-docking center. Consolidation process is neither required nor possible for a TL shipment, which means that for quantity shipped equals, or a multiple of vehicle capacity, direct shipment is the most efficient way of transportation. The loads sent to the cross-dock are consolidated according to their final destination. Figure 2 shows the flow of materials in a typical cross-docking terminal with the assumption that a direct shipment from origin to destination is possible. We also assume that, each direct shipment can only serve a single customer and milk run deliveries are allowable in shipments moving through cross-dock.

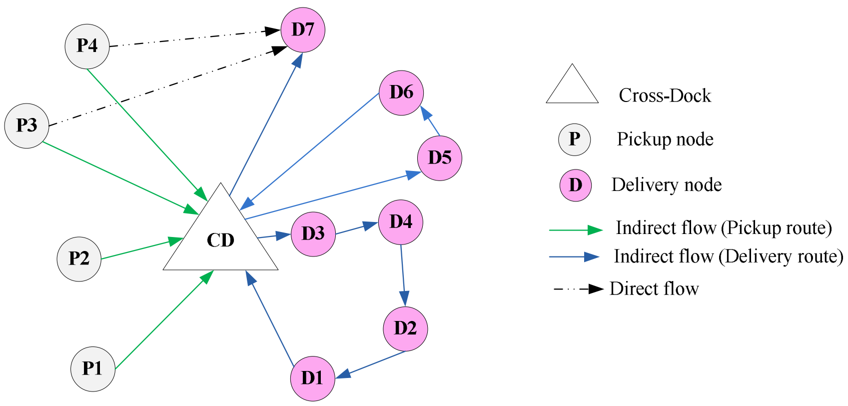

Fig. 2. Considered cross-docking network when direct flow of products is allowable

Other assumptions of problem are as the following:

- All vehicles have the same capacity (Q). If this assumption is relaxed, one could simply add an index to each vehicle for its capacity (Q_m).
- Each retailer has a soft time window for receiving service where LE_i is the allowable earliest time to begin service and UE_i is the latest time. There is a desired time window $[e_i, l_i]$ and exceeding this interval $[e_i , l_i]$ will be considered as earliness or tardiness of visiting nodes. Figure 3 shows the soft time window for servicing nodes.

- There is one cross-dock in the network.The cross-dock is denoted by 0 as an initial node of tour in the delivery process, with s_0= p_0=0.

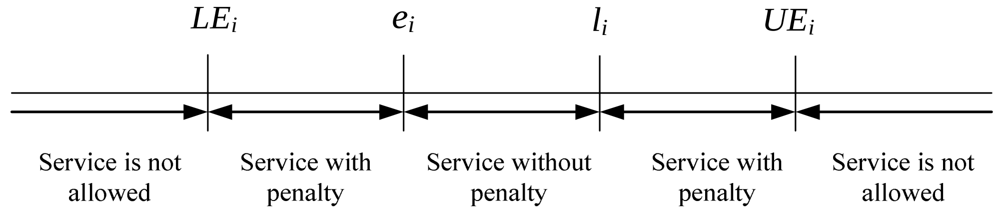

Fig. 3. Soft time window for a service operation

- There is no route between suppliers, which means that every supplier sends products directly to its customers or to the cross-docking center.
- It is assumed that the product load to be sent from supplier *i* to retailer *j* is known in advance (s_{ij}).If s_{ij} is more than the vehicle capacity, multiple vehicles are needed to go directly from *i* to *j*, since consolidation is not possible for TL vehicles and it is more economical to send them directly to destination.
- Splitting services is allowable which means that if demand of a customer is more than capacity of vehicles, more than one vehicle can be applied to meet it.
- There is no inventory kept at the cross-dock, which means that total quantity of a particular product picked up at source nodes should be equal to the total amount of that product that is delivered to retailer locations.
- There is a time horizon for operation, which is not expandable.

3-1- Indices and sets

i, j, k	Index for nodes
m	Index for vehicles (m =1, ...,V)
0	Index for cross-docking center
P	Set of pickup nodes
D	Set of delivery nodes
N	Set of all nodes including cross-dock; $N = P \cup D \cup 0$

3-2- Parameters

V	Number of available vehicles
Q	Maximum capacity of vehicles
tc_{ij}	Transportation cost from node *i* to node *j*
Co	Operational cost of vehicles
F	Variable cost per unit quantity at the cross-dock
t_{ij}	Travel time from node *i* to node *j*
s_{ij}	The flow of products from supplier *I* to retailer *j*
Ser_i	Service time of node *i*
p_i^m	Beginning of the service at node *i* by vehicle *m*
Tr_j	Tardiness of visiting node *j*
Er_j	Earliness of visiting node *j*

3-3- Decision variables

x_{ij}^m	1: if vehicle *m* moves from retailer *i* to retailer *j* in delivery process; 0: otherwise
u_{ij}^m	1: if the flow of products moves directly from pickup node *i* to delivery node *j*
y_i^m	1: if vehicle *m* visits node *i*; 0: otherwise
R_{ij}^m	Quantities shipped directly from pickup node *i* to delivery node *j* by vehicle *m*

a_{ij}^m	Loaded quantity from supplier *i* to retailer *j* by vehicle *m* moving through cross-dock
z_j^m	Unloaded quantity in retailer *j* by vehicle *m* moving through cross-dock
$£_{i0}$	Number of vehicles used on arc from supplier *i* to cross-dock
λ_{0j}	Number of vehicles used on arc from cross-dock to retailer *j*
Ψ_{ij}	Number of vehicles used on direct movement from supplier *i* to retailer *j*.

3-4- Mathematical model

This section presents a mixed integer nonlinear model for given problem with the notations introduced above. The problem has two objective functions to minimize : i) the total travel cost (i.e., cost of direct transportation from suppliers to retailers and indirect transportation via cross-dock) and total operational cost of vehicles by determining the best route assigned to each vehicle and ii) the time of violating the time window of each node (earliness and tardiness), considering every node in a special time window. Moreover, we consider various constraints for the problem that can be described as follows:

$$\mathbf{min} \sum_{i\epsilon P}\sum_{j\epsilon D}(tc_{ij}+Co)\Psi_{ij}+\sum_{j\epsilon D}(tc_{0j}+Co)\,\lambda_{0j}+\sum_{i\epsilon P}(tc_{i0}+Co)£_{i0}+\sum_{i\epsilon(0\cup D)}\sum_{j\epsilon(0\cup D)}\sum_{m} tc_{ij}x_{ij}^m + \sum_{m}\sum_{j\epsilon D} Co \quad x_{0j}^m+\sum_{j\epsilon D}\sum_{m} \mathrm{F}\times z_j^m \tag{1}$$

$$\mathbf{min} \sum_{j\epsilon D}\sum_{m} \quad y_j^m(Er_j \quad + Tr_j \ \ ) \tag{2}$$

s.t.

