



Genetic and Improved Shuffled Frog Leaping Algorithms for a 2-Stage Model of a Hub Covering Location Network

M. Mohammadi, R. Tavakkoli-Moghaddam^{*}, A. Ghodratnama & H. Rostami

Mehrdad Mohammadi is a master student in Department of Industrial Engineering, College of Engineering, University of Tehran, Iran.

Reza Tavakkoli-Moghaddam is a professor in Department of Industrial Engineering, College of Engineering, University of Tehran, Iran.

Ali Ghodratnama is a Ph.D. student in Department of Industrial Engineering, College of Engineering, University of Tehran, Iran.

Hamideh Rostami is a student in Department of Industrial Engineering, College of Engineering, University of Tabriz, Iran.

KEYWORDS

Hub covering location problem,
Network design,
Single machine scheduling,
Genetic algorithm,
Shuffled frog leaping algorithm

ABSTRACT

Hub location problems (HLP) are synthetic optimization problems that appears in telecommunication and transportation networks where nodes send and receive commodities (i.e., data transmissions, passengers transportation, express packages, postal deliveries, etc.) through special facilities or transshipment points called hubs. In this paper, we consider a central mine and a number of hubs (e.g., factories) connected to a number of nodes (e.g., shops or customers) in a network. First, the hub network is designed, then, a raw materials transportation from a central mine to the hubs (i.e., factories) is scheduled. In this case, we consider only one transportation system regarded as single machine scheduling. Furthermore, we use this hub network to solve the scheduling model. In this paper, we consider the capacitated single allocation hub covering location problem (CSAHCLP) and then present the mixed-integer programming (MIP) model. Due to the computational complexity of the resulted models, we also propose two improved meta-heuristic algorithms, namely a genetic algorithm and a shuffled frog leaping algorithm in order to find a near-optimal solution of the given problem. The performance of the solutions found by the foregoing proposed algorithms is compared with exact solutions of the mathematical programming model.

© 2011 IUST Publication, IJIEPR, Vol. 22, No. 3, All Rights Reserved.

1. Introduction

1.1. Hub Location Problem

Hub-and-spoke systems are common in many areas of everyday life from passenger travel through an airline's network in airports, to postal delivery system,

communication, and public transportation networks. Hub networks have their origin in transportation and telecommunication systems where several origin/destination points transfer and receive a number of commodities.

The key feature of these networks is to route products via a specific subset of links, rather than routing each product with a direct link from its origin to its destination point because direct links have expenses more than indirect links network. These applications have led to increase the related research and the

^{*} Corresponding author. R. Tavakkoli-Moghaddam

Email: tavakoli@ut.ac.ir

Paper first received July 13, 2011, and in revised form Sep. 07, 2011.

number of researchers on hub location problems. In particular, hub networks use a set of hub nodes to consolidate and reroute the flows, and a reduced number of links, where economies of scale are applied, to connect the (usually large) set of origins/destination points.

Also Hub Location Problem (HLP) considers the location of a set of hub nodes and the design of the hub network.

In the literature, four major types of hub location problem exist, namely capacitated and uncapacitated hub location problem, p -hub median problem, p -hub center problem, and hub covering location problem. In hub location problem (HLP), the objective is to minimize the total cost of locating hubs and transporting cargo flows through the hub network. The capacity of each hub may be limited (LHLP) or unlimited (UHLP).

Most real-case problems are LHLP. In the p -hub median problem (p HMP), the objective is to locate p hubs in the network so that the total cost of transporting flows through the network is minimized. Unlike the UHLP, the number of hub is given as input. In the p -hub center problem (p HCP), the objective is to find the optimal location of p hubs and the allocation of non-hub nodes to the hubs and minimize the longest path in the network.

In the hub covering location problem (HCLP), the number of hubs is not given and the objective is to find the best location of hubs in the network and allocation of nodes to hubs such that the total cost of locating the hubs is minimized.

The HCLP contains cover constraints, which limit the number of non-hub nodes that can allocate to each hub. Three coverage criteria for hubs have been defined as follows [1]. The origin destination pair (i, j) is covered by hubs k and m if:

- I. The cost from i to j via k and m does not exceed a specified value,
- II. The cost for each link in the path from i to j via k and m does not exceed a specified value,
- III. Each of the origin-hub and hub-destination links meets separate specified values.

All of the above types are divided in two major parts; namely single and multiple allocations hub location problems.

In single allocation hub networks, each non-hub node is allocated to exactly one hub; in multiple allocation networks, a non-hub node can be allocated to more than one hub. The single and multiple allocations are shown in Fig. 1.

In this paper we consider a single allocation capacitated hub covering location problem. Besides, the hub nodes are assumed to be fully interconnected as shown in Fig. 1, where the squares and circles represent hub and non-hub nodes, respectively.

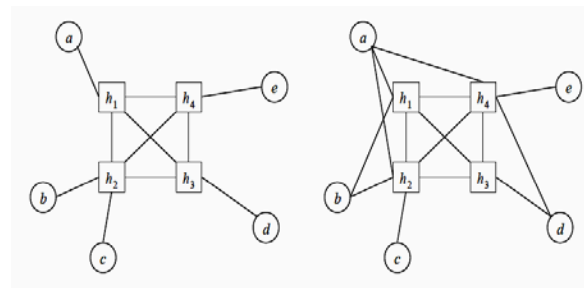


Fig. 1. Hub-and-spoke network designs.

