A New Multi-Objective Inventory-Routing Problem by an Imperialist Competitive Algorithm

Rahmat Arab[1], Seyed Farid Ghaderi[2], Reza Tavakkoli-Moghaddam[3,*]

Received: 24. 11. 2017 Accepted: 25. 09. 2018

Abstract

One of the most important points in a supply chain is customer-driven modeling, which reduces the bullwhip effect in the supply chain, as well as the costs of investment on the inventory and efficient transshipment of the products. Their homogeneity is reflected in the inventory-routing problem, which is a combination of distribution and inventory management. This paper considers a multi objective IRP in a two-level supply chain consisting of a distributor and a set of retailer. This problem is modeled with the aim of minimizing bi-objectives, namely the total system cost and risk-based transportation cost. Products are delivered to customers by some heterogeneous vehicles with specific capacities through a direct delivery strategy. Additionally, storage capacities are limited and the shortage is assumed to be impermissible. To validate this model, the epsilon constraint method is used for solving the model. Since problems without distribution planning are very complex to solve optimally, the problem considered in this paper also belongs to a class of NP-hard ones. Therefore, a multi-objective imperialist competitive algorithm (MOICA) as a well-known multi-objective evolutionary algorithm is used and developed to solve a number of test problems. Furthermore, the computational results are compared to show the performance of the proposed MOICA.

Keywords: Inventory-routing problem; Multi-objective optimization; Epsilon constraint method; Meta-heuristics algorithm.

[*] Corresponding author. E-mail: tavakoli@ut.ac.ir
[1] PhD Student, School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran.
[2] Professor, School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran.
[3] Professor, School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran.

1. Introduction

In the recent years, the development of the chains and the created competition among them, and also the development of the information management and the greater awareness of the companies of their chain performance, have led to a severe consideration for the coordination, and integration of the various elements of the supply chain in order to achieve the competitive advantage.

An inventory-routing problem (IRP) is derived from a vehicle routing problem (VRP), in which inventory control and routing decisions are merged. The IRP is mostly used in vendor management inventory (VMI) systems, in which a vendor is responsible for controlling the timing and size of deliveries to customers. In return for this benefit, the vendor ensures that customers are not faced with shortages. Due to the timing of placed orders in the conventional vendor-customer relationship, in which customers ordered products from vendors, efficiency could decrease drastically and as a result, inventory and distribution costs would increase significantly. Nevertheless, cost reductions resulting from the implementation of VMI systems are not achieved easily in practice. Especially since the increasing number and variety of customers are only adding to this complexity. However, the IRP can make this achievement feasible through an optimal distribution plan, which minimizes total system costs.

We limit our literature review to investigations addressing one or many of the following issues: period type, product type, fleet type, and objective function type that mostly solved by meta-heuristic algorithms.

Having analyzed the industrial aspects of the problem, an inclusive classification and review of previous studies have been presented direct shipment is one of many distribution strategies used in the IRP problem, in which each vehicle only delivers products to retailers once during each period. For example, Barnes and Bassok [Barnes and Bassok, 1997] studied the efficiency of the direct shipment strategy in an IRP on an unlimited time horizon, in which possible retailer demand is considered through a distribution function and shortage is allowed

Many authors have used their meta-heuristics to solve their problem. In the following, we will briefly mention them.

Zhao et al. [Zhao et al., 2008] proposed a partition approach to an IRP and applied a tabu search algorithm (TS) to find the retailers' optimal partition regions. Also, hunag and Lin [Huang and Lin, 2010] used a modified ant colony optimization (ACO) algorithm for an IRP with demand uncertainty. Their feature is multi-product, which uses the adaptive ACO to solve it. In the following Moin et al. [Moin et al., 2011] presented an efficient hybrid genetic algorithm (HGA) for the multi-product multi-period IRP. They applied this algorithm and demonstrated the effectiveness of algorithm with numerical examples.

Other meta-heuristic methods, such as variable neighborhood search (VNS) was also used for the IRP in a supply chain and compared with other existing methods [Liu and Chen, 2012]. Also, Popovic et al. [Popovic et al., 2012] applied the VNS algorithm for the IRP in fuel delivery and developed the stochastic VNS algorithm to solve their MILP model. Other authors like Nekooghadirli [Nekooghadirli et al, 2014] proposed a new bi-objective location-routing-inventory problem in a distribution network by meta-heuristics. They used four multi-objective algorithms; namely multi-objective imperialist competitive algorithm (MOICA), multi-objective parallel simulated annealing (MOPSA), non-dominated sorting genetic algorithm (NSGA-II) and Pareto archived evolution strategy (PAES). The results showed that the MOICA has the best performance.

In recent years, some authors presented a multi-objective IRP model that balances the transportation cost, holding cost and lost sale. They used simulated annealing (SA) and TS to solve the model [Mirzaei and Seifi, 2015]. Also Ghorbani and Akbari Jokar [Ghorbani and Akbari Jokar, 2016] presented a hybrid imperialist competitive-simulated annealing (ICA-SA) algorithm for a multi-source multi-product location-routing-inventory problem. The results showed that the ICA-SA algorithm from the view of solving time was better than the available algorithm.

Tavakkoli-Moghaddam et al. [Tavakkoli-Moghaddam et al., 2016] proposed a fuzzy method to solve a bi-objective multi-product VRP with heterogeneous fleets. The proposed fuzzy approach was used to solve the bi-objective mixed-integer linear problem to find the most preferred solution. Also, they used a Pareto-optimal solution with the ε-constraint method, in order to show the conflict between two objectives. Also, in order to show the conflict between two objective functions in an excellent fashion, they used the Pareto-optimal solution with the ε-constraint method to demonstrate the efficiency and validity of the presented model. Also, Azadeh et al. [Azadeh et al., 2017] used a GA-Taguchi based approach for an IRP of a single perishable product with transshipment. They assumed that product deteriorates at the exponential rate during the time at the warehouse.

Rayat et al. [Rayat et al., 2017] proposed a bi-objective reliable location-inventory-routing problem with partial backordering under disruption risks. Finally, they used an archived multi-objective simulated annealing (AMOSA) and NSGA-II to solve their model. Also, Alinaghian and Shokouhi [Alinaghian and Shokouhi, 2017] presented a multi-depot multi-compartment VRP. Because their model was NP-hard, they used the hybrid ALNS to solve the model. Also, the results demonstrated the good performance of the proposed algorithm. Some authors used probabilistic methods due to the uncertain nature of the model parameters. For example, Rahimi et al. presented the stochastic model to the IRP and considered profit, service level and green criteria. Because the demand and transportation costs were considered uncertain, they used the fuzzy method to solve the model [Rahimi et al., 2017].

Nikhah et al. [nikkhah, Hoseini Motlagh and joker, 2017] presented a Two-Phase Hybrid Heuristic Method for a Multi-Depot Inventory-Routing Problem. The objective function of their problem was to minimize sum of the holding cost at distributer centers and the customers, and of the transportation costs associated to the preformed routes. In the proposed hybrid heuristic method, after a Construction phase (first phase) a modified VNS, with distinct neighborhood structures, was used during the improvement phase (second phase). Moreover, they used SA concept to avoid that the solution remains in a local optimum for a given number of iterations.

Also, Gatreh Samani and Hoseini Motlagh [Gatreh Samani and Hoseini Motlagh, 2017] proposed a hybrid algorithm for a two echelon location-routing problem with Simultaneous pickup and delivery under fuzzy demand. They use a mixed integer linear programming model for a two-echelon location-routing problem with simultaneous pickup and delivery. Also, a combined heuristic method based on simulated annealing (SA) algorithm and genetic algorithm (GA) is devised for solving the presented model.

Ghannadpour [Ghannadpour, 2018] compiled an evolutionary approach for energy minimizing vehicle routing problem with time windows and customers' priority. He tried to maximize the customers' satisfaction using their preference and considers the customers' priority for servicing. Also, in this paper, a multi-objective evolutionary algorithm was proposed and its performance on several completely random instances is compared with NSGA II and CPLEX Solver.

To summarize this paper, in Table 1, the methodology of the work of other authors is described. Also, this table illustrates the classification of the previously presented papers in the IRP literature.

This paper examines a multi-product, multi-period IRP with the aim of minimizing the total system costs and transportation risks. System costs include implementation and start-up, distribution, inventory storage and maintenance costs. The IRP is studied in a supply chain consisting of two parts, a supplier, and a set of retailers (i.e., customers), in which a certain amount of each product is distributed between retailers using direct delivery strategy and some heterogeneous fleet of vehicles with limited capacities. Providing service to each retailer takes one period; meaning that each retailer is visited by a vehicle utmost once during each period.

The mentioned problem also includes the recursive mode. In other words, vehicles travel a trip during each period, and so customers are divided into two groups; namely, customers on inhaul and backhaul trips.

It is to be noted that customers on the inhaul trip are always the first priority. Customers on the inhaul line refer to customers to whom products are being delivered, and customers on the backhaul line are those from whom vehicles receive products on their return trip. Furthermore, vehicles always start a round trip from the depot station and return to this station after visiting one or more customers. Backhaul subject can be observed in the supply of automobile parts or gas delivery to subscribers.

For example, a company supplying automobile parts the number of pieces delivered to their customers and some pieces from other customers received that these received parts could be defective parts.

The important subject is that except for the investigations by Nolz et al. [Nolz et al., 2014] and Niakan and Rahimi [Niakan and Rahimi, 2015], who introduced a risk factor in the IRP by considering it in their problems, there was no other study that considered such issues in IRPs. In this paper, we define the transportation risk in the objective function.

So my contribution in this journal is:

- Consider the transportation risk in the model.
- Consider the backhaul transportation option on the IRP model.
- Consider the problem in multi-objective form, multi-product, multi-period, and heterogeneous fleet.

