Advances in Industrial Engineering, Autumn 2020, 54(4): 365-379
DOI: 10.22059/jieng.2021.325287.1775
RESEARCH PAPER

Comparing Multi-Objective Meta-Heuristics for Multi-Commodity Supply Chain Design Problem with Partial Coverage

Mehdi Seifbarghy [a,*], Meisam Soleimani [b], Mohsen Jabbari [c]

a. *Department of Industrial Engineering, Faculty of Engineering, Alzahra University, Tehran, Iran.*
b. *Department of Industrial and Mechanic Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran.*
c. *Department of Social Science, Imam-Khomeini International University, Qazvin, Iran.*

Received: 08 June 2021, Revised: 03 July 2021, Accepted: 04 July 2021

Abstract

A three-echelon multi-commodity supply chain including manufacturers, distribution centers (DCs) and customers is considered. Customers may be partially or fully covered by the DCs which should be opened in some candidate locations. A two-objective model is developed to find the locations of DCs and the flows of commodities in the whole supply chain considering a pre-determined number of DCs. The first objective function minimizes the total operation costs including transportation, inventory holding, production and site opening costs while the second objective maximizes the customers' partial coverages. Since the presented problem is NP-hard in nature, three metaheuristic algorithms of NSGA-II, NRGA and MOPSO are developed to find the Pareto-optimal solutions and are compared using some standard criteria for multi-objective algorithms. Numerical examples are designed to assess the performance of the model and the developed metaheuristic algorithms. Considering different criteria for comparing the algorithms, the superiority of some algorithms against others are reported.

Keywords:
Supply Chain;
Partial Covering;
NSGA-II;
NRGA;
MOPSO

Introduction

Making decision about selecting facility sites is one of the critical problems in the strategic planning of either private or public companies. This problem is of great importance in the supply chain structure with multiple commodities. Current et al. [1] surveyed the location models in discrete networks. Multi-product facility location models were presented by Warszawski and Peer [2]. Syam [3] proposed two heuristic algorithms based on Lagrangian relaxation and simulated annealing for multiple commodities supply chain model with several producers. Church and ReVelle [4] introduced location problems with maximum coverage. The maximal covering location problem (MCLP) addresses the issue of locating a limited number of distribution centers (DCs) that are going to cover a number of demand zones. Schilling et al. [5] explored the application of coverage problems in facility locating. Galvao and ReVelle [6] introduced a methodology for solving MCLP using the Lagrangian relaxation method. Gendreau et al. [7] applied MCLP for locating emergency vehicles. ReVelle [8] proposed some methodologies for solving MCLP based on heuristic algorithms.

* Corresponding author: (M. Seifbarghy)
Email: m.seifbarghy@alzahra.ac.ir

The given model in this paper is on partial coverage in a supply chain structure. MCLP with partial coverage was introduced by Karasakal et al. [9]. They surveyed the effect of partial coverage on the solutions of classic MCLP. Classical MCLP assumed full coverage of demand points by DCs. This full-coverage was assumed to be represented by the value of 1. In partial coverage, values between 0 and 1 can be assigned to the coverage. Fig. 1 shows the concept of partial covering besides the best solution generated by MCLP.

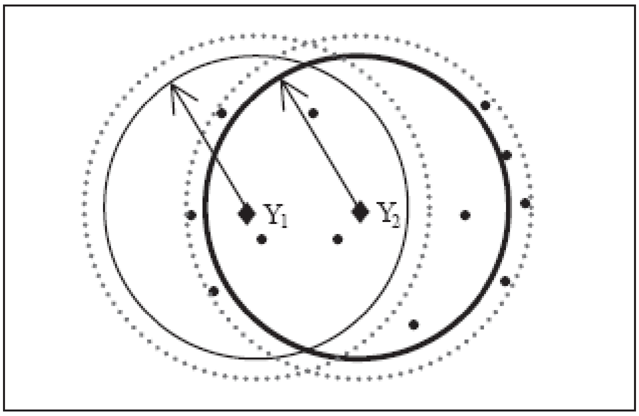

Fig. 1. MCLP with partial coverage [9]

Suppose there are two options to locate facilities and we are eager to have maximal coverage. The solid and dotted lines demonstrate minimum and maximum critical distances, respectively. Points of Y_1 and Y_2 can fully cover six and five demand points; thus, to have maximal coverage, Y_1 is the best solution. If we consider partial covering as the goal of the problem, then Y_2 is the better solution since it covers five demand points completely and seven demand points partially as well.

On the difference between the applications of partial and full coverages in the real world, we can say that full coverage is usually used for very emergency situations; for example, the location of fire stations. In this case, there is a coverage radius; for example, 10 minutes which means that if a demand point (in this case, the point of fire) is within 10 minutes of a station, then, the station can send the fire trucks and the firefighting operation will be successful, otherwise, it will not help. In less emergency situations, such as locations of schools, restaurants and so on, partial coverage can be used. In this case, some of the demand points are fully covered. Some others are not covered since they are far from the selected locations while some demand points are partially covered.

Mestre presented a method to solve partial covering problems using the Lagrangian relaxation method [10]. MCLP in supply chain structure can be one of the most attractive research areas in the field of supply chain and facility location. Melo et al. [11] presented a literature review for the combination of supply chain and facility location problems.

Pereira et al. [12] gave a hybrid algorithm combining a metaheuristic and an exact method in order to solve the probabilistic maximal covering location–allocation problem. To tackle larger instances, a flexible adaptive large neighborhood search heuristic was proposed to obtain location solutions, whereas the allocation sub-problems are solved to optimality. Seifbarghy et al. [13] studied a three-echelon multiple commodity supply chain model with a maximal covering approach with two objectives of maximizing the coverage of customer demand and minimizing the associated transportation cost required for meeting the customer demand. Considering that the given model is an NP-Hard problem, they applied a customized version of the greedy heuristic and clearly indicated its robustness.

