

## A sustainable transportation-location-routing problem with soft time windows for distribution systems

Masoud Rabbani<sup>a\*</sup>, Fatemeh Navazi<sup>a</sup>, Hamed Farrokhi-Asl<sup>b</sup> and Mohammad Hasan Balali<sup>c</sup>

<sup>a</sup>*School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran*

<sup>b</sup>*School of Industrial Engineering, Iran University of Science & Technology, Tehran, Iran*

<sup>c</sup>*Department of Industrial engineering, University of Wisconsin-Milwaukee, Milwaukee, WI, USA*

### CHRONICLE

#### Article history:

Received March 2, 2017  
Received in revised format  
October 10, 2017  
Accepted December 3 2017  
Available online  
December 4 2017

#### Keywords:

*Transportation-location-routing  
problem  
Soft time windows  
Sustainability  
Distribution network*

### ABSTRACT

Increasing in attentions to the environment, city legislative and social problems make companies change their prospects towards supply chain management and design sustainable transportation networks. In this paper, two-stage problem have been investigated in which the transportation stage is considered before Location-Routing Problem, so we call it Transportation-Location-Routing Problem (TLRP). It is an extension of the two-echelon Location-Routing Problem. In the first stage, there is a transportation problem with truck capacity limitation. Furthermore, customers' time windows should be met in the second stage to make the mode more realistic. Minimization of distribution cost, fuel consumption, and carbon dioxide emission along with balancing the workloads for city drivers are considered as the objective functions of the mathematical model to design a sustainable distribution network. To tackle these conflicting objectives, Non-dominated Sorting Genetic Algorithm (NSGA-II) and Multi-Objective Particle Swarm Optimization (MOPSO) are applied to solve the problem. A new customized chromosome based on a priority based technique is presented for the problem. Due to the three comparison metrics for multi-objective problems, with tolerating a little more computational time, MOPSO has the better performance in this problem than NSGA-II.

© 2018 Growing Science Ltd. All rights reserved.

## 1. Introduction

Designing an efficient distribution system for delivering the final products to the customers involves all decisions about transportation, routing, and location of intermediate facilities such as warehouses or urban distribution centers. The existence of different stakeholders in supply chain caused the sustainability approach in Supply Chain Management (SCM), because this approach tries to preserve the interests of all stakeholders together. Stakeholders in this problem could be divided into three categories: customers, the people who are working in the distribution system, and the local people who are affected by this system (Eskandarpour et al., 2015). Thus, a sustainable distribution system should consider all stakeholders' interests. Since the separate design of the location for the distribution centers and vehicles' route from these facilities causes higher costs, the Location-Routing Problem (LRP) is

\* Corresponding author.

E-mail address: [mrabbani@ut.ac.ir](mailto:mrabbani@ut.ac.ir) (M. Rabbani)

established as a combination of Facility Location Problem (FLP) and Vehicle Routing Problem (VRP). Decisions in supply chain comprise strategic decisions such as location decisions, operational decisions like inventory policies, and tactical decisions such as vehicle routing decisions (Prodhon & Prins, 2014). LRP integrates the strategic and tactical level of decisions to avoid generating sub-optimal results by solving them separately (Nagy & Salhi, 2007). Moreover, according to NP-hard nature of FLP and VRP (Azadeh & Farrokhi-Asl, 2017), LRP that is also an extension to aforementioned is NP-hard problem (Farrokhi-Asl et al., 2016).

This study is inspired from a real-world problem, soft drink industry, where the extension of two-echelon LRP is encountered. There are some factories out of the city and some stores, restaurants, and hotels in the city. Because of the city traffic regulations for transportation of large vehicles like trucks, the direct connections between plants and clients are prohibited. Therefore, the necessity of intermediate facilities such as distribution centers (DCs) is felt. To design the distribution system, it is needed to locate some distribution centers beyond the determined potential places in the city's outskirts. Then, deciding about the number of trucks, which are needed to be sent from different plants to each distribution center leads to a transportation problem in the first stage. After that, the allocation of clients to each distribution center and their sequence in the vehicle visiting brings on a vehicle routing problem in the second stage. Combination of these two stages introduce a new field in Operations Research (OR) called transportation location routing problem.

To assess the performance of a supply chain or the distribution network, the cost of providing customers' demands such as the cost of transportation, distribution facilities, etc. should be assessed as well. Also, in this research, the customer's other demands that should be satisfied such as diversity of products are considered. Nowadays, plants consider this necessity and offer a product range (even one product in different package or volume) and a factory with a single product barely could be found. Hence, in this study it has been decided to consider the problem as a multi-product model to be more realistic unlike Martinez-Salazar et al. (2014) that is a single product. For instance, a beverage factory usually produces drinks in different sizes (for example in the cans and in the 1-liter bottles). Although each customer will receive all kinds of its required products on a one visit (Zhang & Chen, 2014), but different weights or volumes of various products occupies different sizes of the trucks, so it should be addressed because of its impact on the final number of required trucks.

Since time is always considered as a valuable resource (Moshref-Javadi et al., 2016), soft time windows are considered for each client to make the model more realistic. For example, restaurants must receive the drinks at the specific time interval before the lunch time, grocery stores must receive their orders at their opening time in the morning and so forth. Violation from each customer's time windows is highly penalized. The penalty as a cost of losing fame, future lost sales due to this competitive condition among different distribution systems, etc. is taken into account. Therefore, on-time delivery as one of the customer interests is paid attention (Nasherahkami et al., 2016).

Increasing environmental concerns in the last decades leads to optimize a green objective function in this problem (Rabbani et al., 2017). So, minimizing the fuel consumption rate based on the weight of the vehicle and the traveled distance is reckoned (Xiao et al., 2012). Also, emission of greenhouse gasses such as CO<sub>2</sub>, which depends on the fuel consumption rate is minimized. By these considerations, the interests of local people are considering. The goal to design a sustainable distribution system is achieved more by considering social objective besides environmental concerns. Balancing the workload of the city drivers causes work equity and fairness for employees as other stakeholders. It is used as a third objective function accompanied by minimizing operation costs of the distribution system. Even though, this problem is based on the soft drink industry, it could be used for other industries such as detergents, home appliances and distribution of perishable foods, which have multiple products with different weights and each customer's demands of all products should be met in a specific time windows.

The rest of the paper is organized as follows: Section 2 involves a comprehensive review of related literature. In Section 3, the introduced problem is described and its mathematical formulation is presented. After reviewing two multi-objective metaheuristic methods in Section 4, the customized solution representation is specified for the model. In the first part of the next section, a small example is solved to verify the model. Then, the metaheuristic parameters are tuned. Thereafter, numerical results for several problem instances and metaheuristics comparison by comparative metrics appear. Ultimately, the summary of conclusions and some future directions are reported in Section 6.

## 2. Literature review

Location routing problem was introduced by Maranzana (1964). He proved that location of facilities such as factories, warehouses, etc. influenced the transportation costs. In LRP there are some potential places for locating depots and some vehicles to deliver goods from depots to customers in a sequence that cause the minimum cost. For studying more about LRP variants interested readers are referred to Prodhon and Prins (2014).

Most of the LRP studies were in single stage like Moshref-Javadi and Lee (2016). They considered to open the determined maximum number of depots between some predicted places, assign customers to them and diagnose the vehicle open routes from depots to customers. There is a predetermined number of vehicles that the problem uses all of them. Their study is a single objective with a new objective of minimizing the total latency at all customers by cumulating arrival time at each recipient in the objective function. Multi-objective approach for commercial distribution networks design is declared as a future research in their study. But considering incapacitated depots may make their problem far from real word problems.

By adding the maximum possible time, where each route could last to a single stage problem, which decides about the location of distribution centers among the potential places and then the vehicle routing problem among customers and opened depots, Khalili-Damghani et al. (2015) customized the LRP for perishable products such as fish, dairy products, etc. They used heterogeneous vehicles and solved a bi-objective problem. The first objective was a classic objective of LRP; minimizing the operational cost of the distribution network. The second objective was a social objective; balancing the workload for transportation employees by minimizing the difference between the longest and shortest route length. They used an evolutionary algorithm called Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) with customized chromosome and genetic operators for solving the problem. To assess the performance of their model performance it was compared with  $\epsilon$ -constraint (an efficient exact Multi-objective method) by accuracy and diversity metrics.

This social objective was also presented by Martínez-Salazar et al. (2014) in a two stage transportation location routing problem. They tried to minimize the number of trucks based on the truck capacity in the first stage and balancing the routes for city drivers in the second stage. In the first stage, there are multiple distribution centers, which can be satisfied from more than one manufacture in case their demand exceeded the manufacture's capacity. It is like the split delivery models, where each distribution center could feed from multiple sources (Ray et al., 2014). A priority base solution representation is proposed and Scatter Tabu Search Procedure for Non-Linear Multi-Objective Optimization (SSPMO) is developed for their problem. By using NSGA-II the efficiency of SSPMO for small instances is proved. The other social objective; the negative impact of treatment facilities on close areas is utilized in location routing problem by waste collection practice. A clustering approach is used for generating initial solutions for the NSGA-II algorithm (Rabbani et al., 2017).

Combination of different variants of vehicle routing problem with facility location problem create various types of location routing problems. A vehicle routing problem with the soft time windows, which is the relaxed form of the hard time windows with violation penalty in the cost function is solved by the new swarm based Artificial Bee Colony (ABC) algorithm combined with two-step constrained

local search for neighborhood selection by Iqbal et al. (2015) in a Multi-objective problem. A location routing problem with the hard time windows, which customer's service cannot start before the lower bound of time window or after the upper bound of that, was investigated in an uncertain environment with fuzzy travel times by Fazel Zarandi et al. (2017). Minimizing the total additional travel distance due to the rout failures was the other objective added to traditional cost minimization objective.

The two-echelon LRP with the soft time windows was studied for perishable foods distribution by considering sustainability in decision making (Govindan et al., 2014). Although they minimized the energy consumption of distribution centers and manufacturers besides VRP green objectives but social objectives did not consider in their work to complete sustainability in decision making. To simplify the model, they used a parameter called average environmental impact for each arc instead of calculation of fuel consumption and CO<sub>2</sub> emission costs. A novel multi-objective hybrid algorithm called MHPV that is the hybrid of MOPSO and Adapted Multi-Objective Variable Neighborhood Search (AMOVNS) was proposed for the problem. A closed-loop location routing inventory problem by sustainability consideration was modeled in an uncertain environment by Zhalechian et al. (2016). The social impacts of created job opportunities and economic development were considered as a societal objective in their study.