$$\sum_{j\epsilon D} x_{0j}^m \le 1\,;\ \forall m \tag{3}$$

$$\sum_{j\epsilon D} x_{j0}^m \le 1;\ \forall m \tag{4}$$

$$\sum_{i\epsilon(D\cup 0)} \quad x_{ij}^m = y_j^m;\ \ \forall m, j\epsilon D \tag{5}$$

$$\sum_{i\epsilon(D\cup 0)} x_{ik}^m = \sum_{j\epsilon(D\cup 0)} x_{kj}^m\ \ ;\ \ \forall m, k\epsilon N-\{0\} \tag{6}$$

$$x_{ij}^m + x_{ji}^m \le 1\,;\ \forall i, j\epsilon D, m \tag{7}$$

$$\sum_{i\epsilon P}\sum_{j\epsilon D}\Psi_{ij} + \sum_{i\epsilon P} \quad £_{i0} + \sum_{j\epsilon D} \quad \lambda_{0j} \le V; \tag{8}$$

$$z_j^m \le \sum_{i\epsilon P} a_{ij}^m \times y_j^m;\ \forall j\epsilon D, m \tag{9}$$

$$\sum_{m} \quad \sum_{j\epsilon D} a_{ij}^m \le Q£_{i0}\ \ ;\ \forall i\epsilon P \tag{10}$$

$$\sum_{j\epsilon D} \quad z_j^m \le Q\ \ ;\ \forall m \tag{11}$$

$$\sum_{m} R_{ij}^{m} \times u_{ij}^{m} \leq Q\Psi_{ij} \; ; \; \forall i\epsilon P, j\epsilon D \tag{12}$$

$$\sum_{m} a_{ij}^{m} + \sum_{m} R_{ij}^{m} = s_{ij}; \; \forall i, j \tag{13}$$

$$(p_i^m + Ser_i + t_{ij})x_{ij}^m \leq p_j^m; \; \forall m, i\epsilon N, j\epsilon N \tag{14}$$

$$LE_j \leq p_j^m \leq UE_j; \;\; \forall m, j\epsilon D \tag{15}$$

$$\sum_{i\epsilon(0\cup D)} \sum_{j\epsilon(0\cup D)} x_{ij}^m \times t_{ij} + \sum_{i\epsilon(0\cup D)} \sum_{j\epsilon(0\cup D)} x_{ij}^m \times Ser_j \leq T; \quad \forall m \tag{16}$$

$$Er_j^m = max\{0, e_j \; - p_j^m\}; \; \forall m, j\epsilon D \tag{17}$$

$$Tr_j^m = max\,\{0, p_j^m - l_j \;\}; \forall m, j\epsilon D \tag{18}$$

$$y_i^m, u_{ij}, x_{ij}^m \epsilon\{0,1\} \; ; R_{ij}, p_j^m \geq 0 \tag{19}$$

Constraints (3) and (4) insure that all delivery routes start from and end at the cross-dock. They also imply that all vehicles are not necessarily applied in the process. Constraint (5) determines that each node is visited and served by each vehicle. Constraint (6) guarantees the consecutive movement of vehicles. Constraint (7) prevents backward movement in routes and guaranties that each vehicle continues its route. Constraint (8) prevents the number of vehicles in direct and indirect trips from exceeding the total that are available. Constraint (9) states that at each delivery node *j*, the delivery amount by vehicle m,z_j^m, should utmost be equal to its demand provided by different suppliers. Constraint (10) calculates the total number of vehicles for pickup process that transport products to cross-docking center. In Constraint (11), the total delivery of the vehicle cannot exceed its capacity. Constraint (12) calculates the total number of vehicles used in direct movement from pickup to delivery nodes.

Constraint (13) guaranties that all requests to be satisfied either through the direct link or through the cross-dock. If there is no flow from i to *j*, then there will be no allocation for direct or indirect travel. Constraint (14) sets a minimum time for beginning the service in each node and guarantees that there will be no sub tour. Constraints (15) and (16) represent a time window for each node and a time horizon to schedule the vehicles, respectively. Eq. (17) and (18) calculate the value of earliness and tardiness for each node, respectively. Finally, Constraint (19) shows the binary and non-negative variables.

4- Solution methodology

The multi-objective optimization problem (MOP) arises when in correspondence of each point *x* in the search space several objective functions $f_i(x)$, (i=1, 2, …) should be considered simultaneously and identified which set of solutions give rise to the best trade-off among the various objective functions (x^*). The comparison of two solutions with respect to several objectives may be obtained by applying the concepts of Pareto optimality and dominance that enable solutions to be compared and ranked without imposing any a priori measure as to the relative importance of individual objectives, neither in the form of subjective weights, nor arbitrary constraints (Marseguerra et al., 2002).

Recently, Evolutionary Algorithms (EAs) have been widely applied to MOP's, since they are successful in finding good solutions to problems with appropriate structure. The main idea of Pareto ranking is to keep the independence of individual objectives, which can be done by treating the current candidate solutions as categorized sets or ranks of possible solutions. The individuals in each rank set demonstrate solutions that are in some sense incomparable with other ones. Pareto ranking only differentiates solutions that are clearly superior to others in terms of all objective functions (Ombuki, et al., 2006).

4-1- Non-dominated sorting genetic algorithm II

Non-dominated sorting genetic algorithm II (NSGA-II), one of the most well-known and efficient multi-objective evolutionary algorithms, was first introduced by Deb et al. (2000, 2002). Non-dominance technique and a crowding distance implement ranking and selecting the population fronts. The algorithm also utilized crossover and mutation operators, as implemented in genetic algorithm (GA), to generate new individuals known as offspring. Then, the current population and generated offspring are combined together, and the best solutions in terms of non-dominance and crowding distance are selected from a combined population as the new population. The following sections explain the non-dominated technique, the calculation of the crowding distance and crowding selection operator.

4-1-1- Non-dominance technique

Let us consider *r* different objective functions $f_i(x)$, *i*=1, 2, …,r_f, where *x* represents a solution. When the following conditions are met, solution x_1 dominates another solution x_2. If x_1 and x_2 do not dominate each other, they are placed in the same front.

(a) For all the objective functions, solution x_1 is not worse than another solution x_2.

(b) For at least one of the *r* objective functions, x_1 is extremely better than x_2.

Front number 1 is formed of all solutions that are not dominated by any other solutions (non-dominated solutions). In addition, front number 2 is made by all solutions that are only dominated by solutions in front 1. Figure 4 indicates the relation between decision and solution space in multi-objective problems.

