Engineering Science and Technology, an International Journal xxx (xxxx) xxx

HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

journal homepage: www.elsevier.com/locate/jestch

Full Length Article

Experimental evaluation of meta-heuristics for multi-objective capacitated multiple allocation hub location problem

İbrahim Demir [a], Berna Kiraz [b,*], Fatma Corut Ergin [a]

[a] Marmara University, Engineering Faculty, Department of Computer Engineering, Istanbul, Turkey
[b] Fatih Sultan Mehmet Vakif University, Engineering Faculty, Department of Computer Engineering, Istanbul, Turkey

ARTICLE INFO

Article history:
Received 4 March 2021
Revised 1 June 2021
Accepted 19 June 2021
Available online xxxx

Keywords:
Capacitated hub location problem
Meta-heuristic algorithms
Multi-objective optimization
Routing and network design

ABSTRACT

Multi-objective capacitated multiple allocation hub location problem (MOCMAHLP) is a variation of classic hub location problem, which deals with network design, considering both the number and the location of the hubs and the connections between hubs and spokes, as well as routing of flow on the network. In this study, we offer two meta-heuristic approaches based on the non-dominated sorting genetic algorithm (NSGA-II) and archived multi-objective simulated annealing method (AMOSA) to solve MOCMAHLP. We attuned AMOSA based approach to obtain feasible solutions for the problem and developed five different neighborhood operators in this approach. Moreover, for NSGA-II based approach, we developed two novel problem-specific mutation operators. To statistically analyze the behavior of both algorithms, we conducted experiments on two well-known data sets, namely Turkish and Australian Post (AP). Hypervolume indicator is used as the performance metric to measure the effectiveness of both approaches on the given data sets. In the experimental study, thorough tests are conducted to fine-tune the proposed mutation types for NSGA-II and proposed neighborhood operators for AMOSA. Fine-tuning tests reveal that for NSGA-II, mutation probability does not have a real effect on Turkish data set, whereas lower mutation probabilities are slightly better for AP data set. Moreover, among the AMOSA based neighborhood operators, the one which adds/removes a specific number of links according to temperature (NS-5) performs better than the others for both data sets. After analyzing different operators for both algorithms, a comparison between our NSGA-II based and AMOSA based approaches is performed with the best settings. As a result, we conclude that both of our algorithms are able to find feasible solutions of the problem. Moreover, NSGA-II performs better for larger, whereas AMOSA performs better for smaller size networks.

1. Introduction

Hub-and-spoke networks constitute a type of distribution networks which is mostly adapted by transportation services. In hub-and-spoke networks, hubs are the nodes that are used for collection, processing and distribution of flow and spokes are the non-hub nodes allocated to hubs. Hub location problem (HLP) deals with locating the hubs, assigning spokes to hubs, and routing the flow between each node pair in the network.

There are several variations of HLP [19]. In this paper, we deal with the multi-objective capacitated multiple allocation HLP. We assume that the number of hubs is not determined in the problem definition and the hubs are not fully interconnected. The two objectives are minimization of cost for establishing hub-and-spoke network and minimization of maximum travel time required to route the flow between all node pairs. In this problem, the number and the location of hub nodes must be determined, network design including inter-hub and hub-spoke connections must be made, and the routing of flow must be established. The mathematical model for the multi-objective capacitated multiple allocation hub location problem (MOCMAHLP) and a basic meta-heuristic approach based on the non-dominated sorting genetic algorithm (NSGA-II) can be found in [11].

Similar to single objective optimization problems, multi-objective optimization problems can be solved using exact methods. However, for NP-complete problems it becomes hard to solve large scale systems with these methods. On the other hand, meta-heuristic algorithms such as simulated annealing, ant colony optimization, and genetic algorithms are effective in solving these problems [22–24,31,33]. Simulated annealing (SA) [21] is a heuris-

* Corresponding author.
E-mail addresses: bkiraz@fsm.edu.tr (B. Kiraz), fatma.ergin@marmara.edu.tr (F. Corut Ergin).

https://doi.org/10.1016/j.jestch.2021.06.012
2215-0986/

Please cite this article as: İ. Demir, B. Kiraz and F. Corut Ergin, Experimental evaluation of meta-heuristics for multi-objective capacitated multiple allocation hub location problem, Engineering Science and Technology, an International Journal, https://doi.org/10.1016/j.jestch.2021.06.012

tic algorithm that is inspired by the metal annealing process. SA has been successfully applied to many combinatorial optimization problems including single-objective and multi-objective problems [4,28,32]. There is an extensive number of multi-objective simulated annealing approaches in literature [4]. Archived Multi-Objective Simulated Annealing (AMOSA) [5] algorithm is an improved version of multi-objective simulated annealing algorithm. It is a single-point search algorithm utilizing an archive in which non-dominated solutions are stored for later use. On the other hand, NSGA-II is one of the most commonly used genetic algorithm methods for multi-objective optimization problems [22,27].

In this paper, we present NSGA-II based and AMOSA based algorithms for solving MOCMAHLP and provide an experimental evaluation of these multi-objective optimization techniques. These algorithms are selected since they are well-known multi-objective heuristic methods in combinatorial optimization. In MOCMAHLP, there are three design criteria: (1) the number and location of hub nodes, (2) network design including inter-hub and hub-spoke connections, and (3) the routing of flow. In this study, for both NSGA-II and AMOSA, we use the solution representation proposed in [11], which offers a solution for the first two design criteria. This solution representation contains two arrays, one for hub decision and one for network design. For the last design criterion, i.e., routing of flow, A^* algorithm is applied. To explore new solution spaces, mutation operator is used for NSGA-II and neighborhood operator is used for AMOSA. In this study, we propose two novel problem-specific mutation operators for NSGA-II and five problem-specific neighborhood operators for AMOSA. To validate the effectiveness of our algorithms, we conducted experiments on Turkish and Australian Post (AP) data sets to statistically analyze the behavior of both algorithms. We investigated the effect of operators under different parameter settings to determine the best settings for both algorithms. Moreover, the performance of both algorithms with best setting were examined on different size networks.

The main contributions of this paper are given as follows:

- Two new problem-specific mutation operators are proposed for NSGA-II for which a basic mutation operator is presented before.
- AMOSA is adapted to find feasible solutions for the MOCMAHLP. We design five neighborhood operators specific to the problem and select the best performing neighborhood operator for AMOSA.
- A comprehensive experimental analysis of both approaches are conducted.

The rest of the paper is organized as follows: The next section provides a brief literature survey on the studies of meta-heuristics for location and routing problems. Section 3 describes the multi-objective capacitated multiple allocation hub location problem addressed in this paper. In Section 4, details of the employed heuristic approaches are presented. The experimental design and experimental results are presented in Sections 5 and 6, respectively. Finally, Section 7 highlights the conclusion and future work.

2. Related work

Hub location problem (HLP) is an NP-complete combinatorial optimization problem [20]; therefore, many meta-heuristic approaches have been proposed in the literature to solve the HLPs [9,12,13,15,19,25,30,34,35]. In this paper, we focus on recently published simulated annealing (SA) and evolutionary algorithms based approaches for the location and routing problems.