Unlike this study, where customers should be met once like Zhang and Chen (2014), a multi-product model for a two-echelon location routing problem with pickup and delivery was studied in which each client could be visited more than once for satisfying demand of different types of products. To study more about the two-echelon location routing problem readers are referred to Rahmani et al. (2016) and Cuda et al. (2015).

Green vehicle routing problem is a vehicle routing problem by minimizing the environmental damages. There are different aspects for achieving this goal. One of them is minimizing energy consumption in the network (Erdoğan & Miller-hooks, 2012). Fuel consumption rate (FCR) based on the distance of arc between two nodes and a load of the vehicle is taken into account by Kara et al. (2007) in order to minimize the fuel consumption of the capacitated vehicles. Kuo (2010) developed a speed based FCR for vehicles with time-dependent speed. Because of the congestion during a day, the time-dependent speed is the more realistic model of real-world. The time-dependent speed leads to savings in CO<sub>2</sub> emissions about 7% (Erdoğan & Miller-hooks, 2012). Considering the cost of FCR may raise the value of the total objective function, but leads to more realistic cost estimate (Faulin et al., 2011).

The other facet of the green objective is to minimize the emission of greenhouse gasses (GHG) such as carbon dioxide to the environment. Bektas and Laporte (2011) state that the using one-liter fuel releases 2.3 kg amount of CO<sub>2</sub>. There are a lot of views in calculating the cost of CO<sub>2</sub> emission. According to Forkenbrock (2001), each ton of the CO<sub>2</sub> has 10 - 20 dollars cost. In other study, each ton costs 27 pounds in 2010 and its increases 2% each year (Price et al., 2011). Due to the environmental consideration in this study; readers can read the work presented by Lin et al. (2014) as a survey on green vehicle routing problem. Significant features of this study versus other recent related studies are shown in Table 1.

**Table 1**  
Significant features of this study versus other recent related studies

papers	Features											
	LRP	Two echelon transportation	City traffic regulation	Fuel tank limitation	Soft time windows	Product range	Sustainability objectives				Others.	
							Min FCR	Green objective based on weight	Min CO2 emission	Social objective Balance work load		
This paper	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Martinez et al. (2014)	✓	✓	✓	✓								✓
Khalili et al. (2015)	✓			✓								✓
Hossein et al. (2013)	✓				✓							
Govindan et al. (2014)	✓	✓			✓					✓		
Zhalechian et al. (2016)	✓	✓				✓				✓		✓
Xiao (2012)										✓		✓

<sup>1</sup> H: Hard time window

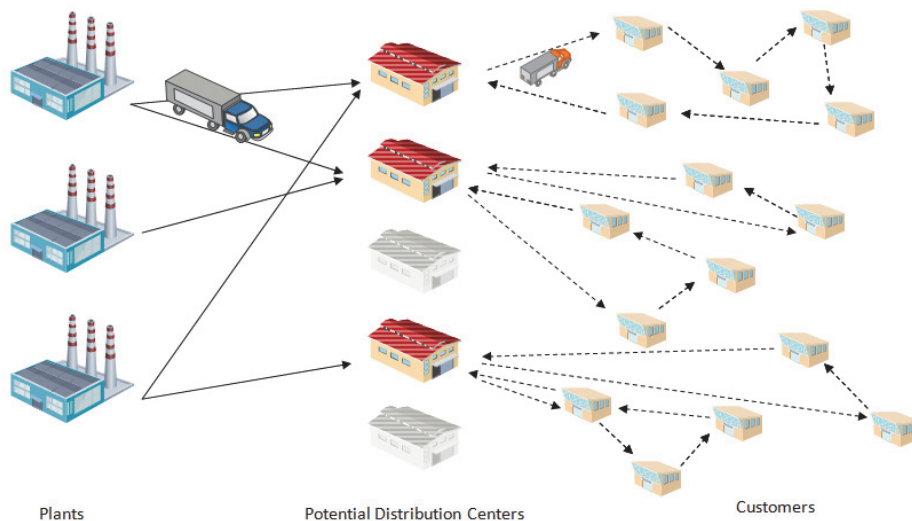
### 3. Problem description and mathematical formulation

Traffic restrictions on the entry of the trucks in populated cities became an inspiration of two echelons location routing problems. In this problem, the distribution centers should be located as some intermediate facilities, which receive goods from plants and deliver them to the customers, among some potential points in city's outskirts. Therefore, there is a facility location problem with defined maximum capacity. Each DC has a fixed cost for its opening. Capacity limitation of trucks, which transfer goods from known plant locations caused transportation problem in the first echelon. As mentioned before, the importance of satisfying customer demands, plants are multi-products. Customers (retailers) soft time windows make vehicle routing problem more complicated but practical in the real world. In the second echelon, vehicles are homogeneous and after delivery, they should return to the departing distribution center. Customer demands are known and deterministic. A sample solution for the transportation problem in stage one associated with vehicle routing problem in stage two through a facility location problem for DCs is shown in Fig. 1.

Sustainability in this distribution network has been considered by reduction of economic cost of routing problems, opening distribution centers and time windows violations penalties, reduction of vehicles energy consumption and CO<sub>2</sub> emission as an environmental objective; balance of driver's workloads as a social objective because of considering the high value of the human resource.

For calculating fuel consumption rate (FCR) of trucks in the first stage and for vehicles in the second stage the relationship of fuel consumption rates by the vehicle's weight and traveled distance introduced by Xiao et al. (2012) is utilized in this study. The FCR is calculated by Eq. (1), which is the best linear regression line equation fitted on real statistic data, which are some points with vehicles weight on the horizontal axis and their FCR per kilometer on the vertical axis (Xiao et al., 2012). With this formulation, the FCR for unloaded and fully-loaded vehicles could be easily obtained. Therefore, the FCR for a partially-loaded truck is caught by linear interpolation between FCR for fully-loaded vehicle and FCR for the empty vehicle.

$$Y(\text{FCR per km}) = 0.0000793 \times X(\text{weight}) - 0.026 \quad (1)$$



**Fig. 1.** Schematic view of the problem

### 3.1. Assumptions

- There are  $p$  plants to manufacture different types of the products
- $G$  different sets for type of the goods are considered
- Each plant can work with its maximum capacity for each product  $a_{gk}$  expressed in units of product  $g$
- The finished goods inventories from plants depot must transfer to DCs by trucks with maximum capacity  $Qt$  expressed in weight
- Customer demands from each good are known  $d_{gi}$
- Vehicles have a maximum capacity  $Qv$  expressed in weight
- logically  $Qv \leq Qt$
- The travel distance for each vehicle cannot exceed a fixed value  $R$  as its fuel tank limitation
- Retailers have a soft time windows based on their customer's behavior for receiving the demands. There is a penalty if the vehicles violate the retailers' due dates.

### 3.2. Notations

Sets:

$K$	$\in \{1, 2, \dots, p\}$	Set of indexes for plants
$J$	$\in \{1, 2, \dots, m\}$	Set of indexes for DCs
$I$	$\in \{m + 1, m + 2, \dots, m + n\}$	Set of indexes for clients
$G$	$\in \{1, 2, \dots, g\}$	Set of indexes for products

Parameters:

$P$	Number of plants
$M$	Number of potential locations for DC location
$N$	Number of clients
$G_i$	Number of goods in the product range
$w_g$	Weight of good $g$
$a_{gk}$	Production capacity in plant $k$
$h_{kj}$	Cost of sending a truck from plant $k$ to DC $j$
$r_{kj}$	Distance from plant $k$ to DC $j$
$Qt$	Capacity of Trucks
$Wt_0$	Weight of no load truck
$Wt^*$	Weight of full load truck ( $wt^* = wt_0 + Qt$ )
$\rho t_0$	Fuel consumption rate of empty truck
$\rho t^*$	Fuel consumption rate of fully loaded truck
$f_j$	Fixed cost for opening and operate DC $j$
$b_{jg}$	Capacity of DC $j$ for product $g$
$d_{gi}$	Demand of customer $i$ for Good's $g$
$c_{ij}$	Cost of visiting client/DC $j$ right after client/DC $i$ in the route
$QV$	Vehicle weight capacity
$Wv_0$	Weight of empty vehicle
$Wv^*$	Weight of fully loaded vehicle ( $wv^* = wv_0 + Qv$ )
$\rho v_0$	Fuel consumption rate of empty vehicle
$\rho v^*$	Fuel consumption rate of fullyloaded vehicle
$r_{ij}$	Distance from client/DC $i$ to client/DC $j$
$R$	Maximum possible total traveling distance of each vehicle

$ff$	Fixed cost of 1 liter fuel
$ce$	CO <sub>2</sub> emission cost of 1 liter fuel consumption
$t_{ij}$	Traveling time from client/DC $i$ to client/DC $j$
$e_i$	Earliest time of customer's $i$ time windows
$l_i$	Latest time of customer's $i$ time windows
$p_e$	Violation penalty of earliest time of time windows
$p_l$	Violation penalty of latest time of time windows
$M$	Big number

Variables:

$LR_{min}$	Length of the shortest route
$LR_{max}$	Length of the longest route
$s_i$	Starting service time at customer $i$
$Ve_i$	Violation amount of earliest time window at customer $i$
$Vl_i$	Violation amount of latest time window at customer $i$
$V_{gkj}$	Amount of product $g$ sent from plant $k$ to DC $j$
$Nt_{kj}$	Number of trucks sent from plant $k$ to DC $j$
$u_i$	Load of vehicle after visiting customer $i$
$o_i$	The distance which vehicle traveled after visiting customer $i$
$y_j = \begin{cases} 1 \\ 0 \end{cases}$	If DC $j$ is opened, Otherwise.
$z_{ij} = \begin{cases} 1 \\ 0 \end{cases}$	if client $i$ is assigned to DC $j$ , Otherwise.
$x_{oi}^j = \begin{cases} 1 \\ 0 \end{cases}$	if client $i$ is the first client in any route of DC $j$ Otherwise.
$x_{io}^j = \begin{cases} 1 \\ 0 \end{cases}$	if client $i$ is the last client in any route of DC $j$ Otherwise.
$x_{il}^j = \begin{cases} 1 \\ 0 \end{cases}$	if client $l$ is visited just after client $i$ in any route of DC $j$ Otherwise.