Li et al. [14] studied the general structure in the humanitarian relief network and developed a maximal cooperative covering model with budget constraints. They maximized the coverage of the people in disastrous regions with uncertainty. They studied the effect of items' availability in the given relief chain management and compared the performance of the proposed model under cooperative and non-cooperative conditions. Eidy and Torabi [15] proposed a bi-objective mixed-integer nonlinear programming model in order to obtain the optimal number, locations and capacities of plants, DCs, and retailers, transportation modes and evaluate the coverage radius of retailers, in such a way as to minimize total transportation costs, maximize demand coverage, and achieve gradual coverage of facilities. Since the problem was NP-Hard, the NSGA-II algorithm was proposed to solve it. Vatsa and Jayaswal [16] examined the problem of assigning doctors to non-operational Primary Health Centers (PHCs) considering the maximum population which could be served by any PHC and the availability of doctor uncertainty. The problem was formulated as a robust capacitated multi-period MCLP with server uncertainty.

Karasakal and Silav [17] proposed a bi-objective facility location model which considered both partial coverage and service to uncovered demands. They assumed that demand nodes within the predefined distance of opened facilities are fully covered, and after that distance, the coverage level decreases linearly. The objectives were maximization of full and partial coverage and the minimization of the maximum distance between uncovered demand nodes and their nearest facilities. Cordeau et al. [18] introduced a novel exact algorithm for two coverage problems including MCLP which required determining a subset of facilities that maximized the amount of customer demand covered subject to a budget constraint on the cost of the facilities and the partial set covering location problem (PSCLP), which minimized the cost of open facilities while forcing a certain amount of customer demand to be covered. They applied a decomposition approach to the two addressed problems based on the branch-and-Benders-cut reformulation. El-Hosseini et al. [19] proposed a partial coverage and a power-aware internet of things (IoT)-based fire detection model with various multi-functional sensors for smart cities. The sleep scheduling approach was utilized for saving the energy of sensors and the need for any extra number of nodes needed for continuously covering the targeted area.

The aforementioned review showed that there was only one research (i.e. [13]) on the issue of partial coverage in the supply chain design. The addressed research studies this problem for the case of regular coverage. The current research in this paper extends and studies the partial coverage for a supply chain network design with two objectives of maximal coverage and minimal cost [13]; furthermore, the given model tries to find the optimum locations of DCs and the flows of materials from manufacturers to DCs and then to customers in a three-echelon supply chain. The research questions can be:

1-What is the optimum location for each opened DC?

2-What are the optimum value of flows from Manufacturers to DCs?

3- What are the optimum value of flows from DCs to customers?

This paper is organized as follows: problem description and formulation are presented in Section 2. In Section 3 we discussed the solution heuristics. Some numerical examples are given in Section 4. Conclusions, managerial insights and future research ideas are given in Section 5.

Problem description and formulation

We consider a three-level supply chain including a number of manufacturers, a number of potential locations for opening some DCs and finally a number of customers. An arbitrary number of products can be produced by each manufacturer. The products in the manufacturers are given to DCs and then from DCs to customers. The location of DCs should be determined due to the limited number of DCs which can be opened and the opening costs; furthermore, the location of DCs should be found with regard to the critical distance of customers' demand coverage by DCs so that the maximum number of demand points should be covered by DCs.

For this reason, this problem is formulated as a bi-objective model. One of the objectives is cost minimization while the other one is coverage maximization.

After opening DCs, customers who are being covered, due to their demands for each product and capacity of DCs and transportation costs, should be assigned to DCs; for this reason, the addressed cost objective includes the operational costs of the model composed of transportation, production and inventory holding costs. On the other hand, the DCs should be assigned to the manufacturers considering the transportation costs and production capacity of manufacturers for each product. Note that some of the customers' demands may be fully covered while others may be partially covered. In other words, as well as defining two values of maximal distance of partial covering (T) and maximal distance of full coverage (S), we define the parameter (G_{ij}) as in Eq. 1 as a function of distance between DC j and customer i (D_{ij}) [9].

$$G_{ij} = \begin{cases} 1 & if \quad D_{ij} \le S \\ f(D_{ij}) & if \quad S < D_{ij} \le T \;, (0 < f(D_{ij}) < 1) \\ 0 & if \quad D_{ij} > T \end{cases} \tag{1}$$

Parameter (G_{ij}) represents the coverage level that DC j gives to customer$i$; it is a value between 0 and 1. The formula for $f(D_{ij})$ can be as in Eq. 2 [9].

$$f(D_{ij}) = \frac{T - D_{ij}}{T - S} \tag{2}$$

Assumptions

- The DCs are assumed to be unlimited in capacity.
- Production capacities of products at each factory are independent.
- Direct product transport from factory to the customer is not authorized.
- All products are allowed to be produced by all factories.
- All customers can receive products of each factory from all DCs.
- Each customer is only covered by one DC.

Notation

The indices are defined as follows:
i: Index of customers ($i=1,2,\ldots,M$)
j: Index of potential locations for DCs ($j=1,2,\ldots,N$)
m: Index of manufacturers ($m=1,2,\ldots,P$).
k: Index of products ($k=1,2,\ldots,Q$).

The parameters are defined as follows:
C_{mj}^{k}: Costs of producing one unit of product k in manufacturer m and sending it to DC j
D_{ji}^{k}: Costs of holding one unit of product k in DC j and sending it to customer i.
S_{m}^{k}: Total capacity of manufacturer m for product k
H_{i}^{k}: Demand of customer i for product k
f_j: Cost of establishing a DC in potential point j
A: Maximum Number of DCs to be opened
R_1: A very large positive number less than 1
G_{ij}: covering level provided by DC j for customer i

The Decision variables of the model are defined as follows:
U_{mj}^{k}: The amount of product *k* which is produced in manufacturer *m* and will be sent to DC *j*.
T_{ji}^{k}: The amount of the demand of customer *i* for product *k* which is supplied by DC *j*.

$$y_j = \begin{cases} 1 & \text{if a DC locates at potential point j} \\ 0 & \text{Otherwise} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{if customer i is covered whether partially or completely by DC j} \\ 0 & \text{Otherwise} \end{cases}$$

The mathematical model

The mathematical model of the problem can be stated as:

$$MaxZ_1 = \sum_{k=1}^{Q}\sum_{i=1}^{M}\sum_{j=1}^{N} G_{ij}\, x_{ij} H_i^k \tag{3}$$

$$MinZ_2 = \sum_{j=1}^{N} f_j\, y_j + \sum_{m=1}^{P}\sum_{j=1}^{N}\sum_{k=1}^{Q} C_{mj}^k\, U_{mj}^k + \sum_{j=1}^{N}\sum_{i=1}^{M}\sum_{k=1}^{Q} D_{ji}^k\, T_{ji}^k \tag{4}$$