Geramianfar et al. [17] presented a multi-objective hub covering location problem under congestion. The problem involves two objective functions: (1) minimize total transportation cost and (2) minimize total waiting time. SA was applied to the problem and was compared to particle swarm optimization algorithm and NSGA-II. They used four different metrics, i.e., quality metric, mean ideal distance, diversification metric and spacing metric, to compare the performance of the algorithms. Based on the results, SA yielded better performance in terms of quality metric; however, similar results are obtained in terms of other metrics.

Alizadeh et al. [2] proposed a multi-objective model for a capacitated, multiple allocation hub maximal covering problem with the objectives of maximizing hub covering, minimizing the maximal capacity of hubs, and minimizing the total cost. They applied two multi-objective genetic algorithms, namely NSGA-II and the non-dominated ranking genetic algorithm (NRGA) to the problem. The experimental results showed that NSGA-II provided better performance than NRGA with respect to solution quality and computational time.

In [1], a hybrid algorithm combining genetic algorithm and simulated annealing was proposed for the competitive hub location and pricing problem. The hybrid algorithm is based on a genetic algorithm including parent selection, crossover and mutation operator. However, the best solution is updated based on move acceptance step in the SAs, i.e., the improving moves are always accepted while the non-improving moves are accepted with a probability. The proposed hybrid algorithm was compared with GA. Based on the experimental results, the hybrid algorithm yielded better performance than GA.

Rayat et al. [26] presented a bi-objective, multi-product and multi-period location-inventory-routing problem taking disruption risks into consideration. They proposed an algorithm based on the archived multi-objective simulated annealing (AMOSA) for solving the problem. In the algorithm, six different arrays were used to represent a solution. The proposed method was compared to three well-known multi-objective evolutionary algorithms. The results showed that AMOSA performed better in terms of quality metric.

In [18], mathematical models for the competitive single and multiple allocation HLP in a duopoly market were presented. To solve the problems in single and multiple allocation cases, a simulated annealing based approach was proposed. For multiple allocation SA, two different swap operators were used to generate neighboring solution. On the other hand, the authors used three different operators for single allocation SA. The empirical results indicate that the proposed approach is capable of getting the optimal solutions in short time.

In [16], the authors proposed two simulated annealing approaches for capacitated location-routing problem. The same initialization procedure, neighborhood structures and diversification procedure are used in both approaches. Initial solution is created by the greedy randomized method in which the customers are allocated to the closest facilities. They considered four different neighborhood structures based on insertion and swapping moves. In diversification phase, the state (open/close) of the facilities are changed. The first approach further includes a local search procedure. The results show that the proposed approaches yield better performance according to the objective function and computational time.

Rathore et al. [25] presented a method to solve p-hub median problems using diversification-based learning in simulated annealing. This method utilizes diversification-based learning neighborhood structures to generate a new candidate solution. The results state that the proposed method is able to find the precise solutions for larger instances of HLPs.

3. Multi-objective capacitated multiple allocation hub location problem

In this study, we present meta-heuristic techniques for solving the Multi-objective Capacitated Multiple Allocation Hub Location Problem (referred to as MOCMAHLP) presented in [11]. The problem is considered as a bi-objective discrete optimization problem. In this problem, the number and location of hubs, design of hub-level network (connections between hubs), and allocation of spoke nodes to hub nodes (connections between hubs and spokes) are determined along with routes of the flow.

The features of the problem are listed as follows:

- The problem is a discrete location problem.
- The set of nodes in the network is given.
- The number of hubs to be constructed is not predetermined at the beginning of the optimization process.
- The multiple allocation version of HLP is utilized, that is a spoke can be connected to more than one hub.
- Fixed costs are taken into consideration for establishing hubs and arcs through the network.
- Connection between spokes is not allowed, i.e., the flow of demands must be routed via at least one hub.
- The graph between hubs (i.e., the hub-level network) is not necessarily complete, provided that hub-level network forms a connected graph.
- The flow balance constraint must be satisfied for each node. Total amount of flow entering a node has to be equal to total amount of flow leaving it.
- All hubs and arcs are capacitated. Therefore, in case of capacity overflow, the demand between origin - destination pair (referred to as commodity) may be split into several packages, so that it may follow different paths.

Fig. 1 illustrates an example hub network. This network includes three hubs and seven spokes. The path between two spokes contains three parts: collection (spoke to hub), transfer (between hubs), and distribution (hub to spoke). Due to the capacity constraints, the commodity may be split into several packages. As seen in the figure, the commodity from node 2 to node 10 is divided into two packages: the first and second packages follow the paths indicated by green color and blue color, respectively.

Since the objectives and the features of the problem are the same, in this study, we consider the model formulation given in [11].

Let a_u and a_v be the initial and terminal nodes of arc $a \in E$; F_a, C_a, and T_a be the total amount of flow traversing arc a, unit cost of delivery on arc a, and time required to transfer over arc a, respectively; λ_a be the fixed cost and Q_a the capacity of arc a; i_k and j_k be the source and the destination nodes for commodity k, respectively; for commodity k, D_k be the total demand of flow, while, D_{k_p} be the demand of flow for package p; f_a^k be the flow of commodity k on arc a; κ_h be the fixed cost and Q_h be the capacity of hub h; χ, α, and δ be the discount factor of collection, transfer, and distribution, respectively; $Z_a = 1$ if arc a is used, and 0 otherwise; $x_i = 1$ if node i is hub, and 0 otherwise; $x'_i = 1$ if node i is spoke, and 0 otherwise; $A^a_{k_p} = 1$ if the flow at package p of commodity k is routed through arc a with $Z_a = 1$, and 0 otherwise; $D_{k_p} = n$ where $0 \leqslant n \leqslant D_k$. The mathematical formulation for the MOCMAHLP is as follows [11]:

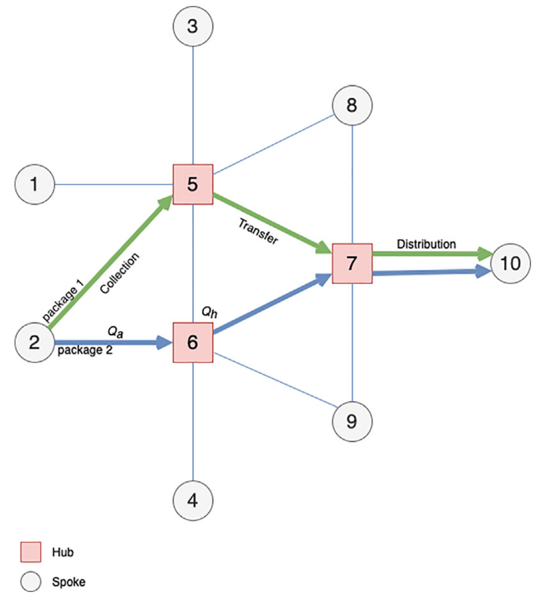

Fig. 1. An example for a hub network.