### 3.3. Mathematical formulation

Objective Functions:

#### 1. Economic objectives

$$\min f_1 = \sum_{k \in K} \sum_{j \in J} h_{kj} nt_{kj} + \sum_{j \in J} f_j y_j + \sum_{j \in J} \sum_{i \in I} (c_{ji} x_{oi}^j + c_{ij} x_{io}^j) + \sum_{j \in J} \sum_{i \in I} \sum_{\substack{l \in I \\ l \neq i}} c_{il} x_{il}^j + \sum_{i \in I} (p_e ve_i + p_l vl_i) \quad (2)$$

#### 2. Green objectives

$$\begin{aligned} \min f_2 = & \sum_{k \in K} \sum_{j \in J} (ff + ce) r_{kj} pt^* \left\lceil \frac{\sum_{g \in G} w_g \times v_{gkj}}{Qt} \right\rceil \\ & + \sum_{k \in K} \sum_{j \in J} (ff + ce) r_{kj} (\rho t_0 + \left( \frac{pt^* - \rho t_0}{Qt} \right) \left( \sum_{g \in G} w_g v_{gkj} - Qt \left\lceil \frac{\sum_{g \in G} w_g \times v_{gkj}}{Qt} \right\rceil \right)) \\ & + \sum_{k \in K} \sum_{j \in J} (ff + ce) r_{jk} \rho t_0 nt_{kj} \end{aligned} \quad (3)$$



$$\begin{aligned}
 & + \sum_{j \in J} \sum_{i \in I} (ff + ce) r_{ji} \left( \rho v_0 x_{oi}^j + \left( \frac{\rho v^* - \rho v_0}{Qv} \right) \left( u_i + \sum_{g \in G} w_g d_{gi} \right) x_{oi}^j \right) \\
 & + \sum_{\substack{i \in I \\ i \neq l}} \sum_{i \in I} (ff + ce) r_{il} \left( \rho v_0 \left( \sum_{j \in J} x_{il}^j \right) + \left( \frac{\rho v^* - \rho v_0}{Qv} \right) u_i \left( \sum_{j \in J} x_{il}^j \right) \right) \\
 & + \sum_{j \in J} \sum_{i \in I} (ff + ce) r_{ij} \rho v_0 x_{i0}^j
 \end{aligned}$$

3. Route balance objective as a social objective

$$\min f_3 = LR_{\max} - LR_{\min} \tag{4}$$

Constraints:

$$\sum_{j \in J} v_{gkj} \leq a_{gk} \quad \forall k \in K, \forall g \in G \tag{5}$$

$$nt_{kj} \geq \frac{\sum_{g \in G} w_g v_{gkj}}{Qt} \quad \forall k \in K, \forall j \in J \tag{6}$$

$$\sum_{k \in K} v_{gkj} \leq b_{jg} y_j \quad \forall j \in J, \forall g \in G \tag{7}$$

$$\sum_{k \in K} v_{gkj} = \sum_{i \in I} d_{gi} z_{ij} \quad \forall j \in J, \forall g \in G \tag{8}$$

$$\sum_{j \in J} z_{ij} = 1 \quad \forall i \in I \tag{9}$$

$$\sum_{i \in I} x_{oi}^j = \sum_{i \in I} x_{i0}^j \quad \forall j \in J \tag{10}$$

$$\sum_{\substack{i \in I \cup \{0\} \\ i \neq l}} x_{il}^j = z_{lj} \quad \forall l \in I, \forall j \in J \tag{11.1}$$

$$\sum_{\substack{i \in I \cup \{0\} \\ l \neq i}} x_{il}^j = z_{ij} \quad \forall i \in I, \forall j \in J \tag{11.2}$$

$$u_i - u_l + Qv \left( \sum_{j \in J} x_{il}^j \right) \leq Qv - \sum_{g \in G} w_g d_{gl} \quad \forall i, l \in I, i \neq l \tag{12}$$

$$\sum_{g \in G} w_g d_{gi} \leq u_i \leq Qv \quad \forall i \in I \tag{13}$$

$$o_i - o_l + (R + r_{il}) \sum_{j \in J} x_{il}^j + (R - r_{li}) \sum_{j \in J} x_{li}^j \leq R \quad \forall i, l \in I, i \neq l \quad (14)$$

$$\sum_{j \in J} r_{ji} x_{oi}^j \leq o_i \leq R + \sum_{j \in J} (r_{ji} - R) x_{oi}^j \quad \forall i \in I \quad (15)$$

$$o_i \leq R - \sum_{j \in J} r_{ij} x_{io}^j \quad \forall i \in I \quad (16)$$

$$LR_{\max} \geq o_i + \sum_{j \in J} r_{ij} x_{io}^j \quad \forall i \in I \quad (17)$$

$$LR_{\min} \leq o_i + \sum_{j \in J} (r_{ij} - R) x_{io}^j + R \quad \forall i \in I \quad (18)$$

$$s_i + M(1 - x_{oi}^j) \geq 0 + t_{ji} \quad \forall i \in I, \forall j \in J \quad (19)$$

$$s_l + M(1 - x_{il}^j) \geq s_i + t_{il} \quad \forall i, l \in I, i \neq l, \forall j \in J \quad (20)$$

$$ve_i \geq e_i - s_i \quad \forall i \in I \quad (21)$$

$$vl_i \geq s_i - l_i \quad \forall i \in I \quad (22)$$

$$y_j \in \{0,1\} \quad \forall j \in J \quad (23)$$

$$z_{ij} \in \{0,1\} \quad \forall i \in I, \forall j \in J \quad (24)$$

$$x_{oi}^j \in \{0,1\} \quad \forall i \in I, \forall j \in J \quad (25)$$

$$x_{io}^j \in \{0,1\} \quad \forall i \in I, \forall j \in J \quad (26)$$

$$x_{il}^j \in \{0,1\} \quad \forall i, l \in I, i \neq l, \forall j \in J \quad (27)$$

$$nt_{kj} \in Z_+ \quad \forall k \in K, \forall j \in J \quad (28)$$

$$v_{gkj} \geq 0 \quad \forall k \in K, \forall j \in J, \forall g \in G \quad (29)$$

$$u_i \geq 0 \quad \forall i \in I \quad (30)$$

$$o_i \geq 0 \quad \forall i \in I \quad (31)$$

$$s_i \geq 0 \quad \forall i \in I \quad (32)$$

$$LR_{\max} \geq 0 \quad (33)$$

$$LR_{\min} \geq 0 \quad (34)$$

In the first objective function, the first term is the cost of transferring goods from manufactories to DCs. The second term states the cost of opening and operating a distribution center. The next two terms are the cost of distributing goods to the clients from DCs. The last term is the penalty for violating the customers' time windows involving both waiting and delay penalty.

In second objective function, all terms consider the cost of fuel consumption and CO<sub>2</sub> emission, for fully loaded trucks, part-loaded trucks, which are leaving the firms, and the empty trucks coming back from DC  $j$  to plant  $k$ , respectively. Also, cost of fuel consumption and CO<sub>2</sub> emission based on both weight of vehicles and distance are measured in the second stage, for loaded vehicle from DC  $j$  to first customer  $i$ , loaded vehicles in optimal routes, which lose their weight by visiting each customer, and empty vehicles coming back from customer  $i$  to DC  $j$ , in order.

The last objective ensures the route lengths are balanced to divide the workload fairly on city drivers. It makes  $LR_{max}$  to take the shortest and  $LR_{min}$  to take the longest possible value.

Constraint (5) prevents exceeding the plants' capacity. Set of constraints (6) are related to the amount of all goods sent from each plant to each DC with the maximum capacity of the trucks. The limitation of different goods, which are sent to each DC is stated by constraint (7) due to the maximum capacity of each DC. The number of products received by each DC must be equal to the sum of the assigned customer demands (8). In the other word, keeping inventory is forbidden. Constraint (9) shows that each customer should assign to just one DC. Numbers of vehicles, which exit a DC must be equal to the number of vehicles returning to that DC (10). Constraint (11.1) ensures that each client must be visited immediately after exactly one DC or after another client, and constraint (11.2) imposes that exactly one client or DC must be visited immediately after. Also, they enable routes only between clients assigned to the same DC. Constraint (12) is a sub-tour elimination constraint and calculates  $u_i$ , which is needed in carbon footprint calculation objective function. Constraint (13) has been presented to prevent from loading more than vehicle capacity. Calculation of variable  $o_i$  takes place in Eq. (14). When customer  $i$  is the first in a route of DC  $j$ , constraint (13) forces  $o_i$  to be equal to  $r_{ji}$ . Constraint (16) remembers the vehicle fuel tank limitation translated to maximum length limit  $R$  that a vehicle could travel. Constraint (17) ascertains the longest route in the second stage While, constraint (18) ascertains the shortest route in the second stage. Constraints (19)-(24) are soft TW models. Starting service time at first customer is calculated by constraint (19). Starting service time at customers in a route is calculated by constraint (20). The amounts of violations from customer's time windows are counted in (21) and (23). At the end, the types of variables are mentioned in constraints (24)-(34).

#### 4. Solution approach

Two solution algorithms have been designed for this problem, one based on the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) introduced by Deb et al. (2002) and the other based on Multi-Objective Particle Swarm Optimization (MOPSO) algorithm (Rabbani et al., 2016).

##### 4.1. Non dominated sorting genetic algorithm

The evolutionary population-based metaheuristic, Non-Dominated Sorting Genetic Algorithm- II (NSGA-II) is the superior multi-objective version of Genetic Algorithm (GA) presented by Deb et al. (2002). The similarity of NSGA-II and GA is high; the only difference is in the sorting and selection method. In the single objective problem, the best possible answer is found but in the multi-objective problem, a set of non-dominated solutions called Pareto front are found. Due to domination concept,  $x$  dominates  $y$  if and only if in all objectives,  $x$  is not worse than  $y$  and at least in one of the objectives,  $x$  is certainly better than  $y$ .

This algorithm starts with the initialization and in the first step, the initial population is generated. Then, the fitness functions, which are the objective functions are calculated for each of the individuals. Based on the results and the concept of non-dominance, a rank will be assigned to each chromosome for sorting purpose. Rank one is better than rank two, rank two is better than rank three, and so on. But for sorting chromosomes with the same rank, which are in the same front, the other measure called crowding distance is used. Crowding distance declares an estimate of the solution's density surrounding a specific solution. The amount of this metrics is equal to the average distance between the two

neighboring solutions of a specific solution. This measure prefers the uniformly spread of solutions in the objective space and prioritizes solutions with lower crowding distance. Fig. 2 illustrates a schematic view of the NSGA-II sorting procedure.