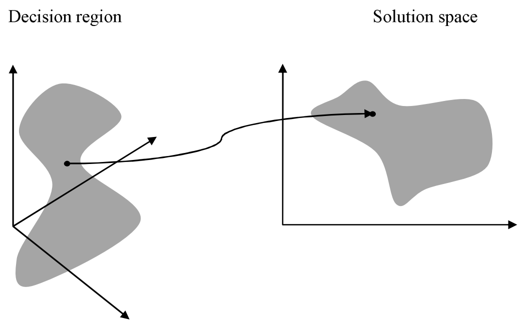

Fig. 4. The representation of decision and solution space

4-1-2- Crowding distance

Crowding distance (CD) is a measure of the density of solutions that represents an estimation of the density of solutions surrounding a particular solution. The crowding distance applied in the NSGA-II is shown in equation (20).

$$CD_i = \sum_{k=1}^{r} \frac{f_{k,i+1}^{p} - f_{k,i-1}^{p}}{f_{k,total}^{p,max} - f_{k,total}^{p,min}} \tag{20}$$

Where *r* shows the number of objective functions, $f_{k,i+1}^{p}$ is the *k*-th objective function of the (*i*+1)-th solution and$f_{k,i-1}^{p}$is the *k*-th objective function of the (*i*-1)-th solution after sorting the population based on crowding distance of the *k*-th objective function. Also, $f_{k,total}^{p,max}$ and $f_{k,total}^{p,min}$ are defined as the maximum and minimum value of objective function *k*, respectively. Whenever two different solutions have the same rank, the solution with a higher value of the CD is preferred. In figure 5 the non-dominated Pareto set are shown by set of black solutions. In addition, the area surrounded by the dotted line illustrates the value of crowding distance measure for solution *i*.

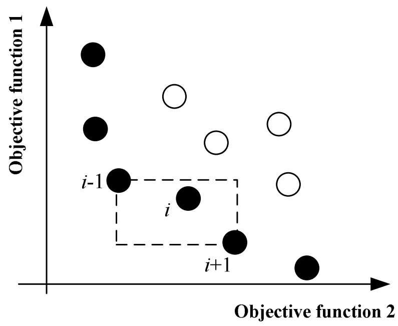

Fig. 5. A Schematic of crowding distance for a bi-objective minimization problem

4-1-3- Tournament selection operator

A binary tournament selection procedure has been utilized in choosing individuals for both crossover and mutation operators. In this procedure, two solutions of the population size are first selected, and then the lowest front number is chosen if two individuals are from different fronts. If they are from the same front, the solution with the highest crowding distance is selected.

4-2- Pareto archive evolution strategy

The Pareto archived evolution strategy (PAES) is another multi-objective evolutionary algorithm (Knowles and Corne, 1999) that uses a simple (1+1) local search evolution strategy to find diverse solutions in the Pareto optimal set. This algorithm starts with the initialization of a single solution evaluated by using the multi-objective cost function. In each iteration, a new solution is generated by applying a mutation operator. Then, the new and current solutions are compared together and afterward, the new solution and archive are updated. The comparison is done based on what is mentioned in Knowles and Corne (1999). This process continues until a stopping criterion (i.e. predefined iteration numbers) is met.

4-3- Multi-objective imperialist competitive algorithm

Atashpas-Gargari and Lucas (2007) to solve continuous optimization problems with one objective function first introduced the Imperialist Competitive Algorithm (ICA). This algorithm uses socio-political evolution of human as a source of inspiration for establishing a strong optimization strategy. They described the algorithm as follows. Like other evolutionary ones, this algorithm starts with an initial population considered as countries in the world. According to fitness function value, countries are divided in two types: imperialists (some of the best countries) and colonies (remaining ones) which all together form empires. Then, imperialistic competition among these empires begins, in which weak empires collapse and powerful ones take possession of their colonies. The competition gradually converges to a state in which there exist only one empire and its colonies are in the same position and have the same cost as the imperialist.

In this study, a multi-objective imperialist competitive algorithm (MOICA) is proposed which its description is presented in the following sections.

4-3-1- Solution representation (Generating initial countries)

The first step of MOICA is to generate initial solutions. Every solution (i.e., country) corresponds to a "*chromosome*" in GA. At first, a set of random matrices are generated (equal to the number of initial countries) as explained below.

According to Gumus and Bookbinder (2004) if the quantity shipped from a supplier is equal to (a multiple of) vehicle capacity, direct shipment is the most cost effective way to transport. The TL portion is transported directly in one or more full vehicles, and removed from model. The remainder, considered as a "separate" demand, is kept for consolidation process.

The first stage of generating initial solutions is based on a heuristic proposed by Ma et al. (2011). Once TL shipments are determined, LTL quantities at cross-dock require to be consolidated to decrease vehicle operational costs. The solution approach consists of two stages: (1) a full truck load plan (TL

Plan), and (2) a less than truckload plan (LTL Plan). The basic concept is to separate vehicles that can be fully loaded at the supply locations at first, and then search for solutions that link together different smaller shipments which would have been sent in partially loaded vehicles. The TL Plan can be achieved by solving a network flow problem. In the second stage, a meta-heuristic is applied to improve the LTL Plan.

(1)TL Plan: A TL Plan, used to establish an initial LTL Plan, is developed in this way: Split the demand of each retailer from each supplier into a TL part and an LTL part, and remove the LTL part (remainders) from the total demand. Then, form the network with suppliers and retailers, without cross-dock, where all the demands are TL. The TL Plan cannot guarantee all demand is met; thus, unsatisfied demands are allowed to be met by the LTL Plan. Assume pickup nodes, P_1, P_2, $P_3$and delivery nodesD_1, D_2, D_3,D_4 and vehicle capacity Q=20. The demand quantity of each retailer from each supplier is shown in table 1.

Table 1. The flow of products from supplier *i* to retailer *j*

s_{ij}		Retailer*j*			
		1	2	3	4
Supplier*i*	1	107	16	0	27
	2	24	0	15	20
	3	0	43	34	2

The resulting TL plan is using five vehicles from supplier P_1 to D_1 and one vehicle from P_1 to D_4, one vehicle from P_2 to D_1, one vehicle from P_2 to D_4, two vehicles from P_3 to D_2 and one vehicle from P_3 to D_3.As indicated in figure 6, each arc shows a direct route from a supplier to a customer in the network. Once the TL Plan is determined, we construct an initial LTL Plan.

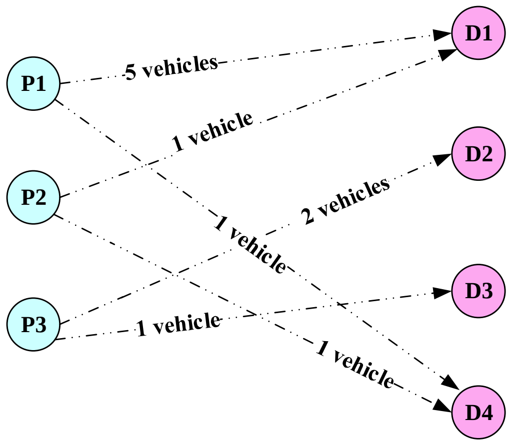

Fig. 6. The network constructed for TL plan.

(2) LTL Plan: With a TL Plan in hand, the quantities remaining at supply and demand nodes can be decided. The remaining supply and demand quantities are as shown in table 2.