O'Kelly [2] first introduced the hub location problem. Further, O'Kelly [3] presented the first recognized mathematical formulation for a hub location problem by studying airline passenger networks.

The first formulation for the multiple allocation case was given by Campbell [1]. The rest of the literature on the hub location problem primarily focused on the linearization of the quadratic model proposed in O'Kelly [2], for example, Campbell [4], Ernst and Krishnamoorthy [5], O'Kelly et al. [6], and Skorin-Kapov et al. [7]. These studies introduce different mathematical formulations and solution procedures for the minimization of the total transportation cost.

Campbell [3] introduced different hub location problems (e.g., p -hub center and p -hub covering) to the literature and considered different objective functions. In particular, the hub covering problem minimizes the total cost of establishing hub networks, so that the cost (or travel time) between any origin-destination pair is within a given bound. Campbell [3] proposed linear and quadratic formulations for both single and multiple allocation variants of the given problem. The first attempt to provide computational results for the single allocation hub covering problem was taken from Kara and Tansel [8].

Ernst et al. [9] proposed a better mathematical formulation for the hub covering problem using the "radius" idea.

For the uncapacitated single allocation hub location problem, Labbe' and Yaman [10] derived a family of valid inequalities that generalizes the facet-defining inequalities and can be separated in a polynomial time. The capacitated multiple allocation case was studied by Aykin [11], Ebery et al. [12], Campbell [3], Boland et al. [13], and Marín [14]. In addition, the capacitated single allocation problem has been studied by Ernst and Krishnamoorthy [15], Labbé et al. [16], Contreras et al. [17], [18].

The interested reader may refer to comprehensive surveys on this matter given by Campbell et al. [19] and Alumur and Kara [20].

Similar to other hub location studies, we use a constant cost discount factor $\alpha \in [0,1]$ to represent the economies of scale in hub-to-hub connections. In addition, we use a constant cost discount factor $\gamma \in [0,1]$ to represent the economies of scale in hub-to-

node or node-to-hub connections (so much that $\gamma > \alpha$), in which we do not allow direct connections between the non-hub nodes. Unlike Ernst et al. [9], we consider the sum of the cost of all flows between each pair of nodes and the fixed cost of opening a new hub for the objective function. Also in this problem, we consider the capacity constraint for each hub that has a radius used for covering the non-hubs. Non-hub nodes can be allocated to one hub when they are in the coverage area of that hub. Unlike Ernst et al. [9], we consider the hub covering version of the given problem. We do not locate a fixed number of hub arcs and we force the hub arc network to be connected. In addition, we do not impose any structure on the hub network in advance.

1.2. Single Machine Scheduling

The aim of scheduling is to plan and arrange jobs in an orderly sequence of operations in order to meet customer's requirements [21]. Scheduling of jobs and the control of their flows through a production process are the most significant elements in any modern manufacturing systems. The single-machine environment is the basis for other types of scheduling problems. In single-machine scheduling, there is only one machine to process all jobs optimizing the system performance measures, such as makespan, completion time, tardiness, number of tardy jobs, idle times, sum of the maximum earliness and tardiness [22], [23]. In this paper, we consider a single machine scheduling problem with release dates and both earliness and tardiness penalties. Now, we illustrate the 2-stage model through Fig. 2 to Fig. 4. Fig. 2 includes central mine and nodes. At first, the hub network for the existing nodes will be designed as depicted in Fig. 3 (stage-1).

After designing the hub network, we schedule the material transportation from the central mine to the five hubs or factories as Fig. 4. We consider that just one truck exist in the network because of large expenses of using costly trucks. Each hub has its own distance from the central mine. As mentioned above, the raw material will be transferred to each hub with just one truck. Now, this fact can be considered as a single machine scheduling with five jobs. As shown in Fig. 4, each black line shows a straight path between central mine and the hubs. The truck should carries raw material through these paths. The optimal order of transferring materials from central mine to the hubs will be obtained according stage-2 mathematical formulation.

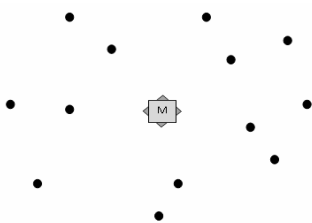


Fig. 2. Position of nodes and central mine

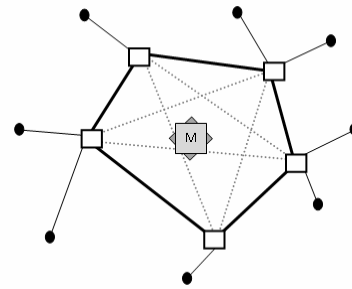


Fig. 3. Hub network

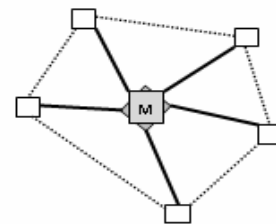


Fig. 4. Links between central mine and hubs

The paper is organized as follows. Section 2 describes 2-stage model for the hub covering location problem. Section 3 explains two proposed meta-heuristic algorithms used to calculate optimal solutions. Section 4 presents and analyzes computational results. Finally, some conclusions are drawn in Section 5.

