IJICTR
International Journal of Information & Communication Technology Research

Volume 9- Number 4 – Autumn 2017 (29-36)

Hub location Allocation Problem in Computer Networks Using Intelligent Optimization Algorithms

Armond Hartoonian
Department of Computer Engineering
Islamic Azad University South Tehran Branch
Tehran, Iran
st_a_hartoonian@azad.ac.ir

Ahmad Khademzadeh*
Iran Telecommunication Research Center
ITRC
Tehran, Iran
zadeh@itrc.ac.ir

Received: April 8, 2017 - Accepted: April 23, 2017

Abstract— **One of the new issues that have been raised in recent years is the hub network design problem. The hubs are collection and distribution centers that are used for the purpose of less connections and more of indirect than direct communications. They are interface facilities which are used as switch centers to collect and distribute flows in the network. They determine routes and organize traffic between source-destination in order to provide high performance and be more inexpensive. In the hub location problem, the aim is to find a suitable location for the hub and routes for sending information from a source to a destination, in order to reduce costs and gain desired purpose by multiple transfers between the hubs. In this paper, teaching and learning based optimization, particle swarm optimization and imperialist competitive algorithm were studied for locating optimally hubs and allocating nodes to the nearest located hub nodes. Experimental results show that optimal location for hubs by using cluster-based optimization algorithm (TLBO) successfully has been performed with extreme accuracy and precision.**

Keywords-Hub Location-allocation; Network; Optimization algorithm; TLBO

I. INTRODUCTION

The Facility location-allocation problems is concerned with selecting the best location for deploying of the service provider facilities, and is trying to make a rational allocation of the demand centers to the facilities in order to reduce the lost demands [1]. The hub location problems are subset of facility location issue. Selecting a node as a hub can thus be very effective in reducing the costs [2]. All nodes are linked to their hubs and hubs are linked together. Hubs are responsible for collecting flows from the source nodes and to distribute them to the destination nodes. Such a hub network allows many source nodes and destination nodes to be connected with links fewer than those required with direct connections. In the hub network the flows between nodes are shown on the edges, which can be defined as cost, time, and distance [3]. The hub location problem is a part of network design issues. This problem arises when there is a need to transfer some information flow between the source and destination nodes where establishes direct connections between all nodes is too costly [4]. The aim of a hub location problem is to find the suitable location of the hubs and to allocate non-hub nodes to the located hub nodes in order to reduce costs and achieve the desired benefits. This topic is important because of a wide range of applications including air travel, postal services,

* Corresponding Author

transportation, telecommunication, and computer networks [5]. In this paper, intelligent optimization algorithms are used to solve the hub location-allocation problem in the computer networks and connect the non-hub nodes to the nearest hub nodes in order to minimize the connection costs and hub creation costs.

The rest of this paper is organized as follows. In second section, the literature review on Hub location-allocation problem is presented. In the third section, optimization algorithms are discussed. The Fourth section presents the proposed model and results evaluation of using optimization algorithms in order to solve the problem. Finally, conclusion is given in the last section.