$$\text{minimize} \sum_a F_a C_a (\chi x'_{a_u} x_{a_v} + \alpha x_{a_u} x_{a_v} + \delta x_{a_u} x'_{a_v}) + \sum_a \lambda_a Z_a + \sum_{i=1}^{N} \kappa_i x_i \quad (1)$$

$$\text{minimize} \max_{k_p} \sum_a T_a A^a_{k_p} \quad (2)$$

subject to

$$1 \leqslant \sum_i x_i \leqslant N \quad (3)$$

$$F_a \leqslant Q_a \quad \forall a \in E \quad (4)$$

$$\sum_{a \in V^+(h)} F_a \leqslant Q_h \quad \forall h \in H \quad (5)$$

$$\sum_p D_{k_p} = D_k \quad \forall k \quad (6)$$

$$A^a_{k_p} \leqslant x_{a_v} Q_h, \quad a_v \neq j_k, \quad \forall k \quad (7)$$

$$0 \leqslant f_a^k \leqslant F_a \quad \forall a \in E \quad (8)$$

for each commodity k

$$\sum_{a \in V^+(v)} f_a^k - \sum_{a \in V^-(v)} f_a^k = \begin{cases} D_k, & \text{if } v = i_k \\ -D_k, & \text{if } v = j_k \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

The two objectives, given in Eqs. (1) and (2) are minimization of total cost and minimization of the maximum travel time required to route flow between all node pairs, respectively. The total cost includes transportation cost (i.e., the sum of collection, transfer and distribution costs) and costs for establishing hubs and operating links (arcs).

The constraints are given in Eqs. (3)–(9). The first constraint (Eq. (3)) assures that at least one hub is established in the network, while second and third constraints (Eqs. (4) and (5)) are capacity constraints for arc and hub, respectively. The summation of flow demand for all packages of commodity k is confirmed to be equal to the total demand of flow for the commodity with the next constraint (Eq. (6)). The constraint given in Eq. (7) guarantees that only hubs are used for routing the demand and Eq. (8) limits the flow of commodity k on arc a to be less than the total amount of flow traversing arc a. Finally, the last constraint given in Eq. (9) enforces the flow balancing constraints.

4. Proposed techniques for solving the MOCMAHLP problem

In this study, the archived multi-objective simulated annealing (AMOSA) [5] and the non-dominated sorting genetic algorithm (NSGA-II) [10] are utilized for solving the MOCMAHLP. In this section, the details of these algorithms are presented. The same solution representation and repair function are considered for both algorithms and presented in the relevant subsections as well. In addition, before evaluating new candidate solutions, A* algorithm is applied for routing in both algorithms.

4.1. Solution representation and constraint handling

The solution representation designed for MOCMAHLP in [11] is used in this study. Two arrays with length equal to the number of nodes in the network are used for each candidate solution. The first array which is used for hub decision contains 0's and 1's, where value 1 denotes that the corresponding node is selected as a hub and value 0 denotes that it is a spoke node. The second array, called link decision array, represents the network design including the inter-hub connections and the spoke-to-hub connections. It is an adjacency list representation in which each entry in the array represents the list of nodes adjacent to the corresponding node.

A sample 10-node network is given in Fig. 1. This network has three hubs (node 5, 6 and 7) and seven spoke nodes. This network is represented by the hub and link decision arrays given in Figs. 2 and 3, respectively. As seen in Fig. 2, the value is 1 for hub nodes and 0 for spoke nodes. For the link decision array given in Fig. 3, for example, the 6^{th} entry represents the list of nodes ($\{2, 4, 5, 7, 9\}$) adjacent to node 6.

The repair operator presented in [11] is used for all infeasible solutions generated by the operators. This repair operator consists of two parts; one used for hub decision array and the other for link decision array. In the given network, there must be at least one hub. Therefore, if there is no hub in the hub decision array of the candidate solution, it is rejected and another random array is created. On the other hand, five different repair cases may occur for link decision array: **(1)** If a link between two nodes is not mutually established, the corresponding link is added to network; **(2)** If a hub does not have connection to any other hub, a link to the nearest hub is established; **(3)** If two spoke nodes are connected, the link between them is deleted; **(4)** If a spoke node is not connected to any of the hubs, a connection to the nearest hub is added; **(5)** In case of a disconnected network, the nearest hub nodes in the connected components of the resulting network are connected.

4.2. AMOSA based approach

The AMOSA [5] based algorithm proposed in this study utilizes an archive to store the non-dominated solutions and a clustering technique to maintain diversity. The parameters of the AMOSA are as follows: hard limit (HL) is the maximum number of non-dominated solutions in the archive; soft limit (SL) is the maximum number of solutions that may be included in the archive before clustering; α is the cooling rate; t_{max} and t_{min} are the initial and final temperature values, respectively.

The pseudo-code of AMOSA algorithm is presented in Algorithm 1. The algorithm starts with archive initialization procedure (line 1). In this procedure, after generating a number of random initial solutions with size $\gamma \times SL$, a hill-climbing method is applied to each solution. Then, non-dominated solutions are stored in the archive for later use. If the number of non-dominated solutions exceeds HL, the size of archive is reduced to HL by clustering. A solution is randomly selected from the archive as the initial solution and named *current solution* (line 2). For each temperature value, hub decision array of the current solution is perturbed once (line 4) while the link decision array is perturbed for some number of iterations (line 6) by the corresponding neighborhood structure to create a candidate solution and temperature value is reduced at the end of the outer loop (line 13). After perturbation, a repair function is applied to the candidate solution to make the solution feasible. A* algorithm is applied for routing; then, the candidate solution is evaluated. The candidate solution is either accepted (and replaced with current solution) or rejected according to acceptance rules as defined in [5]. These acceptance rules include three different cases:

Fig. 2. The hub decision array for the network given in Fig. 1.

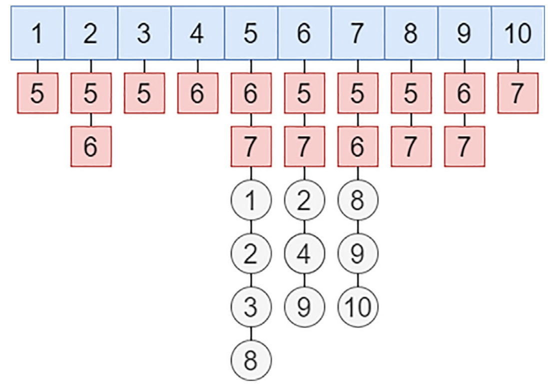

Fig. 3. The link decision array for the network given in Fig. 1.

Case 1. *Current solution dominates candidate solution*: Acceptance probability is calculated according to average domination of candidate solution among solutions in the archive. Then, candidate solution is accepted as current solution with the calculated probability.

Case 2. *Current solution and candidate solution are non-dominating to each other*: The domination status of the candidate solution and the solutions in the archive is tested:

a) If a number of solutions in the archive dominate the candidate solution, then the candidate solution is accepted as the current solution with probability of the average amount of domination.
b) If none of the solutions in the archive dominates the candidate solution, it becomes the current solution and is inserted into the archive. If the number of solutions in the archive reaches SL, clustering is applied to reduce the number of solutions to HL.
c) If the candidate solution dominates a number of solutions in the archive, it is accepted and inserted into the archive. These dominated solutions are deleted from the archive.