$S.T$

$$\sum_{j=1}^{N} y_j = A \tag{5}$$

$$x_{ij} \le y_j G_{ij} + R_1 \quad \forall\, i = 1,2,\dots,M \quad ,\forall j = 1,2,\dots,N \tag{6}$$

$$\sum_{j=1}^{N} x_{ij} \le 1 \qquad \forall\, i = 1,2,\dots,M \tag{7}$$

$$\sum_{j=1}^{N} U_{mj}^k \le S_m^k \qquad \forall\, m = 1,2,\dots,P \quad ,\forall k = 1,2,\dots,Q \tag{8}$$

$$T_{ji}^k \ge H_i^k x_{ij} G_{ij} \qquad \forall\, i = 1,2,\dots,M \quad ,\forall j = 1,2,\dots,N, \forall k = 1,2,\dots,Q \quad \backslash \tag{9}$$

$$\sum_{m=1}^{P} U_{mj}^k \ge \sum_{i=1}^{M} T_{ji}^k \quad \forall j = 1,2,\dots,N \quad ,\forall k = 1,2,\dots,Q \tag{10}$$

$$U_{mj}^k, T_{ji}^k \ge 0 \qquad \forall i, j, m, k \tag{11}$$

$$x_{ij}, y_j \epsilon \{0,1\} \qquad \forall i = 1,2,\dots,M \quad ,\forall j = 1,2,\dots,N \tag{12}$$

The first objective function maximizes customers' coverage by DCs while the second objective function minimizes operational costs including opening DCs, production, holding and transporting. Constraint (5) indicates the number of opened DCs should be equal to a predetermined value. Constraint (6) makes sure that each customer is covered by one of the DCs which has the qualification for coverage. Each customer can be covered by only one DC; this condition is given in Constraint (7). Constraint (8) gives the production capacity restriction of the manufacturers. Constraint (9) assures that the coverage amount of each customer is not higher than the flow arrived at the customer. Constraint (10) ensures that the output flow to each DC of each product is not higher than its corresponding input flow. Constraints (11) and (12) give the status of the decision variables.

Solution heuristics

Since the model is a bi-objective one, we have developed three different heuristics which are described in this section. In general, genetic algorithm (GA) has shown its performance as a very powerful algorithm for most facility location problems. We can extend this result for even bi-objective problems. There are a number of researches using the Non-dominated Sorting Genetic Algorithm (NSGA) and its variations for tackling with bi-objective problems. Since GA and NSGA are population-based heuristics, we have selected particle swarm optimization (PSO) and its bi-objective version for comparing with NSGA. The Multi-Objective Particle

Swarm Optimization (MOPSO) which is the multi-objective version of PSO, is a population-based heuristic like NSGA and can be a good rival for NSGA or its variations.

Non-dominated Sorting Genetic Algorithm II (NSGA-II)

NSGA-II is one of the efficient and popular multi-objective evolutionary algorithms which was introduced by Deb et al. [20]. The basic information to start NSGA-II is as follows:
- Initializing population size (*nPop*) which indicates the number of chromosomes to be kept at each stage.
- The probability of crossover operator (P_c) that is the number of parents who participate in the mating pool divided by the total number of parents.
- The probability of mutation operator (P_m) that is the probability of participation of a gen of solution in the mutation process
- Number of iteration (*nIt*)

The chromosome structure is defined as follow dependent on the variables

The structure of variable $X = [x_{ij}]$ is represented by an $M\times N$ matrix. The elements of such a matrix are binary values equal to 0 or 1. The structure of variable $Y = [y_j]$ is represented by a $1\times N$ binary matrix. In this matrix, elements are binary values equal to 0 or 1.

The structure of variable $U = [U_{mj}^k]$ is represented by a $P\times N$ matrix for each product. This structure can be extended to all products; the result will be a $P\times(Q.N)$ matrix. The value of each element indicates the quantity of product k produced and sent from manufacturer m to DC j. Fig. 2 shows the addressed matrix.

$$\begin{bmatrix} U_{11k} & \cdots & U_{1nk} \\ \vdots & \ddots & \vdots \\ U_{p1k} & \cdots & U_{pnk} \end{bmatrix}$$

Fig. 2. Representation of U_{mj}^k for each product

The structure of variable $T = T_{ji}^k$ is represented by an $N\times M$ matrix for each product. This structure can be extended to all products; the result will be a $N\times (Q.M)$ matrix. The value of each element indicates the quantity of product k sent from DC j to customer i, of course if this customer is covered by that DC. Fig. 3 shows the addressed matrix.

$$\begin{bmatrix} T_{11k} & \cdots & T_{1mk} \\ \vdots & \ddots & \vdots \\ T_{n1k} & \cdots & T_{nmk} \end{bmatrix}$$

Fig. 3. Representation of T_{ji}^k for each product

In all, we can compress all the matrices in a $N\times(Q.M+Q.P+M+1)$ Matrix. Let's give an example of solution representation for a two manufacturers, three potential DCs, five customers and only one product; i.e. $N=3$, $P=2$, $M=5$, $Q=1$. As is clear from Fig. 4, the first and second DCs are opened. The first and second manufacturers give the product to DC 1 while the second DS is only receives the product from manufacturer 1.

$$\begin{bmatrix} 4 & 0 & 8 & 0 & 2 & 7 & 7 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 6 & 0 & 3 & 0 & 9 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 4. Solution representation for the given example

It should be noticed that the population of parents to apply crossover and mutation operators is selected by binary tournament operator. We used uniform crossover and mutation in the proposed algorithm. Finally, the algorithm will stop after reaching a number of predefined iteration.

Non-dominate Ranked Genetic Algorithm (NRGA)

Researchers are always seeking for introducing better operators for multi-objective evolutionary algorithms to improve their efficiency. Among those operators, the focus is on the selection operator. Improving this operator result in better convergence of the algorithm. Al-Jadaan et al. [21] applied NRGA for solving multi-objective optimization problems. They developed the algorithm by combining ranked based roulette wheel selection operator and Pareto-based population ranking algorithm. The structure of the chromosome, crossover and mutation operators is the same as that of considered for NSGA-II. Fig. 5 shows the flowchart of NRGA and NSGA-II.

