

DESIGNING A NEW MEDICINE SUPPLY CHAIN NETWORK CONSIDERING PRODUCTION TECHNOLOGY POLICY USING TWO NOVEL HEURISTIC ALGORITHMS

FARIBA GOODARZIAN^{1,2,3,*}, HASSAN HOSEINI-NASAB³, MEHDI TOLOO⁴
AND MOHAMMAD BAGHER FAKHRZAD³

Abstract. The role of medicines in health systems is increasing day by day. The medicine supply chain is a part of the health system that if not properly addressed, the concept of health in that community is unlikely to experience significant growth. To fill gaps and available challenging in the medicine supply chain network (MSCN), in the present paper, efforts have been made to propose a location-production-distribution-transportation-inventory holding problem for a multi-echelon multi-product multi-period bi-objective MSCN network under production technology policy. To design the network, a mixed-integer linear programming (MILP) model capable of minimizing the total costs of the network and the total time the transportation is developed. As the developed model was NP-hard, several meta-heuristic algorithms are used and two heuristic algorithms, namely, Improved Ant Colony Optimization (IACO) and Improved Harmony Search (IHS) algorithms are developed to solve the MSCN model in different problems. Then, some experiments were designed and solved by an optimization solver called GAMS (CPLEX) and the presented algorithms to validate the model and effectiveness of the presented algorithms. Comparison of the provided results by the presented algorithms and the exact solution is indicative of the high-quality efficiency and performance of the proposed algorithm to find a near-optimal solution within reasonable computational time. Hence, the results are compared with commercial solvers (GAMS) with the suggested algorithms in the small-sized problems and then the results of the proposed meta-heuristic algorithms with the heuristic methods are compared with each other in the large-sized problems. To tune and control the parameters of the proposed algorithms, the Taguchi method is utilized. To validate the proposed algorithms and the MSCN model, assessment metrics are used and a few sensitivity analyses are stated, respectively. The results demonstrate the high quality of the proposed IACO algorithm.

Received February 4, 2020. Accepted February 28, 2021.

Keywords. Medicine supply chain network, production-distribution problem, heuristic algorithms, Taguchi method.

¹ Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, 11, 3rd Street NW, P.O. Box 2259, Auburn, WA 98071, USA.

² School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran.

³ Department of Industrial Engineering, Yazd University, Yazd, Iran.

⁴ Republic Department of System Engineering, Faculty of Economics, Technical University of Ostrava, Ostrava, Czech Republic.

⁵ Department of Operations Management & Business Statistics, College of Economics and Political Science, Sultan Qaboos University, Muscat, Oman.

*Corresponding author: fariba.goodarzian@mirlabs.org

1. INTRODUCTION

Medicines are very important in each medical and health care systems (*e.g.* hospitals, pharmacies, and laboratories). Then, each risk/crisis (shortage of medicines) can affect the whole medicine supply chain instantaneously. This problem can not only waste the resources but can also jeopardize patients' lives by making trouble for access to medical products [22, 39].

The World Health Organization (WHO) defines a medicine preparation as: "any substance or mixture of substances manufactured, sold, offered for sale or represented for use in the diagnosis, treatment, mitigation or prevention of disease, abnormal physical state or the symptoms thereof in man or animal; restoring, correcting or modifying organic functions in man or animal" and it mentioned access to medicine is the one of important human right that must be the main objective of any healthcare systems [3, 44].

The field of health and treatment is one of the most important fields in any country and the supply chain of this field is part of the community health system and has a strategic position [39, 44]. Iran's pharmaceutical industry has gone through various and volatile conditions over the past few decades. In the years after the revolution, this industry has had significant quantitative and qualitative growth. Numerous factors such as profitability, job creation, the existence of a suitable domestic and foreign market, communication with other industries, etc. have made the position of this industry in comparison with other industries privileged and strategic. Commodity investments and government support have also increased the importance of this industry in each country. Medicine use is high in Iran and in this respect, it is the second-largest consumer in Asia and the tenth-largest in the world [5]. The need to pay attention to the MSCN can be traced to the change in the behavior of managers in terms of supply chain optimization. The MSCN must deliver medicines to customers with acceptable quality, in the right quantity, in the right place, and at the right time at the suitable price, in a way that is compatible with the goals of the health system and also beneficial to the beneficiaries. Because supply chain costs have a direct impact on the price of pharmaceutical items. This field must be able to cover the pharmaceutical needs of the society with the highest speed and accuracy, and for this purpose, the integration of the MSCN seems to be essential [5, 38, 39, 43, 44].

Challenges and various methodologies in MSCNs widely have been addressed in the literature on managerial subjects in the medicine industry [5, 9, 26, 29, 38, 43]. Rotstein *et al.* [35] developed an optimization-based method for choosing product development strategies and also capacity planning and investment strategy that the demand has related to the clinical results for a pharmaceutical supply chain. Papageorgiou *et al.* [34] formulated an MILP model in a pharmaceutical system for the optimization of product strategies and also an investment strategy and capacity planning. Gatica *et al.* [14] extended an MILP mathematical model in a multi-stage, multi-period stochastic pharmaceutical inventory system problem under uncertainty. Levis and Papageorgiou [33] suggested an MILP model in a long-term, multi-site capacity planning pharmaceutical firm under uncertainty by considering the trading structure of the company. Amaro and Barbosa-Póvoa [2] presented a new mathematical model for the sequential planning and scheduling problem in the reverse supply chain and employed it for a real pharmaceutical case study. Sousa *et al.* [40] offered a planning and allocation problem for the pharmaceutical supply chain purposing to increase the net profit value. Moreover, two decomposition algorithms were used to solve the problem. Susarla and Karimi [41] formulated a MILP model for multi-period planning in an integrated procurement-production-distribution pharmaceutical company. Jetly *et al.* [23] presented a multi-agent mathematical model in the pharmaceutical supply chains. Alnaji and Ridha [1] described the supply chain management role applications in the pharmaceutical industry. Mousazadeh *et al.* [32] addressed a robust possibilistic programming method for the pharmaceutical supply chain network. Furthermore, they formulated a bi-objective MILP mathematical model. Zahiri *et al.* [47] designed a novel analytical model for the pharmaceutical supply chain network under uncertainty with perishability and substitutable products. Goodarzian *et al.* [16] presented a new multi-objective multi-echelon multi-product multi-period pharmaceutical supply chain network along with production-distribution-purchasing-ordering-inventory holding-allocation-routing

problem under uncertainty. Also, an MINLP model is formulated and a novel robust fuzzy programming approach is developed. To solve the model, multi-objective meta-heuristic algorithms are used. Goodarzian *et al.* [17] developed a multi-objective sustainable MILP model in the MSCN problem considering three aims including economic, environmental, and social impacts. Additionally, a new hybrid meta-heuristic algorithm is provided.

In order to fill the aforementioned gaps and to get closer to real-world applications, this work develops and solves a new multi-level bi-objective multi-period multi-product mathematical model namely Medicine Supply Chain Network (MSCN). The considered network contains three levels: manufacturing centers, main and local distribution centers, and final customers (hospitals, clinics, and pharmacies). In this regard, a new location, production, distribution, inventory holding, and transportation problem are developed to minimize the total cost by using several types of vehicles and the total time of the transportation in the planning horizon. According to the literature and to the best of our knowledge, the production technology policy has not been considered for the MSCN location, production, distribution, inventory holding, and transportation problem, which this technology uses in each distribution center. Another innovation, first of all, a new mixed-integer linear programming model (MILP) for an integrated location, production, distribution, inventory, and transportation problem in the MSCN is formulated. To solve the model and to find Pareto solutions, due to NP-hard and complexity of the model [17, 45], four multi-objective meta-heuristic algorithms including simulated annealing (SA), variable neighborhood search (VNS), harmony search (HS), and ant colony optimization (ACO) are utilized. In this regard, the second contribution of this paper, two novel Improved Ant Colony Optimization (IACO) and Improved Harmony Search (IHS) algorithms are developed to solve the MSCN problem. The third innovation in this paper, in order to more performance and efficiency of the IACO algorithm in the local search, the three important parameters local searches containing dual enhancer, swap move, and insert move are employed. The Taguchi method is used to control and tune the parameters of the meta-heuristic and heuristic algorithms. Additionally, to evaluate the proposed algorithms, three assessment metrics including the number of Pareto solutions, mean ideal distance, and spread of non-dominance solution metrics are employed. Finally, a few sensitivity analyses are explained to validate the MSCN model in this paper.

The rest of this paper is organized as follows. Section 2 describes the addressed MSCN design problem and devotes it to the model formulation. In Section 3, four meta-heuristic algorithms and two new heuristic algorithms are provided. Section 4 reports computational experiments and results. Section 5 illustrates some managerial insights into the MSCN problem. Eventually, concludes this paper with important findings as well as directions and recommendations for future work in Section 6.

2. MATHEMATICAL MODELLING

2.1. Problem description

In this section, a new four-level, multi-period, and multi-product MSCN are considered in Figure 1 that contains manufacturing centers, main distribution centers, local distribution centers, and final customers (hospitals, clinics, and pharmacies) with potential lateral transmission flows between local distribution centers. Geographically, there are many potential locations to build manufacturing centers and main and local distribution centers but decision-makers must according to the different criteria, and also a special focus on the requirements of final customers choose the best location or locations for each facility. Besides, among the available technologies to produce each product family, only one technology per potential manufacturing center can be used that directly affects the production capacity at the per time period, however, the double construction cost is imposed on the whole system. Inventory can be preserved at the main and local distribution centers at the end of each period, while its level has limited based on the storage capacity of each center. However, the capacity levels of the main and local distribution centers are considered as the decision variables of the model. Also, in order to control the essential requirements for a particular medicine at per time period, a level of confidence inventory

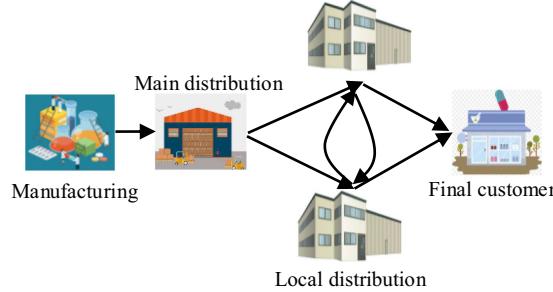


FIGURE 1. The structure of MSCN problem.

is assumed for local distribution centers. Demand for each customer zone can only be fulfilled by transferring products from local distribution centers. Moreover, different modes of transportation systems are available for transporting products between various levels or amongst local distribution centers. Finally, the important and significant aims in this problem including minimizing the fixed cost of construction of centers (manufacturing, main and local distribution), production costs, transportation costs, inventory holding cost in main and local distribution centers, and the total transportation time along the planning horizon.

2.2. Model assumption

- Final customers are considered as multi-types system, but they are treated as one because only the whole demand is important.
- There are no bottleneck sources in the presented model.
- There is no unmet demand in the system meaning that there is no shortage in the overall network.
- According to the transportation type of medical products in Iran, only one type of (ground) transportation is considered to transport the manufactured products between different levels of the network.