Table 2. The remaining demand of products from supplier *i* to retailer *j*

The remaining demand		Retailer *j*			
		1	2	3	4
Supplier *i*	1	7	16	0	7
	2	4	0	15	0
	3	0	3	14	2

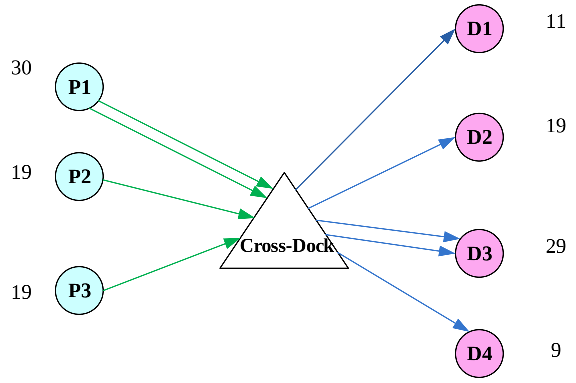

Fig. 7. Consolidation

Now regarding to conditions (capacity of vehicles, time window of delivery nodes, operational cost of vehicles, geographical distribution of delivery nodes, etc.) some scenarios can be defined. For example (i) Using separate vehicles to service each delivery node D_1, D_2, D_4 and two vehicles for delivery node D_3, or (ii) Using a vehicle in order to service D_1 and D_4, one vehicle for D_2 and two vehicles for D_3. (iii) Servicing D_3 and D_1 in a route (with the LTL vehicle goingtoD_3) or (iv) servicing D_3 and D_4 in a route (with the LTL vehicle going to D_3). It should be noted that there is no route between suppliers, so 2, 1 and 1 vehicle are used for supplier 1, 2, and 3, respectively.

Hence, a solution to the problem is coded as a string which represents the delivery routes and order of visiting customers. For instance the second scenario can be coded as Fig. 8,in which each route is differentiated from others by the position of cross-dock that is shown by 0.According to the number of vehicles needed to satisfy LTL demand of a retailer, the corresponding number of retailer should be repeated in the string. For example in the figure 8 delivery node 3 is repeated 2 times. Based on routes, the arrival time of vehicles to delivery nodes and thus, the earliness and tardiness of visiting customers can be calculated.

$$\overbrace{\boxed{1}\boxed{4}\boxed{0}\boxed{2}\boxed{0}\boxed{3}\boxed{3}}^{\text{Customer part}}$$

Fig. 8. Solution representation for the problem

4-3-2- Generating Initial empires

To form the initial empires, we use the approach proposed by Mohammadi et al. (2013). After generating initial countries, a non-dominance technique and a crowding distance are applied to rank and select the population fronts. The members of front one are saved in archive, the best solutions in terms of the non-dominance and crowding distance are selected from population as the imperialists, and the remaining countries form colonies. In order to find the cost value of each imperialist, the value of each objective function for each imperialist is calculated by**:**

$$C_{i,n} = \frac{|f_{i,n}^{p} - f_{i}^{p,best}|}{f_{i,total}^{p,max} - f_{i,total}^{p,min}} \tag{21}$$

where $C_{i,n}$ shows the normalized value of objective function i for imperialist n,$f_{i,n}^{p}$*is* the value of objective function i for imperialist n.$f_{i}^{p,best}$,$f_{i,total}^{p,max}$and$f_{i,total}^{p,min}$ are the best, maximum and minimum values of objective function i in each iteration, respectively. Finally, the normalized cost value of each imperialist is obtained by:

$$C_n = \sum_{i=1}^{r} c_{i,n} \tag{22}$$

where r is the number of objective functions. Afterward the power of each imperialist is computed as indicated in equation (23) and the colonies distributed among the imperialist according to power of each imperialist country.

$$P_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right| \tag{23}$$

Eventually, the number of colonies that an empire can possess is as follows:

$$N.C._n = round\{P_n . N_{col}\} \tag{24}$$

Where P_n is the initial number of colonies of the n-th imperialist and N_{col} is the number of all colonies. After allocating the colonies to imperialists, the imperialism competition commences and each emperor tries to develop its colonies.

4-3-3-Evaluation function

The evaluation function is an operation to evaluate how good a solution is, making the comparison between different possible solutions. In order to have a better diversification in the proposed algorithm, in the whole search space we allow infeasible solutions in term of capacity constraint and time horizon to be generated. But whenever the constraints are violated; a penalty is added to evaluation function which penalizes infeasible solutions. Thus the evaluation function consists of calculating the value of two objective functions and penalty of violating capacity and time horizon constraints of each country.

4-3-4- Assimilation methods

After forming initial empires, their colonies start moving toward the respective imperialist. This movement is known as assimilation policy and is illustrated in Fig. 9, in which y is the distance between imperialist and colony and θ is the direction of the movement. The colony moves toward the imperialist by d units, which is a random variable with uniform distribution between 0 and $\beta \times y$ and β is a number greater than 1. θ is a random number with uniform distribution ($\theta \sim U(-\gamma, \gamma)$), where γ is a parameter that adjusts the deviation from the original direction.

In assimilation phase, we have utilized a crossover operator, in which a crossover points is selected randomly. From parent 1, one will copy in the offspring, at the same absolute positions, the part in the left side of cross over point. From the second parent, one will start from the starch and pick the elements that are not already selected from parent 1 to fill them in the offspring. The population percent sharing information is shown by *p Crossover*. The utilized crossover operator is indicated in figure 10.

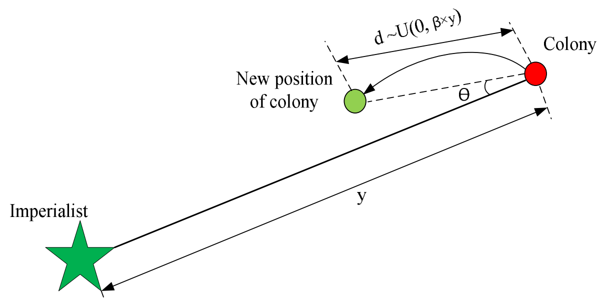

Fig. 9. Moving colonies toward their relevant imperialist with a random angle θ

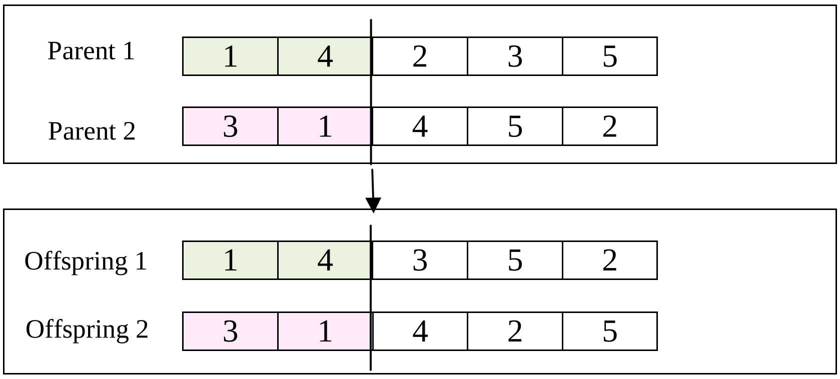

Fig. 10. The order crossover

Exchanging Positions of the Imperialist and a Colony

While moving toward the imperialist, a colony may reach to a position with a lower cost (or equivalently, higher power) than that of its imperialist. In this case, the imperialist will be superseded by that colony and consequently the position of colony and imperialist is changed.