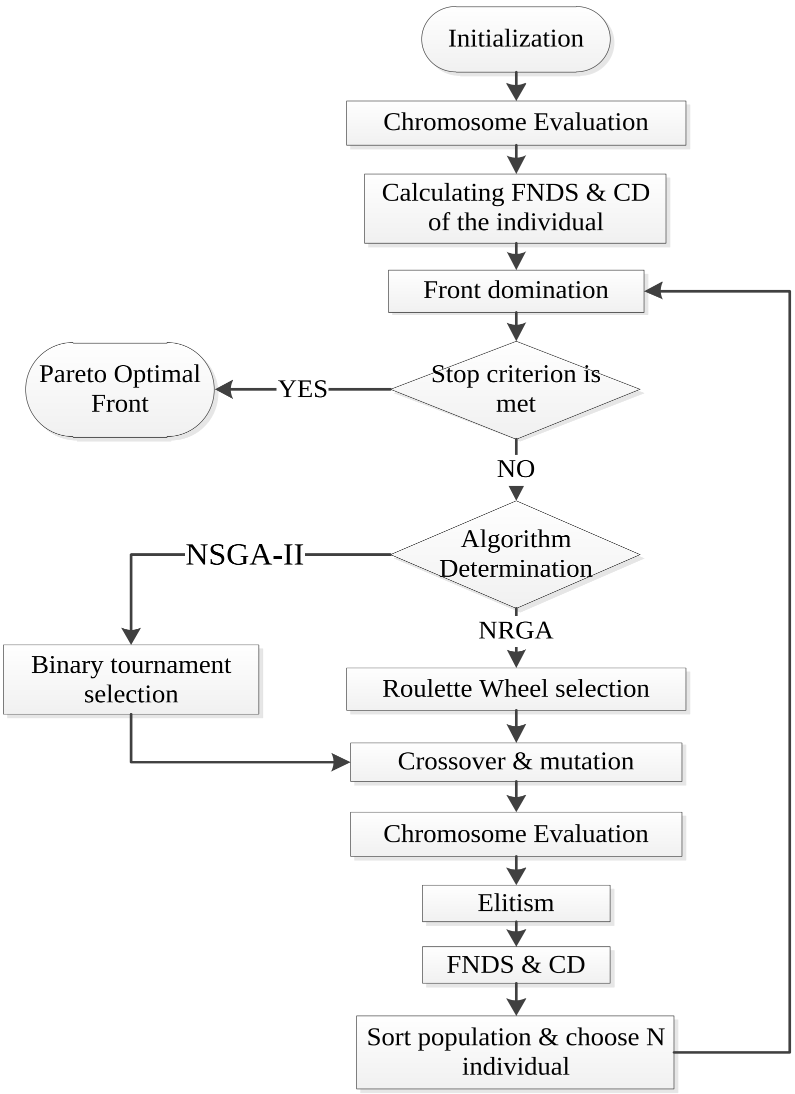

Fig. 5. Simultaneous Flowchart of NRGA and NSGA-II

Multi Objective Particle Swarm Optimization (MOPSO)

One of the best multi-objective evolutionary optimization algorithms is the MOPSO algorithm which was introduced by Coello Coello et al. [22]. In this method, individual memory is allocated to each particle to save the best position achieved in the searching process. Each particle corresponds to a point in the solution space. Particle movement is done in two directions: (1) move toward the best position captured so far by itself and (2) move toward the best position captured so far. In other words, the position changing of each particle is affected by itself and its neighbor experience. Fig. 6 indicates the flowchart of MOPSO.

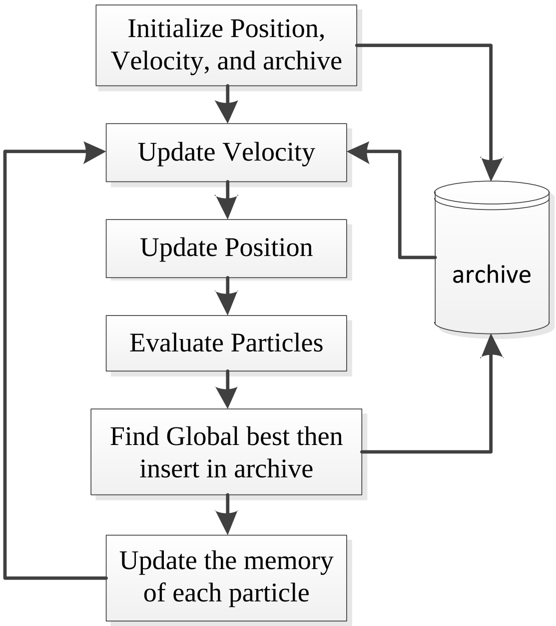

Fig. 6. Flowchart of MOPSO [23]

The major parameters of the MOPSO algorithm are as follow:

W: is an inertia weight that is applied to assure convergence in particle movement.

C_1, C_2: are individual and social parameters, respectively which determine how close the particle is to the best solution a particle has achieved so far (*pbest*) and the global best solution of all particles (*gbest*) (it is assumed that $C_1 + C_2 \leq 4$).

Furthermore, *nPop* is considered as the size of initializing population; nIt as the number of iterations and *nRep* as the size of the repository population.

Numerical Analysis

Initially, we give descriptions of parameters tuning for the given algorithms and then design some numerical examples to assess the performance of the model.

Parameters tuning

In NSGA-II and NRGA, there are four parameters to be tuned. We consider three levels for them. In MOPSO, six parameters are tuned considering three levels. We have done the tuning using the Taguchi method. The parameters and different levels are given as in Table 1.

The Taguchi experiment tables to determine optimum input parameters of the algorithms are as Tables 2 and 3. We have considered mean ideal distance (MID) criterion to compare the result [24].