Table 1. Classification of the literature review for this article

Author(s)	Objective Function	Date Nature	Heterogeneous Fleet	Multi-Product	Multi-Period	Solving Method
Aziz and Moin (2007)	Single	Certain		✓	✓	Hybrid GA
Zhao et al. (2008)	Single	Certain			✓	TS algorithm
Huang and Lin (2010)	Single	Uncertain		✓		ACO algorithm
Moin et al. (2011)	Single	Certain		✓	✓	Hybrid GA
Liu and Chen (2012)	Single	Certain			✓	VNS
Popovic et al. (2012)	Single	Certain		✓	✓	VNS
Nekooghadirli and Tavakoli-Moghadam (2014)	Multi	Uncertain	✓	✓	✓	MOICA-NSGAII-MOSA-PAES
Amorim and Almada (2014)	Multi	Certain		✓	✓	Epsilon constraint-NSGAII
Nolz et al. (2014)	Multi	Certain		✓	✓	ALNS
Mirzaei and Seifi (2015)	Multi	Certain		✓	✓	Hybrid (TS-SA)
Niakan and Rahimi (2015)	Multi	Uncertain		✓	✓	Exact- Fuzzy
Ghorbani and Akbari Jokar (2016)	Single	Certain		✓	✓	Hybrid (ICO-SA)
Azadeh et al. (2017)	Single	Certain	✓		✓	GA
Alinaghian and Shokouhi (2017)	Single	Certain		✓	✓	ALNS
Rahimi et al. (2017)	Multi	Uncertain		✓	✓	Fuzzy
Rayat et al. (2017)	Multi	Certain		✓	✓	MOSA-NSGAII
This paper	Multi	Certain	✓	✓	✓	MOICA

2. Model Description

This section presents the mathematical model of the multi-period, multi-product IRP with backhaul transportation, which can be formulated in the form of a mixed-integer programming model.

The proposed model is based on the following assumptions:

- Model is a single depot that services to all of the customers (retailers).
- Distribution fleet is heterogeneous, but each of them has a limited and specific capacity.
- Distances between points are known.
- The customer's stock is limited.
- Inhaul customers are preferred to backhaul customers to receive or send goods.
- There is no route that merely includes the customers of the backhaul line.
- Demands of customers are predetermined.
- Shortages are not allowed.
- The strategy of sending or receiving goods is direct shipment. In other words, in a given period, the total customer demand must be done and cannot be splitted.

Before describing the model, the notations used to describe the model are defined below.

2.1 Sets and indices

i, j, μ, λ: Demand nodes index

A: Total number of customers

u: Index of customers on the inhaul trip

w: Index of customers on the backhaul trip

{0,U+W+1}: Depot index

v: Vehicle index

p: Distributable product index

t: Index of time periods

2.2 Parameters

cf_v^t : Fixed costs of using vehicle v for a period of t.

cv_v^t : Variable costs of using vehicle v for a period of t.

du_{it}^p : Demand of the i-th customer on the initial trip for product p in a period of t.

dw_{it}^p : Demand of the i-th customer on return trip for product p in a period of t.

Q_v : Capacity of vehicle v per unit weight.

r_{ijvt} : Risk between customers i and j by vehicle v in a period of t.

C_i : Inventory capacity (storage capacity) of the i-th customer.

S_{ij} : Edge length between customers i and j.

C_{ijvt} : Costs of traverse between customers i and j by vehicle v in a period of t.

y_i^p : Storage and maintenance costs undergone by the i-th customer for product p.

q^p : Weight of the p-th product.

2.3. Variables

x_{ijvt} *:* Binary variable (1, if vehicle v travels from i to j; 0, otherwise).

M_{ivt}^p : Amount of product p delivered to the i-th customer by vehicle v at the beginning of the t-th period (inhaul trip).

N_{ivt}^p : Amount of product p received from the i-th customer by vehicle v at the beginning of the t-th period (backhaul trip).

I_{it}^p : Amount of inventory of product p held by the i-th customer at the end of period t.

B_{ijvt}^p : The amount of product p transported from customer i to customer j by vehicle v during period t.

2.4. Mathematical model

$$\text{Min } Z_1 = \sum_{t=1}^{T}\sum_{v=1}^{V}\sum_{j=1}^{U+W} cf_v^t . x_{0jvt} + \sum_{t=1}^{T}\sum_{i=0}^{U+W}\sum_{j=1}^{U+W+1}\sum_{v=1}^{V}(cv_v^t . S_{ij}) x_{ijvt} + \sum_{t=1}^{T}\sum_{i=1}^{U+W}\sum_{p=1}^{P} y_i^p . I_{it}^p \ ; \ (1)$$

$$\text{Min } Z_2 = \sum_{t=1}^{T}\sum_{i=0}^{U+W}\sum_{j=0}^{U+W}\sum_{v=1}^{V} r_{ijv} . x_{ijvt} \ ; \ (2)$$

Subject to:

$$I_{i,t-1}^p - I_{i,t}^p + \sum_{v=1}^{V} M_{ivt}^p = du_{it}^p \ ; \forall i \in \{1,\dots,U\}, p, t \ ; \ (3)$$

$$I_{i,t-1}^p - I_{i,t}^p - \sum_{v=1}^{V} N_{ivt}^p = -dw_{it}^p \ ; \forall i \in \{U+1,\dots,U+W\}, p, t \ ; \ (4)$$

$$\sum_{v=1}^{V}\sum_{\mu=0}^{U} B_{\mu ivt}^p - \sum_{v=1}^{V}\sum_{\lambda=1}^{U+W+1} B_{i\lambda vt}^p = \sum_{v=1}^{V} M_{ivt}^p \ ; \ (5)$$
$$\forall i \in \{1,\dots,U\}, p, t \ , \ i \neq \mu \neq \lambda$$

$$\sum_{v=1}^{V}\sum_{\lambda=U+1}^{U+W+1} B_{i\lambda vt}^p - \sum_{v=1}^{V}\sum_{\mu=1}^{U+W} B_{\mu ivt}^p = \sum_{v=1}^{V} N_{ivt}^p \ ; \ (6)$$
$$\forall i \in \{U+1,\dots,U+W\}, t, p \ ; \ i \neq \mu \neq \lambda$$

$$\sum_{v=1}^{V}\sum_{i=0}^{U+W} x_{ijvt} \leq 1 \ ; \quad \forall j \in \{1,\dots,u+w+1\}, t \ ; \ (7)$$

$$\sum_{j=1}^{U+W} x_{0jvt} \leq 1 \quad ; \; \forall\, v,t \;;\; (8)$$

$$\sum_{i=1}^{U+W} x_{i,(U+W+1)v,t} = \sum_{j=1}^{U+W} x_{0jvt} \;;\; \forall\, v,t \;;\; (9)$$

$$\sum_{i=1}^{U+W} x_{i\mu vt} - \sum_{j=1}^{U+W} x_{\mu jvt} = 0 \;;\; (10)$$

$$\sum_{i=U+1}^{U+W} \sum_{j=1}^{U} \sum_{v=1}^{V} x_{ijvt} = 0 \quad ; \quad \forall\, t \;;\; (11)$$

$$\sum_{p=1}^{P} I_{it}^{p} \leq C_i \quad ; \quad \forall\, i \in A\,, t \;;\; (12)$$

$$\sum_{i \in H} \sum_{j \in H} x_{ijvt} \leq |H| - 1 \;;\; (13)$$

$$\forall\, v,t\,,\; H \subset A\,, 2 \leq |H| \leq u + w - 2$$

$$\sum_{p=1}^{P} q^p\, M_{jvt}.\, x_{ijvt} \leq \sum_{p=1}^{P} q^p\,.\, B_{ijvt}^{p} \leq \left(Q_v - \left(\sum_{p=1}^{P} q^p.\, M_{ivt} \right) \right) x_{ijvt} \;;(14)$$

$$\forall i,j \in \{0,1,\dots,U\}\,, j \neq 0\,, v\,, t$$

$$0 \leq \sum_{p=1}^{P} q^p\,.\, B_{ijvt}^{p} \leq Q_v.\, x_{ijvt} \;;\; (15)$$

$$\forall\, i \in \{U,\dots,U+W\}\,, j \in \{U+1,\dots,U+W+1\}, v{,}t$$

$$S_{0,U+W+1} = S_{U+W+1,0} = 0 \;;\; (16)$$

$$C_{0,U+W+1,v,t} = 0 \;;\; (17)$$

$$S_{ii} \gg \infty \;;\; (18)$$

$$M_{ivt}^{p} \geq 0\,, N_{ivt}^{p} \geq 0\,, I_{it}^{p} \geq 0\,, B_{ijvt}^{p} \geq 0\,, x_{ijvt} \in \{0,1\} \;;\; (19)$$

The first objective function includes the fixed routing costs, shipment and delivery, and maintenance costs undergone by customers. The second objective function minimizes transportation risks on routes taken by vehicles. Equations (3) and (4) express the inventory balance for customers on a round trip with respect to their demand, respectively. Equations (5) and (6) represent the difference between the input and output of each node for customers on a round trip. Equation (7) shows that each customer is visited by a vehicle utmost once during each period. Equations (8) and (9) indicates that each vehicle starts at the central depot and returns to that after each trip. Equation (10) shows the continuity of the travel path. Equation (11) indicates that customers on the inhaul trip are to be visited and provided with service to customers on the backhaul trip. Equation (12) indicates compliance with the allowed storage capacity limit for each customer. Equation (13) is used to sub-tour elimination from vehicle routing problems. Equations (14) and (15) show the maximum and minimum load of variable products for each vehicle. Also, other constraints show assumptions and type of variables.

3. Solution Approach

In recent years, multi-objective optimization problems (MOP) and related solution methods have received extensive attention. When solving a multi-objective optimization problem, we search a set of solutions, known as non-dominated solutions (or Pareto-optimal solutions), in which none of the members is better than the others. In general, the MOP can be solved with three groups of methods.