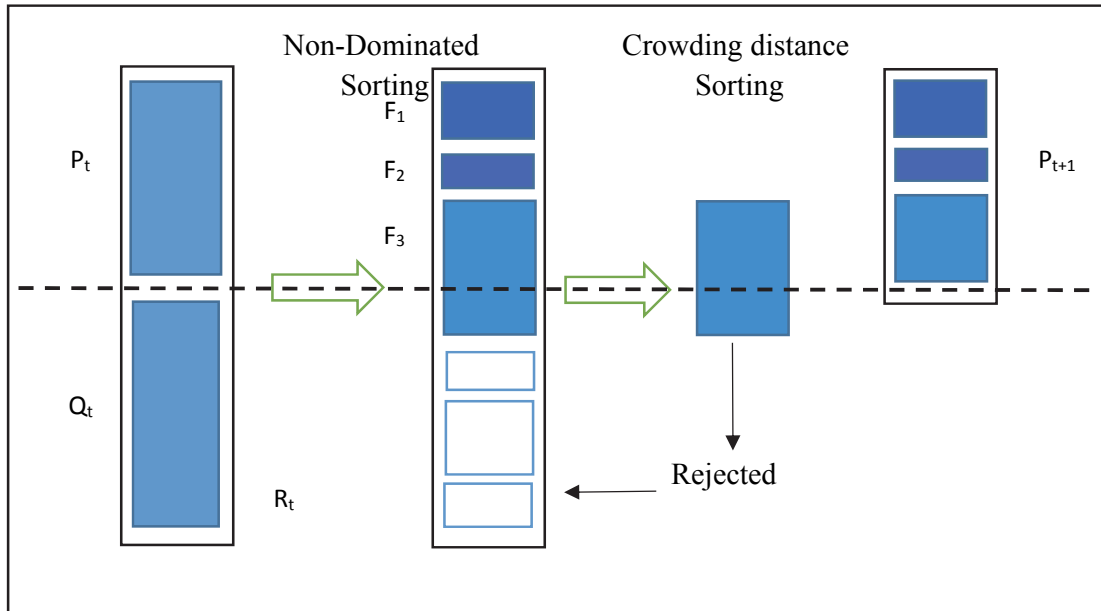


Fig. 2. NSGA-II sorting procedure

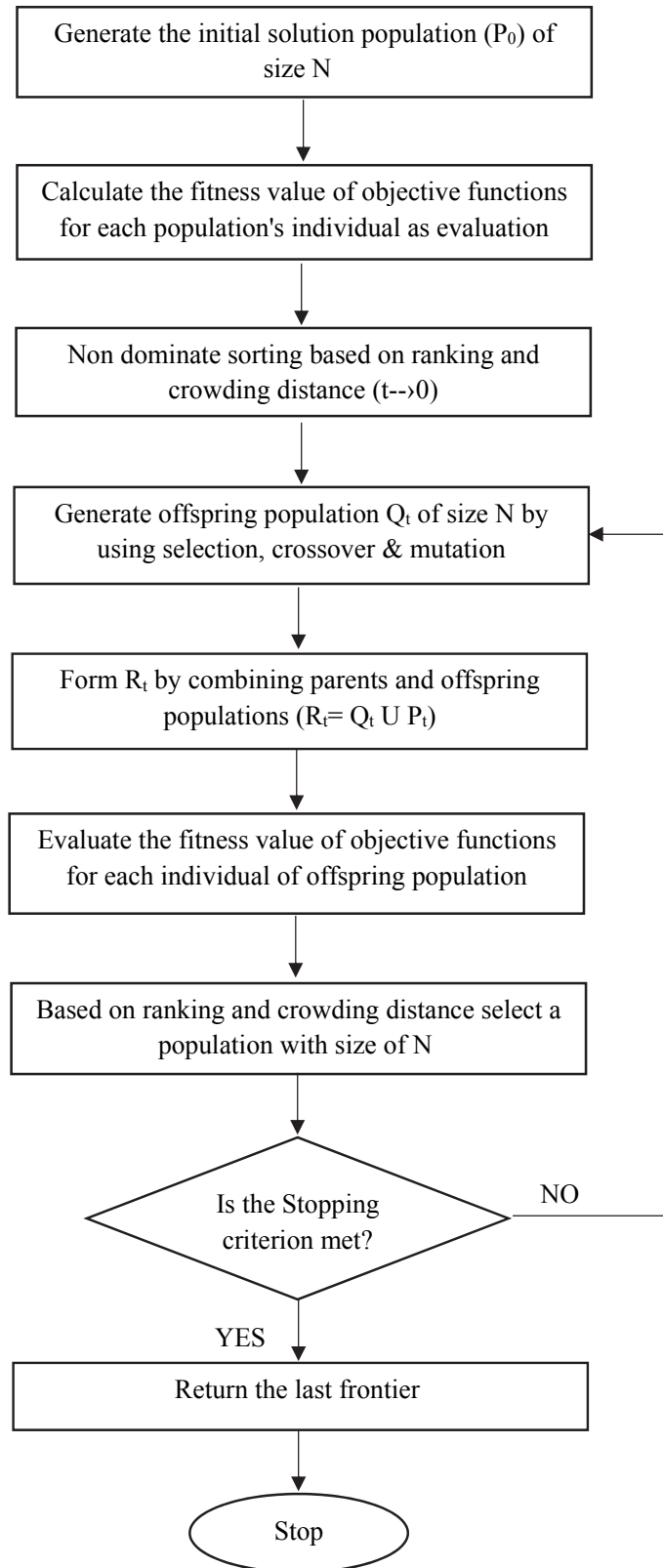
After sorting, best solutions will be selected as parents with the selection method. By means of crossover operator with crossover rate,  $P_c$  and mutation operator with mutation rate,  $P_m$ ; next population called offspring population  $Q_t$  is generated with the size of  $N$ .

The combination of this new population and the former one organizes the union population  $R_t$ . By calculating the fitness value of the offspring population plus the fitness value of the parent population, which form the whole union population  $R_t$  non-dominated sorting is applied with respect to its two criteria: first the domination and second the crowding distance.

In the next pace, the first  $N$  solutions from the sorted union population  $R_t$  will be selected as the best solutions to form the next iteration population  $P_{t+1}$ . These steps will be accomplished until reaching the stopping criterion. The steps of NSGA-II procedure are presented in Fig. 3.

#### 4.2. Multi objective particle swarm optimization

Particle Swarm Optimization (PSO), which is inspired by the movement of a flock of birds for finding food is an appropriate method for single objective optimizations. To overcome the insufficiency of Particle Swarm Optimization (PSO) algorithm in solving multi-objective problems, the other optimization technique for multiple objective problems is proposed named Multi-Objective Particle Swarm Optimization algorithm (MOPSO) (Reyes-sierra & Coello, 2006). MOPSO is the other evolutionary population based algorithm. In this algorithm, each particle uses its previous best memory ( $x_{pbest}$ ) and global best memory ( $x_{gbest}$ ) of the whole swarm to find the best movement for flight route (Rabbani et al., 2016). It means that each particle uses the knowledge of the intelligence group.



**Fig. 3.** Flowchart of NSGA-II procedure

The velocity of each particle  $p$  for  $i$  th dimension at iteration  $t$   $v_{pi}(t)$  is calculated using following equation.

$$v_{pi}(t) = w \times v_{pi}(t-1) + c_1 \times r_1 \times (x_{pbest}(t) - x_{pi}(t)) + c_2 \times r_2 \times (x_{gbest}(t) - x_{pi}(t)) \quad (35)$$

where,  $x_{pbest}$  is the best position of the particle, which has been founded so far,  $x_{gbest}$  is the best position of all particles have found it so far,  $w$  is the inertia weight that maintains the impact of the last velocity on the new velocity,  $C_1$  and  $C_2$  are cognitive and social learning coefficients, which are associated with the particle success and neighborhood success, in order and handle their influence on the new velocity, and  $r_1$  and  $r_2$  are random numbers in the interval of 0 and 1. The position of each particle  $p$  for  $i$  th dimension at iteration  $t$   $x_{pi}(t)$  is calculated using following equation.

$$x_{pi}(t) = x_{pi}(t-1) + v_{pi}(t) \quad (36)$$

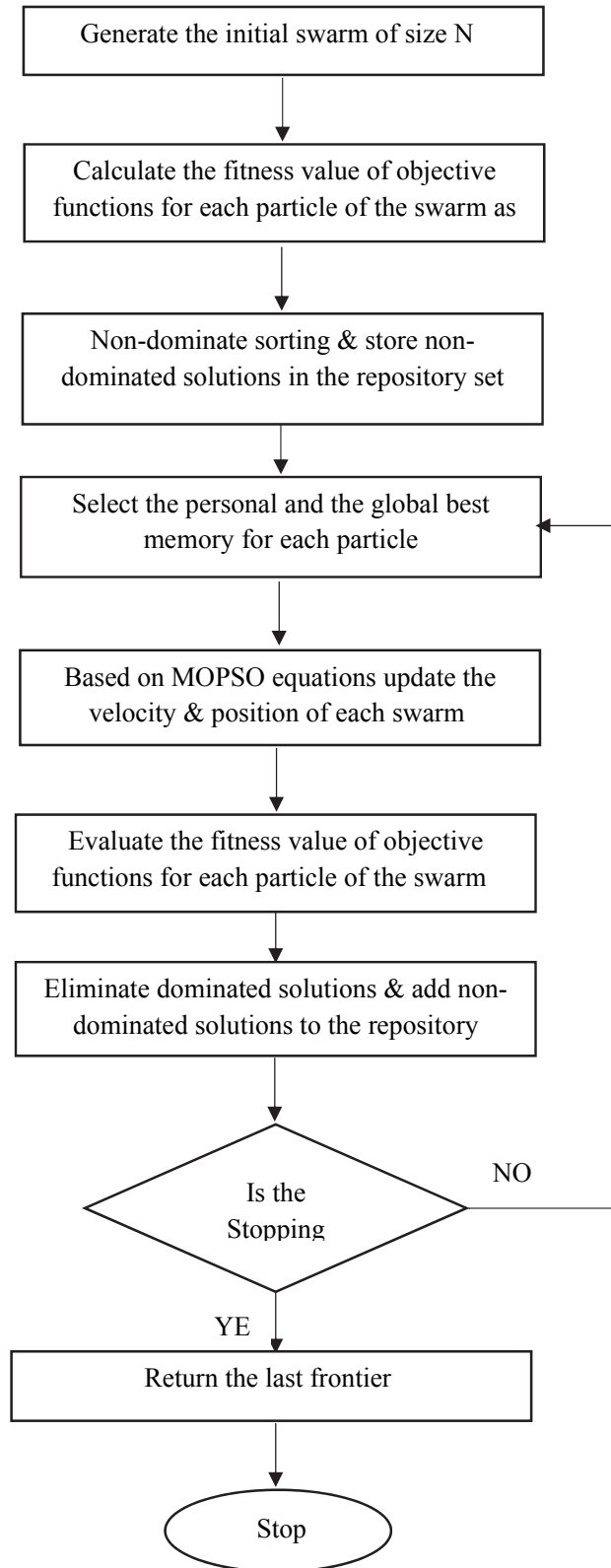
The procedure steps of MOPSO are presented in Fig. 4. In the first step, the initial swarm consist of  $N$  particles is generated. In the next step, the value of fitness functions for each particle will be calculated for determining the non-dominated solutions in the next pace and storing them in the repository set. Then the best position for each particle is selected as the particle best position, which for the particles of the first swarm is their current position. The best position of the non-dominated solutions as leaders of the swarm is selected as the global best position. According to gained information, the velocity vector of each particle and its position will be calculated by the means of Eq. (35) and Eq. (36), respectively for the next iteration. Evaluate the fitness value of objective functions for each particle of the swarm, add new non-dominated solutions to the repository of non-dominated solutions and eliminate dominated solutions to amend it. It is necessary to note that in selecting the best memory of position if the new position is dominated by its position in the memory, the position in the memory must be kept. Otherwise, the memory should be updated. These steps will be accomplished until reaching the stopping criterion.