4-3-5- Total Power of an Empire

The total power of an empire is mainly affected by the power of an imperialist country; however, the power of the colonies of an empire has also an effect on the total power of that empire. Thus, the total cost of *n*-th empire can be obtained by:

$$T.C._n = Cost(Imperialist_n) + \xi mean\{cost(coloniesofempire_n)\} \qquad (25)$$

where $T.C._n$ is the total cost of the *n*-th empire and ξ is a positive number considered to be less than 1.

4-3-6- Imperialistic Competition

All empires try to take possession of colonies of other empires and control them. This competition gradually causes an increase in the power of more powerful empires and a decrease in the power of weaker ones. The competition is modeled by picking one of the weakest colonies of the weakest empires and making a competition among other empires to possess it. During this competition, each emperor who cannot develop its colonies will be eliminated. Each empire will have a likelihood of taking possession of the weakest colonies, based on its total power. In other words, the most powerful empires will not possess these colonies; however, these empires will have more probability to possess them. In order to compute the power of imperialist, the normalized cost of each empire is calculated by:

$$N.T.C_n = max\{T.C._i\} - T.C._n \qquad (26)$$

where $T.c._n$ is the cost of *n*-th imperialist and $N.T.C_n$is its normalized cost. Now, a normalized power for each imperialist for allocating the colonies to them is computed as follows:

$$P_n = \left| \frac{N.T.C_n}{\sum_{i=1}^{N_{imp}} N.T.C_i} \right| \qquad (27)$$

This process will be continued until it remains only one imperialist. Also, the algorithm can be stopped after a predefined iteration.

4-3-7- Revolution policy

A revolution policy tries to generate a new solution and is similar to the mutation operator in the GA. The number of colonies that will be revolted is limited. Revolution policies used in this paper are insertion, inversion and swap.

4-3-8- Eliminating the powerless empires

Powerless empires will collapse in the competition and their colonies will be distributed among other empires. We assume an empire collapsed when it loses all of its colonies.

4-3-9- Stopping criteria

In this paper, the algorithm is finished when the number of function calls (NFC) is set to a predefined value. The pseudo code of the proposed MOICA is presented in figure 11.

```
NFC ← 0
Set the parameters of MOICA (n-Pop, N-imp,ξ,β, P-Assimilation, P-Crossover, P-Revolution, n-Archive)
Generate initial Countries (Randomly) ←n-Pop
Evaluate fitness of each country
update NFC
Form initial empires:
        a)  Choose most powerful countries as the imperialists ←N-imp
        b)  Assign other countries to imperialists based on imperialist power (pop1)
terminate ←false
while (terminate = false) do at each Imperialist
    Move the colonies of an empire toward the imperialist (pop2) ←P-Assimilation (P_A), β
    Crossover some colonies with Empire (pop3) ←P-Crossover (P_C)
    Revolution among colonies (pop4) ←P-Revolution (P_R)
    Evaluate fitness of each country
    Update NFC
    Merge all created population
    Update Colonies
    Update Archive ←n-Archive
    if (Cost of colony is lower than its own Empire) then
        Exchange the positions of the imperialist and a colony
    end
        Calculate Total power of the empires ←ξ
        Perform imperialistic competition
        Eliminate the powerless empires (the imperialist with no colony)
    if (NFC = predefined value) then
        terminate = true
    end if
end while
```

Fig. 11. Pseudo code for the proposed MOICA

5- Numerical experiments

This section presents the results of computational experiments to investigate the performance of the model and solving methods. First a simple instance is presented; then, the intervals used for generating problem instances and comparison metrics are described.

5-1- A simple instance

A simple example with the data presented in section 4.3.1 is described and analyzed here. In this instance, three suppliers and four retailers are distributed as is shown in figure 12.

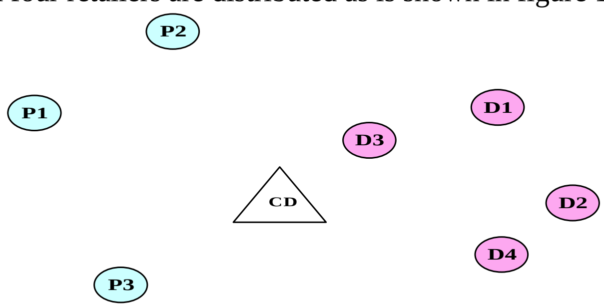

Fig.12. The distribution of nodes in the simple instance

In order to solve this problem, we define three scenarios: in the first one, it has been assumed that direct shipment is the only strategy to transfer parts. In the second one, just using cross-dock is allowed. In the last scenario, both cross-docking and direct shipment is permitted as considered in this study. In order to make the problem more simple, the time windows of retailers and thus the second objective function is not considered and just transportation cost in three scenarios are compared. Operational cost of vehicles is set to 100 and x-y coordinate of all nodes is assumed to be in a plane of [0, 100]. Transportation cost between each couple of nodes is as their distance. Capacity of vehicles is assumed 20 and variable cost of each unit of product at cross-docking center is considered as 0.2.

Figure 13 indicates the first scenario for flow of parts in the network, in which totally 19 vehicles are used and travel cost (objective function 1) is 3351. Cost terms of objective function 1 are reported in details in table 3.

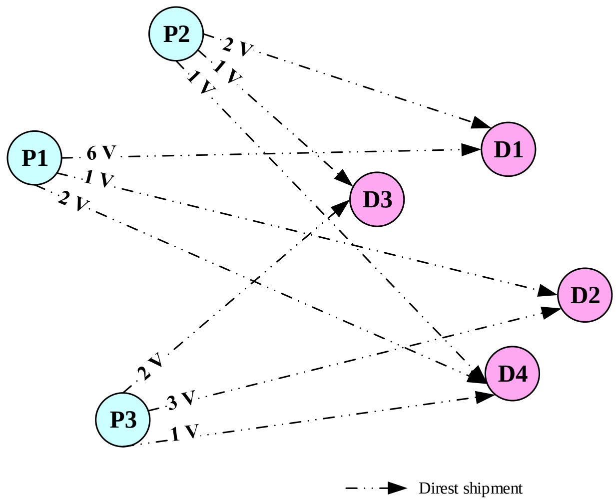

Fig. 13. The flow of parts in the first scenario (direct shipment is permitted).

Table 3. Cost terms of objective function 1 for simple instance (first scenario)

Cost type	**Value**
Direct transshipment cost (including transportation cost and operational cost of vehicles)	3351
Transportation cost from suppliers to cross-dock (including transportation cost and operation cost of vehicles)	0
Transportation cost from cross-dock to retailers (including transportation cost and operation cost of vehicles)	0
Variable cost in cross-dock	0
Total travel cost	3351
Number of used vehicles	
Vehicles in direct transshipment	19
Vehicles in pickup process	0
Vehicles in delivery process	0

In figure 14 the flow of parts is depicted in a network in which all shipments are forced to pass through the cross-docking center. It is obvious that for consignments that are TL, it adds an additional cost to the problem. Going directly to retailers is an economical solution for TL consignments. Table 4 presents the cost terms of objective function 1 in the second scenario.