2.3. Notation

Indices

m Index for The candidate locations to establish manufacturing centers $m \in \{1, 2, \dots, M\}$.
 i Index for The candidate locations to establish main distribution centers $i \in \{1, 2, \dots, I\}$.
 j Index for The candidate locations to establish local distribution centers $j \in \{1, 2, \dots, J\}$.
 c Index for final customer $c \in \{1, 2, \dots, C\}$.
 v Index for the vehicle of types $v \in \{1, 2, \dots, V\}$.
 q_p The potential Construction technology for producing the family product $q_p \in \{1, 2, \dots, Q_p\}$.
 n Index for capacity levels possible for main distribution centers $n \in \{1, 2, \dots, N\}$.
 p Index for product families $p \in \{1, 2, \dots, P\}$.
 t Index for time period $t \in \{1, 2, \dots, T\}$.

Parameters

φ_{pij}^v Transportation cost per unit product p from main DC i to local DC j with the vehicle type v .
 φ_{pmi}^v Transportation cost per unit product p from manufacturing m to main DC i with the vehicle type v .
 d_{cp}^t Final customer demand c for product p at period t .

φ_{pjc}^v	Transportation cost per unit product p from local DC j to final customer c with the vehicle type v .
$\beta_{pq,m}$	Construction cost for the product production p with construction technology q_p in the manufacturing center m .
β_{ni}	Construction cost with capacity level n in the main DC i .
β_j	Construction cost in the local DC j .
μ_{pi}^t	Storage cost of product p in the main DC i at the end of the period t .
μ_{pj}^t	Storage cost of product p in the local DC j at the end of the period t .
I_{pj}^t	Inventory of product p in the local DC j at the end of the period t .
$\varphi_{pj,j'}^v$	Transportation cost per unit product p from local DC j to local DC j' with the vehicle type v .
γ_{in}	Storage capacity of main DC i with capacity level n .
α_i	Available storage capacity in the main DC i .
θ_p	The relative importance of the product p .
β_{pmq_p}	Construction cost of the product p at the manufacturing center m with construction technology q_p .
ψ_{pmi}^v	Transportation time of per unit product p from manufacturing center m to main DC i with the vehicle type v .
ψ_{pij}^v	Transportation time of per unit product p from main DC i to local DC j with the vehicle type v .
$\psi_{pj,j'}^v$	Transportation time of per unit product p from local DC j to local DC j' with the vehicle type v .
ψ_{pjc}^v	Transportation time of per unit product p from local DC j to final customer c with the vehicle type v .
M	The big number.

Decision variables

η_{pi}^t	Inventory level of product p in the main DC i at the end of the period t .
η_{pj}^t	Inventory level of product p in the local DC j at the end of the period t .
Ω_{pjc}^{tv}	The quantity of shipped product p from local DC j to final customer c at the end of the period t with the vehicle type v .
Ω_{pij}^{tv}	The quantity of shipped product p from main DC i to local DC j at the end of the period t with the vehicle type v .
$\Omega_{pmq_p}^{tv}$	The quantity of produced product p in the manufacturing center m with construction technology q_p and shipped to the main DC i with the vehicle type v at period t .
$\Omega_{pj,j'}^{tv}$	The quantity of shipped product p from local DC j to local DC j' at the end of the period t with the vehicle type v .
ϖ_{mpq_p}	If manufacturing center m for producing of product p with the use of construction technology q_p is constructed 1; otherwise 0.
δ_{in}	If potential main DC i with capacity level n is constructed 1; otherwise 0.
δ_j	If potential local DC j is constructed 1; otherwise 0.
π_{pmi}^v	If product p from manufacturing center m to main DC i with the vehicle type v is transported 1; otherwise 0.
π_{pij}^v	If product p from main DC i to local DC j with the vehicle type v is transported 1; otherwise 0.
$\pi_{pj,j'}^v$	If product p from local DC j to local DC j' with the vehicle type v is transported 1; otherwise 0.
π_{pjc}^v	If product p from local DC j to final customer c with the vehicle type v is transported 1; otherwise 0.

2.4. Mathematical modeling

$$\begin{aligned}
 \text{Min } of_1 = & \sum_p \sum_m \sum_{q_p} \beta_{pq,m} \varpi_{mpq_p} + \sum_i \sum_n \beta_{ni} \delta_{in} + \sum_j \beta_j \delta_j + \sum_p \sum_m \sum_{q_p} \sum_i \sum_v \sum_t (\beta_{pmq_p} + \phi_{pmi}^v) \\
 & \times \Omega_{pmq_p}^{tv} + \sum_p \sum_i \sum_j \sum_t \sum_v \phi_{pij}^v \Omega_{pij}^{tv} + \sum_p \sum_j \sum_{j'} \sum_t \sum_v \phi_{pj,j'}^v \Omega_{pj,j'}^{tv} \sum_p \sum_j \sum_c \sum_t \sum_v \phi_{pjc}^v \Omega_{pjc}^{tv}
 \end{aligned}$$

$$+ \sum_p \sum_i \sum_t \mu_{pi}^t \eta_{pi}^t + \sum_p \sum_j \sum_t \mu_{pj}^t \eta_{pj}^t \quad (2.1)$$

$$\begin{aligned} \text{Min of } f_2 = \min \sum_p \sum_m \sum_i \sum_j \sum_{j'} \sum_c \sum_v & (\psi_{pmi}^v \pi_{pmi}^v + \psi_{pij}^v \pi_{pij}^v + \psi_{pj'j}^v \pi_{pj'j}^v), (\psi_{pmi}^v \pi_{pmi}^v + \psi_{pij}^v \pi_{pij}^v \\ & + \psi_{pj'j}^v \pi_{pj'j}^v + \psi_{pjc}^v \pi_{pjc}^v) \end{aligned} \quad (2.2)$$

$$\eta_{pi}^t = \eta_{pi}^{t-1} + \sum_m \sum_{q_p} \sum_v \Omega_{pmq_p i}^{tv} - \sum_j \sum_v \Omega_{pij}^{tv}, \quad \forall i, p, t \quad (2.3)$$

$$\eta_{pj}^t = \eta_{pj}^{t-1} + \sum_i \sum_v \Omega_{pij}^{tv} - \sum_i \sum_v \Omega_{pjc}^{tv} + \sum_{j'} \sum_v \Omega_{pj'j}^{tv} - \sum_{j'} \sum_v \Omega_{pj'j}^{tv}, \quad \forall j, p, t \quad (2.4)$$

$$\sum_i \sum_{q_p} \sum_v \Omega_{pmq_p i}^{tv} \leq \varpi_{mpq_p} \beta_{mpq_p}, \quad \forall p, m, t \quad (2.5)$$

$$\sum_p \left(\eta_{pi}^{t-1} + \sum_m \sum_{q_p} \sum_v \Omega_{pmq_p i}^{tv} \right) \leq \sum_n \delta_{in} \gamma_{in}, \quad \forall i, t \quad (2.6)$$

$$\sum_p \left(\eta_{pj}^{t-1} + \sum_i \sum_v \Omega_{pij}^{tv} + \sum_{j'} \sum_v \Omega_{pj'j}^{tv} \right) \leq \alpha_i \delta_j, \quad \forall j, t \quad (2.7)$$

$$\eta_{pj}^t \geq I_{pj}^t \delta_j, \quad \forall j, p, t \quad (2.8)$$

$$\sum_{q_p} \varpi_{mpq_p} \leq 1, \quad \forall m, p \quad (2.9)$$

$$\sum_n \delta_{in} \leq 1, \quad \forall i \quad (2.10)$$

$$\Omega_{pmq_p i}^{tv} \leq \pi_{pmi}^v M, \quad \forall p, m, i, q_p, t, v \quad (2.11)$$

$$\Omega_{pij}^{tv} \leq \pi_{pij}^v M, \quad \forall p, j, j, t, v \quad (2.12)$$

$$\Omega_{pj'j}^{tv} \leq \pi_{pj'j}^v M, \quad \forall p, j, j', t, v \quad (2.13)$$

$$\Omega_{pjc}^{tv} \leq \pi_{pjc}^v M, \quad \forall p, j, c, t, v \quad (2.14)$$

$$\sum_i \pi_{pmi}^v \geq 1, \quad \forall m, p, v \quad (2.15)$$

$$\sum_j \pi_{pij}^v \geq 1, \quad \forall p, i, v \quad (2.16)$$

$$\sum_c \pi_{pjc}^v \geq 1, \quad \forall p, j, v \quad (2.17)$$

$$d_{cp}^t = \sum_j \Omega_{pjc}^{tv}, \quad \forall c, p, t, v \quad (2.18)$$

$$\varpi_{mpq_p}, \delta_{in}, \delta_j, \pi_{pmi}^v, \pi_{pij}^v, \pi_{pj'j}^v, \pi_{pjc}^v \in \{0, 1\}, \quad \forall m, i, j, j', c, p, q_p, n, v \quad (2.19)$$

$$\Omega_{pmq_p i}^{tv}, \Omega_{pij}^{tv}, \Omega_{pjc}^{tv}, \Omega_{pj'j}^{tv}, \eta_{pi}^t, \eta_{pj}^t \geq 0. \quad (2.20)$$

The first objective function (2.1) asserts to minimize the total cost including the fixed cost of construction of centers (manufacturing, main and local distribution), production costs, transportation costs, inventory holding cost in main and local distribution centers. The second objective function (2.2) seeks to minimize the total transportation time during the planning horizon.

Constraints (2.3) and (2.4) indicate the flow balance of each product at the main and local distribution centers, respectively. Constraint (2.5) ensures that the total manufacturing level of a product at each constructed manufacturing center at each period does not exceed the corresponding manufacturing capacity based on the construction technology selected for that product. The storage capacity constraints for main and local distribution centers are provided in constraints (2.6) and (2.7), respectively. In other words, the sum of the remaining products from the previous period plus the total of newly received products in the main and local distribution centers in each period cannot exceed the relevant storage capacity. Constraint (2.8) guarantees that the inventory level of each product at each period and at each local distribution center is equal to or greater than the level of confidence inventory previously allocated to it. This constraint partly protects the network against unexpected demands. The inventory confidence level can be estimated as part of the total environmental demand of customers for each product at each time period using predetermined service levels. It should also be noted that local distribution centers serve customers directly, while they receive from the main distribution centers. Moreover, each local distribution center will serve final customers that are adjacent to it, while one or more local distribution centers may be in accordance with the optimal allocation pattern of the main and local distribution centers that are determined to serve by each constructed distribution center. Hence, to estimate a reasonable value for the confidence level of local distribution centers is somewhat possible, but it is difficult or even impossible to estimate this value for the original distribution centers. Therefore, the predetermined service level only corresponds to the confidence level in local distribution centers. Also, the inventory level of each main distribution center is not limited by any level of confidence inventory and be able to be freely determined by the model. According to the potential transmission flows between local distribution centers to overcome the demand fluctuations, it is a significant reason for not considering the level of confidence inventory for the main distribution centers. Constraint (2.9) shows that a maximum of one construction technology is used to produce each product at each constructed manufacturing center. Similarly, to select a maximum of one capacity level for each main distribution center is indicated in constraint (2.10). Constraints (2.11)–(2.14) indicate that the number of shipped products at different levels of the chain must be lower than the binary variables. Constraints (2.15)–(2.17) guarantee that products must be moved between main levels of the chain including manufacturing centers, main and local distribution centers, and final customers to satisfy customer demand. Constraint (2.18) indicates that the flow balance at the end level of the chain, in which the amount of final customer demand in the network should be equal to the shipped amount from local distribution centers to final customers. Finally, constraints (2.19) and (2.20) state the type of decision variables.