Case 3. *Current solution is dominated by candidate solution*: The domination status of the candidate solution and the solutions in the archive is tested:

a) If the candidate solution is dominated by a number of solutions in the archive, the solution in the archive with the minimum domination difference with candidate solution is picked as the current solution. Otherwise, the candidate solution becomes the current one.

b) If none of the solutions in the archive dominates the candidate solution, the candidate solution is accepted and inserted into the archive. In case the archive includes the current solution, it is deleted; otherwise, if there is no available room in the archive ($archive\_size > SL$), clustering is applied to reduce the number of solutions to HL.
c) If the candidate solution dominates a number of solutions in the archive, the candidate solution is accepted and inserted into the archive. These dominated solutions are deleted from the archive.

Algorithm 1 The pseudo-code of AMOSA based solution

```
Initialize the archive
Select a solution from the archive, i.e., current solution
while minimum temperature not reached
    Apply the neighborhood structure to the hub decision
    array of the current solution and generate a candidate
    solution
    while maximum number of iterations not reached
        Perturb the candidate solution by using a
        neighborhood structure to the link decision array
        Apply repair function
        Apply A* algorithm for routing
        Evaluate the candidate solution
        Accept/reject the candidate solution according to
        selection rule
        Update archive if needed
    end while
    Update temperature
end while
```

4.2.1. Neighborhood structures

The solution representation consists of a hub decision and a link decision array, thus, we consider different types of neighborhood structures for each array. We propose one neighborhood structure for hub decision array and five different neighborhood structures for link decision array.

The neighborhood structure for hub decision array selects a random node. If the selected node is hub, then it becomes a spoke and vice versa. If there is only one hub in the hub decision array, a spoke node is picked randomly and converted to a hub node.

Five problem-specific neighborhood structures (NS) designed for link decision array are listed below:

- **Swap all links (NS-1)**: This method selects a node randomly. If the selected node is a hub, then another hub node is also selected randomly. If there is no other hub in the solution, this method is not applied. If selected node is a spoke, then another spoke node is selected. After selecting two hub/spoke nodes, all of their links are exchanged.
- **Add/Remove K links (NS-2)**: In this method, first, we decide whether to add or remove links with equal probability. Based on the decision, the predetermined number (K) of links are either added to or removed from the candidate solution. The value of K can be determined empirically based on the number of nodes in the network.
- **Adaptive add/remove K links (NS-3)**: This method starts with an initial value of K. At each step, K links are either added to or removed from the candidate solution. The value of K is increased if there is no improvement for a fixed number of iterations; otherwise, it is decreased. Also, it is allowed to vary within a predetermined range ($[K_{min}, K_{max}]$).
- **Random add/remove K links (NS-4)**: This method is the same as the *add/ remove K links* except that the value of K is determined randomly within a predetermined range ($[K_{min}, K_{max}]$) at each iteration.
- **Add/Remove K links according to the temperature (NS-5)**: The value of K is decreased according to temperature value starting with a maximum value of K. In this method, first, the number of different temperature values (denoted by $t_{changed}$) between maximum temperature and minimum temperature is calculated:

$$t_{changed} = \frac{t_{max} - t_{min}}{\alpha} + 1 \tag{10}$$

Then, the number of different K values is computed:

$$K_{changed} = K_{max} - K_{min} + 1 \tag{11}$$

Finally, the value of K is decreased by 1 for every i iterations:

$$i = \left\lceil \frac{t_{changed}}{K_{changed}} \right\rceil \tag{12}$$

That is, the temperature is at its maximum (minimum) for the maximum (minimum) value of K.

4.3. NSGA-II based approach

In this study, two new problem-specific mutation operators for NSGA-II-based algorithm presented in [11] are introduced. In our NSGA-II algorithm, the initial population is created randomly. Aforementioned A* algorithm is applied to each candidate solution for routing, the objective values of each candidate solution are then calculated. The algorithm uses basic genetic algorithm operators to generate new solutions, i.e., binary tournament selection, one-point crossover, and bit mutation and exchange mutation for hub decision array and link decision array, respectively. The mutation operator including bit mutation and exchange mutation is referred to as swap all links mutation (SALM) in this paper. The non-dominated and crowding distance sorting method is applied to pick the individuals for the next generation. In the new mutation operators proposed in this paper, namely single-link mutation and centrality based mutation, the operator is applied to all nodes in both hub decision array and link decision array.

In **single-link mutation**, bit-flip mutation is applied to hub decision array, which changes a spoke to hub or a hub to spoke. For link decision array, a link is either added to or removed from the node to be mutated with equal probability. However, if there is no available link (i.e., if the node to mutate is a hub, either hub or spoke node can be connected; otherwise a hub node must be added) to be added for the node, a link will be removed from it. On the other hand, if there is no available link to be removed from the node, a link will be added to that node.

In **centrality based mutation**, status of a node (spoke to hub or hub to spoke) is changed based on the centrality of a node (mutation on hub decision array). The centrality of node i is calculated as follows: First, the average distance of node i to all other nodes ($Dist_i$) is calculated as:

$$Dist_i = \frac{\sum_{j=1}^{N} dist_{ij}}{N} \tag{13}$$

where $dist_{ij}$ is the distance between node i and node j. Then, $Dist_i$ values are normalized according to min–max normalization. The average demand of node i (W_i) is computed as:

$$W_i = \frac{\sum_{j=1}^{N} w_{ij} + w_{ji}}{N} \tag{14}$$

where w_{ij} is the demand for flow between node i and node j. W_i values are also normalized according to min–max normalization. Finally, the centrality of node i is:

$$centrality_i = Dist_i + W_i, \quad centrality_i \in [0,2] \tag{15}$$

In centrality based mutation, for the hub decision array, if the node to mutate is a hub, another hub is selected randomly. These two hubs are compared according to their centrality values. Then, the hub with lower centrality value becomes a spoke. Similarly, if the node to mutate is a spoke, another spoke is selected at random and the one with higher centrality value is changed to be a hub. On the other hand, in the link decision array, for the node to mutate, two available nodes are selected randomly. The node is either connected to the closer of these two nodes or disconnected from the further one.

5. Experimental setup

In this section, we present the performance evaluation metric and the data sets used in the experimental study.

In evolutionary multi-objective optimization problems for which the true Pareto front is not known, the most widely used metric for performance evaluation is hypervolume indicator [36]. Thus, we used hypervolume ratio to evaluate the performance of our solution approaches. Hypervolume ratio is calculated as the ratio of the area covered by the obtained Pareto front to the area covering all possible objective values. Before conducting the experimental study, we executed the proposed approaches 10 times with each data set for 100000 fitness evaluation counts. The worst objective obtained in these 5 runs is selected as the reference point for the hypervolume indicator.