Table 1. NSGA-II, NRGA and MOPSO parameters levels

Heuristics	Parameters	Low	Medium	High
NSGA-II	P_c	0.5	0.7	0.9
	P_m	0.01	0.2	0.4
	nIt	100	200	300
	nPop	100	150	200
NRGA	P_c	0.5	0.7	0.9
	P_m	0.01	0.2	0.4
	nIt	100	200	300
	nPop	100	150	200
MOPSO	*W*	0.4	0.65	0.9
	C_1	1	1.5	2
	C_2	1.5	2	2.5
	nIt	100	200	300
	nPop	100	150	200
	nRep	30	40	50

Table 2. The computation results of MID criterion when implementing NSGA-II and NRGA algorithms

Run No.	NSGA-II & NRGA Parameters				MID Measure	
	P_c	P_m	*nIt*	*nPop*	NSGA-II	NRGA
1	0.5	0.01	100	100	19570	19565
2	0.5	0.2	200	150	19410	19478
3	0.5	0.4	300	200	19410	19503
4	0.7	0.01	200	200	19385	19374
5	0.7	0.2	300	100	19819	19616
6	0.7	0.4	100	150	19483	19400
7	0.9	0.01	300	150	19684	19695
8	0.9	0.2	100	200	19423	19337
9	0.9	0.4	200	100	19934	19611

Table 3. The computation results of MID criterion when in implementing MOPSO algorithm

Run No.	MOPSO Parameters						MID Measure
	W	C_1	C_2	*nIt*	*nPop*	*nRep*	MOPSO
1	0.4	1	1.5	100	100	30	20503
2	0.4	1	1.5	100	150	40	20081
3	0.4	1	1.5	100	200	50	20199
4	0.4	1.5	2	200	100	30	20162
5	0.4	1.5	2	200	150	40	19874
6	0.4	1.5	2	200	200	50	19756
7	0.4	2	2.5	300	100	30	20531
8	0.4	2	2.5	300	150	40	20249
9	0.4	2	2.5	300	200	50	19966
10	0.65	1.5	1.5	300	100	40	19592
11	0.65	1.5	1.5	300	150	50	19518
12	0.65	1.5	1.5	300	200	30	20226
13	0.65	2	2	100	100	40	20029
14	0.65	2	2	100	150	50	19872
15	0.65	2	2	100	200	30	20291
16	0.65	1	2.5	200	100	40	19936
17	0.65	1	2.5	200	150	50	19676
18	0.65	1	2.5	200	200	30	20686
19	0.9	2	1.5	200	100	50	19577
20	0.9	2	1.5	200	150	30	20201
21	0.9	2	1.5	200	200	40	19669
22	0.9	1	2	300	100	50	19645

23	0.9	1	2	300	150	30	20292
24	0.9	1	2	300	200	40	20396
25	0.9	1.5	2.5	100	100	50	19893
26	0.9	1.5	2.5	100	150	30	20715
27	0.9	1.5	2.5	100	200	40	20051

The parameters obtained from the Taguchi method for each algorithm are shown as in Figs. 7-9.