2.5. Linearization

In this sub-section, since the second objective function is nonlinear with respect to the Min–Min structure, it is linearized by defining a free variable in the θ sign and is formulated as follows:

$$\text{Min } of_2 = \theta \quad (2.21)$$

$$\theta \geq (\psi_{pmi}^v \pi_{pmi}^v + \psi_{pij}^v \pi_{pij}^v + \psi_{pj}^v \pi_{pj}^v), \quad \forall m, i, j, c, v \quad (2.22)$$

$$\theta \geq (\psi_{pmi}^v \pi_{pmi}^v + \psi_{pij}^v \pi_{pij}^v + \psi_{pj}^v \pi_{pj}^v + \psi_{pjc}^v \pi_{pjc}^v), \quad \forall m, i, j, j', c, v. \quad (2.23)$$

3. SOLUTION METHODOLOGY

It has been proven that MSCN models are NP-hard [7, 12, 15–17, 27, 36, 37, 46]. Hence, the literature review has seen several meta-heuristic algorithms that were suggested to solve these NP-hard problems [16, 17]. Furthermore, the No Free Lunch (NFL) theory says that there is no meta-heuristic algorithm to indicate good efficiency for all optimization problems [45]. A number of NFL theorems are provided which prove that for any algorithm, any elevated efficiency over one category of problems is offset by efficiency over another category. The results of this theory in a geometric interpretation of what it means for an algorithm to well suited to an optimization problem and also show the significance of incorporating problem-specific knowledge into the behavior of the algorithm. The geometric perspective also presents a number of measures to evaluate the similarity of different optimization

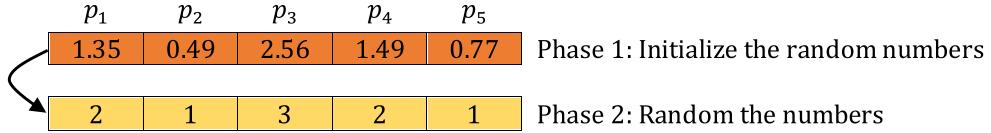


FIGURE 2. The utilized approach to shipping products by sort of transportation systems from manufacturer to main DC.

algorithms. Applications of the NFL theorems to benchmark measures of performance and information-theoretic aspects of optimization are also provided. In this paper, the CPLEX Solver by GAMS Software is used to solve the small-sized problem [11, 16], and also meta-heuristic algorithms are utilized to solve the large-sized problem. Therefore, another main novelty of this paper is to consider and to introduce four original and two improved heuristic algorithms (IACO and IHS) for the first time. In this regard, solution representation and algorithms' descriptions are reported. According to the two objective functions of the developed MSCN model, the proposed algorithms are examined by multi-objective optimization techniques detailed as follows.

3.1. Multi-objective optimization

The presented MSCN model has two objective functions. Then, based on the Pareto optimal set, the interactions between the solutions are observed, which this set contains the non-dominated solutions [16]. Moreover, two solutions are considered that include solutions of_1 and of_2 . Solution of_1 dominates Solution of_2 when all objectives of of_1 are not worse than of_2 and there exists at least one of_1 that is better than of_2 [16, 17]. Then, this paper uses three assessment metrics to evaluate the quality of Pareto fronts according to the Pareto optimal set like several recent studies [7, 11, 16, 17]. In the next section, the solution representation of suggested multi-objective meta-heuristic algorithms is provided.

3.2. Solution representation

In this sub-section, a plan should be designed to encode the problem in order to implement the meta-heuristic algorithms [16]. In this regard, a two-stage approach called Random-Key is employed [11, 16]. The use of random keys is useful for problems that require permutations of the integers. In the random key method, we assign each gene a random number uniformly. Then, we visit the nodes in ascending order of their genes to decode the chromosome. The technique is best illustrated with an example in Figure 2. Then, in two steps, this approach converts an unfeasible solution to a feasible one by a set of procedures [11]. Researchers have used this approach in several contents of engineering design in recent years *e.g.* Supply Chain Network Design [16] and Home Health Care Problem [11], etc. Moreover, firstly, this paper utilizes this approach in the literature review for the developed MSCN model. Hence, a numerical example to encode the solution representation is provided as follows. Five products (p), two manufacturers (m), two main DCs (i), and three vehicles (v) in this example are considered. Firstly, the sort of utilized transportation system for shipping products should be shown. Then, an array by a length of p is created by a uniform distribution: $U(0, v)$. Next, the sort of transportation system travels from the manufacturer to the main DC should be specified. Therefore, a set of procedures has been indicated in Figure 2. It is clear that the first sort of transportation system is used for products p_2 and p_5 for shipping products from the manufacturer to the main DC. Additionally, the second and third sort of transportation systems are utilized for products p_1 , p_4 , and p_3 , respectively.

Then, we have provided the SA, VNS, HS, and ACO algorithms to solve the model and to find Pareto solutions in Appendix A. In the following, the developed heuristic algorithms (IACO and IHS) to find Pareto solutions and to solve the proposed model are illustrated.

3.3. Improved ant colony optimization algorithm

In the Improved Ant Colony Optimization (IACO) algorithm, the concept of neighborhood and its creation in this algorithm is very important. To obtain the neighborhood, all possible transitions can be carried out to the current S solution and determine the elements of $N(s)$. It should be noted that in many cases the number of these transfers may be very high and it is not possible to calculate them all in terms of computational cost. For this reason, in this paper, first of all, a few numbers of the neighborhood are considered. Thus, the algorithm becomes a heuristic algorithm, and in a short time can achieve a near-optimal and feasible solution. Hence, this algorithm can change its neighborhood diversity if it is constant of the best solution in the number of iterations. For this reason, at this stage, the IACO algorithm tries to add quality neighbors through the neighborhood concept of the VNS algorithm to its previous neighbors. This process goes on for so long that the algorithm can either achieve a better solution or establish the algorithm's final condition. Moreover, how to choose this neighborhood is described in full in the next section.

3.4. The rule of decision

This paper is inspired by the ACO algorithm. In the IACO algorithm is used the transfer formula (3.1). This rule can directly be made by creating a balance between the search for new edges and exploit the accumulated knowledge about the problem. In this formula, N_i^k is very significantly important. This set has nodes that have the following two conditions:

- (1) There are nodes that have been met by ant k th in the current iteration.
- (2) The nodes that add their demand to the capacity of the ant k th, which plays the role of the vehicle k th, do not exceed the capacity of the vehicle Q .

Therefore, at any one time, the number of 1 (the number of the variable neighborhood) of nodes that apply to conditions 1 and 2, and give the maximum value of $\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t) \mu_{ij}^\lambda(t)$ is selected and the N_i^k in (3.1) is made. It should be noted that the value of variable 1 makes the algorithm able to modify the ratio of the importance of search *versus* solution extraction.

$$P_{ij}^k(t) = \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t) \mu_{ij}^\lambda(t)}{\sum_{r \in N_i^k} \tau_{ij}^\alpha(t) \eta_{ij}^\beta(t) \mu_{ij}^\lambda(t)} \quad (3.1)$$

where τ_{ij}^α is the amount of pheromone poured on the (i, j) edge. η_{ij}^β is the problem heuristic information that is normally considered equal to $\frac{1}{c_{ij}}$, which c_{ij} is the cost of moving on the (i, j) edge. It should be noted that the smaller the cost on the (i, j) edge, the higher the $\frac{1}{c_{ij}}$ value and the higher the probability of the (i, j) edge being selected. μ_{ij}^λ is the saving amount that is inspired by the efficient saving algorithm and is equal to $\mu_{ij} = c_{i0} + c_{j0} - c_{ij}$. α, β, λ are positive parameters by which the user can modify the relative importance of the (i, j) edge size *versus* the amount of pheromone on it and the saving formula.

3.5. Intensification mechanism

To find sub-optimal solutions in meta-heuristic methods, the search process must be able to dig deeper into parts of the solution space that include higher-quality solutions. This concept is known as the intensification mechanism. According to the intensification mechanism, solutions, and transitions that lead to higher quality solutions are reinforced. In other words, this strategy is to return to elite solutions and further searches within their range, such as insertion, displacement, dual local search, triple local search, and cross. In the proposed algorithm, this is accomplished by two local algorithms of Swap move and Insert move and also is activated when the best solution is obtained. In the Swap move, one node is removed from one route and sent to another route provided that firstly the new solution applies to the problem constraints and secondly the new solution has a better value than before (Fig. 3), while in the Insert move, two nodes move in two different routes (Fig. 4). It should be noted that in the Insert move and Swap move, nodes are considered that are in two adjacent routes

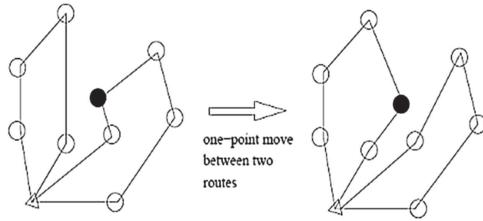


FIGURE 3. Swap move.

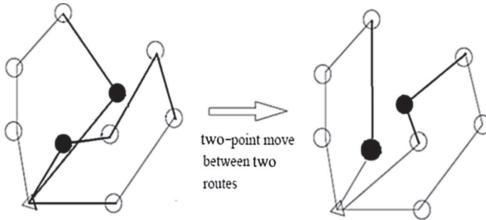


FIGURE 4. Insert move.

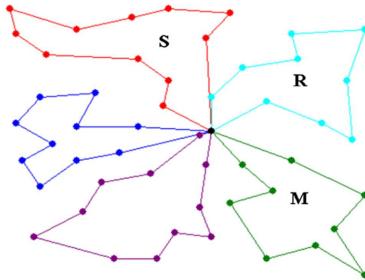


FIGURE 5. The location of Insert move and Swap move.

as is shown in Figure 5. The Insert move is one of the most efficient methods to remove excess vehicles. In this way, if the transfer of the last customer in a vehicle causes a cost reduction for the objective function, this action is performed. These two consecutive operations continue until there is no new recovery algorithm for the algorithm.

3.6. Pheromone update

In this proposed algorithm, the local pheromone is avoided and in each iteration, only the best current iteration solution and the best solution obtained so far are pheromones. It should be noted that local pheromones are not efficient because of the pheromones without considering whether the edge is put in the elite solution. This results in a lack of foresight in the algorithm, and the pheromone is poured over many edges that are not known to give a robust solution. Therefore, the pheromone shedding is done after all the ants have produced their nets and their edges are updated according to formula (3.2).