The first group consists of the prior methods before the start of a solution. The problem is transformed into a single-objective problem and preferences of the decision-makers should be the priority. The second group comprises the posterior methods. These methods attempt to consider several objective functions simultaneously and generate a set of Pareto-optimal solutions, which ultimately allow the decision-maker to choose the most desirable solutions according to his preferences. The third group consists of the iterative methods, in which the decision-maker states his/her preferences during of solution. After several iterations, the method converges toward the preferable results.

In this section, at first, a solution representation and creating population are described. Then, the MOICA algorithm is presented followed by the ε-constraint method used in this paper. In the last section, the proposed solving methodology based on mentioned methods is illustrated.

3.1 Solution Representation

In the given model, the solution of the problem is represented by the matrix with one row and *U*+*W*+*V*-1 columns. The rows show how the vehicles are viewed by customers and the columns show the customers assigned to each vehicle and prioritize the customer visited by vehicles. Figure 1 depicts a typical solution representation.

Vehicle move from the depot	5	2	1	0	4	3	0	0

Figure 1. Solution representation for five customers and four vehicle

As noted above, the chromosome is represented by a matrix with an order of order (1×(*U*+*W*+*V*-1)). To clarify the subject, consider Figure 2, which is the matrix of 1×8 and shows the schedule of customer visits by vehicles. The blue color shows the sequence of the first vehicle movement. Typically, the first vehicle visit customers 5, 2, and finally 1. The red color is related to the second vehicle that visits customers 4 and 3, and the last two vehicles will not visit any customer. Figure 2 shows an initial population method.

3.2 Multi-Objective ICA

An imperialist competitive algorithm (ICA), which was first presented by Atashpaz and Lucas [Atashpaz and Lucas, 2007], is an evolutionary algorithm inspired by the social phenomenon, called colonialism. It is the population-based, similar to the other meta-heuristic algorithms, in which solution space is searched by points in the name of the country (similar to a chromosome in the GA). One part of these countries is called imperialist, and some elements act as colonies. The competition between imperialists to develop their power by taking over colonial-dominated countries of other imperialists and changing the colonies' position during this competition are two key principles of this algorithm.

In recent years, this algorithm was used to solve single-objective problems. The success of this algorithm also encourages solving multi-objective problems. Most researchers have reported this algorithm's success in solving problems with high complexity. Also, the novelty of this algorithm and the existence of a higher research potential are the other reasons for the recent attention to this algorithm.

1. Create the count of time periods and start with the first period.
2. Create a random sequence of inhaul customers.
3. On the basis of capacity limitations and inventory limitations, vehicles are randomly assigned to customers.
4. The process of step 3 applies to all inhaul customers.
5. After assigning inhaul customers, create a random sequence of backhaul customers.
6. On the basis of capacity limitations and inventory limitations, vehicles are randomly assigned to customers.
7. The process of step 3 applies to all backhaul customers.

8. Go to the next period.
9. This process applies to all periods, and then calculate the objective functions.

Figure 2. Creation method of the primary population in this model

On the other hand in a similar article [Tavakkoli et al., 2016] and other papers, for example [Zhu et al. 2016], [Zhang et al., 2015], [Chen et al. 2015], [Shim et al. 2015] and [Li et al. 2015] which was mentioned in this article, the authors used the MOICA, NSGA-II and MOSA methods to solve the multi-objective IRP model. Finally, the MOICA algorithm had the best performance in terms of multi-objective criteria, such as the number of Pareto's solution and spacing metric. Also, in terms of computational time, the MOICA algorithm is a better method that in short time has given acceptable quality in providing solutions.

Therefore, in this paper, the MOICA algorithm is used to solve the model. The steps of this algorithm are as follows:

Step 1: Initializing empires

Since the ICA is a population-based approach, so as the first step in this algorithm, we create the N_{pop} number of the country, which includes N_{imp} countries as imperialist and the rest are colonies. The countries under the domination of each imperialist are proportional to its power. For this purpose, first the rank of each country is calculated according to the FNDS Index.

All countries on the Pareto-optimal front are considered rank 1. The imperialist countries are selected from this set, which can greatly impact the coverage and diversity of solutions. Also, we consider the following two assumptions:

- Assumption 1: The power of each country is mainly associated with its rank. Considering this fact, the weakest country in a higher rank is more powerful than the strongest in a lower rank.
- Assumption 2: Countries with the same rank are compared with the Sigma method (Equation 20)

After applying fast non-dominated sorting, assuming that country *c* lies in Rank (*C*).

In the following, using the Sigma method, the fitness value and power of each country are derived from the following equation.

$$Power_c = \frac{1}{Fitness\ Value_c} = \left\{ \sum_{j=1}^{K} \left[\frac{f_j(C)}{\sum_{i=1}^{N_{Rank(C)}} f_j(i)} \right] (Rank(C) - 1) \times K \right\} \quad (20)$$

where *K* is the number of objectives, $f(i)$ is the value of the *i*-th objective, and $N_{Rank(C)}$ is the number of countries lying in $Rank(C)$. Finally, the power of the *c*-th country is $Power_c$. After determining the most powerful countries as imperialists, the number of countries under their power is obtained by:

$$NC_n = Round(P_n * N_{col}) \quad ; \quad (21)$$

That;

$$P_n = \left| \frac{Power_n}{\sum_{i=1}^{N_{imp}} Power_i} \right| \quad ; \quad (22)$$

where N_{imp} is the number of imperialists and N_{col} is the number of all colonies.

We select NC_n of colonies and assign them to the imperialists. Finally, the more powerful imperialist takes a greater number of colonies than the weaker imperialist.

Step 2: Moving the colonies towards their imperialists

In this step, some colonies take some of the characteristics of their imperialists and move with a little deviation towards their imperialists. These movements are calculated by:

$$x \sim U(0, \beta \times d) \qquad (23)$$

$$\theta \sim U(-\gamma, \gamma) \qquad (24)$$

Also, Figure 3 shows that the colonies movement to a new position in line with the imperialist country, where parameter β is greater than one and close to 2.

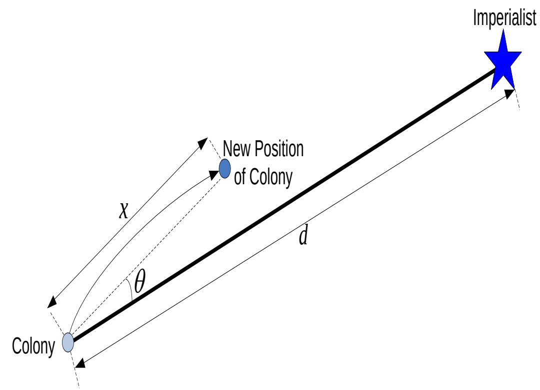


Figure 3. Moving colonies to the relevant imperialist

In this figure, parameter d is the distance between the colony and the imperialist and x is the amount of movement of a colony toward the imperialist that is a random number with a uniform distribution. Also, θ shows the movement angle and γ is an arbitrary parameter that its increase raises searches around the imperialist and vice versa. Usually, in most implementations, $\theta=\pi/4$ is a good choice.

Step 3: Change the Colonies and Imperialist position

During this step, a colony may achieve a better position than the relevant imperialist due to an improvement of its position. So, after Step 2, the power of each country is calculated again in each empire and the strongest country acts as the new imperialist.

Step 4: Calculate the total cost of empires

In order to determine the strongest and weakest empires, the total cost of each empire is calculated, which is commensurate with the cost of the concerned imperialist and the proportion of the average cost of the colonies. In order to calculate the total cost, all objective functions should be considered according to the two criteria of FNDS and Sigma as follows:

$$TC_n = Cost(imperialist_n) + \xi.mean\{Cost(colonies\ of\ empire_n)\ ; (25)$$

Where ξ is a number smaller than one.

Step 5: Competition of the Imperialists

In this step, the imperialists compete to get each other colonies. For this reason, the strongest empire takes the weakest colony of the weakest empire under his/her control. The power of each empire is normalized with respect to its total cost relative to other empires as shown below.

$$N.TC_n = TC_n - max_i\{TC_i\} \quad ; \quad (26)$$

Step 6: Elimination powerless Empires

If an empire loses all its colonies, it will be eliminated and competition between other empires will continue.

Stopping Criteria

If the stopping condition is not met (remaining only one empire in the world), return to Step 2 and continue the algorithm with new empires. Therefore, the flowchart of the algorithm used in this paper is shown in Figure 4.

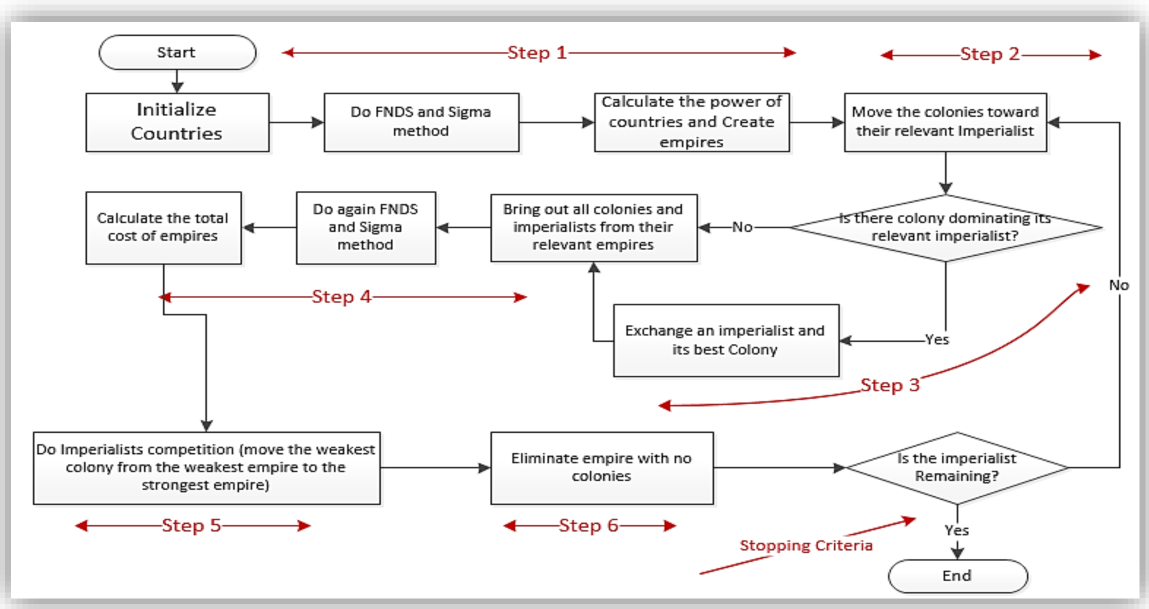


Figure 4. Flowchart of the proposed MOICA

3.3 Epsilon constraint method

The epsilon constraint (i.e., ε-constraint) method is one of the best-known methods for solve MOPs. In this method, one of the objective functions must be considered as the main objective function (randomly) and other objective functions must be converted to model constraints [Mavrotas, 2009]. For more explanation, consider the following two functions:

The ε-constraint method is also one of the more common methods used in obtaining the Pareto frontier, for which one of the objective functions is considered as the main objective function, and all other functions are limited with the allowable ε_j. ε_j may be subjected to change in order to generate Pareto solutions. A description of the epsilon-constraint method can be found below: Suppose that the objective of simultaneously maximizing two objective functions f_1 and f_2, is as follows:

Where, S is the problem solution space.

$$\begin{array}{ll} \text{Max } f_1(X) & \text{Max } f_2(X) \\ \text{s.t.} & \text{s.t.} \\ X \in S & X \in S \end{array} \qquad (27)$$

According to the Epsilon constraint method, firstly, each of these two problems will be solved separately, and an optimal solution of each problem will be found while taking all limits and constraints into account. Calculating the values of each objective function for an optimal solution of the other objective functions will result in the Nadir values of each objective function, as follows:

$$\begin{array}{ll} X_1 \rightarrow f_1^*(X_1) & X_2 \rightarrow f_2^*(X_2) \\ \downarrow & \downarrow \\ X_1 \rightarrow f_2(X_1) & X_2 \rightarrow f_1(X_2) \end{array} \qquad (28)$$

Where, f^* indicates the optimal solutions separately obtained for each objective function. Next, starting from one of these solutions, Pareto solutions can be obtained using an iterative algorithm. Suppose that the first solution (X_1) is selected as a starting point. In this case, the following mathematical programming (29) will be solved to obtain the first Pareto solution.

$$\begin{aligned} &\text{Max } f_1(X) \\ &\text{s.t.} \\ &\quad X \epsilon S \\ &\quad f_2(X) \geq f_2(X_1) + \varepsilon \end{aligned} \tag{29}$$

The purpose of solving this problem is to obtain the best worse answer and the lowest decline in the optimal solution objective function $f_1^*(X_1)$ for f_1, such that a better answer can be achieved for f_2. Supposing that this answer will be called X_3, the next step of the algorithm will be to solve the following mathematical programming:

$$\begin{aligned} &\text{Max } f_1(X) \\ &\text{s.t.} \\ &\quad X \epsilon S \\ &\quad f_2(X) \geq f_2(X_3) + \varepsilon \end{aligned} \tag{30}$$

This procedure will continue until no worse solution can be found for f_1 to cause an improvement in f_2. The last solution is the answer that provides the best value for f_2 (i.e., point X_2). In this method, each of the initial solutions that lead to the optimization of one of the objective functions may be considered as the starting point. This selection is completely optional and does not impact the obtained Pareto points.

Finally, the following steps are necessary to apply the proposed ε-constraint method:

- Create the payoff table. To do this, optimize each objective function individually, and calculate the value of other objective functions at this optimal point. For each objective function, call the interval between the ideal value and the worst value (nadir value. In cases where finding nadir value is not straightforward, generate the payoff table with a lexicographic method.
- Choose one of the objective functions (f_j(x)) as the main objective function to be optimized, and transform other functions into constraints.
- Solve the mathematical model as provided in Equation (30). In this model, e_i is the nadir value of the objective function.

3.4 Measuring metrics

In this section, we introduce the main measuring metrics used in the proposed meta-heuristic algorithms.

- *Number of Pareto solutions (NP)*: This criterion indicates the number of Pareto solutions obtained by each algorithm. According to this index, a higher number of Pareto solutions are associated with a higher algorithm quality.
- *Spacing metric (SM)*: This criterion measures the uniformity of non-dominated solutions distribution within solution space, and can be defined by:

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{d} - d_i)^2} \; ; \; d_i = min_{j \neq i}\left(\sum_{m=1}^{M}\left|f_i^m - f_j^m\right|\right)$$

Where $\bar{d}$ stands for the average d_i. It is evident that the lower the spacing metric index, the better the algorithm.

- *Diversity Metric (DM)*: This criterion evaluates the diversity and distribution of Pareto solutions, and is defined by:

$$D = \left[\sum_{i=1}^{n} \max(\|\hat{x_i} - \hat{y_i}\|\right]^2 ; (32)$$

Where $\|\hat{x_i} - \hat{y_i}\|$ presents the Euclidean distance between two adjacent solutions $\hat{x_i}$ and $\hat{y_i}$ on the optimal boundary. It is also clear that a higher DM index is preferable.

- *Index of Mean Ideal Distance (MID)*: This criterion measures the average distance of Pareto solutions from the origin and is preferred to be as little as possible. In this equation, number OF indicates the number of solutions, g stands for the targets and sol represents the solutions.

$$MID = \sum_{sol=1}^{n} \sqrt{\sum_{g=1}^{number\ OF} {F_{sol.g}}^{2}} \ ; \ (33)$$

3.5 Parameters setting

The performance of the meta-heuristic algorithms is usually sensitive to the settings of the parameters that can influence the search behavior. In this section, the parameters of the model are explained. For this purpose, we use the Taguchi method that is one of the most powerful statistical methods used to set parameters.

3.5.1 Proposed MOICA

Table 2 shows to tune six parameters, namely the number of countries, number of imperialists, number of iterations, assimilation coefficient, assimilation angle coefficient and alpha coefficient used in three levels. Thus, the number of the selected factors, analysis levels, and standard orthogonal table L27 provided by the Taguchi method are chosen for this study (see Figures 5). The final results are obtained through a procedure similar to that of the previous algorithm as shown in Table 3.

Table 2. Parameters values at different levels of the proposed algorithm

Parameter	Level		
	1 (Low)	2 (Medium)	3 (High)
Number of countries (*NC*)	20	50	100
Number of imperialists (*NI*)	5	7	10
Number of iterations (*I*)	50	100	150
Assimilation coefficient (β)	1.5	2	3
Assimilation angle coefficient (γ)	0.1	0.5	π/4
Alpha coefficient (α)	0.05	0.075	0.1

Table 3. Levels and amounts results of parameters for the MOICA

	Parameters					
	Number of countries	Number of imperialists	Number of iterations	Assimilation coefficient (β)	Assimilation angle coefficient (γ)	Alpha coefficient (α)
Level	2	1	2	2	2	3
Amount	50	5	100	2	0.5	0.1

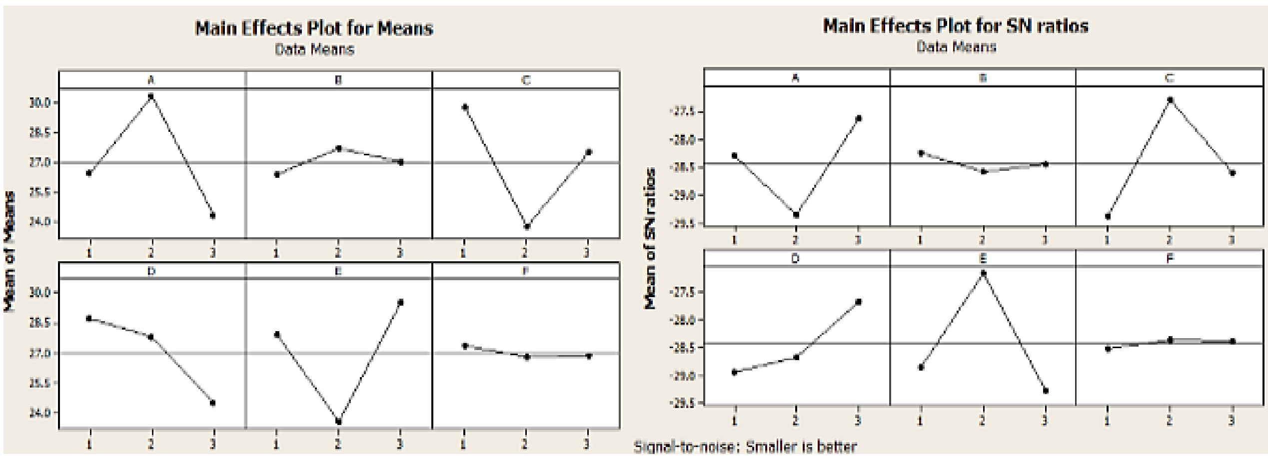


Figure 5. Main Effects plot for means and Main effect plot for S/N ratios (for the MOICA)

4. Discussion

In this section, the computational results derived from the model presented in this paper. Therefore, solutions of the problems are carried out in small and large sizes using the ε-constraint method. Furthermore, in order to validate the proposed meta-heuristic algorithm, the results obtained from the ε-constraint method are compared to those obtained from the proposed MOICA that seen in next section.

The problems and solutions are carried out on a personal computer with an Intel Core i5 processor 2.4 GHz and 4GB of internal memory, using GAMS and MATLAB software.

4.1 Creating sample problems

The sample problems are accidentally created in two large and small sizes. To do so, twenty five problems are created, among which fifteen problems are in small to medium sizes and ten problems are in large sizes.

To distinguish these groups, the time duration for solving the problem by the ε-constraint method is employed so that those problems which can be solved less than 30 minutes are categorized in the small-size group, whereas those that can be solved within maximum three hours are classified as the large size group. In addition, a method of coding representation including the number of customers, number of products, number of vehicles and number of planning periods is used to create the sample problems. Table 4 represents the values of the parameters of the problem.