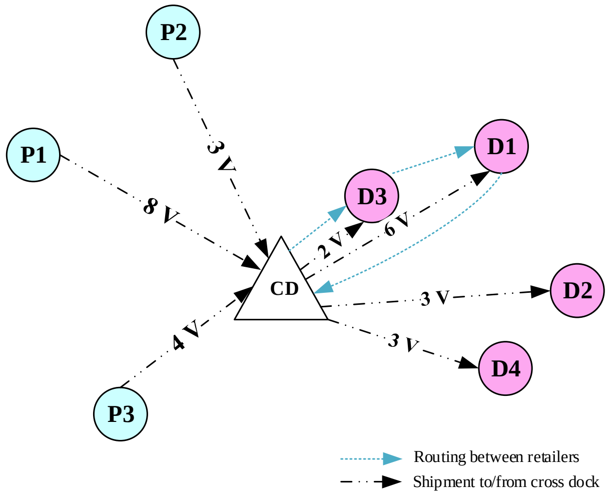

Fig. 14. The flow of parts in the second scenario (cross-docking is permitted).

Table 4. Cost terms of objective function 1 for simple instance (second scenario)

Cost type	**Value**
Direct transshipment cost (including transportation cost and operational cost of vehicles)	0
Transportation cost from suppliers to cross-dock (including transportation cost and operation cost of vehicles)	2185
Transportation cost from cross-dock to retailers (including transportation cost and operation cost of vehicles)	2150
Variable cost in cross-dock	57.6
Total travel cost	4392.6
Number of used vehicles	
Vehicles in direct transshipment	0
Vehicles in pickup process	15
Vehicles in delivery process	15

As mentioned before, in the third scenario both transportation strategies are permitted, therefore in TL consignments, the vehicles move directly to retailers and in LTL, consolidation is done. For the simple instance, applying this scenario reduces the cost about 11.6% compared to direct shipment and 46.2% compared to pure cross-docking strategy. Figure 15 shows the flow of parts in the third scenario and the cost term of objective function 1 are summarized in table 5.

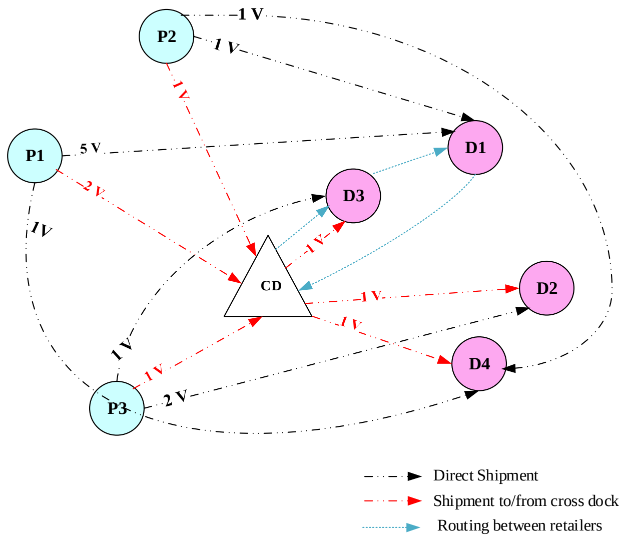

Fig. 15. The flow of parts in the third scenario (both scenarios are permitted).

Table 5. Cost terms of objective function 1 for simple instance (third scenario)

Cost type	Value
Direct transshipment cost (including transportation cost and operational cost of vehicles)	1966
Transportation cost from suppliers to cross-dock (including transportation cost and operation cost of vehicles)	583
Transportation cost from cross-dock to retailers (including transportation cost and operation cost of vehicles)	441
Variable cost in cross-dock	13.6
Total travel cost	3003.6
Number of used vehicles	
Vehicles in direct transshipment	11
Vehicles in pickup process	4
Vehicles in delivery process	4

5-2- Numerical results

In this section, the performances of the proposed MOICA is evaluated and compared with PAES and NSGA-II, in terms of the solution quality and some other metrics. To do so, we perform some numerical experiments on a set of randomly generated problem instances in small and large sizes. The programming model is coded and implemented in Matlab. All experiments are performed on a PC with a Core 5 Duo CPU processor and 4 GB of RAM. TL capacity and operational cost of the vehicles were set to 200 and 1000 units, respectively. The generation of test problems in terms of quantity of demand and time window is based on a paper of Ma et al. (2011). The values of parameters are distributed randomly in U~ [$\frac{4}{5}$parameter, $\frac{6}{5}$ parameter]. Thus, by setting the ratio of $\frac{s}{Q}$, we generate normal demand quantity ($\frac{300}{200}$) and high demand quantity ($\frac{500}{200}$). Besides, consider (d, t)—time window length (l_i- e_i,) and distances between suppliers, cross-dock and customers. By setting the ratio of$\frac{d}{t}$, wegenerate ‘‘normal time window’’$\frac{40}{30}$and ‘‘narrow time window’’($\frac{20}{30}$) cases. Thus, the length of desired interval is randomly generated from $[\frac{4}{5} \times \frac{4}{3} \times 110, \frac{6}{5} \times \frac{4}{3} \times 110]$ = [117, 176] for “normal time window” and [59, 88] for “narrow time window”. For convenience, *LE* and *UE* are assumed 0 and 1080 for all nodes, respectively. It is also assumed that the planning horizon, *T*, is 18 h.

Table6. Parameter values

Parameters	Value		
Problem size	N=10	N=30	N=50
Number of pickup nodes	4	10	15
Number of delivery nodes	6	20	35
V	10	20	40
F	2	2	2
Ser_i	U~ (10,80)	U~ (10,80)	U~ (10,80)
tc_{ij}	U~ (48,560)	U~ (48,560)	U~(48,560)
t_{ij}	U~ (20,200)	U~ (20,200)	U~ (20,200)
s_{ij}	U~ (240,360)	U~ (240,360)	U~ (400,600)

5-3- Comparison metrics

According to table 3, we define six test problems and repeat each of them for five times. Then we list the average values in tables related to comparison metrics. There are a number of methods to compare the performance of different algorithms. To validate the reliability of the proposed MOICA, we use four comparison metrics in this study.

- *Quality metric (QM)*: This metric indicates the number of non-dominated solutions found by each algorithm. We compare the number of solutions belonging to each algorithm with the total number of non-dominated solutions. Higher quality shows better performance.

Table 7. The value of quality metric (QM) for problem sets

Problem instance	Number of nodes	Type of Time window	NSGA-II	PAES	MOICA
1	10	normal	0	0	1
2	10	narrow	0.328	0	0.672
3	30	normal	0.094	0	0.906
4	30	narrow	0.215	0	0.785
5	50	normal	0	0	1
6	50	narrow	0	0	1

As indicated in the table, the proposed algorithm can find a greater number of non-dominated solutions compared with NSGA-II and PAES especially in large-sized test problems; hence, the quality metric of it is equal or near to 1 in most of test problems. It should be noted that NSGA-II can also find Pareto solutions in some cases.