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \rho w. \quad (3.2)$$

In which if the corresponding edge belongs to two routes then the value of w is considered equal to be $(1/C_b)$, otherwise, it is equal to zero. The C_b cost of these the best two routes is in the current iteration and of the

```

Initial pheromone  $\tau_{ij}$ ;
Repeat for all ants  $i$ : construct solution  $(i)$ ;
  For all ants  $i$ : global pheromone update  $(i)$ ;
  For all ants' edges: evaporate pheromone;
    
$$\tau_{i-j}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \rho w$$

  construct solution  $(i)$ ;
  Initial ant;
  While not yet a solution:
    Expand the solution by one edge probabilistically according to the pheromone;
    Global_pheromone_update  $(i)$ :
      For all edges in the solution:
        Increase the pheromone according to the quality:
        
$$(\Delta\tau_{i-j} = 1/\text{length of the path stored})$$

       $(V = \{0, \dots, N\}, E = \{i \rightarrow j\})$ =directed acyclic graph
       $N$ = food source
       $\tau_{i-j}$ = initial pheromone
       $\rho$  = pheromone deposited
      VNS neighborhood []
      End for
    End for
  End for
End while
End

```

FIGURE 6. The pseudo-code of IACO algorithm.

algorithm since it started. It is important to note that the global update of pheromones encourages ants to neighborhood search for the best nets in subsequent iterations, notable in this formula is the use of dynamic pheromone shedding because of increased repetition and better solutions for C_b , edges belong to the best nets in the end iterations are more encouraged than the primary iterations. This definition is important because the more time the algorithm goes through, the better the results are likely to be. As a result, the amount of pheromone shedding is better suited to the algorithmic process, and if the algorithm is found to be better solutions, the algorithm with more pheromone shedding tends to search for the neighbors of those solutions. The IACO algorithm pseudo-code is shown in Figure 6.

3.7. Improved harmony search algorithm

In the Improved Harmony Search (IHS) algorithm, a new algorithm for generating a new solution vector is presented which improves its accuracy and convergence compared to the basic HS algorithm. The role of the PAR parameter in increasing the diversity in the onset of the algorithm search in the solution space and the role of BW in the local search of the algorithm shows to increase the efficiency convergence rate.

One of the disadvantages of the HS method is the use of constant values of PAR and BW, which are difficult to adjust for these parameters. Another disadvantage of HS is that the number of iterations the algorithm needs to find the optimal solution is not appropriate. If the PAR is small and the BW is large, the algorithm performs poorly and needs to be resolved to increase the number of improvisation (NI) to obtain the optimal solution (Fig. 7). The larger the value of the BW parameter in the elementary iterations, the greater the algorithm's

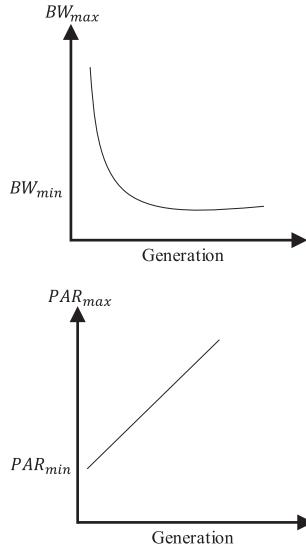


FIGURE 7. The changes of PAR and BW parameters in IHS algorithm.

search diversity across the solution space, and the subsequent iterations in order to local search the less value is appropriate. The IHS is similar to HS with a slight difference according to which the values of PAR and BW parameters are obtained dynamically in each separate iteration according to the equations (3.3) and (3.4).

$$PAR(t) = PAR_{\min} + \frac{PAR_{\max} - PAR_{\min}}{NI} \times t \quad (3.3)$$

$$BW(t) = BW_{\max} \times e^{\left(\frac{\ln\left(\frac{BW_{\min}}{BW_{\max}}\right)}{NI} \times t \right)} \quad (3.4)$$

where PAR_{\min} , PAR_{\max} , BW_{\min} , BW_{\max} , are constant values defined at the beginning of the IHS algorithm. T is the iteration number of the new solution vector of the algorithm and then NI is the number of improvisation to reach a stop condition. The IHS algorithm pseudo-code is indicated in Figure 8.

4. COMPUTATIONAL RESULTS

In this section, firstly, the writers generate the experiment problems for the MSCN model. Three assessment metrics are provided to evaluate the quality of non-dominated solutions of the algorithms according to the two objective functions. Additionally, the Taguchi approach is used to set the proper values for algorithm's parameters. Taguchi method is based on four basic and simple concepts including (i) factor: a factor is an experimentally controllable variable that affects the output. It can be considered as an independent variable. (ii) factor level is a specific amount of a factor, (iii) answer: answer (output) is a measurable phenomenon that the effect of factors on it is considered in a series of experiments, and (iv) experimental design: an experimental design is the result of the arrangement of factors and their levels, in a specific way. Taguchi combined and created special groups of orthogonal arrays to present his experiments. Orthogonal arrays facilitate the experimental design process. Designing an experiment involves selecting the most appropriate orthogonal array, determining the factors with the appropriate columns, and finally the position of the experiments (test conditions). The Taguchi procedure is utilized to tune the parameters of the SA, VNS, HS, ACO, IACO, and IHS algorithms, due to the algorithms parameter values on the quality of the solution is affected. Table 1 indicates various levels of the factors for SA, VNS, HS, ACO, IACO, and IHS. In the present paper based on the number of factors

Begin

 Define PAR and BW parameters

$$PAR(t) = PAR_{min} + \frac{PAR_{max} - PAR_{min}}{NI} \times t$$

$$BW(t) = BW_{max} \times e^{\left(\frac{\ln(BW_{min})}{BW_{max}} \times t \right)}$$

 Define objective function $f(x)$, $x = (x_1, x_2, \dots, x_d)^T$

 Define harmony memory accepting rate (r_{accept})

 Define pitch adjusting rate (r_{pa}) and other parameters

Generate Harmony Memory with random harmonies

While ($t < \text{max number of iterations}$)

While ($i \leq \text{number of variables}$)

If ($\text{rand} < r_{accept}$), Choose a value from HM for the variable i
If ($\text{rand} < r_{pa}$), Adjust the value by adding certain amount

End if
Else Choose a random value

End if
End while

Accept the new harmony (solution) if better

End while

Find the current best solution

End

FIGURE 8. The pseudo-code of IHS algorithm.

and the levels, the Taguchi procedure for multi-objective SA, VNS, HS, ACO, IACO, and IHS algorithms are employed for the adjustment and control of the parameters, respectively. For more details about Taguchi method refer to Goodarzian *et al.* [16, 17] and Fathollahi-Fard [11]. Additionally, the Taguchi method decreases the total number of experiments by proposing a set of orthogonal arrays to tune the algorithms in a reasonable time. The Taguchi method for the suggested algorithms proposed the Orthogonal Array L9 (for MOSA and MOVNS), L27 (for MOHS and MOACO), and L27 (for IHS and IACO) with different levels obtained by MINITAB Software. Orthogonal Arrays of the proposed algorithms are reported in Tables 2–4. For experiment analysis by the Taguchi method, signal to noise (S/N) ratio was used. Figures 9–14 display the S/N ratios of proposed algorithms. Consequently, various sorts of algorithms in order to find the best algorithm for the proposed problem are compared. Eventually, a set of sensitivity analyses is used to evaluate the validation of the MSCN model. It should be noted, the related codes were written in GAMS 24.1 and MATLAB R2016 for the proposed meta-heuristic and heuristic algorithms on a computer with Intel® Core™ i5-2400 6 GHz in CPU and 8 GB in RAM and utilizing Windows 10 as an operating system.

According to the novelty of the developed MSCN model, no existing paper has treated a similar model in the literature review. Moreover, the benchmarks existing in the literature review are not available for the suggested model and an approach is required to design the experiment problems. Ten experiment problems including two classifications *i.e.* small: S1–S5 and large sizes: L6–L10 are provided. The sizes of problem samples are introduced in Table 5 as well as the range of other parameters is indicated in Table 6.

In this regard, each instance was repeated 10 times for each run utilizing proposed algorithms for more accurate calculations. Furthermore, the average outcomes of the objective function of ten runs were chosen as the conclusion basis for the IACO algorithm. The objective function outputs for each problem size employing

TABLE 1. Factors and levels of algorithms.

Factor	Algorithm	Level	Value
Max-iteration	MOSA	3	50, 100, 150
Sub-iteration		3	50, 75, 100
Initial temperature		3	0.5, 0.75, 0.99
Max-iteration	MOVNS	3	50, 100, 150
Population size		3	75, 100, 125
Neighborhood size		3	10, 20, 30
Pitch adjustment rate	MOHS	3	2, 3, 4
Harmony memory consideration rate		3	4, 5, 6
Harmony memory size		3	2, 3, 4
Stop criteria		3	10, 15, 20
Number of ants	MOACO	3	50, 100, 150
Pheromone factor		3	0.5, 0.75, 0.95
Heuristic factor		3	2, 3, 4
Evaporation rate		3	0.25, 0.5, 0.75
Max-iteration		3	50, 100, 150
Memory size	IHS	3	5, 10, 15
Memory rate		3	0.5, 0.8, 0.99
Minimum step adjustment rate		3	0.2, 0.4, 0.6
Maximum step adjustment rate		3	0.5, 0.7, 0.9
Minimum bandwidth		3	0.2, 0.35, 0.45
Maximum bandwidth		3	0.5, 0.75, 0.99
Number of ants	IACO	3	50, 100, 150
Impact factor of the pheromone		3	1, 2, 3
Impact factor of the edge length		3	1, 3, 5
Effectiveness factor of pheromone deviation		3	0.25, 0.5, 0.75
Max-iteration		3	50, 100, 150

TABLE 2. The orthogonal array L9 for MOSA and MOVNS algorithm.

A	B	C
1	1	1
1	2	2
1	3	3
2	1	2
2	2	3
2	3	1
3	1	3
3	2	1
3	3	2

the proposed meta-heuristic and heuristic algorithms and CPLEX method are examined in Table 7. Table 8 depicts the Computational (CPU) time of the suggested methods in each example. CPU time of the proposed methods in the small- and medium-scale problems are displayed in Figure 15.

TABLE 3. The orthogonal array L27 for MOHS and MOACO algorithm.

A	B	C	D	E
1	1	1	1	1
1	1	1	1	2
1	1	1	1	3
1	2	2	2	1
1	2	2	2	2
1	2	2	2	3
1	3	3	3	1
1	3	3	3	2
1	3	3	3	3
2	1	2	3	1
2	1	2	3	2
2	1	2	3	3
2	2	3	1	1
2	2	3	1	2
2	2	3	1	3
2	3	1	2	1
2	3	1	2	2
2	3	1	2	3
3	1	3	2	1
3	1	3	2	2
3	1	3	2	3
3	2	1	3	1
3	2	1	3	2
3	2	1	3	3
3	3	2	1	1
3	3	2	1	2
3	3	2	1	3

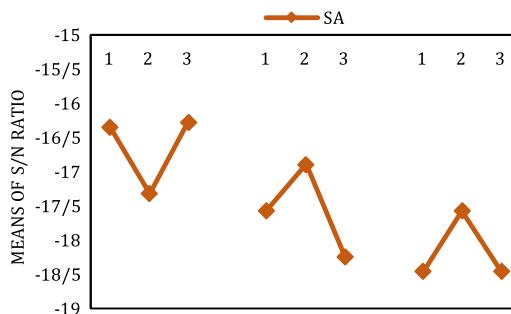


FIGURE 9. S/N ratio for the SA.