In the experimentation, for each setting of parameters, we conducted 30 independent runs. The Average and Std. Dev. columns in the result tables are the average and standard deviation values of hypervolume for these 30 runs, respectively. The hypervolume values given in these tables are smaller than the hypervolume values on [11] since the magnitude of hypervolume indicator mainly depends on the chosen reference point [7].

Both algorithms are implemented in open source Java library, namely the MOEA framework. The source code for the algorithms is publicly available.[1] The experiments are performed on the computer with 8 GB RAM and Intel Core i7-2600K Processor.

5.1. Data sets

In the experimental study, we used two data sets, namely, Australian Post (AP) [14] and Turkish data sets. These data sets are selected since they are two frequently used data sets of the hub location literature. The data sets can be obtained from OR-Library [6].

In the Turkish data set, 81 cities of Turkey are included as nodes. The data set introduces the distance and travel time between nodes, fixed hub costs [29], fixed link costs [3], and flow data [8]. There is no hub and link capacity information in the data set, so we used the capacities as defined in [11]. To generate hub capacity for each city, the population size of the corresponding city is used, while, the link capacities are generated according to the distance and travel time for the corresponding link.

AP data set contains 200 postal districts of Australia. The data set includes the coordinates of the districts, fixed hub costs, hub capacities, and flow data. The Euclidean distance between nodes is calculated using the coordinates, and travel time is calculated according to the distance between nodes (an average speed of 90 km/h is considered as in Turkish data set). Fixed link costs and link capacities are not included in the data set. Since there is a direct correlation between distance and link cost, fixed link cost is set to distance between nodes. Finally, link capacity is calculated as, multiplication of total direct flow between nodes and link capacity multiplier. We set fixed hub cost multiplier as 1, the fixed cost multiplier as 10, and total transfer cost multiplier as 10^{-7}. These values are determined according to results of preliminary experiments. The AP data set is originally a 200-node network and different size networks can be derived from the original data set. The link capacity multiplier is selected as 100 for all size networks. In this paper, for the fine-tuning and selection of the operators in our algorithms, we propose the computational results obtained for 100-node AP data set. For both Turkish and AP data sets, the values of discount factor for collection, transfer and distribution are set to $\chi = 1, \alpha = 0.9$, and $\delta = 1$, respectively.

6. Results and discussion

In this section, we report and discuss the results of the experiments. The experimental study is conducted in three phases. In the first phase, the performance comparison of mutation operators designed for the NSGA-II approach (see Section 4.3) is carried out. In the second phase, the effect of neighborhood operators (see Section 4.2) on the performance of AMOSA is evaluated and in the third phase, the best setting of NSGA-II and AMOSA are compared. We also perform a scalability analysis to our approaches for three different network sizes.

6.1. Comparison of NSGA-II mutation operators

In this part of the experiments, we tested three different mutation operators in NSGA-II. These operators include swap all links mutation (SALM) proposed in [11], single-link mutation (SLM), and centrality based mutation (CBM). We experimented with various mutation probability settings for each mutation operator as:

- **SALM:** $0.005, 0.01, 0.02, 0.05, 0.1, 0.2$
- **SLM:** $0.01, 0.05, 0.1, 0.15, 0.2, 0.3$
- **CBM:** $0.01, 0.05, 0.1, 0.15, 0.2, 0.3$

Higher mutation probabilities are used for SLM and CBM, since they offer a very small change to the individual compared with SALM. Other parameters in NSGA-II are set as recommended in [11]: crossover probability is 0.8, mutation probability for the hub decision array is $1/n$, population size is 200, and the maximum number of fitness function evaluation is 20000.

Table 1 presents the average and standard deviation of hypervolume values obtained for each of 3 different mutation operators with different mutation probabilities. All tests are performed for both Turkish and 100-node AP data sets. As it is stated before, for each probability value of each mutation type, the algorithm is executed 30 times with different seeds. We also carried out Tukey's honestly significant difference (HSD) post hoc test to see if there is any statistical difference between mutation probabilities. As a result of statistical analysis of Turkish data set results, we can say that: **(1)** for SALM, mutation rates higher than 0.02 produce better performance than smaller mutation rates. However, there is no statistically significant difference between these higher mutation rates. **(2)** for SLM, when mutation rate is 0.01, the algorithm performs worse than the others. On the other hand, mutation rates higher than 0.01 have equal performance. **(3)** for CBM, only mutation rate 0.1 performs better than mutation rate 0.01, the other pairs all have the same performance.

[1] https://github.com/MOEAFramework/MOEAFramework/pull/227

Table 1
Average hypervolume values and standard deviation obtained by mutation operators with various mutation probability values on Turkish and 100-node AP data sets.

Mutation	p_m	Turkish data set		AP data set	
		Average	Std. Dev.	Average	Std. Dev.
SALM	0.005	0.6638	0.0004	0.6003	0.0109
	0.01	0.6639	0.0003	**0.6022**	**0.0160**
	0.02	0.6641	0.0002	0.5975	0.0130
	0.05	0.6642	0.0002	0.5880	0.0134
	0.1	0.6642	0.0002	0.5786	0.0127
	0.2	**0.6643**	**0.0001**	0.5659	0.0130
SLM	0.01	0.6629	0.0015	**0.6019**	**0.0127**
	0.05	0.6637	0.0014	0.5928	0.0151
	0.1	**0.6639**	0.0006	0.5854	0.0171
	0.15	**0.6639**	0.0006	0.5768	0.0186
	0.2	0.6638	0.0007	0.5702	0.0161
	0.3	0.6637	0.0004	0.5627	0.0170
CBM	0.01	0.6564	0.0082	0.6151	0.0120
	0.05	0.6612	0.0056	0.6267	0.0139
	0.1	**0.6617**	**0.0052**	0.6345	0.0122
	0.15	0.6612	0.0068	0.6377	0.0125
	0.2	0.6603	0.0067	0.6375	0.0122
	0.3	0.6596	0.0072	**0.6433**	**0.0118**

The best setting for each operator is marked in bold.

For Turkish data set, higher hypervolume values are usually produced with higher mutation rates. However, there is no real effect of mutation rate on hypervolume values.

On the other hand, when we interpret the results of statistical analysis for 100-node AP data set, we can conclude that: **(1)** for SALM, higher mutation rates always deliver poorer performance than the smaller mutation rates. However, there is no statistically significant difference between mutation rates of 0.005, 0.01, and 0.02. **(2)** for SLM, smaller mutation probabilities achieve slightly better performance than higher ones. **(3)** for CBM, higher mutation rates generate better performance, and there is no significant difference between mutation rates higher than 0.1.

Considering the hypervolume values given in the table, we can say that SALM and SLM yield slightly better results with smaller mutation probabilities, for AP data set. However, for CBM hypervolume value increases with increase in the mutation rate. This can be because CBM leads to a small change on the solution.

From the results given in the table and the statistical analysis, we can conclude that there is no intrinsic effect of mutation probability on hypervolume values for this problem. Most of the mutation rates have similar performance. That is most probably because of the repair function, that we apply after each mutation operation resulting in an infeasible solution.