### 4.3. Solution representation

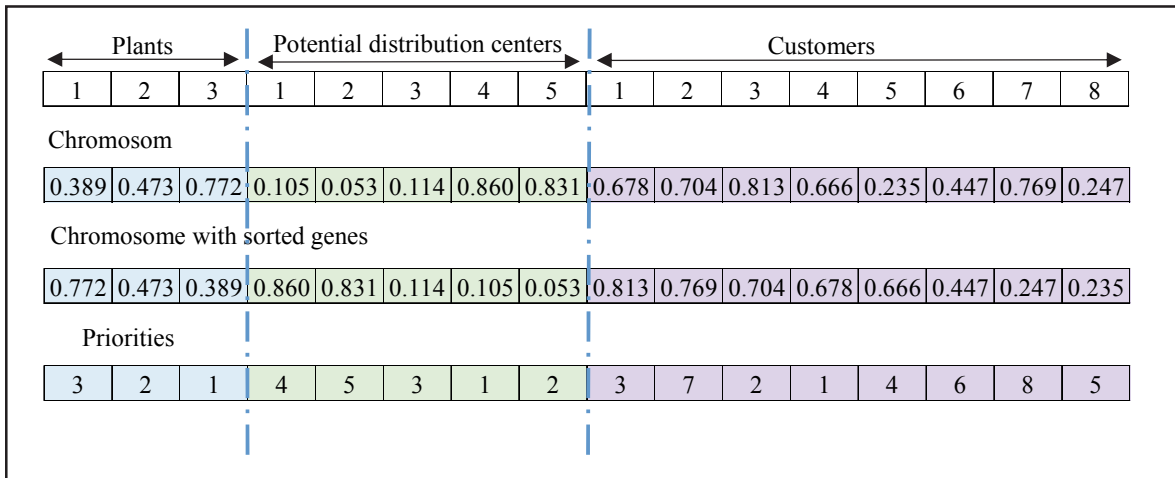
There are different techniques to implement the metaheuristic algorithm for solving a problem. In some techniques, a random solution is generated and then try to make it feasible by penalizing the violation of some constraints in the objective function. But in this paper, a better technique has been used. A feasible solution has been generated from the beginning and then we try to improve it in each iteration. Since both NSGA-II and MOPSO algorithm need the initial solution in a shape of chromosome, the customized chromosome is proposed in the next part.

#### 4.3.1. Customized chromosome

With respect to the proposed problem, the priority-based method is the most adapted method to encode the problem. In this study, the customized chromosome is a  $1 \times (P + M + N)$  string or vector, which is used for generating the solutions. Here  $P$  is the number of plants,  $M$  is the number of potential distribution centers, and  $N$  is the number of customers. It means that the genotype has three genes, the first gene is related to plants, the second gene is for potential DCs, and the third gene is for customers. This chromosome is filled by the real numbers between 0 and 1, randomly. Then, these real numbers are sorted in each gene of the genotype. These sorted numbers show the priority of customers in assigning to the prioritize distribution centers or the priority of distribution centers for linking to the prioritize plants. Changing these priorities by genetic algorithm operators will examine the different scenarios such as opening the different distribution centers to find the minimum cost with the most balanced route length. Fig. 5 demonstrates an instance chromosome for the problem with  $P=3$  plants,  $M=5$  potential distribution center, and  $N=8$  customers. Also, it shows the way of converting it to priorities.



**Fig. 4.** Flowchart of MOPSO procedure



**Fig. 5.** A chromosome instance for this study

To reach to a meaningful solution, which distinguishes decision variables from the proposed chromosome for the problem, a decoding policy is needed. In this study, the policy forms the solution from the last part of the distribution network, means customers. The decoding procedure includes two phase: routing phase and transportation phase. In the routing phase procedure, routes have been determined by assigning customers to them. After sorting the random real numbers of all genes and determining the priority of their components, the unassigned customer will be picked with the higher priority and assign it to the potential DC with the higher priority in the ordered list, if the unassigned capacity of that potential DC could satisfy the demands of the mentioned customer for all products; Otherwise this condition will be checked for the potential DC with the next priority.

In the later step, the next priority customer is picked, if adding this customer to the route does not lead to offend the capacity limitation and maximum possible traveling distance for each second stage's vehicle, this customer will be added to the opened route. Otherwise, if the capacity restriction of DCs or the vehicle limitations will trespass by adding next priority customer to the route; the opened route is closed and the new route will start to construct. It has been noticed that each vehicle should return to its dispatching DC and calculates it with each route length while checking maximum possible traveling distance of vehicles. Until assigning all the customers to DCs, the creation of new routes will be continued. If there is no customer allocated to a distribution center, it will not be opened. Now the total demand, which each distribution center satisfies is defined and the transportation phase is started. The opened distribution center with higher priority will be linked to the first priority plant in the ordered list. If its demand is under the capacity of that plant, that plant sends  $Nt_{kj}$  (=DC's demand/capacity of a truck) number of trucks to the mentioned DC; if not, the excessive demand of the mentioned DC from the capacity of plant will be satisfied from the next priority plant in the ordered list. This procedure will continue until supplying all DCs demands from the plants. Fig. 6 shows the example data for the decoding procedure of the Fig. 5's chromosome.

#### 4.3.2. NSGA-II operators

NSGA-II algorithm uses mutation and crossover operators to find high quality solutions and to produce the new solutions (offspring) from old solutions (parents).

- *Crossover operator*

There are different types of crossover operators, which are used for generating new chromosomes such as single-point, two-point, uniform, half uniform, and three parent crossovers. In this study, single point crossover is used according to its simplicity.



plant	1	2	3
capacity of 1	240	200	296
capacity of 2	180	160	168

DC	1	2	3	4	5
capacity of 1	252	364	324	312	298
capacity of 2	194	152	210	172	264

customer	1	2	3	4	5	6	7	8
demand of 1	59	106	76	73	88	54	72	52
demand of 2	48	32	42	28	50	98	28	84

Capacities and demands of example data

plant	1	2	3
priority	3	2	1

DC	1	2	3	4	5
priority	4	5	3	1	2

customer	1	2	3	4	5	6	7	8
priority	3	7	2	1	4	6	8	5

Obtained priorities based on sorted genes in figure (5)

Capacity of vehicles = 300

DC 4					
customer	4	3	1	5	
demand of 1	73	76	59	88	296
demand of 2	28	42	48	50	168
route	vehicle 1	vehicle 1	vehicle 2	vehicle 2	total demand

Second stage results for DC 4 as the first opened DC

DC 5					
customer	8	6	2	7	
demand of 1	52	54	106	72	284
demand of 2	84	98	32	28	242
route	vehicle 3	vehicle 3	vehicle 4	vehicle 4	total demand

Second stage results for DC 5 as the second opened DC

Capacity of trucks = 400

plant 3	
DC	4
supply 1	296
supply 2	168
number of trucks	2

plant 2	
DC	5
supply 1	200
supply 2	160
number of trucks	1

plant 1	
DC	5
supply 1	84
supply 2	42
number of trucks	1

First stage results for opened DCs 4 & 5

DCs allocation starts from high priority plants

Fig. 6. An example data for decoding procedure

In the single point crossover: first, a random number is generated that shows the position of crossover point in both parents. Then, the first children take the first part of its string from the first parent, which is the beginning part until crossover point of parents' string and it takes the second part of its string from the second parent, which is a copy from the crossover point to the end of parents' string. It should be noted that applying this crossover technique does not have any influence on the feasibility of the newborn child. Fig. 7 shows the operation of single point crossover.

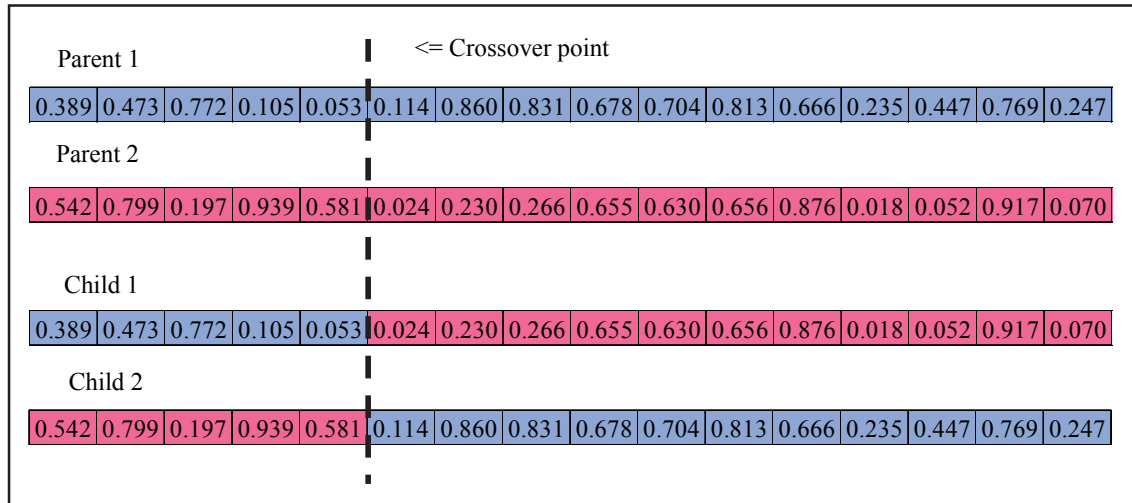


Fig. 7. Single point crossover for this study

- *Mutation operator*

There are several strategies to mutate a solution such as swap, insertion, scramble, and inversion mutation. In the proposed algorithm, the inversion mutation operator is used. In the inversion mutation operator, after selecting two mutation points randomly called  $M_1$  and  $M_2$ , the specified part of the string between these two points will be inverted backward. Fig. 8 depicts the operation of inversion mutation operator.