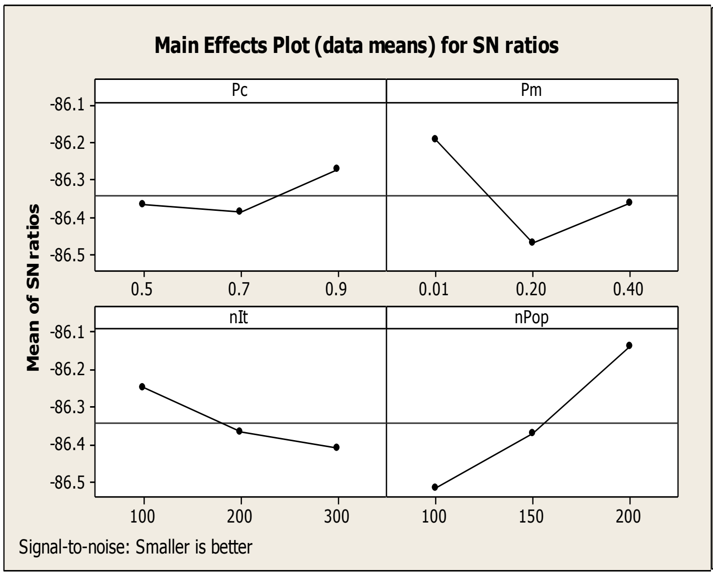

Fig. 7. Result of parameter tuning for NSGA-II

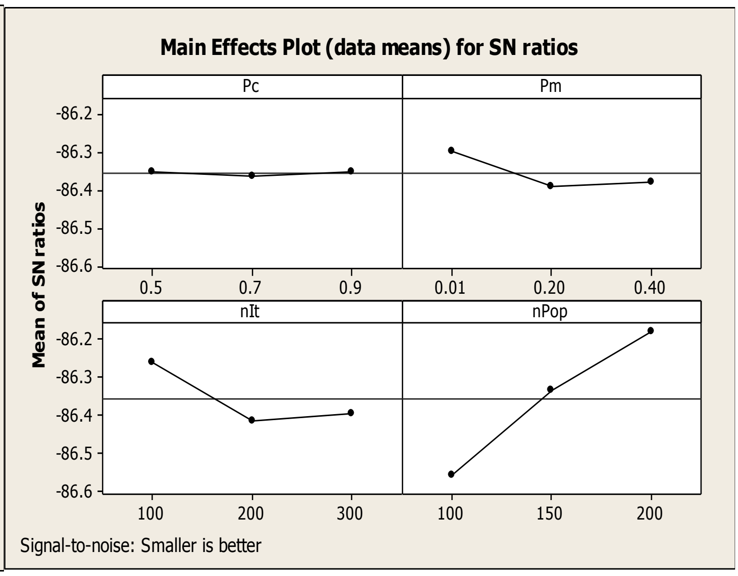

Fig. 8. Result of parameter tuning for NRGA

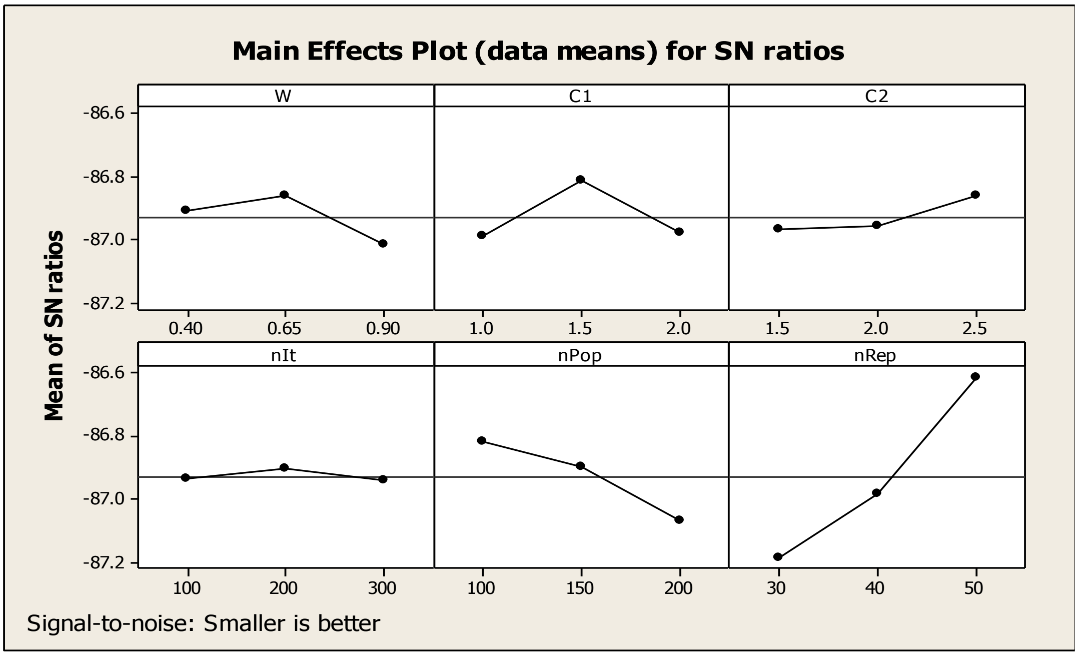

Fig. 9. Result of parameter tuning for MOPSO

The final values of the parameters of the algorithms are given as in Tables 4 and 5.

Table 4. The values of parameters for NSGA-II and NRGA

Parameters	*Pc*	*Pm*	*nIt*	*nPop*
NSGA-II	0.5	0.01	100	200

NRGA	0.7	0.2	100	200

Table 5. The values of parameters for MOPSO

Parameters	W	C_1	C_2	*nIt*	*nPop*	*nRep*
MOPSO	0.65	1.5	1.5	200	100	50

Numerical examples

In this section, 15 numerical examples are designed to study the performances of the proposed heuristics. Some criteria are used to evaluate the results as follow:

- Diversity: This criterion indicates how many of the solutions in the obtained Pareto-optimal set are distributed in the solution space. The larger value is the better one.
- Spacing: This criterion indicates the degree of uniform distribution of the solutions in the solution space. The smaller value is the better one.
- Number of Pareto-optimal Solutions (NOS): This criterion indicates the number of non-dominated solutions in the obtained Pareto-optimal set. The larger value is the better one.
- Mean ideal distance (MID): This criterion measures the proximity degree to the real Pareto-optimal set. The smaller value is the better one.
- CPU Time: This criterion shows the computational time of the algorithm. The smaller value is the better one

All the values for different criteria are given for each numerical example. Tables 6-8 give the results. Note that all algorithms are coded in Matlab version7.11.0 (R2010b) software. All the computational results were performed on a Pentium 4 notebook with Core i7 CPU 2.2 GHz and 8 GB RAM.

Table 6. Result and performance of NSGA-II for different criteria

No.	P	N	M	Q	A	Diversity	Spacing	NOS	MID	Time
1	3	12	30	2	6	14024	127.4	200	20070	535.86
2	3	12	30	2	8	14187	130	200	25538	542.12
3	3	12	50	4	8	51523	489.3	200	30831	597.79
4	3	15	60	4	10	218580	2912.2	200	53245	645.08
5	4	20	80	4	15	260720	1045	200	78454	639.65
6	4	20	100	4	15	188740	1221.3	200	82169	755.93
7	4	25	120	4	15	332690	3406.9	143	131570	762.63
8	4	25	120	4	20	265580	4635.4	200	105950	747.68
9	4	35	150	5	25	386320	4031	200	159970	1127.25
10	5	50	250	5	25	268910	1780.2	180	152200	1241.41
11	5	50	250	5	30	258660	1928.4	118	160930	1263.34
12	5	60	300	5	30	209600	1647.4	142	135980	1271.67
13	5	60	300	5	35	280180	8788.1	155	156860	1272.7
14	5	80	500	5	40	178960	1100.8	115	155260	1779
15	5	80	500	5	50	81663	1293.6	83	169910	1802.87
SUM						3010337	34537	2536	1618937	14985

Table 7. Result and performance of NRGA for different criteria

No.	P	N	M	Q	A	Diversity	Spacing	NOS	MID	Time
1	3	12	30	2	6	12987	145.4	200	20108	376.39
2	3	12	30	2	8	11748	106.7	200	25228	394.86
3	3	12	50	4	8	61179	473.1	200	32216	416.28
4	3	15	60	4	10	179660	1204.9	200	35308	425.67
5	4	20	80	4	15	256740	1021	200	70603	424.22
6	4	20	100	4	15	257420	1192.5	200	79740	508.28
7	4	25	120	4	15	203160	1469.9	153	123270	528.28
8	4	25	120	4	20	221850	1364.2	200	96484	518.55
9	4	35	150	5	25	378790	4012.9	200	117760	729.75
10	5	50	250	5	25	313140	6112.3	117	159550	819.8

11	5	50	250	5	30	254560	1921.3	200	134410	836.3
12	5	60	300	5	30	224720	1589.7	163	130960	851.4
13	5	60	300	5	35	218600	5978.2	167	123510	869.7
14	5	80	500	5	40	174280	1285.4	153	179000	1209.75
15	5	80	500	5	50	65686	540.99	200	150980	1227.88
SUM						2834520	28418.5	2753	1479127	10137.1

Table 8. Result and performance of MOPSO for different criteria

No.	*P*	*N*	*M*	*Q*	*A*	Diversity	Spacing	NOS	MID	Time
1	3	12	30	2	6	11830	345.7	42	20914	98.71
2	3	12	30	2	8	15269	515.1	50	26661	116.13
3	3	12	50	4	8	52079	529.1	50	34632	206.16
4	3	15	60	4	10	179580	5048.3	50	83785	279.04
5	4	20	80	4	15	156400	2310.7	49	94693	423.06
6	4	20	100	4	15	223490	8305.6	49	101300	497.47
7	4	25	120	4	15	204280	5076	38	108030	476.29
8	4	25	120	4	20	214260	4149.2	50	115920	512.22
9	4	35	150	5	25	322340	15156	36	161070	712.95
10	5	50	250	5	25	335720	14690	36	177810	802.43
11	5	50	250	5	30	365960	9615	50	170630	836.3
12	5	60	300	5	30	219020	5095.7	43	162140	853.4
13	5	60	300	5	35	217320	22013	25	159660	860.7
14	5	80	500	5	40	165650	16558	14	166420	1109.75
15	5	80	500	5	50	94034	18174	9	180950	1127.88
SUM						2777232	127581.4	591	1764615	8912.49

As is clear from Tables 6-8, the following results can be found:

- Regarding NOS, NRGA outperforms NSGA-II.
- Regarding Diversity, NSGA-II outperforms NRGA and MOPSO.
- Regarding Spacing and MID, NRGA outperforms NSGA-II and MOPSO
- Regarding CPU Time, MOPSO outperforms the two others.