Table 4. Values of problem parameters

Problem parameter	Parameter distribution
U & W ~ Uniform (3,50)	*P* ~ U (2,15)
V ~ U (2,20)	*T* ~ U (2,15)
Du & *Dw* ~ U (20,60)	*γ* ~ U (0.1,0.9)
Cf ~ U (200,400)	*Cv* ~ U (100,300)
C ~ U (60,100)	*Y* ~ U (30,60)
S ~ U (20,70)	*c* ~ U (20,70)
Q ~ U (1000,2000)	*q* ~ U (0.7,1.4)

4.2 Validation

In order to validate the model, we solve and analyze the small problem of the model with the corresponding methods.

For this purpose, we will consider the sample of problem P01. That is a problem with U=3, W=3, V=3, P=2, T=2. In other words, out of 6 customer sets, 3 customers went for the inhaul, and 3 other customers were considered for the backhaul trip.

In the following, the values for each parameters are produced according to the following tables.

Table 5. Parameter values for example test problem

Du				***Dw***			
T=1		T=2		T=1		T=2	
57	31	59	59	25	52	21	47
45	42	26	39	37	59	54	51
23	59	59	52	57	46	58	50

Cf		***Cv***		***Y***	
T=1	T=2	T=1	T=2	T=1	T=2
2780	3410	109	239	56	35
2310	2060	119	163	58	42
2340	2550	265	290	60	34
				56	30
				54	59
				45	39

Given the above parameters, we solve the model. So, the output of the model is based on the Pareto's solutions as follows:

Table 6. Final Output (for P_{01} test problem)

Pareto 1	T=1	0	0	1	3	2	5	6	4	Z1=5617
	T=2	0	2	3	1	5	6	0	0	Z2=5.3
Pareto 2	T=1	0	0	1	3	2	5	4	6	Z1=5734
	T=2	0	2	3	6	0	1	5	4	Z2=4.9
Pareto 3	T=1	0	0	1	3	2	5	4	6	Z1=6002
	T=2	2	1	3	6	5	0	0	0	Z2=4.3
Pareto 4	T=1	0	0	1	3	2	5	4	6	Z1=6493
	T=2	1	2	3	6	5	0	0	0	Z2=3.7
Pareto 5	T=1	1	2	0	0	3	6	4	5	Z1=6657
	T=2	0	3	2	1	0	5	4	6	Z2=3.6

For further analysis, we consider the Pareto 3, which seems best Pareto solution.

Figures 6 and 7 show that in this optimal Pareto solution, vehicles start moving from Depot, and after going through the route and sending the goods to inhaul customers, and delivery of goods from the backhaul customers, return to Depot. It is also quite obvious that first inhaul customers are served.

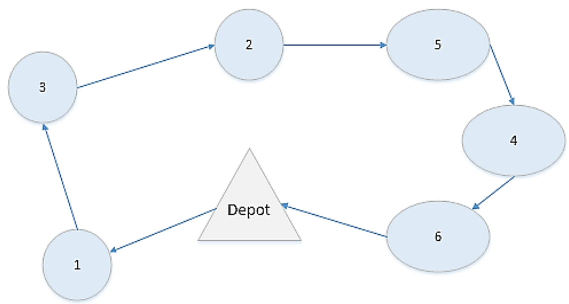


Figure 6. Optimal solution path for the problem (T=1)

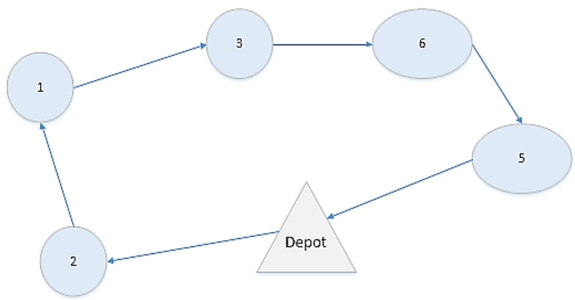


Figure 7. Optimal solution path for the problem (T=2)

4.3 Sample problem results

In this section, we compare the ε-constraint method with the proposed MOICA in solving the model.

4.3.1 Effectiveness of MOICA vs. Epsilon constraint in small and medium-size problems

In this subsection, 15 small and medium-size problems are generated to compare the proposed MOICA with an efficient Pareto set obtained by GAMS software using the ε-constraint algorithm. Table 7 demonstrates the related results. In this table, a maximum of five Pareto solutions is considered.

In the ε-constraint method, the number of Pareto solution is considered to be its input but in the meta-heuristic algorithm, we record a maximum of five points of selective Pareto solutions. Also, the computational time of each algorithm is shown in this table. As can be seen from this table, computational time of the MOICA is much less time than ε-constraint method (about 12.3%).

Table 7. Solving the proposed model for small- and medium-sized problems

Prob. (U/W/P/V/T)	ε-constraint		MOICA	
	Pareto solution	Time (s)	Pareto solution	Time (s)
P01 (3/3/2/3/2)	(5295,5.4), (5431,4.9), (5888,4.2), (6254,3.6), (6581,3.5)	145	(5617,5.3), (5734,4.9), (6002,4.3), (6493,3.7), (6657,3.6)	14
P02 (4/3/3/2/3)	(9894,11.5), (10065,11.3), (11551,10.5), (11430,9.6), (11523,8.9)	157	(10702,11.7), (11040,11.2), (11309,10.5), (11874,9.8), (11933,9.2)	16
P03 (5/5/3/4/4)	(30643,18.2), (33185,16.9), (36482,17.4), (38806,17.1), (44612,16.2)	250	(33213,19), (35228,18.5), (37312,17.8), (41198,17.1), (45765,16.9)	24
P04 (6/4/4/3/3)	(24113,13.7), (27761,13.3), (30631,12.1), (35006,11.5), (38808,10.6)	236	(25361,14.2), (29743,13.3), (32505,12.5), (36733,11.9), (39886,11)	22
P05 (7/5/4/3/4)	(32683,25.3), (33471,24), (38741,21.1), (42959,20.1), (44508,18.6)	333	(35549,25.5), (36498,23.7), (41501,20.2), (44664,19.9), (47357,18.6)	31
P06 (8/6/5/4/4)	(53850,27.2), (55843,25.7), (58371,23.1), (60107,22.3), (63881,21.4)	362	(60248,25.3), (61164,25.2), (63816,22.8), (66425,22)	32
P07 (8/7/5/4/5)	(70613,45.4), (76531,40.2), (81152,36.5), (85578,34.8), (93162,33.9)	425	(76426,46.4), (79487,37.1), (83689,36.3), (90927,35.2), (98995,34.8)	42

P08 (9/8/6/5/5)	(91324,43.6), (97187,41.6), (110438,38.7), (121906,37.2), (128613,35.9)	611	(100649,45.4), (109840,41), (118204,38.6), (126368,38.4), (134751,37.5)	63
P09 (9/9/6/5/6)	(131629,56.2), (148734,52), (160537,51.1), (168784,49.9), (177908,49.5)	704	(143589,59.9), (158295,53.6), (165854,53.6), (177701,53.3), (185063,51.6)	70
P10 (11/11/8/6/7)	(221613,85.3), (245612,84.2), (263907,83), (278816,82.1), (297508,78.8)	989	(252184,87.9), (263691,85.5), (276882,82.6), (295509,86.4), (316447,82.8)	106
P11 (12/11/8/8/8)	(313091,107.3), (320813,105.1), (328718,101.7), (340086,100.6), (358852,98.5)	1116	(333306,115), (345816,110.1), (352930,108.5), (360857,107), (380023,105)	128
P12 (13/11/9/9/8)	(324700,124.5), (348851,117.2), (367565,111.2), (386064,108.3), (392226, 105.1)	1235	(367481,136.9),(377518,124.7), (384402,117.7),(398467,110.5), (409830,109.7)	155
P13 (14/12/9/8/8)	(334872,125.3), (367808,120.5), (388530,116.5), (404373,112), (425771,107.6)	1436	(388391,132.8),(402951,129.6), (418665,123.3),(426638,119.9), (433286,114.2)	207
P14 (14/13/9/9/8)	(359540,120.5), (377802,117.5), (394736,113.3), (418909,110.6), (437746,105.8)	1597	(422522,126.3),(436953,124), (438356,122.4),(446581,119.6), (454746,113.8)	228
P15 (15/14/10/9/9)	(440872,156.1), (479083,153.4), (503612,149.7), (528867,145.2), (556911,137.6)	1795	(525264,168),(548901,158), (562735,154.7),(579249,152.9), (642145,149.3)	258
Mean		12.65 min		1.55 min

To explain more about the ε-constraint method, the problem P_{07} is provided. Additionally, since objective functions appear in a linear form, and hence convex, the application of ε-constraint method seems to be a good choice. First, the optimization of each objective function is considered without regard to other objective functions. Therefore, an optimal solution of each objective function will be calculated individually. Next, the first objective function will be computed based on the optimal solution of the second objective function and vice versa to obtain the solutions of the beginning and ending of the Pareto front with GAMS software. The results are presented in Table 8.

Table 8. Calculation of objective functions for P_{07} (payoff table)

Min Obj_1=Transportation	Min Obj_2=Risk
$Obj_1^*= f_1(X_1^*) =70613$	$Obj_2^*= f_2(X_2^*) =33.9$
$Obj_1= f_1(X_2^*) =93162$	$Obj_2= f_2(X_1^*) =45.4$

One of the functions is then adopted as the basis of movement (i.e., the second objective function) to obtain other Pareto points if exist. The basis objective function is moved away from its optimal value as much as an epsilon and then should be added to the problem as a constraint, and finally the problem has to be optimized based on another objective function. Now, the number of grid values is considered 8. Thus, the epsilon value is calculated by:

$$\varepsilon = \frac{45.4 - 33.9}{8 - 1} = 1.65$$

Now, the following constraint is added to the problem:

$$Min\ Obj_2 = Risk \leq 45.4 - 1.65$$

Then, the problem is optimized according to the first objective function. This will be repeated for the subsequent steps, and the solutions will also be recorded in each step until the problem is proved to be infeasible. Table 9 provides information on the results of the repetition of epsilon constraint method. Therefore, the Pareto solutions of this problem based on the ε-constraint method are as follows: (70613, 45.4), (76531, 40.2), (81152, 36.5), (85578, 34.8) and (93162, 33.9) as shown in Table 7.