- *Mean ideal distance (MID)*: We use MID to determine the closeness between non-dominated set of solutions and ideal pointf_1^{best}, f_2^{best}. The equation of the MID is as follows:

$$MID = \frac{\sum_{i=1}^{n} \sqrt{\left(\frac{f_{1i}-f_1^{best}}{f_{1,total}^{max}-f_{1,total}^{min}}\right)^2 + \left(\frac{f_{2i}-f_2^{best}}{f_{2,total}^{max}-f_{2,total}^{min}}\right)^2}}{n} \tag{28}$$

where n is the number of non-dominated solutions and $f_{i,total}^{max}$ and $f_{i,total}^{min}$ are the maximum and minimum values of each fitness function among the all non-dominated solutions obtained by the algorithms, respectively (Mohammadi et al., 2013). Regarding to definition, a less value for the MID means better performance.

Table 8. The value of mean ideal distance (MID) for problem sets

Problem instance	Number of nodes	Type of Time window	NSGA-II	PAES	MOICA
1	10	normal	0.691	0.832	0.205
2	10	narrow	0.425	0.972	0.364
3	30	normal	0.716	1.031	0.429
4	30	narrow	0.533	0.752	0.484
5	50	normal	0.827	0.594	0.568
6	50	narrow	0.824	0.818	0.342

In all test problems, the MID values of proposed algorithm are smaller than those of others, which shows that solutions found by MOICA are closest to ideal point

- *Diversification metric (DM)*: This metric measures the spread of the solution set and calculated by:

$$DM = \sqrt{\left(\frac{max\, f_{1i} - min\, f_{1i}}{f_{1.total}^{max} - f_{1.total}^{min}}\right)^2 + \left(\frac{max\, f_{2i} - min\, f_{2i}}{f_{2.total}^{max} - f_{2.total}^{min}}\right)^2} \tag{29}$$

Based on this measure, the upper the DM's value, the better the corresponding Pareto set.

Table 9. The value of diversification metric (DM) for problem sets

Problem instance	Number of nodes	Type of Time window	NSGA-II	PAES	MOICA
1	10	normal	0.523	0.405	0.649
2	10	narrow	0.417	0.581	0.426
3	30	normal	0.525	0.588	0.639
4	30	narrow	0.792	0.547	0.656
5	50	normal	0.779	0.499	0.787
6	50	narrow	0.873	0.340	0.587

In 67% of cases, the values of the diversity metric of the proposed MOICA are greater than other algorithms.

- *Spacing metric (SM)*: This metric allows us to measure the uniformity of the spread of the non-dominated set of solutions (Tavakkoli-Moghaddam et al., 2011). The definition of the spacing metric is as follows:

$$SM = \frac{\sum_{i=1}^{n-1} \left|\bar{d} - d_i\right|}{(n-1)\bar{d}} \tag{30}$$

where $d_i$is the Euclidean distance between consecutive solutions in the obtained non-dominated solution set, and $\bar{d}$is the mean value of these distances.

Table 10. The value of spacing metric (SM) for problem sets

Problem instance	Number of nodes	Type of Time window	NSGA-II	PAES	MOICA
1	10	normal	0.592	0.387	0.704
2	10	narrow	0.916	0.285	0.644
3	30	normal	0.674	0.516	0.621
4	30	narrow	0.621	0.327	0. 569
5	50	normal	0.728	1.141	1.273
6	50	narrow	0.963	0.282	0.730

The values of spacing metric in PAES are smaller than those of others, which show that non-dominated solutions are spread uniformly within this algorithm. From this point of view, MOICA does not perform well.

Figure 16 shows the result of problem instance 6. As can be inferred from this figure, all non-dominated solutions belong to MOICA. MID of MOICA is definitely smaller than two other algorithms. In this instance, the value of DMin NSGA-II is the greatest and it is more spread compared to other two algorithms. In instance 6, SM value of PAES is the least and it generates the uniform set of non-dominated solutions.

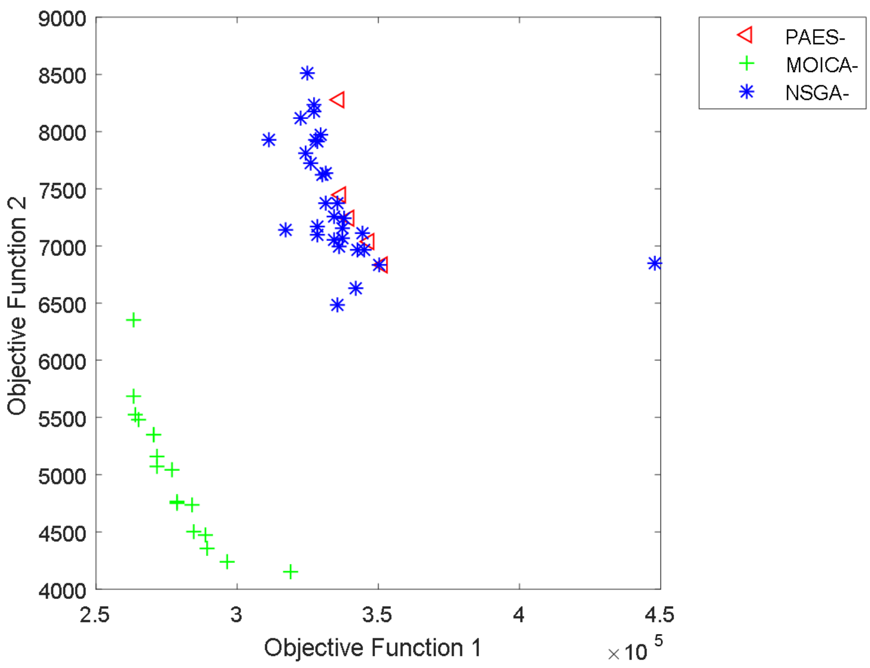

Fig. 16. Non-dominated solutions in problem instance 6

6- Conclusion

Cross docking is a logistics technique aimed to reduce inventory and transportation costs, order picking, and delivery time in the supply chain. This approach is applied to transfer different products from various origins to multiple final destinations. This study addresses the transportation problem of cross-docking network where the shipments are allowed to be transferred from suppliers to retailers directly as well as through cross-docks. The objective functions try to 1) minimize the total travel cost for a number of routes (i.e., the sum of the operational cost of vehicles and the transportation cost) and 2) minimize earliness and tardiness of visiting nodes. In order to solve the problem, three multi-objective algorithms are used and a heuristic based phase is introduced to generate initial solutions. The performance of multi-objective algorithms was analyzed by means of four measures (i.e., quality, spacing, diversity and mean ideal distance) which demonstrated that the MOICA can relatively overwhelm other two algorithms. In all problems, the number of non-dominated solutions of the proposed MOICA is greater than those of other algorithms. It means that its quality metric is more than others are. In addition, the value of MID metric in the proposed MOICA is less than NSGA-II and PAES for all test problems. From the view of the diversity metric, the proposed MOICA can find solutions with greater values in most cases. In fact, the spacing metric of PAES in most cases is less than other metrics. The results show that the proposed MOICA is superior to the NSGA-II and PAES in each test problem.