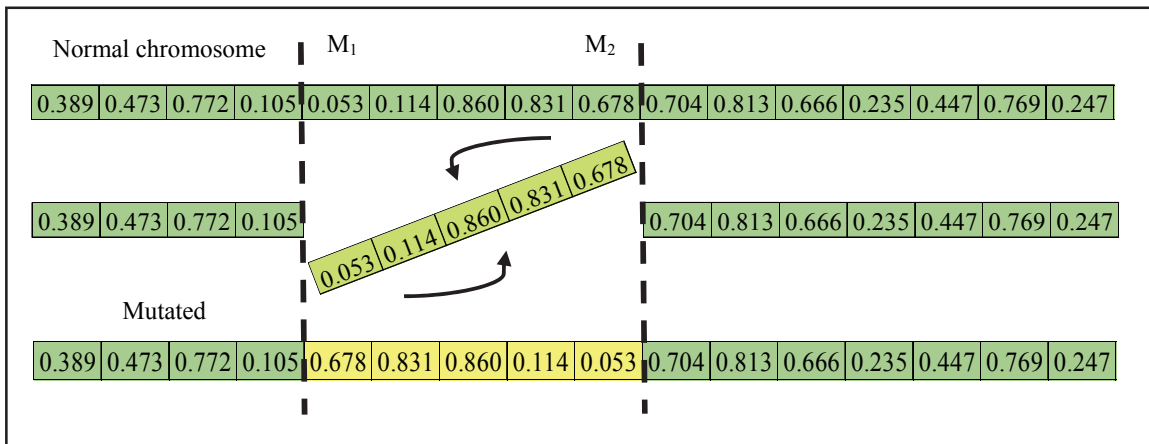


Fig. 8. Inversion mutation for this problem

- *Selection operator*

In this study the roulette wheel selection method is used.

### 4.3.3. Customized particle

Due to high similarity of implementing NSGA-II and MOPSO, the structure of each particle is the same as the structure of each chromosome. Because MOPSO works in a continuous search space real numbers between 0 and 1 are produced in each chromosome instead of producing orders in each gene. For example, integer numbers between 1 and  $P$  for first gene.

## 5. Computational results

With solving a small size instance by means of BARON solver as a powerful solver in solving Mixed Integer Non-Linear Problems (MINLP) by GAMS software, the model is validated. It's good to mention that because of the inability of GAMS in solving multi-objective problems, the global criterion method is used for transforming the multi-objective problem into a single objective problem. This method uses normalization of the objective functions and tries to minimize the differences of the optimal point's objective value from their ideal point's objective value (Meng et al., 2016). Eq. (37) shows this method's performance. By considering  $p$  equal to one, this method turns to one of the LP-metric methods.

$$f = \sum_{i=1}^3 \frac{f_i - f_i^{\min}}{f_i^{\max} - f_i^{\min}} \quad (37)$$

where  $f_i$  is the objective value,  $f_i^{\min}$  is the best objective value and  $f_i^{\max}$  is the worst objective value of objective function  $i$ .

The characteristics of the small instance (P0 problem) is presented in Table 2. The detailed parameters of P0 problem, which is used for validating the model with GAMS are available at <https://www.dropbox.com/s/m6sdiu7as3bt1p0/navazi.xlsx?dl=0>. The results of the software including the number of trucks, opened DCs and second stage routs are reported here in Table 3. Also, the value of three objective functions besides the normalized one are presented in Table 4.

**Table 1**

Features of P0 problem

name	P	M	N	G
P0	2	3	7	2

**Table 2**

Summary results of GAMS for P0 problem

Factory	NT	Distribution centers	Order of customers in routs
1	zero	1	-
		2	-
2	2»3		3»4»10»3
	One (including	3(opened)	3»8»5»3
	652 g1 & 339 g2)		3»9»7»6»3

**Table 3**

Objective values of GAMS for P0 problem

	value
$f_1$	2.400001E+9
$f_2$	4748.753
$f_3$	250
$f$	-4.26433E-7

### 5.1. Parameters tuning

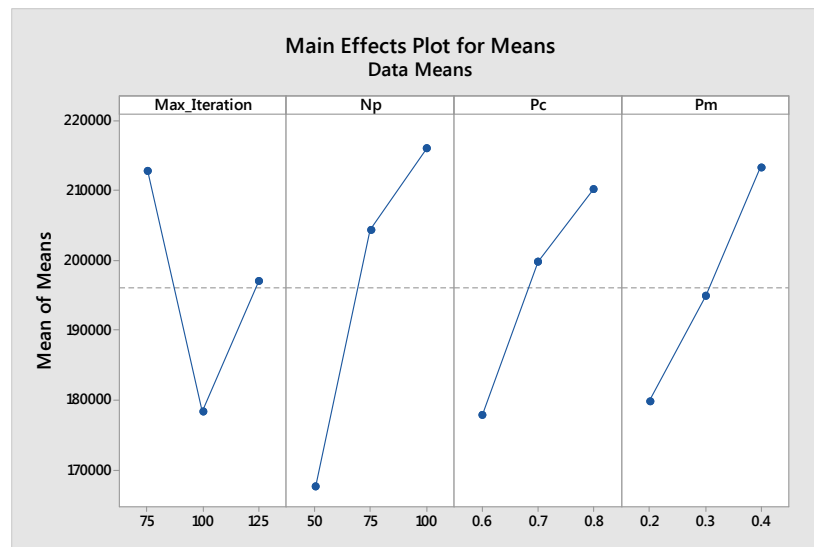
Because the performance of evolutionary algorithms is directly affected by their parameters, the calibration of these parameters become obvious and important. To improve the experimental results of this problem, the design of experiment method called Taguchi method as a powerful tool for parameter tuning is employed for both NSGA-II and MOPSO algorithms to set their parameters. One of the distinctive features of this method is achieving the largest information with generating the least number of experiments. A three-level Taguchi design is applied to analyze the influence of vital NSGA-II parameters involving, population size (Np), the total number of iterations (Max Iteration), crossover rate (Pc) as well as mutation rate (Pm). The examined assuming three levels for NSGA-II are brought in a Table 5. The basis of this investigation is assumed to be the solution quality. In other words, the best value of three objective functions is considered as a response for the Taguchi design.

**Table 4**

Levels of parameters for NSGA-II

parameter	NSGA-II		
	low	mean	High
N population	50	75	100
P crossover	0.6	0.7	0.8
P mutation	0.2	0.3	0.4
max iteration	75	100	125

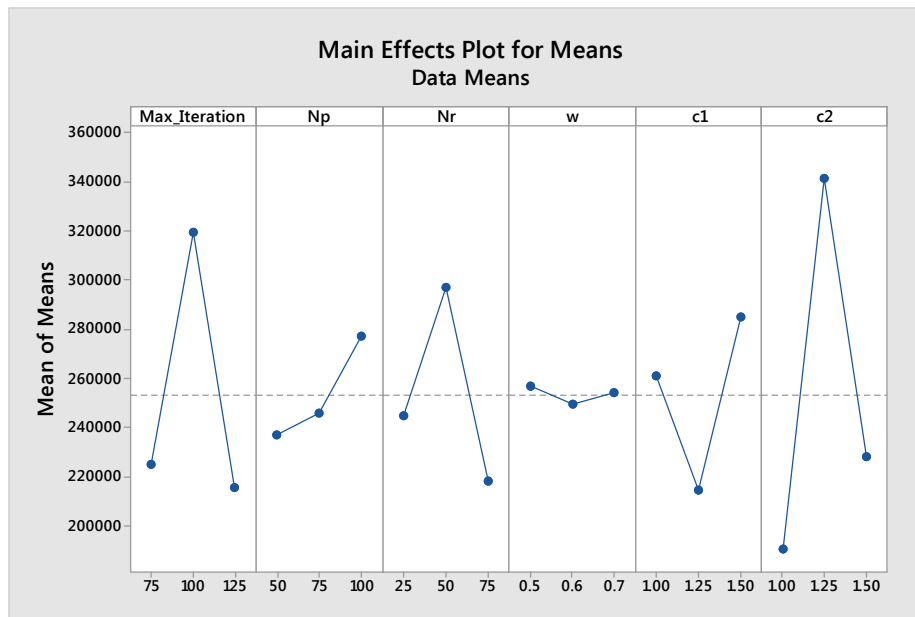
The results obtained through carrying out the Taguchi design for the first problem instance by means of MINITAB software 17 are pictured in Fig. 9.

**Fig. 9.** Analysis diagrams of NSGA-II parameters tuning by Taguchi DOE

The same design of experiment for the MOPSO algorithm is repeated to determine the optimal levels of swarm size (Np), the total number of iterations (Max Iteration), repository size (Nr), inertia weight (w), personal learning coefficient (c1), and global learning coefficient (c2). The examined assuming three levels for MOPSO are brought in Table 6.

**Table 5**  
Levels of parameters for MOPSO

parameter	MOPSO		
	low	mean	high
N population of swarm	50	75	100
N repository	25	50	75
weight of inertia	0.5	0.6	0.7
c <sub>1</sub>	1	1.25	1.5
personal learning	1	1.25	1.5
c <sub>2</sub>	1	1.25	1.5
global learning	1	1.25	1.5
max iteration	75	100	125



**Fig. 10.** Analysis diagrams of MOPSO parameters tuning by Taguchi DOE

Finally, the desirable value of each parameter, which has the minimum mean for both algorithms is reported in the followings:

#### *NSGA-II assumptions*

- Number of population size is 50,
- Crossover coefficient is set 0.6,
- Mutation coefficient is set 0.2,
- The algorithm is terminated after 100 iterations.