Table 9. Epsilon's calculation table

Round	Limitation	Obj_1	Obj_2	Efficiency
1	45.4	70613	45.41	Efficient
2	43.75	76531	40.23	Efficient
3	42.1	76531	40.23	-
4	40.45	76531	40.23	-
5	38.8	81152	36.5	Efficient
6	37.15	81152	36.5	-
7	35.5	85578	34.82	Efficient
8	33.9	93162	33.93	Efficient

Finally, Figure 8 shows the Pareto diagram for two methods (for the sample problem P_{07}). As can be seen, the Pareto solutions of the MOICA show good quality and are almost identical to those obtained using the ε-constraint method. Moreover, as indicated in this figure, increased costs can reduce the total risk, whereas a rise in the total risk will minimize system costs, which indicates the conflict of objective functions and accounts for the use of multi-objective optimization methods. Moreover, as indicated in this figure, increased costs can reduce the total risk, whereas a rise in the total risk will minimize system costs, which indicates the conflict of objective functions and accounts for the use of multi-objective optimization methods.

The second results shown in Table 7 that we present the relative gap. It is necessary to mention that errors of the two algorithms regarding the optimal solution can be found with respect to objective functions that shown below.

$$GAP = \frac{OF_{MOICA} - OF_{Epsilon}}{OF_{Epsilon}} \tag{34}$$

Where OF_{MOICA} and OF_{Epsilon} are the average objective value among the Pareto's solutions that provided by the proposed meta-heuristic algorithms and Epsilon for both objective functions ($\overline{Z_1}$ and $\overline{Z_2}$).

As can be seen in figure 9, with the increase in the dimensions of the model, the average gap is also rising. In other words, in small-size problems, the distance between the solutions of the model based on these two methods is almost negligible and as the problem rises, the gap increases.

4.3.2 Effectiveness of the MOICA vs. Epsilon constraint method in large-sized problems

Finally, for large-size problems (with the computational time less than three hours for the ε-constraint method), we use multi-objective criteria for P_{16} to P_{25} test problems as shown in Table 10.

Table 10. Solving the proposed model for large-size problems

Prob. (U/W/P/V/T)	Epsilon				MOICA			
	DM	SM	MID	Time (min)	DM	SM	MID	Time (min)
P16 (16/16/12/9/9)	55311	1.44	34.5	41.8	84757	0.82	14.9	5.45
P17 (18/18/13/11/10)	46176	1.11	32.3	52.2	64190	0.79	17.4	8.16
P18 (20/20/12/9/8)	69505	0.89	29.5	59.6	102503	0.67	10	7.64
P19 (22/23/10/11/11)	77732	1.45	41.5	58.3	135810	0.72	27.3	8.68
P20 (25/25/11/11/10)	86008	1.85	38.7	67.5	164620	1.16	21.4	10.75
P21 (30/28/10/12/10)	82205	0.65	31	75.3	135772	0.29	33.6	11.81
P22 (30/30/10/15/10)	79703	1.83	35.2	90.1	124087	0.9	16.9	13.58
P23 (35/32/11/15/10)	58333	1.06	26.1	125.2	94211	0.72	22.6	17.13
P24 (40/35/10/16/12)	104609	0.98	42.4	142.4	277824	0.66	17	20.05
P25 (45/45/12/20/12)	131742	1.27	45.3	172.8	265438	0.64	21.6	22.65
Mean	79132	1.25	35.7	88.5	144921	0.74	20.3	12.6

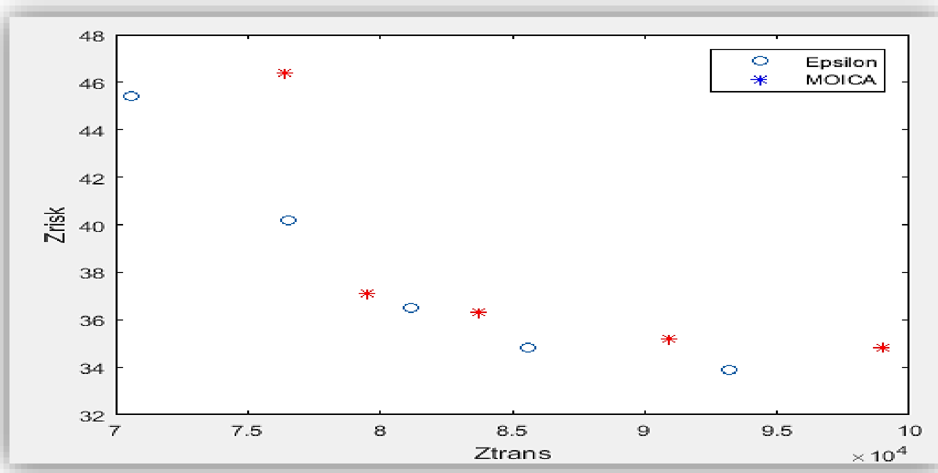
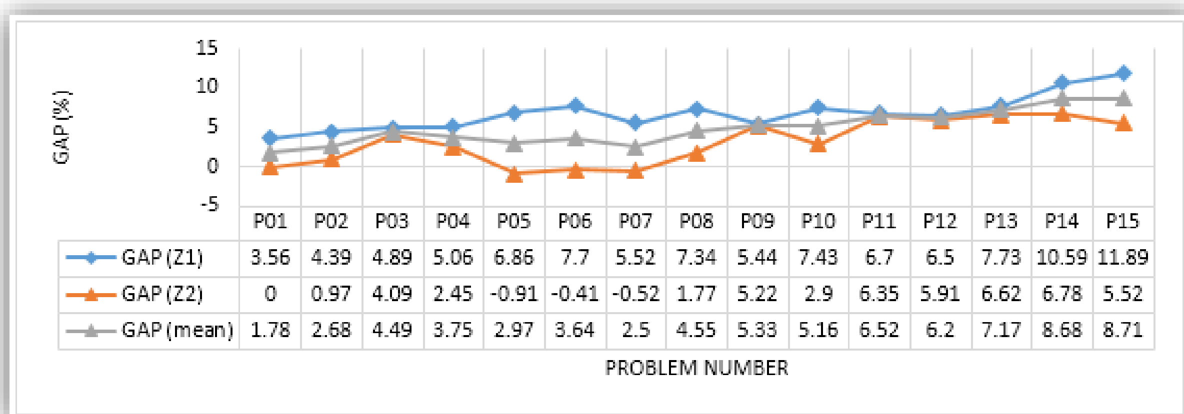


Figure 8. Pareto solutions of two mentioned algorithms for P07 problem



	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	P12	P13	P14	P15
GAP (Z1)	3.56	4.39	4.89	5.06	6.86	7.7	5.52	7.34	5.44	7.43	6.7	6.5	7.73	10.59	11.89
GAP (Z2)	0	0.97	4.09	2.45	-0.91	-0.41	-0.52	1.77	5.22	2.9	6.35	5.91	6.62	6.78	5.52
GAP (mean)	1.78	2.68	4.49	3.75	2.97	3.64	2.5	4.55	5.33	5.16	6.52	6.2	7.17	8.68	8.71

Figure 9. Comparison of MOICA GAP than to Epsilon in mentioned test problems

As can be seen in table 10, spacing metric reveals that the value of the MOICA algorithm is lower than that of the ε-constraint method, thereby proving the superiority of this algorithm. The diversity metric, on the other hand, confirms that the proposed MOICA is more extensive in Pareto solutions than the ε-constraint method. Finally, the MID metric also points to the higher efficiency of the MOICA due to its lower value for this algorithm than the ε-constraint method.

Also, table 11 represents the results of the statistical analysis of the metrics of multi-objective problems. To do so, the following statistical assumption is considered:

$$\begin{cases} H_0 = \bar{d} = 0 \\ H_1 = \bar{d} \neq 0 \end{cases}$$

Where $\bar{d} = d_{Proposed} - d_{Epsilon}$ and the values of d account for multi-objective metrics. It is noteworthy that since the lower values of MID and SM metrics imply better performance of the algorithm, the alternative hypothesis of a problem is changed $\bar{d} < 0$. This hypothesis is tested using *t*-test at the 97.5% level of significance. Additionally, the Kolmogorov-Smirnov (K-S) test is also employed to test the collected data for normality, because *t*-test and ANOVA methods assume that the sample data follow a normal distribution. In the K-S test, if the *p*-value is greater than 0.05, then we can say that the data distribution is normal. This is shown in Figure 10.

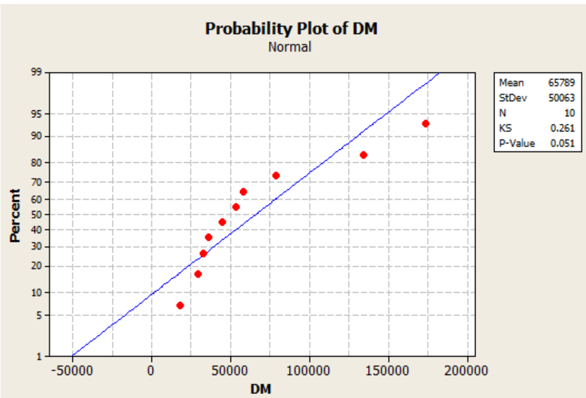


Figure 10. Normal probability plot for the statistical analysis for the DM metric

As an example, let us consider the case of the DM metric. The value of d_i, which is the difference of the values of this metric for the ε-constraint method and the proposed MOICA, is documented and then the mean and variance of the numbers will be calculated. Finally, the test statistic calculation follows the following procedure:

$$t = \frac{\bar{d} - 0}{S/\sqrt{n}} = \frac{65789 - 0}{50062/\sqrt{10}} = 4.155$$

Finally, the obtained result will be compared with $t_{\alpha,9}$ at the 97.5% level of significance. Since 4.155>2.262, the null hypothesis is rejected, while alternative hypothesis is accepted. That is, $d_{MOICA} - d_{Epsilon} > 0$.