2. Mathematical Formulation

2.1. Stage 1: Hub Covering Model

In this paper, we assume that there is a given node set N with n nodes that some of them can be located, such as hubs. The mathematical model locates hubs, constructs the hub network, and allocates the remaining nodes in set N to these hubs, such that any hub can allocate only some nodes that can cover them. Also just the limited number of nodes can be allocated to specific hubs, because all hubs are under capacity constraints. The objective of our mathematical model is to minimize the total cost of the flow between any origin-destination pair and the total cost of establishing hubs.

The parameters of the presented model are as follows. W_{ij} is the flow between nodes i and j , C_{ij} is the transportation cost of a unit of the flow between i and j , F_k is the fixed cost of opening a hub at node k and r_k is the maximum collection/distribution cost between hub k and nodes that are allocated to hub k . $Chub_k$ is the capacity of hub k . And $\alpha \in [0,1]$, is cost discount factor for between tow hubs, and also $\gamma \in [0,1]$ is the same; but, it is for between non-hub nodes and hubs. That is most likely higher than the α , and it is expected to be a number close to 1.

We define the decision variables of the model as follows:

$X_{ik} = 1$, if node i is allocated to hub k ; otherwise, it is 0.

Y_{kl}^i = The total amount of flow of commodity i (i.e., traffic emanating from node i) that is routed between hubs k and l .

b_k = The cost of raw materials transportation from the central mine to hub k .

Let $O_i = \sum_j W_{ij}$ be the total amount of the flow originating at node i .

More specifically, the allocation decisions are taken care by the classical X_{ik} variables. Consistent with the literature, if the variable $X_{kk} = 1$ for some k nodes, it means that node k is a hub node. The objective function of our mathematical model is to minimize the total cost of the flow between any origin-destination pair and the total cost of establishing hubs. With the previously defined parameters and decision variables, the objective function is expressed as follows.

$$\text{Min } Z_1 = \sum_{i=1}^n \sum_{k=1}^n \gamma C_{ik} X_{ik} O_i + \sum_{i=1}^n \sum_{k=1}^n \sum_{l=1}^n \alpha C_{kl} Y_{kl}^i + \sum_{k=1}^n F_k X_{kk} + \sum_{k=1}^n O_k b_k X_{kk} \quad (1)$$

$$\text{s.t. } \sum_{k=1}^n X_{ik} = 1 \quad \forall i \quad (2)$$

$$\sum_{l=1}^n Y_{kl}^i + \sum_{j=1}^n W_{ij} X_{jk} = O_i X_{ik} + \sum_{l=1}^n Y_{lk}^i \quad \forall i, k \quad (3)$$

$$\sum_{\substack{i=1 \\ i \neq k}}^n O_i X_{ik} \leq C_{hub} b_k \quad \forall k \quad (4)$$

$$r_k \geq C_{ik} X_{ik} \quad \forall i, k \quad (5)$$

$$X_{ik} \leq X_{kk} \quad \forall i, k \quad (6)$$

$$Y_{kl}^i \leq M X_{kk} \quad \forall i, k, l \quad (7)$$

$$Y_{kl}^i \leq M X_{ll} \quad \forall i, k, l \quad (8)$$

$$Y_{kl}^i \leq M X_{ik} \quad \forall i, k, l \quad (9)$$

$$X_{ik} \in \{0, 1\} \quad \forall i, k \quad (10)$$

$$Y_{kl}^i \geq 0 \quad \forall i, k, l \quad (11)$$

In the objective function, in the first term, we sum the total cost of the flow between non-hub nodes and hub nodes; in the second term we sum the total cost of flow between each tow hubs, and in the third term we calculate the total cost of establishing hubs. Constraints (2) and (10) ensure that each node is assigned to exactly one hub. Eq. (3) is the flow balance equation (divergence equation) for commodity i at node k where the demand and supply at the node is determined by the allocations X_{ik} .

Constraint (4) ensures that the allowed nominal capacity of the hub is not exceeded by preventing cargo from entering. Constraint (5) makes sure that node i can only be allocated to k , if cost C_{ik} between i and k is at most the radius r_k of k . Constraint (6) states that a node cannot be allocated to another node unless that node is a hub node. Constraints (7) to (9) ensure that Y_{kl}^i can be higher than zero only when k and l are hubs where M is a large number.

Finally, Constraint (11) ensures that variable Y_{kl}^i is bigger than zero, because it is the amount of the flow and need to be more than zero.