We also applied a statistical analysis to compare the three mutation types using Tukey's honestly significant difference (HSD) post hoc test. In the statistical analysis for Turkish data set, we compared the results for mutation rate of 0.1 for all three types of mutation. For AP data set, on the other hand, the results of SLM and SALM with mutation rates of 0.01 and CBM with mutation rate of 0.3 are used. When we check the results of Tukey's post hoc test, for Turkish data set, SALM is seen to have better performance than SLM and CBM. This observation can be seen in Fig. 4, which presents the boxplot of hypervolume values for three mutation operators in NSGA-II on Turkish data set.

For AP data set, the boxplot of hypervolume values for different mutation operators is given in Fig. 5. In this case, CBM has higher hypervolume value than other mutation types. Thus, we used CBM to compare NSGA-II and AMOSA at the end of the experimental study.

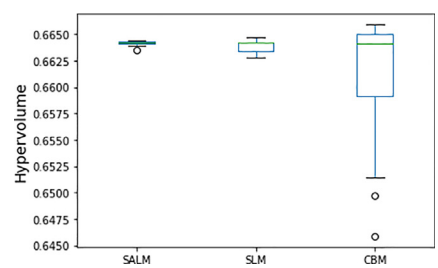

Fig. 4. Boxplot of hypervolume values for a statistical comparison of different mutation operators in NSGA-II on Turkish data set.

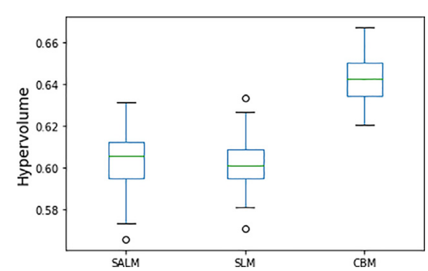

Fig. 5. Boxplot of hypervolume values for a statistical comparison of different mutation operators in NSGA-II on AP data set.

Table 2
Parameter setting of AMOSA.

Parameter	Value
Iteration per temperature	10
Soft Limit	100
Hard Limit	10
γ	2
α	0.9882
t_{min}	10^{-7}
t_{max}	200

6.2. Comparison of neighborhood operators in AMOSA

In this part, we experimented with five neighborhood operators in AMOSA: Swap all links (NS-1), add/remove K links (NS-2), adaptive add/remove K links (NS-3), random add/remove K links (NS-4), and add/remove K links according to temperature (NS-5). We conducted some experiments for the best parameter setting specific to each neighborhood operator. For all neighborhood operators, we

Table 3
Average hypervolume values and standard deviation obtained by NS-2 with different *K* values on Turkish and AP data sets

K	Turkish data set		AP data set	
	Average	Std. Dev.	Average	Std. Dev.
1	0.5773	0.1343	0.4874	0.1453
5	**0.6214**	**0.0230**	0.5843	0.0551
10	0.6106	0.0275	**0.5994**	**0.0463**
15	0.5934	0.1073	0.5826	0.0375
20	0.6078	0.0639	0.5929	0.0400
30	0.5981	0.0657	0.5768	0.0373

Table 4
Average hypervolume values and standard deviation obtained by neighborhood structures on Turkish and AP data sets

NSs	Turkish data set		AP data set	
	Average	Std. Dev.	Average	Std. Dev.
NS-1	0.4036	0.2534	0.1714	0.2677
NS-2	0.6106	0.0275	0.5994	0.0463
NS-3	0.5771	0.1048	0.4759	0.1478
NS-4	0.6058	0.0799	0.6060	0.0440
NS-5	**0.6177**	**0.0324**	**0.6239**	**0.0508**

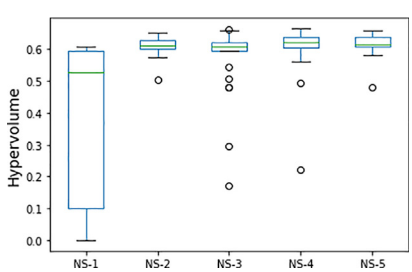

Fig. 6. Boxplot of hypervolume values for a statistical comparison of different neighborhood operators in AMOSA on Turkish data set.

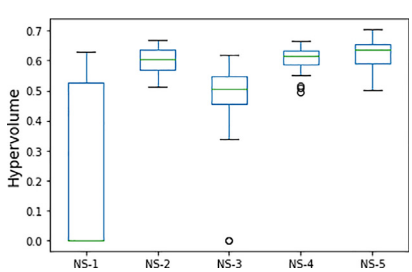

Fig. 7. Boxplot of hypervolume values for a statistical comparison of different neighborhood operators in AMOSA on AP data set.

consider the parameter setting for AMOSA given in Table 2. The parameter setting is decided according to some preliminary experiments based on the setting recommended in [5].

For NS-2, we experimented with six K values: $1, 5, 10, 15, 20$, and 30. Table 3 presents the results of NS-2 with various K values on Turkish and 100-node AP data sets. The results show that the differences between different K values are not statistically significant for Turkish data set. On the other hand, for AP data set, there are no statistically significant difference between most cases except for the setting of $K = 1$. We can conclude that the setting of the parameter is not very sensitive. Therefore, we use $K = 10$ which provides an acceptable performance for both data sets for the rest of the experiments.

Table 5
Average hypervolume values and standard deviation obtained by AMOSA and NSGA-II on Turkish and AP data sets

Algorithms	Turkish data set		AP data set	
	Average	Std. Dev.	Average	Std. Dev.
NSGA-II	**0.6643**	0.0001	**0.6433**	0.0118
AMOSA	0.6177	0.0324	0.6239	0.0508

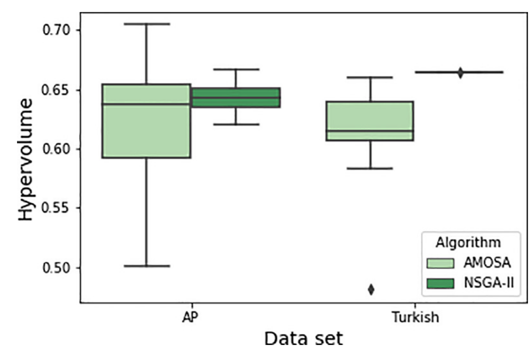

Fig. 8. Boxplot of hypervolume values for a statistical comparison of two approaches for the AP and Turkish data sets.

According to the results obtained in the previous part (see Table 3), we set $k_{min} = 1$ and $k_{max} = 20$ for NS-3, NS-4, and NS-5. Additionally, NS-3 starts with the initial value of $k = 10$. Table 4 provides the average hypervolume results and standard deviation for the neighborhood structures on Turkish and AP data sets. NS-1 yields the worst performance for both data sets, while the best performance is achieved when NS-5 is used as a neighborhood structure.