Table 11. Summary of the statistical analysis results

	DM	SM	MID	Time
Epsilon vs MOICA	$P_{k\text{-}s}$ = 0.051	$P_{k\text{-}s}$ = 0.081	$P_{k\text{-}s}$ = 0.085	$P_{k\text{-}s}$= 0.144
	t= 4.15	t= 6.98	t= 5.58	t= 6.30
	P_{value}= 0.001*	$P_{value} \simeq 0^*$	$P_{value} \simeq 0^*$	$P_{value} \simeq 0^*$

As can be seen, the results prove the superiority of the MOICA to the ε-constraint method within all metrics because the corresponding *p*-value is smaller than 0.05.

Figure 11 demonstrates the comparison of computational time among the mentioned methods. As can be seen in the ε-constraint method, as the number of the dimensions of the problem (particularly the number of customers and periods) increases, the computational time of problem enhances with a deeper slope, whereas it rises with a constant slope in the proposed MOICA. Therefore, it is observed from Figure 9, the time interval between two algorithms with greater intensity is increasing with respect to the dimensions of the problem. Moreover, the average time for solving a large-sized problem using the ε-constraint method in this algorithm is 88.5 minutes, while this is 12.6 min for the MOICA, which is approximately $\frac{1}{7}$ times of the ε-constraint method.

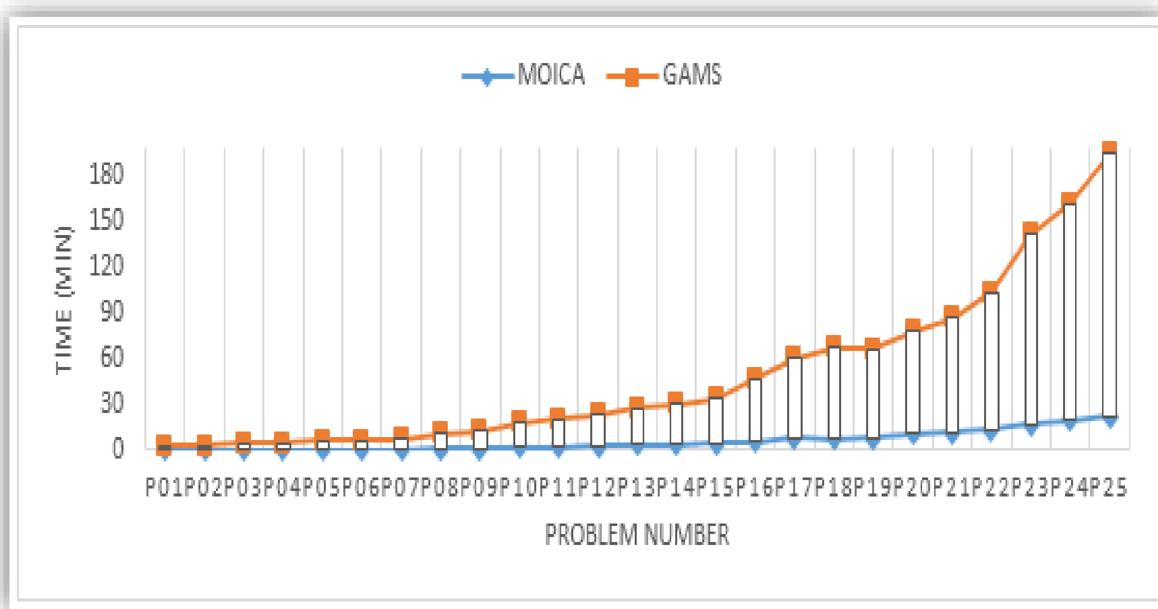


Figure 11. Comparison of the computational time for two algorithms in all test problems

4.4 Sensitivity Analysis

In this section, sensitivity analysis has been used to evaluate the effect of changes in model parameters on the values of objective functions.

For this purpose, we consider some important parameters of the model (*du*, *dw*, y_i, q_p, C_i), and by changing them, we obtain the changes of the objective functions.

Similarly, this analysis can be done for other parameters of this model.

The method of sensitivity analysis is that all the parameters of the problem, except for the parameter under consideration, remain constant and the results of this analysis are examined by making changes in the parameter values selected for the sensitivity analysis.

The results of this sensitivity analysis are shown in Figures 12 and 13.

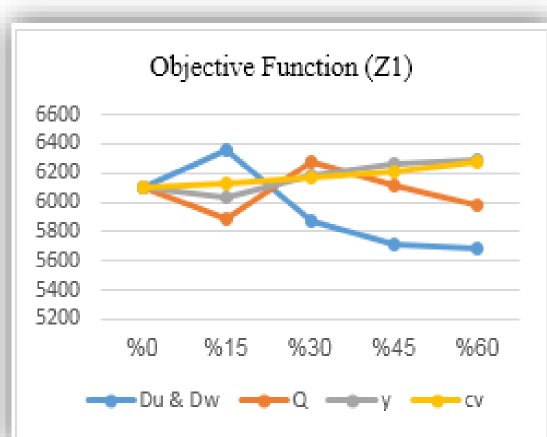


Figure 12. Sensitivity analysis for P_{01} (Objective Function Z_1)

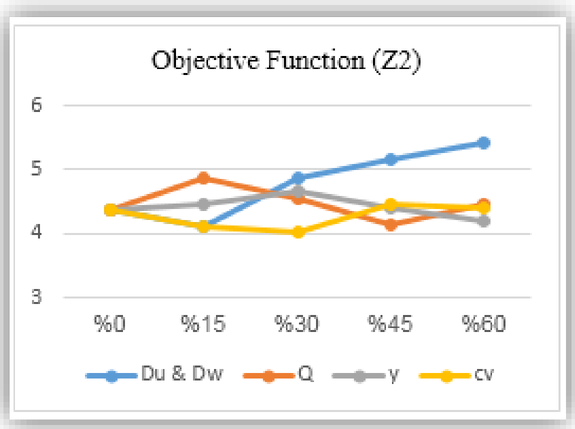


Figure 13. Sensitivity analysis for P_{01} (Objective Function Z_2)

From the above figures, there are several results that are referred to below:

1. By increasing the value of the parameter *du* and *dw* to 60%, the objective function Z_1 increases and the Z_2 is reduced. Only this does not happen when the parameter change is 15%.
2. The change of the *Q* parameter on the objective functions has not produced a definite effect. In other words, with a change of 30% in this parameter, the objective function Z_1, to the maximum and with a change of 45% in this parameter, the second objective function Z_2 reaches its lowest value.
3. By increasing the value of the parameter y_i, the value of the objective function Z_1 increases (except in 15%) and the value of the second objective function Z_2 is reduced (except in 30%).
4. By increasing the parameter value of *cv*, the objective function Z_1 is obviously increased, but the second OF Z_2 does not have a consistent effect.

So, what follows from this sensitivity analysis is that roughly the process of changing the objective functions are inverse. This can also show some degree of confliction in the objective functions.

4.5 Managerial implications

This paper present a new bi-objective inventory-routing problem according to some real cases. In this model, transportation risk has been added to the model. The risk of transportation is determined according to each route and type of vehicle. Experts' opinions can also be effective in calculating risk factors. Furthermore, usually researchers aggregate inventory costs and transportation risk and formulate them as a single objective function. But in non-cooperative real life cases, inventory holding costs are paid by retailers while the transportation related costs are paid by the distributer. In this paper, we separate these two cost elements and introduce a bi-objective IRP formulation in which the first objective will minimize the inventory holding cost and the second one will minimize the transportation risk. To demonstrate the conflict between the proposed objectives functions, the Pareto efficient frontiers of the test problem P_{07} are shown in Figure 8.

Simplicity of implementation in real cases, is another essential benefits of the proposed model. For example, considering the environmental aspects (e.g., minimization of air pollution) to the presented model, applying the model in a real-case industry (e.g., gas, petroleum and transportation of hazardous materials), developing any exact solution method (e.g., branch-and-price), and solving the model under uncertainty by a fuzzy method or robust optimization can be taken into account for future studies.

5. Conclusion

In this paper, a novel mathematical model and solution approach for bi-objective inventory routing problem with transportation risk was presented. Then, a linearization approach was applied to make the model linear. In non-cooperative real life cases, inventory-holding cost is paid by retailers while the transportation related risk are paid by the distributer. In this paper, introduced a bi-objective IRP with backhaul formulation; the first objective function included the system inventory cost and the second objective function minimized transportation risks on routes taken by vehicles. Also, two algorithms, namely MOICA and ε-constraint method, were developed to solve the presented model. For this purpose, 25 random test problem were produced with different sizes (small, medium and large size), and the model was solved based on the mentioned algorithms.

As discussed before, the ε-constraint method is an exact method, but it cannot solve large-size problems within a reasonable time. To overcome this problem, we proposed and used the MOICA to solve the model and then we compared the results of the model solution with the Epsilon constraint method.

It is worth noting that in small-size problems, the method of the MOICA was almost accurate and produced close-to-optimal solutions. But with increasing size of the problem, the Epsilon method was not able to answer in a short time. But, on the other hand, the method MOICA provided a good solution in an appropriate time. As for large-size problems, the Epsilon method was answered for about 1.5 hours, but the MOICA method was answered within 12 minutes. Of course, to clarify the above, we present a figure 9 that makes a comparison of the solutions obtained from the two methods. We also plotted the Pareto solutions of the two methods for the P_{07} problem, and measured the efficiency of the MOICA algorithm based on the criteria of multi-objective problems. In terms of these criteria, the MOICA method is superior to Epsilon's method in large-sized issues, which indicates that the proposed algorithm is much efficient.