2.2. Stage 2: Single Machine Scheduling

The following notations and definitions are used to describe the single-machine scheduling problem [32].

| | |
|-------|--|
| N | Number of jobs |
| P_i | Processing time of job i , ($i=1, 2, \dots, N$) |
| R_i | Release time of job i , ($i=1, 2, \dots, N$) |
| D_i | Due date of job i , ($i=1, 2, \dots, N$) |
| C_i | Completion time of job i , ($i=1, 2, \dots, N$) |
| T_i | Tardiness of job i , ($i=1, 2, \dots, N$) |
| E_i | Earliness of job i , ($i=1, 2, \dots, N$) |
| | Tardiness penalty for job i , ($i=1, 2, \dots, N$) t_i |
| | Earliness penalty for job i , ($i=1, 2, \dots, N$) e_i |
| | A large positive integer value M |
| | If job j is scheduled after job i ; 0 otherwise X_{ij} |

The processing time of each job can be computed by:
 $P_i =$ Travel time from mine to factory i + return time from factory i to mine.

Processing time depends on type of transportation vehicle, type of transportation way and the road or the condition of transporting .

In this model, the objective is to minimize the total tardiness and total earliness. The second model is as follows.

$$\text{Min } Z_2 = \sum_{i=1}^N t_i T_i + \sum_{i=1}^N e_i E_i \quad (12)$$

$$\text{s.t. } T_i = \max \{C_i - D_i, 0\} \quad \forall i \quad (13)$$

$$E_i = \max\{D_i - C_i, 0\} \quad \forall i \quad (14)$$

$$C_i \geq R_i + P_i \quad \forall i \quad (15)$$

$$X_{ij} + X_{ji} = 1 \quad \forall i, j \quad (16)$$

$$C_i - C_j + MX_{ij} \geq P_i \quad \forall i, j \quad (17)$$

$$X_{ij} \in \{0, 1\} \quad \forall i, j \quad (18)$$

Constraints (13) and (14) specify the tardiness and earliness of job i , respectively. Also these constraints ensure that just one of and can exist for an especial job i in the model. Constraint (15) ensures that the completion time of a job i is greater than its release time plus processing time. Constraint (16) specifies the order relation between two jobs scheduled. Constraint (17) stipulates relative completion times of any two jobs. M should be large enough for Constraint (17).

3. Proposed Heuristic Algorithms

Hub location problems are NP-hard problems [20]. The proposed integer programming formulation involves $O(n^3)$ decision variables and $O(n^3)$ constraints. To solve realistically sized instances, meta-heuristic algorithms are used for this problem. According to literature, constructing feasible solutions for the hub covering problem, especially with tight radius and capacity bounds, is challenging.

Many studies in the literature review apply different heuristics to hub location problems, for example, tabu search heuristic method proposed by Klinecicz [24] and Skorin-Kapov [25], the simulated annealing heuristic method by Ernst and Krishnamoorthy [5], and the Lagrangean relaxation-based heuristic method by Pirkul and Schilling [26] for p-hub median problems. Additional contributions include a shortest path-based heuristic method by Ebery et al. [12], a genetic algorithm by Cunha and Silva [27], a hybrid heuristic by Chen [28], and a dual-ascent heuristic method by Cánovas et al. [29] for hub location problems with fixed costs.

Lastly, proposals for p-hub center problems include a tabu search-based heuristic method by Pamuk and Sepil [30] and a greedy heuristic by Ernst et al. [31]. The reader should note that, for hub location problems, the nearest allocation strategy (i.e., assigning a non-hub node to its nearest hub) does not necessarily give optimum solutions for the hub location problem.

3.1. Proposed Genetic Algorithm for Solving Stage-1

It is hard to optimally solve most of the NP-hard problems for realistically sized instances. For the problem at hand, even finding a feasible solution is

challenging. With this motivation, we decide to develop a meta-heuristic algorithm for our problem. In this section, a proposed genetic algorithm for the hub location problem will be presented and described.

3.1.1 Representation of the Solution

Any solution encoding procedure should show the location of hub nodes and the allocation of non-hub nodes to the hubs.

One of the procedures is using integer numbers for presenting given network. The solutions are presented as a vector which the length of the vector is equal to the number of nodes in the network. Each bit shows a node in the network, where its value explains the number of the hub or the node is allocated to. Further, when the value of the bit is equal to the number of the node, the node is considered as a hub. For example, a sample solution is obtained by:

$$\text{Ind} = [1 \ 7 \ 3 \ 7 \ 3 \ 3 \ 7 \ 1 \ 9 \ 9]$$

3.1.2. Initial Population

The creation of initial population is processed in three steps. In the first step, the number of hubs, p , is determined randomly. The number of hubs in this study is not given so it can be any number of nodes in the network with maximum n and minimum 2. In the second stage (i.e., location stage), p number of hubs are located randomly among the n nodes of network. Finally, in the allocation step, remained nodes will be allocated to the located hubs based on their shortest distance from located hubs and the capacity of the mentioned hubs.

The above process is applied iteratively to create the entire population.

3.1.3. Evaluation of Chromosome Fitness

The solutions in the genetic algorithm are named as chromosomes. The evaluation function is an operation to evaluate how good the network configuration of each chromosome is.

Also, it makes the comparison between different solutions possible. In this work, the evaluation function includes of calculating the value of the objective function of the network represented by each chromosome.