4.1. Assessment metrics for Pareto optimal solutions: comparison of meta-heuristics

In this section, the comparison of meta-heuristics in the term of multi-objective programming is difficult. Moreover, scholars have developed a number of metrics to evaluate the quality of Pareto fronts for the algorithms [8, 11, 16, 18, 19]. In this regard, three popular evaluation metrics have been used in this paper.

- Mean Ideal Distance (MID) [16, 17]: This metric represents the mean distance of the Pareto points of the ideal solutions. The ideal value is equal to the best possible value for each of the objective functions used in

TABLE 4. The orthogonal array L27 for IHS and IACO algorithm.

A	B	C	D	E	F
1	1	1	1	1	1
1	1	1	1	2	2
1	1	1	1	3	3
1	2	2	2	1	1
1	2	2	2	2	2
1	2	2	2	3	3
1	3	3	3	1	1
1	3	3	3	2	2
1	3	3	3	3	3
2	1	2	3	1	2
2	1	2	3	2	3
2	1	2	3	3	1
2	2	3	1	1	2
2	2	3	1	2	3
2	2	3	1	3	1
2	3	1	2	1	2
2	3	1	2	2	3
2	3	1	2	3	1
3	1	3	2	1	3
3	1	3	2	2	1
3	1	3	2	3	2
3	2	1	3	1	3
3	2	1	3	2	1
3	2	1	3	3	2
3	3	2	1	1	3
3	3	2	1	2	1
3	3	2	1	3	2

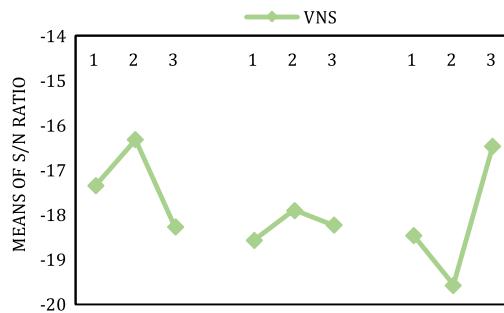


FIGURE 10. S/N ratio for the VNS.

all algorithms. A lower value of the MID means better performance of the algorithm. MID can be formulated as follows:

$$MID = \frac{\sum_{i=1}^{NPS} \left(\sqrt{\sum_{j=1}^{n_{obj}} \left(\frac{f_j^i - f_{best}^j}{f_{max}^j - f_{min}^j} \right)^2} \right)}{NPS} \quad (4.1)$$

TABLE 5. The samples for experiment problems.

Classification	Size	Levels (#m#i#j#c#v#n#p#t)
Small	S1	(#2#2#3#3#4#5#6)
	S2	(#4#4#5#5#7#6#7)
	S3	(#6#6#7#7#6#8#7#8)
	S4	(#8#7#8#8#7#9#8#9)
	S5	(#10#8#9#9#8#10#9#10)
Large	L6	(#12#9#10#10#9#12#10#10)
	L7	(#14#10#12#11#10#14#12#12)
	L8	(#16#12#14#12#11#16#14#12)
	L9	(#18#14#16#13#12#18#16#12)
	L10	(#20#16#18#14#14#20#18#12)

TABLE 6. The generated parameters according to the uniform distribution.

Parameters	Value	Parameters	Value
$\varphi_{pij}^v, \varphi_{pmi}^v, \varphi_{pjc}^v, \varphi_{pj}^v$	$\sim U(50\ 000, 60\ 000)$	$\beta_{pq,m}, \beta_{ni}, \beta_j, \beta_{pmq_p}$	$\sim U(70\ 000, 80\ 000)$
μ_{pi}^t, μ_{pj}^t	$\sim U(40\ 000, 50\ 000)$	d_{cp}^t	$\sim U(25\ 000, 35\ 000)$
I_{pj}^t, M	10 000 000	θ_p	$\sim U(0.3, 0.9)$
$\gamma_{in}, \rho_{mpq_p}, \alpha_i$	5 000 000	$\psi_{pij}^v, \psi_{pmi}^v, \psi_{pj}^v, \psi_{pjc}^v$	$\sim U(2, 8)$ (Hour)

TABLE 7. The average outcomes for the proposed methods.

Example no.	CPLEX	SA	Gap %	VNS	Gap %	HS	Gap %	ACO	Gap %	IACO	Gap %	IHS	Gap %
S1	OF1 2324	3156	0.26	2677	0.13	3788	0.39	3501	0.34	2355	0.013	2467.6	0.058
	OF2 1647.2	2788.1	0.4	1933.2	0.15	3208.1	0.48	2945.7	0.44	1677.1	0.017	1766.5	0.067
S2	OF1 2521.2	4322.1	0.42	3126.2	0.19	4801.3	0.47	4512.6	0.44	2541.3	0.008	2789.4	0.096
	OF2 1827.5	3780.2	0.52	2799.1	0.34	4102.4	0.55	3971.3	0.54	1899.5	0.038	1983.1	0.078
S3	OF1 2629.1	4977.3	0.48	3702.4	0.29	5963.1	0.56	5233.7	0.5	2744.3	0.041	2876.3	0.086
	OF2 2011.5	4105.4	0.51	2967.4	0.32	5288.1	0.62	4677.2	0.56	2112.4	0.048	2234.1	0.099
S4	OF1 2832.2	5477.9	0.48	4012.5	0.29	6512.4	0.57	6122.7	0.54	2899.1	0.023	2944.1	0.038
	OF2 2078.6	4901.3	0.57	3661.2	0.43	5880.2	0.64	5321.7	0.61	2133.6	0.025	2455.8	0.153
S5	OF1 2919.7	6034.7	0.52	4566.2	0.36	7212.4	0.6	6650.3	0.56	3013.7	0.031	3221.7	0.031
	OF2 2219	5632.7	0.6	4012.3	0.45	6770.3	0.67	6190.4	0.64	2344	0.053	2567.9	0.135
L6	OF1 –	7323.5		5334.8		7966.3		7503.2		3221		3455	
	OF2 –	6944.3		4908.2		7239.1		7012.7		2566.7		2899.2	
L7	OF1 –	7456.2		5421.1		7989.4		7545.5		3245		3478	
	OF2 –	6959.1		4934.4		7287.2		7035.6		2578.2		2923.1	
L8	OF1 –	7523.2		5474.6		8129.3		7594.7		3289		3494	
	OF2 –	6978.6		4957.2		7345.1		7089.8		2592.3		2945.3	
L9	OF1 –	7681.2		5534.8		8168.5		7689.1		3322		3534	
	OF2 –	7012.4		5034.2		7439.3		7112.7		2678.6		2987.4	
L10	OF1 –	7692.5		5634.8		8238.3		7761.1		3489		3589	
	OF2 –	7198.3		5089.7		7591.8		7278.2		2678.8		2991.6	

Notes. The bold values are the best answer to the proposed method.

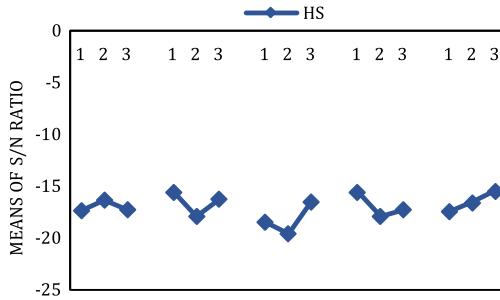


FIGURE 11. S/N ratio for the HS.

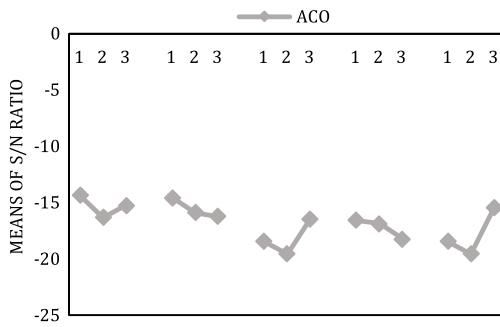


FIGURE 12. S/N ratio for the ACO.

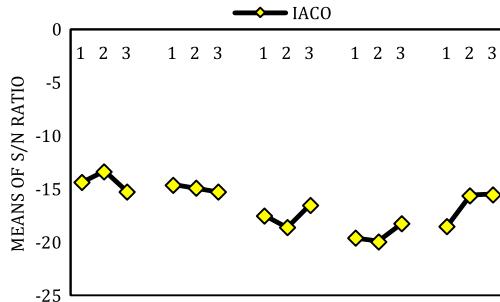


FIGURE 13. S/N ratio for the IACO.

where f_j^i shows i th solution and j th objective function. NPS states the Pareto solution number of the algorithm. Additionally, f_{\max}^j and f_{\min}^j indicate the maximum and the minimum values, respectively, between all Pareto solutions of each objective function. f_{best}^j shows the ideal point for the j th objective function.

- Number of Pareto solution (NPS) [16, 17]: This metric indicates the Number of Pareto solutions obtained through the algorithm, and the higher the number, the better the performance and efficiency of the algorithm.
- Spread of Non-Domination Solution (SNS) [16]: The SNS metric assesses the diversity of Pareto solutions. The higher value of this metric brings better and high quality. SNS can be formulated as follows:

$$\text{SNS} = \sqrt{\frac{\sum_{i=1}^{\text{NPS}} (\text{MID} - \sum_{j=1}^{n_{\text{obj}}} f_j^i)^2}{\text{NPS} - 1}}. \quad (4.2)$$

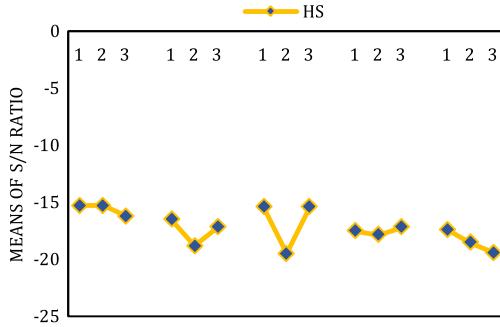


FIGURE 14. S/N ratio for the IHS.

TABLE 8. The results of the CPU time of the proposed methods.

Example no.	CPLEX	SA	VNS	HS	ACO	IACO	IHS
S1	1.56	4.61	3.02	6.73	5.88	1.75	2.45
S2	5.32	15.21	10.23	19.27	17.63	5.67	8.45
S3	9.56	21.43	15.45	38.31	30.76	10.32	13.67
S4	43.67	87.91	63.52	143.48	102.07	45.12	58.23
S5	62.67	94.78	75.33	178.54	144.46	63.87	70.02
L6	—	123.45	91.21	212.27	189.56	78.32	81.56
L7	—	135.12	113.67	312.14	212.23	89.21	95.71
L8	—	178.41	145.19	378.56	278.47	95.12	99.12
L9	—	213.57	169.32	456.12	298.14	126.29	132.18
L10	—	278.12	178.14	532.19	324.17	137.13	142.19

Notes. The bold values are the best answer to the proposed method.