#### *MOPSO assumptions*

- swarm size is 50,

- repository size is 75,
- inertia weight is 0.6,
- personal learning coefficient is 1.25,
- global learning coefficient is 1,
- The algorithm is terminated after 125 iterations.

The both NSGA-II and MOPSO algorithms have been coded in MATLAB R2016b software and executed on an Intel Corei7 laptop with 12 GB of RAM and 2.6 GHz CPU.

### 5.2. Test problems

To see the performance of the metaheuristic algorithms, the test problems have been generated; because to the best of our knowledge, there are no available benchmark examples that suits this model. For this purpose, three faction of problem instances, containing small, medium, and large size problems have been generated using the instance generator code in MATLAB software. The most important characteristics of these instances like the number of plants ( $P$ ), the number of potential distribution centers ( $M$ ), the number of customers ( $N$ ) and the number of goods type ( $G$ ) are presented in the following Table. Their detailed information is available to download by researchers at

<https://www.dropbox.com/s/m6sdiu7as3btlp0/navazi.xlsx?dl=0>.

**Table 6**  
Important features of test problems

number	name	$P$	$M$	$N$	$G$	number	name	$P$	$M$	$N$	$G$
1	s1	2	3	6	2	10	m4	5	9	33	3
2	s2	2	4	8	2	11	m5	6	10	36	3
3	s3	3	4	9	2	12	m6	6	12	40	4
4	s4	3	5	10	2	13	11	8	15	50	4
5	s5	3	5	12	2	14	12	10	20	60	4
6	s6	3	5	15	3	15	13	12	22	70	4
7	m1	4	6	20	3	16	14	15	25	80	4
8	m2	4	7	25	3	17	15	18	27	90	4
9	m3	5	8	30	3	18	16	20	30	100	5

### 5.3. Comparison metrics

For evaluating the performance of customized metaheuristic algorithms for this problem and comparison them together, four comparative metrics are implemented.

(1). The number of Pareto front solutions (NPS): The number of non-dominated points that each algorithm could obtain. The more number of points shows the more ability of each algorithm to find efficient points, so it's a desirable factor.

(2). Computational time (CT): the time that each algorithm spends to find an efficient frontier is saved. The less time shows the better performance so it's an undesirable factor.

(3). Spacing metrics (SM): it means that how the Pareto points in the approximation set are distributed in the objective space. The lower value of SM shows the more uniform distribution of Pareto points. It calculates as follows:

$$SM = \sqrt{\frac{1}{n-1} \times \sum_{i=1}^n (d_i - \bar{d})^2} \quad (38)$$

$$\text{where } d_i = \min_j \left( \left| f_1^i(\vec{x}) - f_1^j(\vec{x}) \right| + \left| f_1^i(\vec{x}) - f_1^j(\vec{x}) \right| \right) \quad i, j = 1, 2, \dots, n \quad (39)$$

and  $\bar{d}$  is the average value of  $d_i$  s.

(4). Diversity metrics (DM): It distinguishes the spread of solution sets and It calculates as follows:

$$DM = \sqrt{\sum_{i=1}^n \max(\|x_t^i - y_t^i\|)} \tag{40}$$

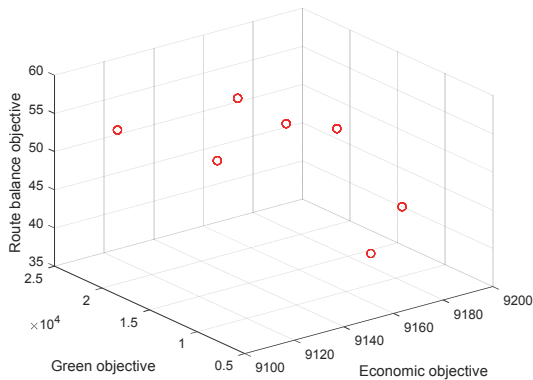
where  $\max(\|x_t^i - y_t^i\|)$  is the Euclidean distance between the non-dominated solutions  $x_t^i$  and  $y_t^i$ .

5.4. Results of the comparison

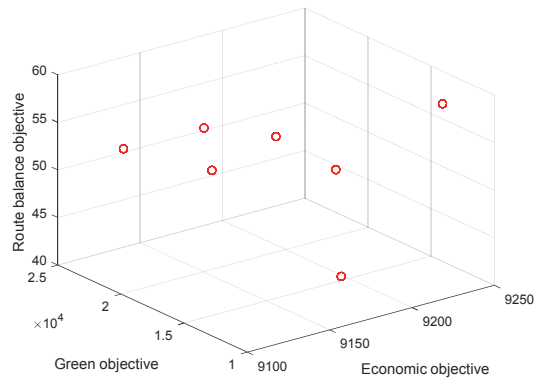
Table 8 shows the comparison metrics for NSGA-II and MOPSO algorithms for small instances. As can be seen, the average number of Pareto solutions for NSGA-II is more than MOPSO, even though the computational CPU time for NSGA-II is a bit lower than the computational CPU time for MOPSO. But the spacing for MOPSO is more uniform than NSGA-II. Also, the diversity of Pareto solutions in objective space for MOPSO is better than NSGA-II. So, it depends on the manager's preferences to choose between quicker answers or uniform answers with wide diversity. The Pareto front solutions of NSGA-II and MOPSO for the sixth problem instance (s6) are shown in Fig. 11 and Fig. 12, in order.

**Table 7**  
Comparison metrics for small instances for NSGA-II & MOPSO

Small	Method							
	NSGA-II				MOPSO			
	NPS	CT	SM	DM	NPS	CT	SM	DM
s1	5	5.9282	638.8626	5770.4276	5	5.8239	674.63	5770.3915
s2	9	5.9405	1023.8379	6300.9816	7	5.4063	1112.2263	6300.9816
s3	6	6.2517	159.1845	3071.6893	4	6.4903	912.3577	3071.6893
s4	8	6.0027	277.3069	8707.0052	7	6.0952	181.4099	8047.9108
s5	6	6.1402	4198.1564	11148.5094	6	6.4034	1530.2649	14264.0646
s6	7	6.2443	966.7764	12258.2275	7	6.439	481.9637	10658.6701
Avg.	6.8333	6.0846	1210.68745	7876.1401	6	6.10968333	815.475417	8018.95132



**Fig. 11.** NSGA-II Pareto front for problem instance s6



**Fig. 12.** MOPSO Pareto front for problem instance s6

Table 9 shows the comparison metrics for NSGA-II and MOPSO algorithms for medium instances. As can be seen, it can be concluded that despite the more computational time of MOPSO, which is no notable difference, it has the better diversification and spacing metrics with a little more number of Pareto solutions than NSGA-II for medium size instances. Nearly, it has been found that the MOPSO has the better performance for medium instances.

**Table 8**

Comparison metrics for Medium instances for NSGA-II &amp; MOPSO

medium	Method							
	NSGA-II				MOPSO			
	NPS	CT	SM	DM	NPS	CT	SM	DM
m1	5	5.5088	1077.0054	25189.225	5	6.7599	5128.784	40133.4369
m2	6	6.5717	8931.4188	23652.351	7	7.1182	3851.321	23652.3519
m3	6	6.884	5111.3649	45818.794	6	7.6146	10852.62	41412.5234
m4	5	6.8389	15275.586	34302.638	7	7.6573	8392809	33930.496
m5	8	7.3257	3945.4386	32206.112	6	7.4938	2840.601	29669.8577
m6	5	7.1445	18668.69	41055.659	6	7.7914	2503.385	36344.0403
Avg.	5.83333333	6.71226667	8834.91744	33704.13043	6.16666667	7.40586667	5594.92193	34190.451

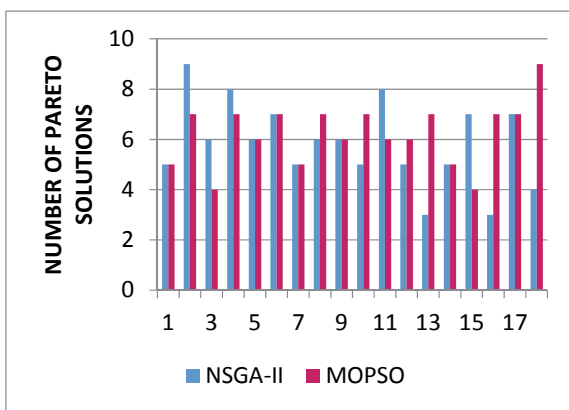
Table 10 shows the comparison metrics for NSGA-II and MOPSO algorithms for large instances. As can be seen, except lower computational time of the NSGA-II algorithm, it has no more advantages in comparison metrics. The number of Pareto solutions, diversity, and spacing assess better for MOPSO. Despite the fact that MOPSO takes more computational time, we might say MOPSO has the better operation for large instances.

**Table 9**

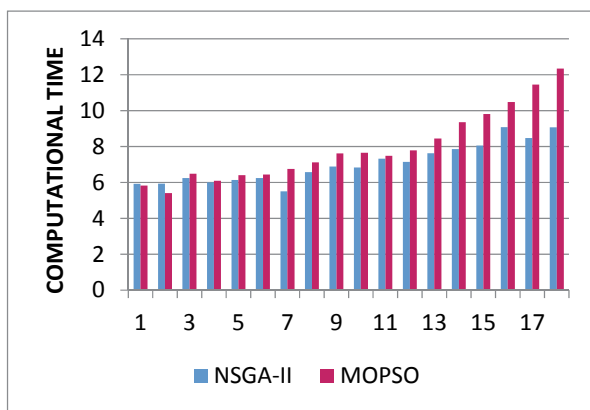
Comparison metrics for large instances for NSGA-II &amp; MOPSO

Large	Method							
	NSGA-II				MOPSO			
	NPS	CT	SM	DM	NPS	CT	SM	DM
l1	3	7.635	16761.2964	5100.3104	7	8.447	3473.042	30807.372
l2	5	7.8632	16290.1387	5900.2227	5	9.3587	12077.14	44306.529
l3	7	8.0661	19820.7727	7680.0933	4	9.8185	7063.925	93627.431
l4	3	9.0903	13647.1041	1602.5598	7	10.487	3201.887	49378.469
l5	7	8.473	2048.6095	3800.5679	7	11.4542	18415.20	102157.73
l6	4	9.0676	25140.3419	118740.26	9	12.3448	13467.69	168089.75
Avg.	4.83333333	8.36586667	15618.0439	23804.0024	6.5	10.3183667	9616.48328	81394.5495

Also, Figs. (13-16) compare the both solution approaches for all instances based on four comparison metrics, graphically. In an average of all sizes, although MOPSO needed more Computational Time (CT) but it diversified solutions in objective space more with more uniformity because it has less Spacing Metric (SM) and more Diversity Metric (DM). Hence, the number of Pareto solutions for MOPSO is a little bit more than NSGA-II.