3.1.4. Crossover

In the crossover operator, information between two selected chromosomes is exchanged in order to create new offspring with better fitness value. For the proposed GA, a tournament selection size of 4 is used to select the individual that will undergo crossover. In this study, one-point crossover was chosen. Due to the individual's structure, every new offspring created by crossover should be passed through a filtering process to guarantee that the individuals have a valid structure at the end of the process.

3.1.5. Mutation

In proposed algorithm, mutation creates the new solution. It likes the population production as mentioned in Section 3.1.1. By this method some solutions will be produced which help others solution to exit from local searches. Besides, by mutation, the proposed algorithm finds optimal solution in wider solution space.

3.2. Proposed Shuffled frog Leaping Algorithm for Solving Stage 2

The shuffled frog leaping algorithm (SFLA) is designed as a population-based meta-heuristic method that benefits from both the genetic-based memetic algorithm (MA) and the social behavior based particle swarm optimization (PSO) algorithm. It is worth noting that the MA is a population-based algorithm for the stochastic search in optimization problems [33]. Dawkins [34] introduced the meme as a unit of intellectual or cultural information that can pass from generation to generation.

The main idea of the SFLA is participating of each member in previous experience of all other members. The SFLA has three main stages, namely partitioning, improvement process and shuffling. In this algorithm, after generating the initial population randomly from search space, members of a population are sorted as a decreasing order based on their value of the function evaluation.

Then population is partitioned into several parallel subsets that are called memplex.

The different memplexes are considered as different cultures of individuals. Each subset performs a local search independently using an evolutionary process to evolve their quality for a predefined maximum number of the iteration.

Then, all memplexes shuffle together and the stopping criteria are checked that if they are not met, the partitioning, local search and shuffling process are continued.

You can see the pseudo code of a hybrid shuffled frog leaping algorithm in Fig. 5.

To represent the solution of the problem for proposed algorithm, suppose that we have n job and one machine.

Therefore, n is equal to a number of hubs in each solution. At first, initial solution will be generated as a $1 \times n$ matrix. For example, a sample solution with $n=5$ nodes, is randomly presented as follow:

$$\text{Solution} = [0.1877 \ 0.8838 \ 0.6314 \ 0.6224 \ 0.6583]$$

Now, initial solution is sorted in ascending order. The place of their indexes will be the sequence of machines in each solution. So, the sequence of machines is as follow:

$$\text{Solution} = [1 \ 4 \ 3 \ 5 \ 2]$$

1. Set m Number of memplex
2. Set n Number of frogs in each memplex
3. Set Number of local iteration in each memplex
4. Set Number of algorithm iteration
5. **Begin;**
6. Generate random population of P solution (individuals).
7. For each individual $i \in P$: calculate the fitness (i);
8. Sort the whole population P in descending order of their fitness;
9. Determine X_b, X_w and X_g ;
10. Divide the population P into m memplex;
11. For each memplex;
 - 11.1. Shuffled leaping method
 - a. Determine the best (X_b) and worst individuals (X_w);
 - b. Change in frog position (D_i) = $\text{rand} \times (X_b - X_w)$;
 - c. New position of worst frog;
 $X_w = \text{current position } X_w + D_i$;
 - d. **If** $X_{w\text{new}} \leq X_{w\text{current}}$ **then** exchange this two individuals; **otherwise** use X_g instead of X_b . (X_g is the best position which has visited by all frogs) Now **if** the last condition does not meet, **then** generate the new individual;
 - e. Repeat for a specific number of iteration;
 - 11.2. Crossover method
 - a. Determine the best and worst individuals;
 - b. Improve the worst individual position using GA crossover;
 - b.1. Select a crossover point;
 - b.2. Produce offspring 1 and offspring 2;
 - b.3. For each offspring: calculate fitness;
 - b.4. If any of the offspring are better than those existing in P , replace them; if not, generate a new random individual;
 - c. Repeat for a specific number of iterations;
12. End.
13. Combine the evolved memplexes;
14. Sort the population P in descending order of their fitness;
15. Check if termination = true;
16. End.

Fig. 5. Pseudo code of the hybrid shuffled frog leaping algorithm

Above illustration means the first hub which should be served is hub 1, then hubs 4, 3, 5 and 2, respectively. By this method, the filtering steps for producing feasible solution are omitted and infeasible solutions are not produced during the algorithm. By this approach, the computational time is decreased efficiently.

4. Computational Results

In order to test the performance of the proposed algorithms, we compared the quality of their solutions with those of optimal solutions obtained by the developed mathematical programming for small size instances. Table 1 shows the parameter values and probability distributions used for randomly generated test instances. For running this problem we used a PC computer with bellow configuration:

Pentium 4, CPU 1.4GHz, RAM 1 MB.

In this study for the optimal solution in small sizes, we used the LINGO software. Also for the meta-heuristic we used the MATLAB software. The results of the comparison with the different value of parameters are shown in Table 2 and Table 3. For each instance we run the problem by meta-heuristics four times and inserted the best one of them in the Table 2.

In Table 2, the first three columns are the input parameters of the problem. The fourth column is the objective function of the optimal solution which obtained by the LINGO software.