Fig. 6 shows the boxplot for different neighborhood operators on Turkish data set. Based on the statistical results of Turkish data set, it can be said that: **(1)** NS-1 is significantly outperformed by all other methods. **(2)** Among the remaining neighborhood structures (NS-2 to NS-5), NS-5 seems to generate better results than the others; however, the performance differences are not statistically significant. On the other hand, for AP data set, we can say that: **(1)** NS-1 is again the worst performing neighborhood structure. **(2)** NS-3 performs significantly worse than NS-2, NS-4 and NS-5. **(3)** There are no statistically significant differences between the results of NS-2, NS-4 and NS-5. This results are illustrated as a boxplot of hypervolume values given in Fig. 7.

As mentioned before, NS-5 produces better performance for both data sets. At the beginning of the search, NS-5 starts with the highest K value which results in more changes in the network, which leads to diversification. During the search, decreasing K value (less change in the network) provides intensification. This is the main reason behind the success of NS-5.

6.3. Comparison of NSGA-II and AMOSA

In this part, we compare the performance of NSGA-II and AMOSA with the best parameter setting provided in previous sections. Based on the results, SALM with the probability of 0.2 is selected as mutation operator in NSGA-II and NS-5 is selected as the neighborhood structure in AMOSA for Turkish data set. On the other hand, for AP data set, NS-5 and CBM with the probability of 0.3 are selected in AMOSA and NSGA-II, respectively. The corresponding results are provided in Table 5 for both data sets. More-

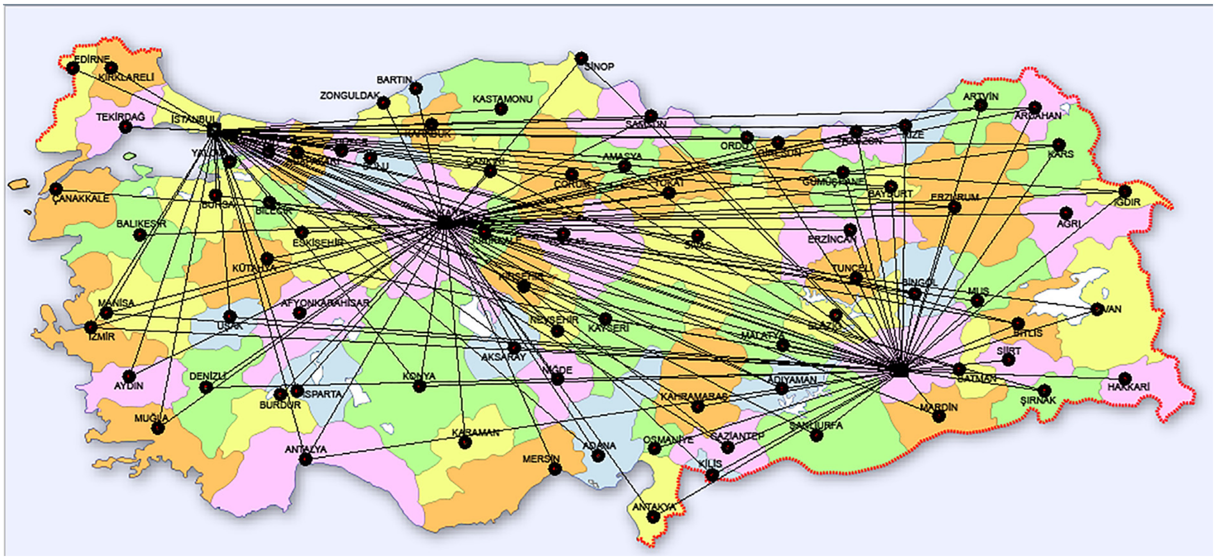

Fig. 9. A Pareto optimal solution indicating a network with all hub-to-hub and hub-to-spoke connections on Turkish data set produced by AMOSA based approach.

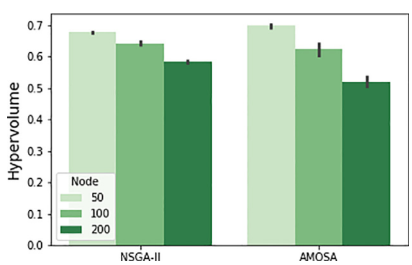

Fig. 10. Bar plot of average hypervolume values with error bars obtained by AMOSA and NSGA-II for analyzing the effect of number of nodes.

over, Fig. 8 presents the boxplot of hypervolume values for the statistical comparison of two approaches for the AP and Turkish data sets. Based on the results, it can be seen that the NSGA-II outperforms the AMOSA.

A visualization of one of the Pareto optimal solutions is given in Fig. 9 indicating a network with all hub-to-hub and hub-to-spoke connections on Turkish data set produced by AMOSA based approach. As seen in the figure, the hubs are chosen as (İstanbul, Ankara, and Diyarbak), which are the most populated cities in different geographical regions (Marmara, Central Anatolia, and Southeastern Anatolia) of Turkey. In the data set, it is noticed that most of the flow of eastern cities are to west part of Turkey. Accordingly, in this solution, the cities in the eastern part of Turkey are mostly connected to two or more hubs in the solution.

As the last part of experimental study, we investigate the scalability of the proposed approaches for AP data set. We perform experiments with different number of nodes: $50, 100, 200$. Fig. 10 presents the bar plot with error bars of the results for analyzing the effect of number of nodes on the performance. Based on the results, for 50-node instance, AMOSA based approach delivers better performance. On the other hand, NSGA-II outperforms AMOSA for larger data sets. According to the results, we conclude that both NSGA-II and AMOSA approaches can find a feasible solution for any size of data sets.

7. Conclusion and future work

In this study, we presented two new mutation operators in NSGA-II and an AMOSA-based algorithm to solve the multi-objective capacitated multiple allocation hub location problem. Among different objectives, in this study, we considered two objectives: the minimization of total cost and the minimization of the maximum travel time. In the given problem, the decisions to be made are as follows: the number and location of hubs, the design of hub-level network, the allocations of spoke to hub nodes, and the routing of the flow. For both approaches, the solution representation consisted of two arrays: hub decision and link decision array. A^* algorithm was used for routing. For NSGA-II, three different problem-specific mutation operators, two of which proposed in this study, were considered. In this study, AMOSA was adapted to solve the MOCMAHLP for the first time and five problem-specific neighborhood operators were presented. The performance of meta-heuristic approaches addressed in this study were tested on Turkish and AP data sets.

As the experimental study, first, we performed analysis of mutation operators in NSGA-II under different mutation probabilities for both data sets. For Turkish data set, the results showed that the algorithms are not very sensitive to mutation probabilities. Among mutation operators, SALM is the best performing mutation operator. On the other hand, for AP data set, the mutation operator which results in small changes in the network produces better performance. SLM and SALM with lower mutation probability deliver better performance than those with higher probabilities. In addition, CBM significantly outperforms the other mutation operators.

In the second part of the experimental study, we conducted experiments using different neighborhood operators to determine the best one in AMOSA for both data sets. In this part, first, the best parameter settings for each neighborhood operator were determined. Then, the performance of the neighborhood operators with best parameter settings were compared. Based on the results, NS-5 yielded better performance for both data sets. NS-5 achieved a good balance between intensification and diversification.