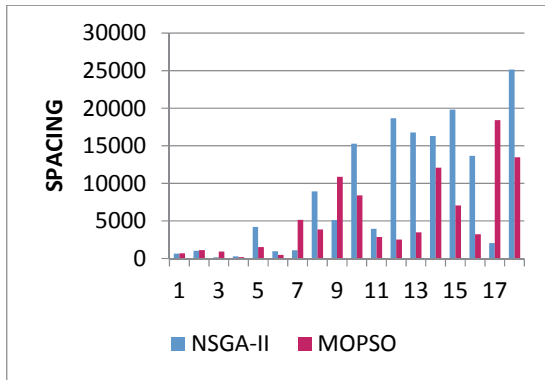


**Fig. 23.** Comparison of NSGA-II and MOPSO in terms of number of Pareto solutions

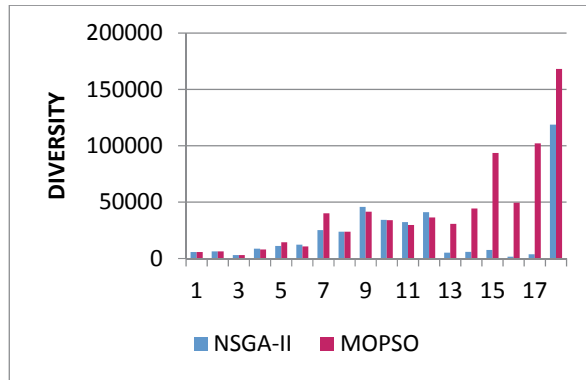


**Fig. 34.** Comparison of NSGA-II and MOPSO in terms of computational time





**Fig. 45.** Comparison of NSGA-II and MOPSO in terms of spacing metric



**Fig. 56.** Comparison of NSGA-II and MOPSO in terms of diversity metric

## 6. Conclusion

In this paper, a transportation stage was considered before the Location Routing Problem, so a Transportation Location Routing Problem (TLRP) as an extension of the two-echelon Location Routing Problem was solved. To be more realistic, customers' time windows were considered in the second stage. Distribution operational costs, fuel consumption costs, and CO<sub>2</sub> emission costs were minimized also the workloads for city drivers were balanced to design a sustainable distribution network. To tackle these conflicting objectives, NSGA-II and MOPSO were applied for solving the problem. Based on the obtained results, MOPSO algorithm has the better performance in the Number of Pareto front Solutions (NPS), Spacing Metric (SM), and diversification of solutions in the space (DM) than the NSGA-II algorithm, in general. Generally, NSGA-II just has the shorter computational CPU time (CT) than MOPSO. It is clear that considering three objectives had decreased the number of Pareto solutions but those few solutions could maintain the interests of different stakeholders in the distribution network. In this study, equal weights were given to three objectives, but distribution managers by weighting objectives could give higher priority to one or two objectives due to the policy of company in different situations. Researchers could consider a different kind of VRP in the second echelon such as open VRP or VRP with pickup and delivery. Considering heterogeneous vehicles and trucks will be appreciated to be closer to real world problems. The other future direction could be considering the uncertainty, which is correlated with traveling time between two nodes due to the traffic, accidents, etc.

## References

- Azadeh, A., & Farrokhi-Asl, H. (2017). The close–open mixed multi depot vehicle routing problem considering internal and external fleet of vehicles. *Transportation Letters*, 1–15.
- Bektas, T., & Laporte, G. (2011). The Pollution-Routing Problem. *Transportation Research Part B*, 45, 1232–1250.
- Cuda, R., Guastaroba, G., & Speranza, M. G. (2015). A survey on two-echelon routing problems. *Computers and Operation Research*, 55, 185–199.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.
- Demir, E., Bektas, T., & Laporte, G. (2014). The bi-objective Pollution-Routing Problem. *European Journal of Operational Research*, 232(3), 464–478.
- Erdoğan, S., & Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1), 100–114.
- Eskandarpour, M., Dejax, P., Miemczyk, J., & Péton, O. (2015). Sustainable supply chain network design: An optimization-oriented review. *Omega*, 54, 11–32.
- Farrokhi-Asl, H., Tavakkoli-Moghaddam, R., Asgarian, B., & Sangari, E. (2017). Metaheuristics for a bi-objective location-routing-problem in waste collection management. *Journal of Industrial and Production Engineering*, 34(4), 239–252.
- Faulin, J., Juan, A., Lera, F., & Grasman, S. (2011). Solving the Capacitated Vehicle Routing Problem with Environmental Criteria Based on Real Estimations in Road Transportation : A Case Study. *Procedia - Social and Behavioral Sciences*, 20, 323–334.
- Fazel Zarandi, M. H., Hemmati, A., Davari, S., & Burhan Turksen, I. (2013). Capacitated location-routing problem with time windows under uncertainty. *Knowledge-Based Systems*, 37, 480–489.

- Forkenbrock, D. J. (2001). Comparison of external costs of rail and truck freight transportation. *Transportation Research Part A: Policy and Practice*, 35(4), 321–337.
- Govindan, K., Jafarian, A., Khodaverdi, R., & Devika, K. (2014). Two-echelon multiple-vehicle location-routing problem with time windows for optimization of sustainable supply chain network of perishable food. *International Journal of Production Economics*, 152.
- Hossein, M., Zarandi, F., Hemmati, A., Davari, S., & Turksen, I. B. (2013). Capacitated location-routing problem with time windows under uncertainty. *Knowledge-Based Systems*, 37, 480–489.
- Iqbal, S., Kaykobad, M., & Rahman, M. S. (2015). Solving the multi-objective vehicle routing problem with soft time windows with the help of bees. *Swarm and Evolutionary Computation*, 24, 50–64.
- Kara, I., Kara, B. Y., & Yetis, M. K. (2007). Energy minimizing vehicle routing problem. *Combinatorial Optimization and Application, Springer*, 1, 62–71.
- Khalili-Damghani, K., Abtahi, A.-R., & Ghasemi, A. (2015). A new bi-objective location-routing problem for distribution of perishable products: Evolutionary computation approach. *Journal of Mathematical Modelling and Algorithms in Operations Research*, 14(3).
- Kuo, Y. (2010). Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. *Computers & Industrial Engineering*, 59(1), 157–165.
- Lin, C., Choy, K. L., Ho, G. T. S., Chung, S. H., & Lam, H. Y. (2014). Survey of green vehicle routing problem : Past and future trends. *Expert Systems with Applications*, 41, 1118–1138.
- Maden, W., Eglese, R., & Black, D. (2009). Vehicle routing and scheduling with time-varying data : A case study. *Journal of the Operational Research Society*, 61(3), 515–522.
- Maranzana, F. E. (1964). On the location of supply points to minimize transport costs. *Operations Research*, 261–270.
- Martínez-Salazar, I. A., Molina, J., Ángel-Bello, F., Gómez, T., & Caballero, R. (2014). Solving a bi-objective transportation location routing problem by metaheuristic algorithms. *European Journal of Operational Research*, 234(1), 25–36.
- Meng, Z., & Pan, J.-S. (2016). Monkey King Evolution: A new memetic evolutionary algorithm and its application in vehicle fuel consumption optimization. *Knowledge-Based Systems*, 97, 144–157.
- Moshref-javadi, M., & Lee, S. (2016). The latency location-routing problem. *European Journal of Operational Research*, 255(2), 604–619.
- Nagy, G., & Salhi, S. (2007). Location-routing : Issues , models and methods. *European Journal of Operational Research*, 177(2), 649–672.
- Nasherahkami, N., Bashiri, M., & Bagherinejad, J. (2015, December). Modeling of periodic location routing problem with time window and satisfaction dependent demands. In *Industrial Engineering and Engineering Management (IEEM), 2015 IEEE International Conference on* (pp. 1457–1461). IEEE.
- Price, R., Thornton, S., & Nelson, S. (2007). The social cost of carbon and the shadow price of carbon: what they are, and how to use them in economic appraisal in the UK.
- Prodhon, C., & Prins, C. (2014). A survey of recent research on location-routing problems. *European Journal of Operational Research*, 238(1), 1–17.
- Rabbani, M., Farrokhi-Asl, H., & Asgarian, B. (2017). Solving a bi-objective location routing problem by a NSGA-II combined with clustering approach: application in waste collection problem. *Journal of Industrial Engineering International*, 13(1), 13–27.
- Rabbani, M., Mousavi, Z., & Farrokhi-Asl, H. (2016). Multi-objective metaheuristics for solving a type II robotic mixed-model assembly line balancing problem. *Journal of Industrial and Production Engineering*, 33(7), 472–484.
- Rahmani, Y., Ramdane Cherif-Khettaf, W., & Oulamara, A. (2016). The two-echelon multi-products location-routing problem with pickup and delivery: Formulation and heuristic approaches. *International Journal of Production Research*, 54(4), 999–1019.
- Ray, S., Soeanu, A., Berger, J., & Debbabi, M. (2014). The multi-depot split-delivery vehicle routing problem : Model and solution algorithm. *Knowledge-Based Systems*, 71, 238–265.
- Reyes-sierra, M., & Coello, C. A. C. (2006). Multi-objective particle swarm optimizers : A survey of the state-of-the-art. *International Journal of Computational Intelligence Research*, 2(3), 287–308.
- Xiao, Y., Zhao, Q., Kaku, I., & Xu, Y. (2012). Development of a fuel consumption optimization model for the capacitated vehicle routing problem. *Computers and Operations Research*, 39(7).
- Zhalechian, M., Tavakkoli-moghaddam, R., Zahiri, B., & Mohammadi, M. (2016). Sustainable design of a closed-loop location-routing-inventory supply chain network under mixed uncertainty. *Transportation Research Part E*, 89, 182–214.
- Zhang, Y., & Chen, X. D. (2014). An Optimization Model for the Vehicle Routing Problem in Multi-product Frozen Food Delivery. *Journal of Applied Research and Technology*, 12(2), 239–250.