The fifth column is the objective function of proposed genetic algorithm. The sixth column is the number of hubs which located in the network based on proposed algorithm. The seventh column is the gap of the proposed genetic algorithm in comparison with optimal solution. This column is calculated as follow:

$$\frac{Obj_{heuristic} - Obj_{optimalsolution}}{Obj_{optimalsolution}} \times 100$$

In Table 3, the first two columns are the input parameters of the problem. The second column is the number of hubs which is obtained from stage 1 in Table 2. The third column is the objective function of the optimal solution which obtained by the LINGO software. The fourth column is the objective function of proposed genetic algorithm. The fifth column is the gap of the proposed shuffled frog leaping algorithm in comparison with optimal solution. This column is calculated by:

$$\frac{Obj_{heuristic} - Obj_{optimalsolution}}{Obj_{optimalsolution}} \times 100$$

In Table 2, for a size of 30, 40, 50, 60 and 70 nodes, LINGO spends more than 8 hours. Therefore, for mentioned sizes, we just use the proposed algorithm for finding the near-optimal solutions.

Similarly, in Table 3, for a size of 21, 24, 25, 33, 39, 44, 49 and 54, LINGO spends more than 10 hours for obtaining near optimal solutions. So, the proposed shuffled frog leaping algorithm is applied for the mentioned sizes.

According to Table 2, the maximum gap of proposed genetic algorithm is 0.9%. In Table 3, the maximum

gap is 2%. Proposed algorithms spend around 600 seconds for solving the big size of problem. Therefore, those presented algorithm are helpful and efficient in solving big size instances of hub location network problems.

Tab. 1. Generation of random data

| Parameters | Value | | | | |
|---------------------|------------|------|------|------|----|
| <i>n</i> | 10 | 11 | 12 | 15 | 20 |
| | 30 | 40 | 50 | 60 | 70 |
| <i>α</i> | 0.70 | 0.80 | 0.85 | | |
| <i>β</i> | 0.80 | 0.85 | 0.90 | 0.95 | |
| C, F, W, r, P, R, D | U ~ (1,10) | | | | |

Tab. 2. Computational comparison of the stage-1 model and proposed genetic algorithm.

| Parameters | | | OFV with LINGO | OFV with heuristic | No. of hubs | Gap (%) |
|------------|----------|----------|----------------|--------------------|-------------|---------|
| <i>α</i> | <i>β</i> | <i>n</i> | | | | |
| 0.75 | 0.95 | 10 | 2730.80 | 2730.80 | 4 | 0.0 |
| 0.70 | 0.90 | 10 | 2246.00 | 2246.00 | 4 | 0.0 |
| 0.80 | 0.85 | 10 | 2320.00 | 2320.00 | 4 | 0.0 |
| 0.80 | 0.80 | 10 | 2296.00 | 2296.00 | 4 | 0.0 |
| 0.75 | 0.95 | 11 | 3491.25 | 3491.25 | 4 | 0.0 |
| 0.70 | 0.90 | 11 | 2691.40 | 2691.40 | 4 | 0.0 |
| 0.80 | 0.85 | 11 | 2781.10 | 2781.10 | 4 | 0.0 |
| 0.80 | 0.80 | 11 | 2749.40 | 2749.40 | 4 | 0.0 |
| 0.75 | 0.95 | 12 | 4216.75 | 4229.25 | 5 | 0.3 |
| 0.70 | 0.90 | 12 | 3477.30 | 3477.30 | 5 | 0.0 |
| 0.80 | 0.85 | 12 | 3642.95 | 3642.95 | 5 | 0.0 |
| 0.80 | 0.80 | 12 | 3612.40 | 3616.00 | 5 | 0.1 |
| 0.75 | 0.95 | 15 | 7457.25 | 7457.25 | 7 | 0.0 |
| 0.70 | 0.90 | 15 | 5720.40 | 5720.40 | 5 | 0.0 |
| 0.80 | 0.85 | 15 | 5949.60 | 5949.60 | 6 | 0.0 |
| 0.80 | 0.80 | 15 | 5894.80 | 5894.80 | 6 | 0.0 |
| 0.75 | 0.95 | 20 | 14883.00 | 15018.25 | 10 | 0.9 |
| 0.70 | 0.90 | 20 | 11338.00 | 11384.32 | 10 | 0.4 |
| 0.80 | 0.85 | 20 | 10452.64 | 10520.42 | 9 | 0.6 |
| 0.80 | 0.80 | 20 | 9982.00 | 10017.00 | 8 | 0.3 |
| 0.75 | 0.95 | 30 | - | 34434.00 | 25 | - |
| 0.70 | 0.90 | 30 | - | 31247.24 | 24 | - |
| 0.80 | 0.85 | 30 | - | 29514.08 | 21 | - |
| 0.75 | 0.95 | 40 | - | 86801.20 | 39 | - |
| 0.70 | 0.90 | 40 | - | 80712.54 | 33 | - |
| 0.75 | 0.95 | 50 | - | 183762.63 | 44 | - |
| 0.75 | 0.95 | 60 | - | 391485.78 | 49 | - |
| 0.75 | 0.95 | 70 | - | 810179.92 | 54 | - |