In the last part of experimental study, the performance of NSGA-II and AMOSA with their best settings were compared. The results show that NSGA-II significantly outperforms AMOSA for larger size networks. As a future work, we will confirm our findings on other variants of HLP.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. Abbasi, R. Niknam, A combined genetic algorithm and simulated annealing approach for solving competitive hub location and pricing problem, International Journal of Applied Management Science 9 (2017) 188–202.
[2] Y. Alizadeh, R. Tavakkoli-Moghaddam, S. Ebrahimnejad, A new multi-objective model for a capacitated hub covering problem solving by two multi-objective evolutionary algorithms, International Journal of Mathematics in Operational Research 9 (2016) 99–124.
[3] S.A. Alumur, B.Y. Kara, O.E. Karasan, The design of single allocation incomplete hub networks, Transportation Research Part B: Methodological 43 (2009) 936–951.
[4] K. Amine, Multiobjective Simulated Annealing: Principles and Algorithm Variants, Advances in Operations Research 2019 (2019) 8134674.
[5] S. Bandyopadhyay, S. Saha, U. Maulik, K. Deb, A simulated annealing-based multiobjective optimization algorithm: AMOSA, IEEE Transactions on Evolutionary Computation 12 (2008) 269–283.
[6] J.E. Beasley, accessed September, 2019. OR-Library. url:http://people.brunel.ac.uk/ mastjjb/jeb/orlib/phubinfo.html..
[7] Y. Cao, B.J. Smucker, T.J. Robinson, On using the hypervolume indicator to compare Pareto fronts: Applications to multi-criteria optimal experimental design, Journal of Statistical Planning and Inference 160 (2015) 60–74.
[8] S. Çetiner, C. Sepil, H. Süral, Hubbing and routing in postal delivery systems, Annals of Operations Research 181 (2010) 109–124.
[9] K. Danach, S. Gelareh, R. Neamatian Monemi, The capacitated single-allocation p-hub location routing problem: a Lagrangian relaxation and a hyper-heuristic approach, EURO Journal on Transportation and Logistics 8 (2019) 597–631.
[10] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2002) 182–197.
[11] I. Demir, F. Corut Ergin, B. Kiraz, A new model for the multi-objective multiple allocation hub network design and routing problem, IEEE Access 7 (2019) 90678–90689.
[12] A. Ebrahimi Zade, A. Sadegheih, M.M. Lotfi, A modified NSGA-II solution for a new multi-objective hub maximal covering problem under uncertain shipments, Journal of Industrial Engineering International 10 (2014) 185–197.
[13] M. Eghbali, M. Abedzadeh, M. Setak, Multi-objective reliable hub covering location considering customer convenience using NSGA-II, International Journal of System Assurance Engineering and Management 5 (2014) 450–460.
[14] A.T. Ernst, M. Krishnamoorthy, Efficient algorithms for the uncapacitated single allocation p-hub median problem, Location Science 4 (1996) 139–154.
[15] R.Z. Farahani, M. SteadieSeifi, N. Asgari, Multiple criteria facility location problems: A survey, Applied Mathematical Modelling 34 (2010) 1689–1709.
[16] K.M. Ferreira, T.A. de Queiroz, Two effective simulated annealing algorithms for the Location-Routing Problem, Applied Soft Computing 70 (2018) 389–422.
[17] R. Geramianfar, M. Pakzad, H. Golhashem, R. Tavakkoli-Moghaddam, A multi-objective hub covering location problem under congestion using simulated annealing algorithm, Uncertain Supply Chain Management 1 (2013) 153–164.
[18] N. Ghaffarinasab, A. Motallebzadeh, Y. Jabarzadeh, B.Y. Kara, Efficient simulated annealing based solution approaches to the competitive single and multiple allocation hub location problems, Computers & Operations Research 90 (2018) 173–192.
[19] S.Y. Hsieh, S.S. Kao, A survey of hub location problems, Journal of Interconnection Networks 19 (2019) 1940005.
[20] B. Kara, B. Tansel, On the allocation phase of the p-hub location problem. Technical Report, Department of Industrial Engineering, Bilkent University, 1998.
[21] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[22] Q. Liu, X. Li, H. Liu, Z. Guo, Multi-objective metaheuristics for discrete optimization problems: A review of the state-of-the-art, Applied Soft Computing 93 (2020) 106382.
[23] P. Pirozmand, A.A.R. Hosseinabadi, M. Farrokhzad, M. Sadeghilalimi, S. Mirkamali, A. Slowik, Multi-objective hybrid genetic algorithm for task scheduling problem in cloud computing, Neural Computing and Applications (2021).
[24] M. Randall, Solution approaches for the capacitated single allocation hub location problem using ant colony optimisation, Computational Optimization and Applications 39 (2008) 239–261.
[25] H. Rathore, S. Nandi, P. Pandey, S.P. Singh, Diversification-based learning simulated annealing algorithm for hub location problems, Benchmarking: An International Journal 26 (2019) 1995–2016.
[26] F. Rayat, M. Musavi, A. Bozorgi-Amiri, Bi-objective reliable location-inventory-routing problem with partial backordering under disruption risks: A modified AMOSA approach, Applied Soft Computing 59 (2017) 622–643.
[27] S. Shamshirband, M. Shojafar, A.A.R. Hosseinabadi, A. Abraham, A solution for multi-objective commodity vehicle routing problem by nsga-ii, in: 2014 14th International Conference on Hybrid Intelligent Systems, 2014, pp. 12–17.
[28] B. Suman, P. Kumar, A survey of simulated annealing as a tool for single and multiobjective optimization, Journal of the Operational Research Society 57 (2006) 1143–1160.
[29] P.Z. Tan, B.Y. Kara, A hub covering model for cargo delivery systems, Networks 49 (2007) 28–39.
[30] L. Tavakoli, N. Manavizadeh, M. Rabbani, Multi-objective multiple allocation hub location problem with multiple capacity levels considering sustainable development paradigm, International Journal of Applied Decision Sciences (2014) 295–326.
[31] E.B. Tirkolaee, M. Alinaghian, A.A.R. Hosseinabadi, M.B. Sasi, A.K. Sangaiah, An improved ant colony optimization for the multi-trip capacitated arc routing problem, Computers & Electrical Engineering 77 (2019) 457–470.
[32] V. Tongur, M. Hacibeyoglu, E. Ulker, Solving a big-scaled hospital facility layout problem with meta-heuristics algorithms, Engineering Science and Technology, an International Journal 23 (2020) 951–959.
[33] X.S. Yang, Nature-inspired optimization algorithms: Challenges and open problems, Journal of Computational Science 46 (2020) 101104.
[34] Q. Yuchi, N. Wang, S. Li, Z. Yang, B. Jiang, A bi-objective reverse logistics network design under the emission trading scheme, IEEE Access (2019). 1–1.
[35] A. Zabihi, M. Gharakhani, A literature survey of HUB location problems and methods with emphasis on the marine transportations, Uncertain Supply Chain Management (2018) 91–116.
[36] E. Zitzler, D. Brockhoff, L. Thiele, The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration, in: S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, T. Murata (Eds.), Evolutionary Multi-Criterion Optimization, Springer, Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 862–876.