The comparisons of all criteria are depicted as in Figs. 10.a to 10.e.

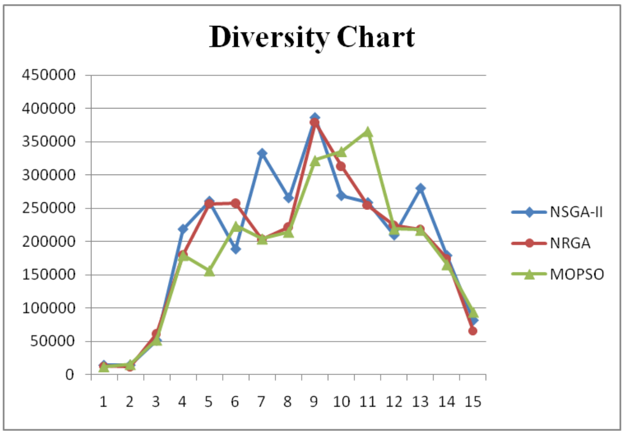

Fig. 10.a. Comparison of Diversity

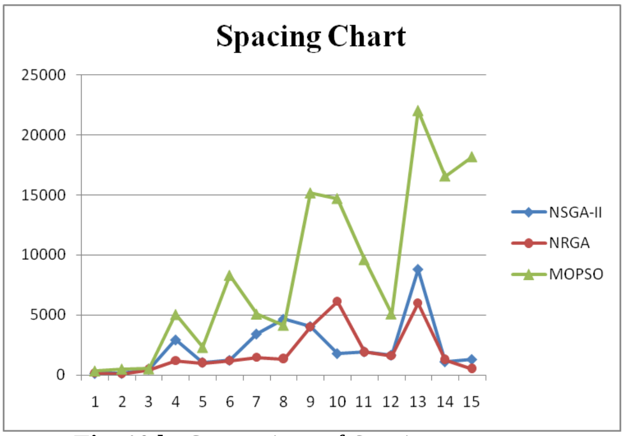

Fig. 10.b. Comparison of Spacing

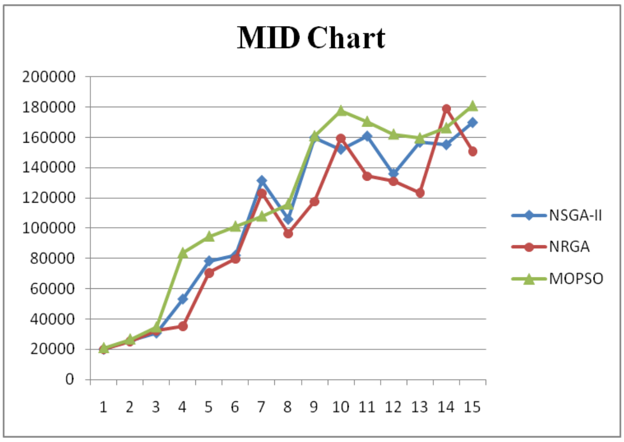

Fig. 10.c. Comparison of MID

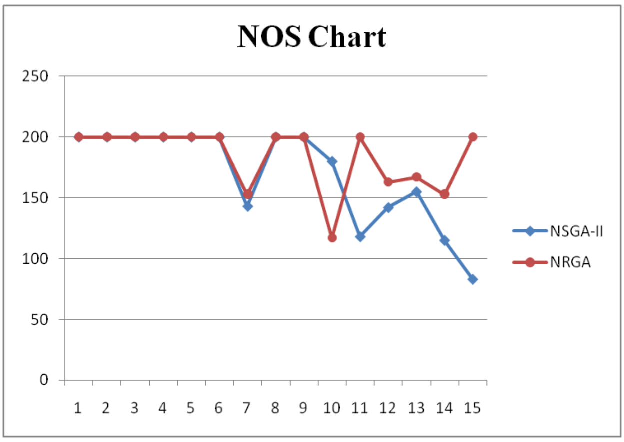

Fig. 10.d. Comparison of NOS

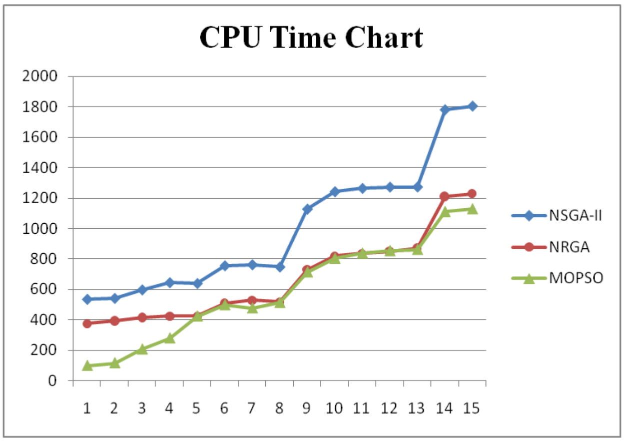

Fig. 10.e. Comparison of CPU time

To compare the performance of the given heuristics, we applied the analysis of variances (ANOVA) technique for different criteria of diversity, spacing, NOS, MID, and CPU time. For each criterion, if P-Value is less than 0.05, it means that there is a significant difference between the heuristics performances; otherwise, there are no significant differences between the performances. The ANOVA results are given in Table 9.