Tab. 3. Computational comparison of the stage-2 model and proposed shuffled frog leaping algorithm

| Parameters | | OFV of LINGO | OFV of Meta-heuristic | Gap (%) |
|-------------|-------------|--------------|-----------------------|---------|
| No. of hubs | No. of hubs | | | |
| 10 | 4 | 46 | 46 | 0.0 |
| 10 | 4 | 64 | 64 | 0.0 |
| 10 | 4 | 53 | 53 | 0.0 |
| 10 | 4 | 86 | 86 | 0.0 |
| 11 | 4 | 80 | 80 | 0.0 |
| 11 | 4 | 110 | 110 | 0.0 |
| 11 | 4 | 94 | 94 | 0.0 |
| 11 | 4 | 173 | 173 | 0.0 |
| 12 | 5 | 148 | 148 | 0.0 |
| 12 | 5 | 220 | 220 | 0.0 |
| 12 | 5 | 189 | 189 | 0.0 |
| 12 | 5 | 204 | 204 | 0.0 |
| 15 | 7 | 374 | 382 | 2.0 |
| 15 | 5 | 114 | 114 | 0.0 |
| 15 | 6 | 217 | 221 | 1.8 |
| 15 | 6 | 270 | 273 | 1.1 |
| 20 | 10 | 942 | 962 | 1.4 |
| 20 | 10 | 1283 | 1324 | 1.6 |
| 20 | 9 | 1524 | 1544 | 0.9 |
| 20 | 8 | 1398 | 1420 | 0.8 |
| 30 | 25 | - | 5663 | - |
| 30 | 24 | - | 5016 | - |
| 30 | 21 | - | 4782 | - |
| 40 | 39 | - | 14653 | - |
| 40 | 33 | - | 10718 | - |
| 50 | 44 | - | 18930 | - |
| 60 | 49 | - | 28372 | - |
| 70 | 54 | - | 40318 | - |

5. Conclusion

In this paper, we have studied the 2-Stage model for scheduling materials transportation from the central mine to factories in the hub covering location network. We have also presented an $O(n^3)$ integer programming formulation for the problem. To solve realistically sized instances, we have proposed a genetic algorithm (GA) and an improved frog shuffled leaping algorithm (FSLA) for stage-1 and stage-2, respectively. In stage-1, for size of 30 nodes, the LINGO software has taken more than 4 hours. Therefore, for sizes larger than 30, the proposed GA has been used. The mean gap for proposed genetic algorithm has been equal to 0.13% with 0.0% and 0.9% gap for the minimum and maximum gaps, respectively.

In stage-2, for size of 21 hubs, the proposed SFLA has spent more than 3 hours. Therefore, for sizes larger than 20 hubs, the proposed SFLA has been used. The mean gap for the proposed SFLA is 0.48% with 0.0% and 2.0% for the minimum and maximum gaps, respectively.

In contrast to other hub location problems, constructing feasible solutions for the hub covering problem is challenging, especially with capacity constraints.

We have used randomly generated data for the proposed meta-heuristics. Finally, the performance of

two foregoing algorithms has been compared with the optimal solution of the LINGO software.

References

[1] Campbell, J.F., *Integer Programming Formulations of Discrete Hub Location Problems*. European Journal of Operational Research 72, 1994, 387–405.

[2] O’Kelly, M.E., *The location of Interacting Hub Facilities*. Transportation Science 20 (2), 1986, 92–105.

[3] O’Kelly, M.E., *A Quadratic Integer Program for the Location of Interacting Hub Facilities*. European Journal of Operational Research 32, 1987, 393–404.

[4] Campbell, J.F., *Hub Location and the p-Hub Median Problem*. Operations Research 44 (6), 1996, 1–13.

[5] Ernst, A.T., Krishnamoorthy, M., *Efficient Algorithms for the Uncapacitated Single Allocation p-Hub Median Problem*. Location Science 4 (3), 1996, 139–154.

[6] O’Kelly, M.E., Skorin-Kapov, D., Skorin-Kapov, J., *Lower Bounds for the Hub Location Problem*. Management Science 41 (4), 1995, 713–721.

[7] Skorin-Kapov, D., Skorin-Kapov, J., O’Kelly, M., *Tight Linear Programming Relaxations of Uncapacitated p-Hub Median Problems*. European Journal of Operational Research 94, 1996, 582–593.

[8] Kara, B.Y., Tansel, B.C., *The Single-Assignment Hub Covering Problem: Models and Linearizations*. Journal of the Operational Research Society 54, 2003, 59–64.

[9] Ernst, A.T., Jiang, H., Krishnamoorthy, M., *Reformulations and computational Results for Uncapacitated Single and Multiple Allocation Hub Covering Problems*. Unpublished Report, CSIRO Mathematical and Information Sciences, Australia. 2005.

[10] Labbe’, M., Yaman, H., *Projecting Flow Variables for Hub Location Problems*. Networks 44 (2), 2004, 84–93.