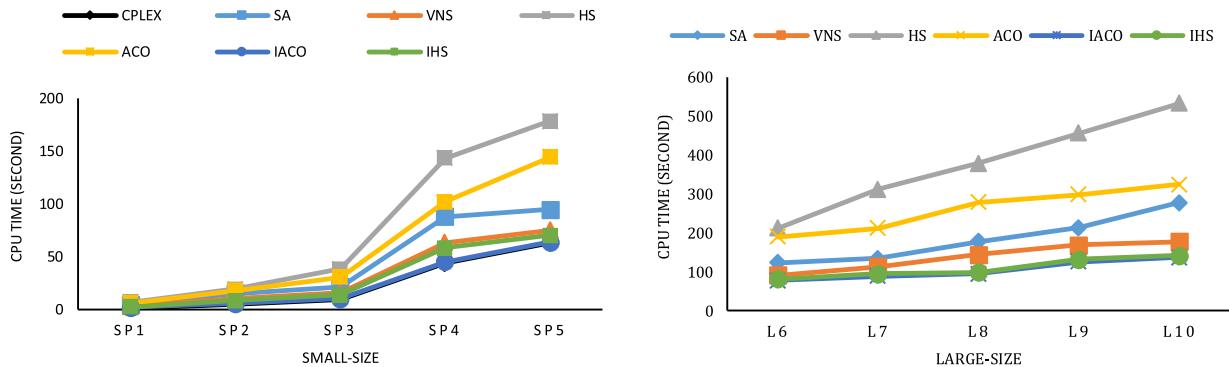


FIGURE 15. The average CPU time of algorithms.

The efficiency and performance of proposed algorithms are evaluated by evaluation metrics *i.e.* MID, NPS, and SNS as the comparison metrics for attained Pareto sets under each experiment problem. Hence, the outcomes of the evaluation metrics are reported in Tables 9–11.

In the following, an example of non-dominated solutions of suggested algorithms in an experiment problem *e.g.* L8 is shown in Figure 16. As provided by this figure, HS indicates the worst efficiency, IACO shows the best performance and more robust. The SA, VNS, and ACO meta-heuristics are the same as each other and close.

TABLE 9. The results of the MID of the proposed algorithms.

Example no.	SA	VNS	HS	ACO	IACO	HIS
S1	4.23	3.15	6.73	5.32	1.49	1.56
S2	4.28	3.23	6.27	5.63	1.12	1.34
S3	4.47	3.37	6.89	5.76	2.34	2.67
S4	4.91	3.52	7.34	6.07	3.18	3.23
S5	4.78	4.33	7.54	6.47	3.67	3.87
L6	4.97	5.21	8.27	7.56	4.45	4.52
L7	5.12	5.67	8.14	7.23	4.89	5.71
L8	5.41	5.19	8.56	7.47	5.12	5.82
L9	6.57	6.32	9.12	8.14	5.29	6.18
L10	7.12	6.89	9.49	8.17	6.34	6.69

Notes. The bold values are the best answer to the proposed method.

TABLE 10. The results of the SNS of the proposed algorithm.

Example no.	SA	VNS	HS	ACO	IACO	HIS
S1	3.35	5.32	2.23	4.34	6.75	6.45
S2	15.67	10.45	4.22	9.12	14.63	13.45
S3	12.89	15.42	5.31	10.58	17.31	16.67
S4	54.91	56.62	5.48	32.07	67.17	65.19
S5	67.78	67.83	18.54	34.41	78.82	76.56
L6	121.45	98.21	22.27	87.58	178.32	176.23
L7	145.13	113.17	34.45	89.22	189.31	187.72
L8	156.43	145.21	78.56	91.48	295.19	293.23
L9	208.57	157.89	96.12	101.24	226.34	219.18
L10	278.34	168.76	112.17	134.67	337.45	312.27

Notes. The bold values are the best answer to the proposed method.

TABLE 11. The results of the NPS of the proposed algorithm.

Example no.	SA	VNS	HS	ACO	IACO	IHS
S1	2	3	1	1	5	5
S2	2	3	1	1	6	4
S3	3	3	2	2	7	4
S4	3	4	2	3	7	5
S5	3	4	3	4	8	6
L6	2	5	3	4	10	9
L7	4	5	4	5	10	7
L8	5	6	3	5	9	8
L9	5	6	2	5	10	9
L10	6	5	3	4	10	10

Notes. The bold values are the best answer to the proposed method.

Further, a set of statistical comparisons among algorithms according to the Pareto optimal analyses taken by measurement metrics to find out the best algorithm is performed. Hence, the available results are provided in Tables 9–11 are transformed into a well-known metric, namely, Relative Deviation Index (RDI) as following formula [8, 11, 16, 18, 19]:

$$RDI = \frac{|\text{Alg}_{\text{sol}} - \text{Best}_{\text{sol}}|}{\text{Max}_{\text{sol}} - \text{Min}_{\text{sol}}} \times 100. \quad (4.3)$$

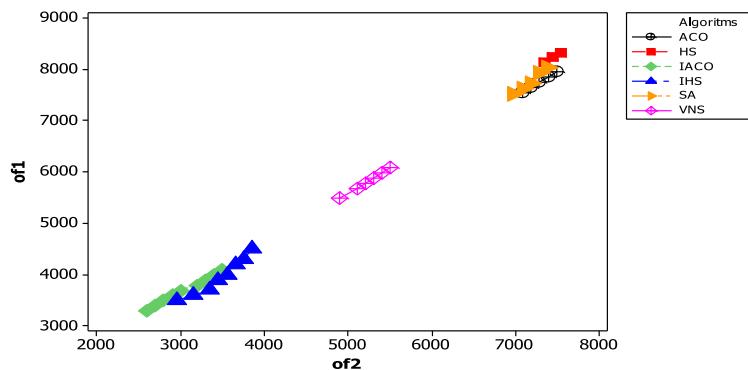


FIGURE 16. Pareto frontier of proposed algorithms in L8.

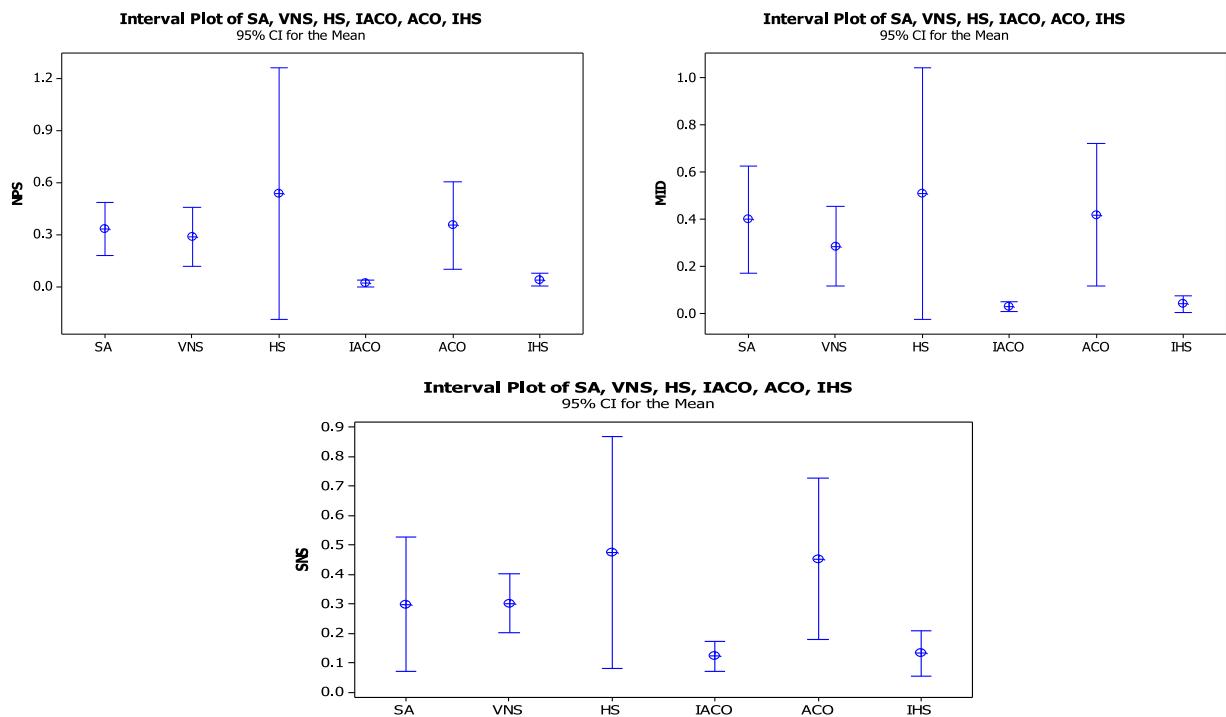


FIGURE 17. ANOVA plots for the three assessment metrics in term of RDI for algorithms.

It is clear that a lower value of RDI provides a higher quality of meta-heuristics and heuristics. Therefore, the Least Significant Difference (LSD) and means a plot for the proposed meta-heuristics and heuristics have shown. The outcomes run by Minitab software are indicated in Figure 17. In terms of MID, NPS, and SNS, and based on Figure 17, the developed IACO and IHS algorithms are better than the suggested algorithms and both of them are close to each other. All in all, first the IACO algorithm and then IHS indicate the best efficiency in all metrics. According to Figure 16, ACO and HS algorithms demonstrate poor efficiency, but HS has the worst performance than the ACO in all metrics. Additionally, SA and VNS algorithms are close to each other, and

TABLE 12. Sensitivity analysis on transportation cost.

Number of instances	$\varphi_{pij}^v, \varphi_{pmi}^v, \varphi_{pjc}^v, \varphi_{pj,j'}^v$	OF1	OF2
I1	50 000	4012.5	3661.2
I2	60 000	4834.2	3661.2
I3	70 000	5667.3	3661.2
I4	80 000	6344.1	3661.2
I5	90 000	7133.8	3661.2

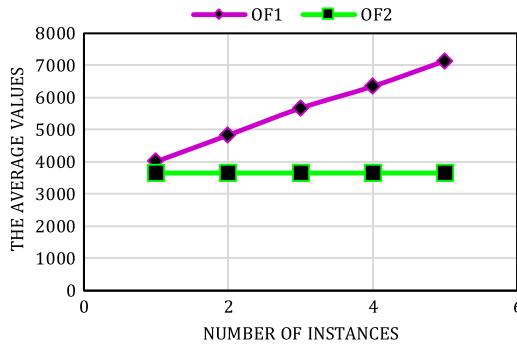


FIGURE 18. The sensitivity analysis of both objective functions.

also VNS is more robust than SA in all metrics. Consequently, IACO is better and more successful than other proposed algorithms, but HS shows the worst behavior in terms of MID, NPS, and SNS metrics.

5. SENSITIVITY ANALYSIS FOR MSCN MODEL

To recognize the behavior of the MSCN model, two sensitivity analyses have been carried out on the significant parameters of the suggested model. Furthermore, an experiment problem such as 4 considering eight manufacturers, seven main DCs, eight local DCs, four final customers, seven types of vehicles, nine capacity levels, eight medicine products, and nine time periods are chosen. Then, the IACO algorithm the most reliable in this paper is considered to handle the presented model. A set of changes containing the transportation costs ($\varphi_{pij}^v, \varphi_{pmi}^v, \varphi_{pjc}^v, \varphi_{pj,j'}^v$) and the transportation times ($\psi_{pij}^v, \psi_{pmi}^v, \psi_{pj,j'}^v, \psi_{pjc}^v$) for the proposed MSCN model is analyzed. Each analysis is divided into five instances numbered as I1 to I5. Eventually, all results are depicted in Tables 12 and 13 as well as Figures 18 and 19.