In the case of future studies, we can suggest other varieties of cross-docking systems, in which one single cross-dock is replaced by several cross-docks. Furthermore, considering other objective functions can help to have a more realistic problem. Using other multi-objective optimization methods and comparing the result can also be done. Finally, considering a queuing theory in the mathematical model for a cross-docking system can be proposed as the future research.

References

Atashpas-Gargari, E., Lucas, C., "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialist competitive," *Proceeding IEEE Congress on Evolutionary computation*, CEC 2007, Singapore, 2007.

Chen,M-Ch, Hsiao,Y-H, , Reddy,R. H., Tiwari,M. K.,"The Self-Learning Particle Swarm Optimization approach for routing pickup and delivery of multiple products with material handling in multiple cross-docks," *Transportation Research Part E: Logistics and Transportation Review*, Vol. 91, pp. 208–226, 2016.

Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T., "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," *In M.S. et al.* (Ed.), Parallel Problem Solving from Nature – PPSN VI, Berlin, Springer, pp. 849–858, 2000.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Transaction on Evolutionary Computation*, Vol. 6(2), pp. 182–197, 2002.

Dondo, R., Méndez, C. A., Cerdá, J., "The multi-echelon vehicle routing problem with cross docking in supply chain management," *Computers and Chemical Engineering*, Vol. 35, pp. 3002– 3024, 2011.

El-Ghazali Talbi, "Metaheuristicsfrom design to implementation", *Wiley*, University of Lille – CNRS – INRIA, 2009

Gumus, M, Bookbinder, J. H. "Cross-docking and its implications in location– distribution systems," *Journal of Business Logistics*; Vol. 25(2), pp. 199–228, 2004.

Hasani Goodarzi,A., Zegordi,S. H., "A Location- Routing Problem for Cross-docking Networks: A Biogeography-based Optimization Algorithm," *Computers and Industrial Engineering*, vol. 102, pp. 132–146, 2016.

Hosseini, S.D., Akbarpour Shirazi, M., & Karimi, B., "Cross-docking and milk run logistics in a consolidation network: A hybrid of harmony search and simulated annealing approach,"*Journal of Manufacturing Systems*, vol. 33, pp. 567–577, 2014.

Knowles, J.D., and Corne, D.W., "The Pareto archived evolution strategy: A new baseline algorithm for Pareto multi-objective optimization," *Congress on Evolutionary Computation (CEC99)*, Vol. 1, Piscataway, NJ, IEEE Pres, pp. 98–105. 1999.

Konur, D., Golias, M.M.,"Cost-stable truck scheduling at a cross-dock facility with unknown truck arrivals: A meta-heuristic approach," *Transportation Research Part E*, Vol. 49, pp. 71–91, 2013.

Ladiera,A-L, Alpan,G., "Cross-docking operations: Current research versus industry practice," *Omega*, Vol. 62, pp. 145–162, 2016.

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, pp. 247–256, 2006.

Liao, C.-J., Lin, Y., and Shih, S.C., "Vehicle routing with cross-docking in the supply chain," *Expert Systems with Applications*, Vol. 37, pp. 6868–6873, 2010.

Ma, H., Miao, Z., Lim, A., Rodrigues, B.,"Crossdocking distribution networks with setup cost and timewindow constraint,"*Omega*, Vol. 39, pp. 64–72, 2011.

Marseguerra, M., Zio, E., and Podofillini, L., "Condition-based maintenance optimization by means of genetic algorithms and Monte Carlo simulation," *Reliability Engineering and System Safety*, Vol. 77, pp. 151–166, 2002.

Mohammadi, M., Jolai, F. and Tavakkoli-Moghaddam, R., "Solving a new stochastic multi-mode p-hub covering location problem considering risk by a novel multi-objective algorithm," *Applied Mathematical Modelling*,Vol. 37 (24), 2013, pp. 10053–10073.

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(1), pp. 85-92, 2010.

Nikolopoulou,A.I., Repoussis,P.P., Tarantilis,Ch.D., Zachariadis,E. E.,“Moving products between location pairs: Cross-docking versus direct-shipping,” *European Journal of Operational Research*, Vol. 256 (3), pp. 803–819, 2017.

Ombuki, B., Ross, B.J., and Hanshar, F., “Multi-Objective Genetic Algorithms for Vehicle Routing Problem with Time Windows,” *Applied Intelligence, Springer*, Vol. 24, pp. 17–30, 2006.

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, pp. 64–79, 2008.

Santos, F.A., Mateus,G. R., Salles da Cunha, A.,“The Pickup and Delivery Problem with Cross-Docking,”*Computers & Operations Research*, Vol. 40,pp.1085–1093, 2013.

Santos, F.A., Mateus, G.R., da Cunha, A.S., “A branch-and-price algorithm for a vehicle routing problem with cross docking. In: LAGOS’11—VI Latin American algorithms, graphs and optimization symposium,” *Electronic notes in discrete mathematics*, Vol. 37, pp. 249–254, 2011.

Santos, F.A., Mateus, G.R., da Cunha, A.S., “A novel column generation algorithm for the vehicle routing problem with cross-docking In: Network optimization,” *Lecture notes in computer science, Springer,* Vol. 6701, pp. 412–425, 2011.

Shi, W., Liu, Z., Shang,J., Cui, Y., “Multi-criteria robust design of a JIT-based cross-docking distribution center for an auto parts supply chain,” *European Journal of Operational Research*, Vol.229, pp. 695–706, 2013.

Tarantilis, Christos D., “Adaptive multi-restart Tabu Search algorithm for the vehicle routing problem with cross-docking,” *Optim Lett*, Vol. 7, pp. 1583–1596, 2013.

Tavakkoli-Moghaddam, R., Azarkish, M. and Sadeghnejad, A., “A new hybrid multiobjective Pareto archive PSO algorithm for a bi-objective job shop scheduling problem,” *Expert System with Applications*. Vol. 38(9), pp. 10812-10821, 2011.

Vahdani, B., Tavakkoli-Moghaddam, R., Zandieh, M., and Razmi, J. “Vehicle routing scheduling using an enhanced hybrid optimization approach,” *Journal of Intelligent Manufacturing*, Vol. 23, pp. 759–774, 2012.

Vasiljevic,D., Stepanovic,M., Manojlovic, O., “Cross docking implementation in distribution of food products,”*Economics of Agriculture*, Year 60, No. 1 (1-216), pp 91-101, Belgrade, 2013.

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 (12), pp. 1708-1718,2008.

Yang, KK., Balakrishnan, J., Cheng, CH. “An analysis of factors affecting cross docking operations,” *Journal of Business Logistics*, Vol. 31(1), pp.121–48, 2010.

Yin,P-Y, Chuang,Y-L, “Adaptive memory artificial bee colony algorithm for green vehicle routing with cross-docking,” *Applied Mathematical Modelling*, Vol. 40 (21–22), pp. 9302–9315,2016.

Yu, V.F., Jewpanya,P., Perwira Redi,A.A.N., “Open vehicle routing problem with cross-docking,” *Computers & Industrial Engineering*, Vol. 94, pp. 6–17, 2016.