[11] Aykin, T., *Lagrangian Relaxation Based Approaches to Capacitated Hub-and-Spoke Network Design Problem*. European Journal of Operational Research 79 (3), 1994, 501–523.

[12] Ebery, J., Krishnamoorthy, M., Ernst, A., Boland, N., *The Capacitated Multiple Allocation Hub Location Problems: Formulations and Algorithms*. European Journal of Operational Research 120, 2000, 614–631.

[13] Boland, N., Krishnamoorthy, M., Ernst, A.T., Ebery, J., *Pre-Processing and Cutting for Multiple Allocation Hub Location Problems*. European Journal of Operational Research 155 (3), 2004, 638–653.

[14] Marin, A., *Formulating and Solving Split Table Capacitated Multiple Allocation Hub Location Problems*. Computers & OR 32 (12), 2005a, 3093–3109.

- [15] Ernst, A.T., Krishnamoorthy, M., *Solution Algorithms for the Capacitated Single Allocation Hub Location Problem*. Annals of Operations Research 86, 1999, 141–159.
- [16] Labbe', M., Yaman, H., Gourdin, E., *A Branch and Cut Algorithm for the Hub Location Problems with Single Assignment*. Mathematical Programming 102, 2005, 371–405.
- [17] Contreras, I., Díaz, J.A., Fernández, E., *Branch and Price for Large-Scale Capacitated Hub Location problems with Single Assignment*. Optimization (Online), 2008.
- [18] Contreras, I., *Network Hub Location: Models, Algorithms and Related Problems*. PhD. Thesis, Department of Statistics and Operations Research, Technical University of Catalonia, Spain, 2009.
- [19] Campbell, J.F., Ernst, A.T., Krishnamoorthy, M., *Hub Location Problems*. In: Drezner, Z., Hamacher, H.W. (Eds.), Facility Location: Applications and Theory. Springer, 2002.
- [20] Alumur, S., Kara, B.Y., *Network Hub Location Problems: The State of the Art*. European Journal of Operational Research 190 (1), 2008, 1–21.
- [21] Sule, D.R., *Industrial Scheduling*, PWC Publishing Company, 1997.
- [22] Tavakkoli-Moghaddam, R., Moslehi, G., Vasei, M., Azaron, A., *Optimal Scheduling for a Single Machine to Minimize the Sum of Maximum Earliness and Tardiness Considering Idle Insert*, Applied Mathematics and Computation 167 (2) (2005) 1430–1450.
- [23] Tavakkoli-Moghaddam, R., Moslehi, G., Vasei, M., Azaron, A., *A Branch-and-Bound Algorithm for a Single Machine to Minimize the Sum of Maximum Earliness and Tardiness with Idle Insert*, Applied Mathematics and Computation 174 (1) (2006) 388–408.
- [24] Klincewicz, J.G., *Avoiding Local Optima in the p-Hub Location Problem using Tabu Search and GRASP*. Annals of Operations Research 40, 1992, 283–302.
- [25] Skorin-Kapov, D., Skorin-Kapov, J., *On Tabu Search for the Location of Interacting Hub Facilities*. European Journal of Operational Research 73, 1994, 502–509.
- [26] Pirkul, H., Schilling, D.A., *An Efficient Procedure for Designing Single Allocation Hub and Spoke Systems*. Management Science 44 (12), 1998, 235–242.
- [27] Cunha, C.B., Silva, M.R., *A Genetic Algorithm for the Problem of Configuring a Hub-and- Spoke Network for a LTL Trucking Company in Brazil*. European Journal of Operational Research 179, 2007, 747–758.
- [28] Chen, J.F., *A Hybrid Heuristic for the Uncapacitated Single Allocation Hub Location Problem*. Omega 35, 2007, 211–220.
- [29] Cánovas, L., Garcia, S., Marin, A., *Solving the Uncapacitated Multiple Allocation Hub Location Problem by Means of a Dual-Ascent Technique*. European Journal of Operational Research 179, 2007, 990–1007.
- [30] Pamuk, F.S., Sepil, C., *A Solution to the Hub Center Problem Via a Single-Relocation Algorithm with Tabu Search*. IIE Transactions 33 (5), 2001, 399–411.
- [31] Ernst, A., Hamacher, H., Jiang H., Krishnamoorthy, M., Woeginger, G., *Uncapacitated Single and Multiple Allocation p-Hub Center Problems*. Unpublished Report, CSIRO Mathematical and Information Sciences, Australia, 2002a.
- [32] Tavakkoli-Moghaddam, R., Javadi, B., Jolai, F., Ghodrathnama, A., *The use of a Fuzzy Multi-Objective Linear Programming for Solving a Multi-Objective Single-Machine Scheduling Problem*. Applied Soft Computing, 2009.
- [33] Rahimi-Vahed, A.R., Dangchi, M., Rafiei, H., Salimi, E., *A Novel Hybrid Multi-Objective Shuffled Frog-Leaping Algorithm for a Bi-Criteria Permutation Flow Shop Scheduling Problem*. Int J Adv Manuf Technol, 2009, 41:1227–1239.
- [34] Dawkins, R., *The Selfish Gene*. Oxford University Press, Oxford, 1976.