According to the transportation cost value, sensitivity analyses by increasing the amount of this parameter have been carried out that details are reported in Table 12. To recognize the two objective functions behavior *i.e.* total cost (OF1) and total transportation time (OF2), meanwhile, the average values are considered in this comparison as indicated in Figure 18. The outcomes illustrate that although by increasing the amount of this parameter the total cost is increased while the total transportation time remains no change.

The transportation times to do some analyses are considered. The outcomes are indicated in Table 13. Moreover, the behavior of both objective functions is shown in Figure 19. With increased transportation times, the behavior of the total transportation time is increased. Hence, the total costs are remained fixed and without change.

TABLE 13. Sensitivity analysis on transportation time.

Number of instances	$\psi_{pij}^v, \psi_{pmi}^v, \psi_{pjj'}^v, \psi_{pjc}^v$	OF1	OF2
1	2	4012.5	3661.2
2	4	4012.5	4667.3
3	6	4012.5	5433.1
4	8	4012.5	6770.4
5	10	4012.5	7233.8

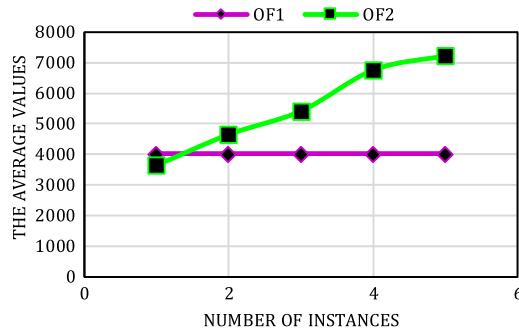


FIGURE 19. The sensitivity analysis of both objective functions.

6. CONCLUSIONS AND FUTURE WORKS

In this paper, the design of bi-objective MSCN for a production-distribution problem with consideration of production technology policy is examined. Firstly, a multi-period multi-level multi-product model is presented to design the MSCN, and also a mixed-integer nonlinear programming model to minimize the total costs and total transportation times is formulated. Another contribution of this paper is to propose two new heuristic algorithms. To solve the presented model, four multi-objective meta-heuristic algorithms (SA, VNS, HS, and ACO) are used. In this regard, two new heuristics including the IACO and IHS algorithm to solve the MSCN problem and to find the Pareto solutions are developed. In this regard, a new method is presented in the solution structure of the HS algorithm which is used to solve continuous mathematical optimization problems. According to the features of the algorithm in early convergence due to falling in local optimality, the re-start phase technique is used. Also, by dynamically adjusting the parameters of the algorithm from the initial iterations to the end iterations, a new structure is created in the algorithm that first high search variability and search intensification is minimized. In order to investigate the whole problem-solving space before achieving local or global optimality, areas of the problem-solving space that have a better objective value are identified. In subsequent iterations of the algorithm, the search diversity is reduced and the search intensification is increased so that the IHS uses local search techniques in the areas it identifies from previous iterations. Also, IACO Algorithm to solve the MSCN problem is developed in which the neighborhood concept of the VNS algorithm to a more robust algorithm has used. This modification is presented in the local search as intensified mechanisms that cause the IACO has the best performance and more robust than other proposed meta-heuristics. The results show that the IACO has the performance and efficiency to escape local optimum points. The Taguchi method has been used to enhance the efficiency of meta-heuristic algorithms and heuristic methods to tune algorithms' parameters. The CPLEX Solver has been employed to validate the presented mathematical modeling in the small-sized problems. Next, to analyze the presented algorithms and heuristic approaches, three assessment metrics have been suggested. Then, outcomes have been indicated the efficiency of the suggested algorithms and their performance to solve

the large-sized problems. According to the three assessment metrics, CPU time, and the LSD and means a plot for the proposed meta-heuristics and heuristics, the results of the IACO has been indicated that it has a more robust and high convergence speed than other presented algorithms. Eventually, in order to illustrate the real characteristics of the model, sensitivity analyses on the MSCN model were performed.

For future works, interested researchers can consider stochastic and robust programming approaches to cope with uncertain parameters such as cost, demand, and capacity. Also, we recommended to researchers developing new heuristic algorithms and compared them with the proposed heuristics in this paper. Scholars can provide exact methods such as logic-based Benders decomposition, Lagrangian relaxation approach, branch-and-price, and branch-and-cut for further work. a set of new assumptions and contributions can be added into the present model for future works such as considering sustainability (economic, environmental, and social), and the level of customer satisfaction are recommended as well.

APPENDIX A.

A.1. Simulated annealing

SA algorithm is an effective meta-heuristic optimization algorithm to solve optimization problems. This algorithm was introduced by Kirkpatrick *et al.* [28]. They proposed a method based on a gradual freezing technique to solve difficult optimization problems. The gradual freezing technique is a method of achieving a state in which the energy of the solid is well and uniformly minimized. This technique involves placing the material at high temperature and then gradually lowering it. For more information about this algorithm refer to Leite [29] and Fakhrzad *et al.* [10]. The SA pseudo-code is shown in Figure A.1.

```

Generate an initial solution (S0)
Initialize the algorithm parameters T,  $\alpha$ , K, and x
// where T is the initial temperature //
//  $\alpha$  is the cooling rate //
// K is the number of iterations at a fixed temperature //
// x is a uniform random number  $\in (0,1)$  //
While stopping condition is not satisfied do
  For  $i \leftarrow 1$ : K
    5: Generate a new neighbor for the current solution via neighborhood structure and evaluate it
    Compare the new neighbor against the current solution
    If the neighbor is non-dominated, Then
      Add the neighbor to the non-dominated frontier
      Replace the current solution with the neighbor
    Else check metropolis criterion
      If  $x < P(A) = \exp((-\Delta E)/T)$ , Then
        Replace the current solution with the neighbor
      End if
    End if
    Next i
     $T \leftarrow T \times \alpha$ 
  End while
Return the final non-dominated frontier

```

FIGURE A.1. The pseudo-code of SA algorithm.

A.2. Variable neighborhood search algorithm

Variable Neighborhood Search (VNS) algorithm is one of the meta-heuristic algorithms based on systematic changes of neighborhood structure which is used to search for optimal solutions in the combinatorial optimization problems [25]. Unlike many other meta-heuristic algorithms, this algorithm is very simple and requires fewer parameters to be adjusted. Therefore, achieving high-quality solutions and in a reasonable time, besides being simple, demonstrates the proper performance of this algorithm. The VNS algorithm was first proposed by Hansen and Mladenovic [20]. The basic idea behind this algorithm is that changes in the neighborhood structure when searching for a solution space avoids the risk of falling into local optimization, which is possible in many combinatorial optimization problems. Especially with the rapid growth of space the solution to the problem size, the probability of being trapped in the local minimum will be inevitable. So the first step in the VNS algorithm is to define neighborhood structures to generate a neighborhood solution. For more information about this algorithm refer to Baniamerian *et al.* [4]. The VNS algorithm pseudo-code is indicated in Figure A.2.

Input: a set of neighborhood structures $N_l, l = 1, 2, \dots, l_{max}$

S=generate initial solution ();

Repeat

```

    l = 1;
    While l ≤ lmax
    {
        S' = shaking (S, Nl)
        S'' = Local search (S')
        If f(S'') < f(S)
            S ← S''
            l = 1;
        Else
            l = l + 1;
    }

```

Until stopping condition are met;

Output: The best solution

FIGURE A.2. The pseudo-code of VNS algorithm.

A.3. Harmony search algorithm

The harmony search (HS) algorithm was introduced by Gams [21]. HS become one of the most widely used optimization algorithms due to its applicability to discrete and continuous optimization problems, low mathematical computation, simple concept, low parameters, and easy implementation. HS algorithm has less mathematical computational than other meta-heuristic algorithms and can be adapted to different engineering problems by changing parameters and operators. One of the features of the HS algorithm is that it identifies better-performing solution spaces at a reasonable time [42].

In this algorithm, each solution is called a harmonic and is represented by a subsequent N vector. This algorithm has three main phases: Initial generation (initialization), improvisation of the new harmonic vector, and algorithm memory update. According to the first phase, an initial generation of harmonic vectors is randomly generated and stored in Harmony Memory Size (HM). In the second phase, a new harmonic vector (new solution) is created using the rules of memory consideration, step matching and random re-generation of available solutions in the memory of the algorithm [48]. For more information about this algorithm refer to Ma *et al.* [31]. The HS algorithm pseudo-code is indicated in Figure A.3.

```

Begin
  Define objective function  $f(x), x = (x_1, x_2, \dots, x_d)^T$ 
  Define harmony memory accepting rate ( $r_{accept}$ )
  Define pitch adjusting rate ( $r_{pa}$ ) and other parameters
  Generate Harmony Memory with random harmonies
  While ( $t < \text{max number of iterations}$ )
    While ( $i \leq \text{number of variables}$ )
      If ( $\text{rand} < r_{accept}$ ), Choose a value from HM for the variable  $i$ 
      If ( $\text{rand} < r_{pa}$ ), Adjust the value by adding certain amount
      End if
      Else Choose a random value
      End if
    End while
    Accept the new harmony (solution) if better
  End while
  Find the current best solution
End

```

FIGURE A.3. The pseudo-code of HS algorithm.

```

Input: Problem size, Population size, The number of ants ( $m$ ), Pheromone reduction coefficient ( $\rho$ ), Heuristic coefficient ( $\alpha$ ), Historical coefficient ( $\beta$ )
Output:  $P_{best}$ 
 $P_{best} \leftarrow \text{create heuristic solution (problem size)}$ ;
 $P_{best\_cost} \leftarrow \text{cost}(S_h)$ ;
Pheromone  $\leftarrow \text{Initialize pheromone (}P_{best\_cost}\text{)}$ ;
While stop condition () do
  Candidates  $\leftarrow \emptyset$ ;
  For  $i = 1$  to  $m$  do
     $S_i \leftarrow \text{probabilistic step wise construction (pheromone, problem size, } \alpha, \beta\text{)}$ ;
     $S_{icost} \leftarrow \text{cost}(S_i)$ ;
    If  $S_{icost} \leq P_{best\_cost}$  then
       $P_{best\_cost} \leftarrow S_{icost}$ ;
       $P_{best} \leftarrow S_i$ ;
    End
    Candidates  $\leftarrow S_i$ ;
  End
  Decay pheromone (pheromone,  $\rho$ );
  For each  $S_i \in \text{Candidates}$  do
    Update pheromone (pheromone,  $S_i, S_{icost}$ );
  End
  End
Return  $P_{best}$ ;

```

FIGURE A.4. The pseudo-code of ACO algorithm.