Table 9. ANOVA results for NSGA-II, NRGA and MOPSO

Response Variable	P-value	Test Result
Diversity	0.922	Null hypothesis is not rejected

Spacing	0.000	Null hypothesis is rejected
NOS	0.952	Null hypothesis is not rejected
MID	0.637	Null hypothesis is not rejected
CPU Time	0.008	Null hypothesis is rejected

As is clear from Table 9, there are only significant differences among the heuristics for Spacing and CPU Time criteria.

Conclusions, managerial insights and future research ideas

We developed a bi-objective model for a three-echelon multi-commodity supply chain including manufacturers, distribution centers (DCs) and customers which might be partially or fully covered by the DCs. The DCs were selected from among a number of candidate points. Furthermore, the flow of commodities in the whole supply chain considering a limited number of DCs was determined by minimizing the total operational costs and maximizing the customers' coverage. Since the presented problem was NP-hard in nature, three metaheuristic algorithms i.e. NSGA-II, NRGA and MOPSO were applied to find the Pareto-optimal solutions. Numerical examples were designed to assess the performance of the model and the developed metaheuristic algorithms. Five different criteria were measured in order to compare the performances of the algorithms. There were only significant differences among the heuristics for Spacing and CPU Time criteria.

From a managerial point of view, this research shows the efficiency and application of partial coverage instead of full coverage which can be costly for some situations in supply chain network design. A supply chain manager should have in mind that it is not mandatory to fully cover customers' demands with higher costs. Furthermore, the managers can learn that in making coverage decisions, they should also have a look at the associated costs with such decisions. For this purpose, the given model suggests two objectives of maximal coverage and minimal cost.

As further research ideas, we can consider the DCs to be capacitated in order to make the problem closer to real-world conditions. The locations of the manufacturers can be determined by the model. The presented model in this research can be combined with the inventory control and routing problems in DCs and from DCs to customers. Furthermore, developing faster heuristic solutions based on Lagrangian relaxation or Benders' decomposition algorithm can be suggested.

References

[1] Current, J.R., Daskin, M.S., Schilling, D. (2001). Discrete network location models, In book: Facility Location; Applications and theory. Editors: Z. Drezner and H. Hamacher, Springer, Heidelberg, 80-118.
[2] Warszawski, A., Peer, S. (1973). Optimizing the location of facilities on a building site. Journal of the Operational Research Society, 24, 35–44.
[3] Syam S.S. (2002). A model and methodologies for the location problem with logistical components. Computers and Operations Research, 29, 1173–1193.
[4] Church, R., ReVelle, C. (1974). The maximal covering location problem. Papers of the Regional Science Association, 32, 101–18.
[5] Schilling, D., Jayaraman, V., Barkhi, R. (1993). A review of covering problems in facility location. Location Science, 1, 25-55.
[6] Galvao, R.D., ReVelle. C. (1996). A Lagrangean heuristic for the maximal covering location problem. European Journal of Operational Research, 88, 114–23.
[7] Gendreau, M., Laporte, G., Semet, F. (2006). The maximal expected coverage relocation problem for emergency vehicles. Journal of Operational Research Society, 57, 22–28.

[8] ReVelle, C. (2008). Solving the maximal covering location problem with heuristic Concentration. Computers & Operations Research, 35, 427 – 435.

[9] Karasakal, O. (2004). A maximal covering location model in the presence of partial coverage. Computers & Operations Research, 31, 1515–1526.

[10] Mestre, J. (2008). Lagrangian relaxation and partial cover. In: Proceedings of the 25th Annual Symposium on Theoretical Aspects of Computer Science, 539–550.

[11] Melo, M.T., Nickel, S., Saldanha-da-Gama, F. (2009). Facility location and supply chain management a review. European Journal of Operational Research, 196, 401-412.

[12] Pereira, M.A., Coelho, L.C., Lorena, L.A.N., De Souza, L.C. (2015). A hybrid method for the probabilistic maximal covering location-allocation problem. Computers & Operations Research, 57, 51-59.

[13] Seifbarghy, M., Soleimani, M., Pishva, D. (2016). Multiple commodity supply chain with maximal covering approach in a three layer structure. International Journal of Mathematical Modeling and Numerical Optimization, 7 (2), 138-161.

[14] Li, X., Ramshani, M., Huang, Y. (2018). Cooperative maximal covering models for humanitarian relief chain management. Computers & Industrial Engineering, 119, 301-308.

[15] Eidy, A., Torabi, H. (2019). A maximal covering problem in supply chain considering variable radius and gradual coverage with the choice of transportation mode. Arabian Journal for science and Engineering, 44, 7219-7233.

[16] Vatsa, A.K., Jayaswal, S. (2021). Capacitated multi-period maximal covering location problem with server uncertainty. European Journal of Operational Research, 289 (3), 1107-1126.

[17] Karasakal, E., Silav, A. (2016). A multi-objective genetic algorithm for a bi-objective facility location problem with partial coverage. TOP, 24, 206-232.

[18] Cordeau, J-F., Furini, F., Ljubic, I. (2019). Benders decomposition for very large scale partial set covering and maximal covering location problems. European Journal of Operational Research, 275(3), 882-896.

[19] El-Hosseini, M., ZainEldin, H., Arafat, H., Badawy, M. (2021). A fire detection model based on power-aware scheduling for IoT-sensors in smart cities with partial coverage. Journal of Ambient Intelligence and Humanized Computing, 12, 2629–2648.

[20] Deb, K., Agrawal, S., Pratap, A., Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi- Objective optimization: NSGA-II. In: Proceedings of the parallel problem solving from nature VI (PPSN-VI) conference, 849-858.

[21] Al jadaan, O., Rao, C.R., Rajamani, L. (2008). Non-dominated ranked genetic algorithm for solving multi-objective optimization problems: NRGA. Journal of Theoretical and Applied Information Technology, 60-67.

[22] Coello Coello, C., Lamont, G.B., Van Veldhuizen, D.A. (2007). Evolutionary algorithms for solving multi-objective problem. 2nd end, Springer, Berlin.

[23] Sharaf, A.M., Elgammal, A. (2009). A multi objective multistage particle swarm optimization MOPSO search scheme for power quality and loss reduction on radial distribution system. In Proceedings of the International Conference on Renewable Energies and Power Quality (ICREPQ'09).

[24] Zitzler, E., Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms- a comparative case study. Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V), Berlin, Germany, 292-301.