For future research, we can use other methodological methods such as NSGA-II, MOPSO and … to compare the results. Also, other exact methods such as the Benders decomposition can be used.

6. References

- Alinaghian, M. and Shokouhi, N. (2017) "Multi-Depot multi-compartment vehicle routing problem, solved by a hybrid adaptive large neighborhood search", Omega, Article in Press.

- Atashpaz, E. and Lucas, C. (2007) "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialist competition." IEEE Congress on Evolutionary Computation, Singapore, 25-28

- Azadeh, A., Elahi, S., Farahani, M.H. and Nasirian, B. (2017). "A genetic algorithm-Taguchi based approach to inventory routing problem of a single perishable product with transshipment", Computer and Industrial Engineering. Vol. 104, pp. 124-133.

- Barnes, D. and Bassok, Y. (1997) "Direct shipping and the dynamic single-depot/multi-retailer inventory system", European Journal of Operational Research. Vol. 101, pp. 509-518.

- Ghannadpour, S. F. (2018) "Evolutionary approach for energy minimizing vehicle routing problem with time windows and customers' priority", International Journal of Transportation Engineering. Vol. 6, pp. 237-264.

- Ghatreh Samani, M. and Hoseini Motlagh, S. (2017) "A Hybrid algorithm for a two-echelon location- routing problem with simultaneous pickup and delivery under fuzzy demand", International Journal of Transportation Engineering. Vol. 5, pp. 59-

85.

- Ghorbani, A. and Akbari Jokar, M.R. (2016) "A hybrid imperialist competitive-simulated annealing algorithm for a multisource multi-product location-routing inventory problem", Computer and Industrial Engineering. Vol. 101, pp. 116-127.

- Huang, S. H. and Lin, P.C. (2010) "A modified ant colony optimization algorithm for multi-item inventory routing problems with demand uncertainty", Transportation Research Part E: Logistics and Transportation Review, Vol. 46, pp. 598-611.

- Liu, S.C. and Chen, A. (2012) "Variable neighborhood search for the inventory routing and scheduling problem in a supply chain", Expert Systems with Applications. Vol. 39, pp. 4149-4159.

- Mirzaei, S. and Seifi, A. (2015) "Considering lost sale in inventory routing problems for perishable goods", Computer and Industrial Engineering. Vol. 87, pp. 213-227.

- Moin, N. H., Salhi, S. and Aziz, N. (2011) "An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem", International Journal of Production Economics. Vol. 133, pp. 334-343.

- Nekooghadirli, N., Tavakkoli-Moghaddam, R., Ghazvini, V.R. and Javanmard, S. (2014) "Solving a new bi-objective location-routing-inventory problem in a distribution network by meta-heuristics", Computers and Industrial Engineering. Vol. 32, pp. 48-63.

- Niakan, F. and Rahimi, M. (2015) "A multi-objective healthcare inventory routing problem; a fuzzy possibilistic approach", Transportation Research Part E: Logistics and Transportation Review. Vol. 80, pp. 74-94.

- Nikkhah Ghamsari, A., Hoseini Motlagh, S. and Jokar, A. (2017) "A Two-Phase Hybrid Heuristic Method for a Multi-Depot Inventory-Routing Problem", International Journal of Transportation Engineering, Vol. 4, pp. 287-304.

- Nolz, P.C., Absi, N. and Feillet, D. (2014) "A Bi-objective inventory routing problem for sustainable waste management under uncertainty", Journal of Multi-Criteria Decision Analysis, Vol. 54, pp. 543-556.

- Popović, D., Vidović, M. and Radivojević, G. (2012) "Variable neighborhood search heuristic for the inventory routing problem in fuel delivery", Expert Systems with Applications, Vol. 39, pp. 90-98.

- Rahimi, M., Baboli, R. and Yacine, R. (2017) "Multi-objective inventory routing problem: A stochastic model to consider profit, service level and green criteria", Transportation Research Part E. Vol. 101, pp. 59-83.

- Rayat, F., Musavi, M. and Bozorgi, A. (2017) "Bi-objective reliable location-inventory-routing problem with partial backordering under disruption risks: a modified AMOSA approach", Applied Soft Computing, Vol. 65, pp. 132-144.

- Tavakkoli-Moghaddam, R., Raziei, Z. and Tabrizian, S. (2016) "Solving a bi-objective multi-product vehicle routing problem with heterogeneous fleets under an uncertainty conditions", International Journal of Transportation Engineering, Vol. 3, pp. 207-225.

- Zhao, Q. H., Chen, S. and Zang, C. X. (2008) "Model and algorithm for inventory/routing decision in a three-echelon logistics system", European Journal of Operational Research, Vol. 191, pp. 623-635.

Appendix

Suppose a distribution system with four customers and a central warehouse for distributing products. The capacity of the warehouse for each customer is 50 units, for which the shortage and maintenance costs based on the ending inventory are presented in table 1.

Table 1. Cost information for instance example

Costs	Customer			
	1	2	3	4
Maintenance Cost (Dollars per unit in each per period)	0.09	0.09	0.13	0.1
Shortage Cost (Dollars per unit in each per period)	2.8	3.4	2.7	3.2

The planning is set for a four period, during each period two vehicles, each with 100 products, are available. The fixed cost of using each vehicle is $10. Table 2 and figure 1 indicate the distribution system and demand system of each customer for each period, respectively.

Table 2. Customer demand in each period

Period	Customer			
	1	2	3	4
1	5	34	38	35
2	15	27	26	42
3	12	30	20	39
4	17	33	25	40
Sum	49	124	109	156

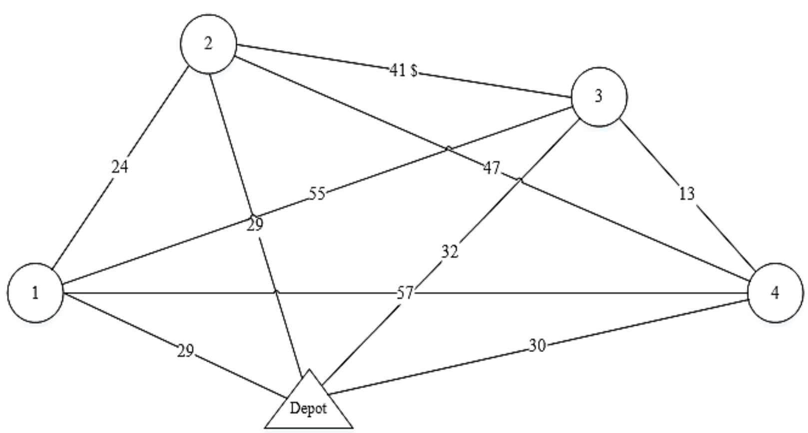


Figure 1. A sample of a distribution network with four customers and a warehouse (depot)

If the above problem is set to be solved in the form of four individual vehicle routing problem, then the sum of transportation costs will be $708 (=4*(157+20))and the inventory costs will be naturally equal to zero. A possible solution for the proposed problem can be found in tables 3, 4 and 5. Tables 6 and 7 represent the cost of this solution.

Table 3. Amounts of delivered goods

Period	Customer			
	1	2	3	4
1	0	61	64	35
2	49	0	0	81
3	0	63	45	0
4	0	0	0	40
Sum	49	124	109	156

Table 4. Ending inventory status

Period	Customer			
	1	2	3	4
1	-5	27	26	0
2	29	0	0	39
3	17	33	25	0
4	0	0	0	0
Inventory	46	60	51	39
Shortage	5	0	0	0

Table 5. Routs selection in each of the periods

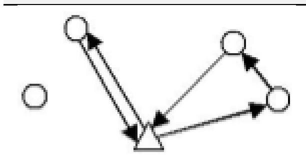
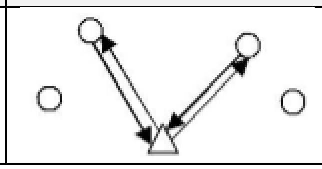
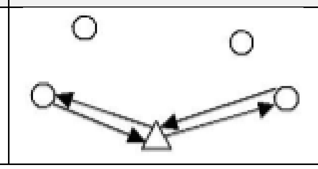
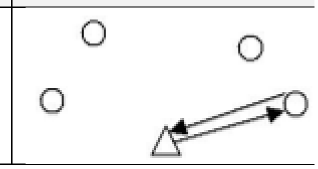
Period 1	Period 2	Period 3	Period 4
			

Table 6. Calculation of inventory costs resulting from the sample feasible solution

Costs	Customer				Sum
	1	2	3	4	
Maintenance Cost	46*0.09=4.14	60*0.09=5.4	51*0.13=6.63	39*0.1=3.9	20.07
Shortage Cost	5*2.8=14	0	0	0	14
Sum	18.14	5.4	6.63	3.9	34.07

Table 7. Calculation of transportation costs resulting from the sample feasible solution

Costs	Customer				Sum
	1	2	3	4	
Fixed transportation cost	20	20	20	10	70
Route Cost	58+75=133	58+60=118	64+60=124	60	435
Sum	18.14	5.4	6.63	3.9	505

Therefore, the total costs of the possible solution is \$539.07 (=505+34.07), which corresponds to 76.1% of the solution of the problem solved in terms of four individual vehicle routing problems.

The optimal solution for the number of delivered goods is presented in table 8. The costs incurred in the optimal condition is \$445.31 (424+21.31), which corresponds to 62.9% of the solution of the problem solved in terms of four individual vehicle routing problems. In other words, using inventory routing concept can save the costs up to 37.1%. The possible saving resulting from using this concept greatly depends on the magnitude of the transportation costs compared to scarcity and maintenance cost.

Table 8. The optimal amount of delivered goods

Period	Customer			
	1	2	3	4
1	20	61	64	35
2	0	0	0	66
3	29	63	45	55
4	0	0	0	0
Sum	49	124	109	156