A.4. Ant colony optimization algorithm

The Ant Colony Optimization (ACO) algorithm is one of the most popular methods of meta-heuristics first introduced by Dorigo and has had many applications to hybrid optimization problems [6]. This algorithm is based on the natural behavior of the ant colonies and the working ant of workers. The process of finding nutrients in the ant colony is highly optimized. When ants begin their exploration to find food sources, they will naturally find an “optimal” and “logical” route from their nest to food sources. In other words, the ant population is always able to find an optimal route to supply the required food. Then, simulating such optimal behavior forms the basis of the ACO algorithm [30]. For more information about this algorithm refer to Forozandeh *et al.* [13] and Jovanovic *et al.* [24]. The ACO algorithm pseudo-code is indicated in Figure A.4.

REFERENCES

- [1] L. Alnaji and M. Ridha, The role of supply chain applications in Jordanian pharmacies: a case study on pharmacies in the capital city Amman. *Ind. Eng. Lett.* **3** (2013) 65–71.
- [2] A.C.S. Amaro and A.P.F. Barbosa-Póvoa, Planning and scheduling of industrial supply chains with reverse flows: a real pharmaceutical case study. *Comput. Chem. Eng.* **32** (2008) 2606–2625.
- [3] Z. Azadehghanbar, A. Bozorgi-Amiri and A. Zandi, Warehouse redesigning in a three-echelon supply chain network with consideration of routing under uncertainty. *RAIRO:OR* **55** (2021) S147–S166.
- [4] A. Baniamerian, M. Bashiri and R. Tavakkoli-Moghaddam, Modified variable neighborhood search and genetic algorithm for profitable heterogeneous vehicle routing problem with cross-docking. *Appl. Soft Comput.* **75** (2019) 441–460.
- [5] S.H. Chung and C. Kwon, Integrated supply chain management for perishable products: dynamics and oligopolistic competition perspectives with application to pharmaceuticals. *Int. J. Prod. Econ.* **179** (2016) 117–129.
- [6] M. Dorigo and G. Di Caro, Ant colony optimization: a new meta-heuristic. In: Vol. 2 of *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*. IEEE (1999) 1470–1477.
- [7] M.B. Fakhrzad and F. Goodarzian, A fuzzy multi-objective programming approach to develop a green closed-loop supply chain network design problem under uncertainty: modifications of imperialist competitive algorithm. *RAIRO:OR* **53** (2019) 963–990.
- [8] M.B. Fakhrzad and F. Goodarzian, A new multi-objective mathematical model for a Citrus supply chain network design: metaheuristic algorithms. *J. Optim. Ind. Eng.* **14** (2020) 127–144.
- [9] M.B. Fakhrzad, P. Talebzadeh and F. Goodarzian, Mathematical formulation and solving of green closed-loop supply chain planning problem with production, distribution and transportation reliability. *Int. J. Eng.* **31** (2018) 2059–2067.
- [10] M.B. Fakhrzad, F. Goodarzian and A.M. Golmohammadi, Addressing a fixed charge transportation problem with multi-route and different capacities by novel hybrid meta-heuristics. *J. Ind. Syst. Eng.* **12** (2019) 167–184.
- [11] A.M. Fathollahi-Fard, A. Ahmadi, F. Goodarzian and N. Cheikhrouhou, A bi-objective home healthcare routing and scheduling problem considering patients’ satisfaction in a fuzzy environment. *Appl. Soft Comput.* **93** (2020) 106385.
- [12] M. Fazli-Khalaf, S.K. Chaharsooghi and M.S. Pishvae, A new robust possibilistic programming model for reliable supply chain network design: a case study of lead-acid battery supply chain. *RAIRO:OR* **53** (2019) 1489–1512.
- [13] M. Forozandeh, E. Teimouri and A. Makui, A mathematical formulation of time-cost and reliability optimization for supply chain management in research-development projects. *RAIRO:OR* **53** (2019) 1385–1406.
- [14] G. Gatica, L.G. Papageorgiou and N. Shah, Capacity planning under uncertainty for the pharmaceutical industry. *Chem. Eng. Res. Design* **81** (2003) 665–678.
- [15] F. Goodarzian and H. Hosseini-Nasab, Applying a fuzzy multi-objective model for a production–distribution network design problem by using a novel self-adaptive evolutionary algorithm. *Int. J. Syst. Sci.: Oper. Logistics* **8** (2019) 1–22.
- [16] F. Goodarzian, H. Hosseini-Nasab, J. Muñozuri and M.B. Fakhrzad, A multi-objective pharmaceutical supply chain network based on a robust fuzzy model: a comparison of meta-heuristics. *Appl. Soft Comput.* **92** (2020) 106331.
- [17] F. Goodarzian, H. Hosseini-Nasab and M.B. Fakhrzad, A multi-objective sustainable medicine supply chain network design using a novel hybrid multi-objective metaheuristic algorithm. *Int. J. Eng.* **33** (2020) 1986–1995.
- [18] F. Goodarzian, A. Abraham and A.M. Fathollahi-Fard, A biobjective home health care logistics considering the working time and route balancing: a self-adaptive social engineering optimizer. *J. Comput. Design Eng.* **8** (2021) 452–474.
- [19] F. Goodarzian, A.A. Taleizadeh, P. Ghasemi and A. Abraham, An integrated sustainable medical supply chain network during COVID-19. *Eng. App. Artif. Intell.* **100** (2021) 104188.
- [20] P. Hansen and N. Mladenović, Variable neighborhood search: principles and applications. *Eur. J. Oper. Res.* **130** (2001) 449–467.
- [21] S.D. Hosseini, M.A. Shirazi and B. Karimi, Cross-docking and milk run logistics in a consolidation network: a hybrid of harmony search and simulated annealing approach. *J. Manuf. Syst.* **33** (2014) 567–577.
- [22] G. Jamali, S.S. Sana and R. Moghdani, Hybrid improved cuckoo search algorithm and genetic algorithm for solving Markov-modulated demand. *RAIRO:OR* **52** (2018) 473–497.
- [23] G. Jetly, C.L. Rossetti and R. Handfield, A multi-agent simulation of the pharmaceutical supply chain. In: *Agent-Based Modeling and Simulation*. Palgrave Macmillan, London (2014) 133–154.

- [24] R. Jovanovic, M. Tuba and S. Voß, An efficient ant colony optimization algorithm for the blocks relocation problem. *Eur. J. Oper. Res.* **274** (2019) 78–90.
- [25] P. Karakostas, A. Sifaleras and M.C. Georgiadis, A general variable neighborhood search-based solution approach for the location-inventory-routing problem with distribution outsourcing. *Comput. Chem. Eng.* **126** (2019) 263–279.
- [26] S. Khalifehzadeh and M.B. Fakhrzad, A stochastic bi-objective mathematical model for optimizing a production and distribution system with stochastic demand and stochastic lead time. *Int. J. Eng. Sci. (2008–4870)* **29** (2018).
- [27] M. Khorbatly, H. Dkhil, H. Alabboud and A. Yassine, A hybrid algorithm Tabu Search-GRASP for wounded evacuation in disaster response. *RAIRO:OR* **54** (2020) 19–36.
- [28] S. Kirkpatrick, Optimization by simulated annealing: Quantitative studies. *J. Stat. Phys.* **34** (1984) 975–986.
- [29] N. Leite, F. Melício and A.C. Rosa, A fast simulated annealing algorithm for the examination timetabling problem. *Expert Syst. App.* **122** (2019) 137–151.
- [30] J. Luan, Z. Yao, F. Zhao and X. Song, A novel method to solve supplier selection problem: hybrid algorithm of genetic algorithm and ant colony optimization. *Math. Comput. Simul.* **156** (2019) 294–309.
- [31] H. Ma, X. Li and Y. Liu, Multi-period multi-scenario optimal design for closed-loop supply chain network of hazardous products with consideration of facility expansion. *Soft Comput.* **24** (2020) 2769–2780.
- [32] M. Mousazadeh, S.A. Torabi and B. Zahiri, A robust possibilistic programming approach for pharmaceutical supply chain network design. *Comput. Chem. Eng.* **82** (2015) 115–128.
- [33] L.G. Papageorgiou, Supply chain optimization for the process industries: advances and opportunities. *Comput. Chem. Eng.* **33** (2009) 1931–1938.
- [34] L.G. Papageorgiou, G.E. Rotstein and N. Shah, Strategic supply chain optimization for the pharmaceutical industries. *Ind. Eng. Chem. Res.* **40** (2001) 275–286.
- [35] G.E. Rotstein, L.G. Papageorgiou, N. Shah, D.C. Murphy and R. Mustafa, A product portfolio approach in the pharmaceutical industry. *Comput. Chem. Eng.* **23** (1999) S883–S886.
- [36] B. Roy and B.C. Giri, A three-echelon supply chain model with price and two-level quality dependent demand. *RAIRO:OR* **54** (2020) 37–52.
- [37] N. Sahebjamnia, F. Goodarzian and M. Hajiaghaei-Keshteli, Optimization of multi-period three-echelon citrussupply chain problem. *J. Optim. Ind. Eng.* **13** (2020) 39–53.
- [38] E. Settanni, T.S. Harrington and J.S. Srai, Pharmaceutical supply chain models: a synthesis from a systems view of operations research. *Oper. Res. Perspect.* **4** (2017) 74–95.
- [39] S.K. Singh and M. Goh, Multi-objective mixed integer programming and an application in a pharmaceutical supply chain. *Int. J. Prod. Res.* **57** (2019) 1214–1237.
- [40] R.T. Sousa, S. Liu, L.G. Papageorgiou and N. Shah, Global supply chain planning for pharmaceuticals. *Chem. Eng. Res. Design* **89** (2011) 2396–2409.
- [41] N. Susarla and I.A. Karimi, Integrated supply chain planning for multinational pharmaceutical enterprises. In: Vol. 29 of *Computer Aided Chemical Engineering*. Elsevier (2011) 1075–1079.
- [42] E. Tsakirkis, M. Marinaki, Y. Marinakis and N. Matsatsinis, A similarity hybrid harmony search algorithm for the team orienteering problem. *Appl. Soft Comput.* **80** (2019) 776–796.
- [43] D. Weraikat, M.K. Zanjani and N. Lehoux, Two-echelon pharmaceutical reverse supply chain coordination with customer's incentives. *Int. J. Prod. Econ.* **176** (2016) 41–52.
- [44] D. Weraikat, M.K. Zanjani and N. Lehoux, Improving sustainability in a two-level pharmaceutical supply chain through Vendor-Managed Inventory system. *Oper. Res. Health Care* **21** (2019) 44–55.
- [45] D.H. Wolpert and W.G. Macready, No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1** (1997) 67–82.
- [46] E. Yadegari, A. Alem-Tabriz and M. Zandieh, A memetic algorithm with a novel neighborhood search and modified solution representation for closed-loop supply chain network design. *Comput. Ind. Eng.* **128** (2019) 418–436.
- [47] B. Zahiri, J. Zhuang and M. Mohammadi, Toward an integrated sustainable-resilient supply chain: a pharmaceutical case study. *Transp. Res. Part E: Logistics Transp. Rev.* **103** (2017) 109–142.
- [48] H. Zarei and M. Rasti-Barzoki, Mathematical programming and three metaheuristic algorithms for a bi-objective supply chain scheduling problem. *Neural Comput. App.* **31** (2019) 9073–9093.