II. Literature Review

Sender and Clausen presented a new heuristic solution approaches based on local improvements for solving hub location problem in network design of German wagonload traffic. They solve the problem by combining the heuristics and CPLEX, and then test it on datasets provided by Deutsche Bahn AG. The results show improvement in hub location-allocation compared to the previous methods in this field [6]. Campbell et al. proposed a new approach in order to solve hub arc location problem on the cluster of workstations. Here, many transportation systems rely on a network of hubs to help ease the traffic of transportation flows to exploit the economies of scale. Therefore, the design of a hub network, including location of the hub facilities, is regarded a key determinant of the cost and competitiveness of a transportation and logistics systems. They present a parallel implementation of this algorithm in an attempt to optimally solve larger hub arc location problems. The results of tests carried out on the cluster of workstations with network traffic data sets show an improvement and better performance of this type of calculation [7]. Aykin studied the hub location and routing problems. The problem was to find the hub locations and at the same time to determine the delivery mode for each demand. This method differs from other methods where demands for services are collected and flows from a source to a destination are considered separately. For each source-destination pair, one-hub, two-hub or giving services directly without a hub, are considered. The hubs interact with each other and the level of interaction between them is determined by the two-hub-stop service routes. A mathematical formulation and an algorithm for solving the hub location problem and the routing sub-problems were also presented separately in an iterative manner. The experimental results of applying this method with four different versions of the algorithm were used for finding the solutions of the problem [8]. Ernst and Krishnamoorthy introduced a new method to solve uncapacitated single allocation p-hub median problem. This proposed method is a different linear integer programming formulation which needs fewer variables than other linear programming formulations and it tries solving larger problems. They developed a heuristic algorithm based on simulated annealing (SA) for this purpose and used it to obtain upper bounds for linear programming based branch and bound. The results show that exact solutions can be found in a reasonable amount of computing time [9]. O'Kelly et al. presented a model with price-sensitive demands for solving hub location problem. This method can be used for computer networks with a large number of computers. The proposed method consists of two distinct working formulations, and an improved Benders decomposition algorithm is deployed in this respect. Here, simulation of customer choice between competing services is examined in the computer experiments. Further, there is a specialized sub-problem solution which uses good Benders cuts in reasonable computing time. The results on standard data sets show that the proposed method contributes well to a better understanding of hub traffic with varying service levels, as well as price equilibrium in competitive markets [10]. Labbé et al. studied several capacitated versions of hub location problems with single allocation in order to minimize the cost of hub installation and the cost of routing the traffic in the network. There may also be capacity restrictions on the amount of traffic that can transit by hubs. Hence, they proposed a branch-and-cut algorithm for this problem. The results of the proposed solution show a significant improvement over previous methods [11]. Randall presented a new method based on ant colony optimization (ACO) for solving capacitated single allocation Hub location problem and developed the four types of the ACO to explore various modeling of the problem. He also used the local search heuristics to improve the solutions provided by the ACO approach. The results show that this approach is good enough for small size networks only [12].

III. Intelligent Optimization Algorthims

The optimization concept is to find some values among the parameters of a function to optimize that function. All the proper values for this are possible solutions and the best of them is known as the optimal solution. Optimization algorithms cover both types of problems which need to be minimized or maximized. Optimization has always been associated with many problems. The former methods for solving optimization problems need the countless computational effort. Algorithms such as collective intelligence algorithms have solved part of the problem. By these algorithms, solutions can be found which are almost close to the answer [13].

A. *Teaching & Learning Based Optimization (TLBO)*

The Teaching and learning based optimization (TLBO) is a new meta-heuristic algorithm presented by Rao et al [14]. TLBO like other nature-inspired algorithms is population-based optimization algorithm but it is parameter less and that means it does not require any specific parameters and just needs common controlling parameters including population size and number of generations for its working. TLBO algorithm simulates a class teaching scenario where a teacher (the best learner) who outperforms others in terms of grades, shares his/her knowledge with the other learners, and the learners also learn from initiative interaction among themselves. The procedure of TLBO is divided into two sequential parts, (1)

teacher phase and (2) learner phase, these phases are explained below.

1) Teacher Phase

The first part of TLBO algorithm is teacher phase where students learn through the teacher. During this phase, a teacher is trying to enhance the mean result of the classroom from any value depending on his or her capability. Since it is practically impossible, a teacher can improve the mean of the classroom to other better value depending on the class capability. Consider $Mean_i$ is the mean and $Teacher_i$ is the teacher at each iteration i. $Teacher_i$ will try to improve existing mean $Mean_i$ towards to the new mean. The solution is updated according to the difference between the existing and the new mean which is given by:

$$Difference\_Mean_i = Teacher_i - TF_i \times Mean_i \quad (1)$$

TF, is the teaching factor which determines the value of mean to be changed. Teaching factor is not a parameter of the TLBO algorithm and its value can be either 1 or 2. The value of $TF$is randomly decided as,

$$TF = round[(1 + rand(0,1)\{2 - 1\}] \quad (2)$$

Based on the Difference Mean, the existing solution is updated as follows:

$$X_{new}^i = X_{old}^i + rand(.) \times Diference_{Mean}^i \quad (3)$$

Where $X_{new}^i$and X_{old}^i are the new and old positions of the learner. The output of teacher phase is considered as the input of the learner phase.

2) Learner Phase

The second part of TLBO algorithm is learner phase where learners improve their knowledge via interaction between themselves. A learner interacts randomly with other learners in order to increase his or her knowledge. Learner learns new things if the other learner has more knowledge than him or her as explained, learners increase their knowledge with the help of their mutual interaction.

In the case that X_j is better than X_i , X_i is moved toward X_j.Otherwise it is moved away from X_j:

$$X_{new} = \begin{cases} X_i + rand(.) \times (X_i - X_j) & if\ f(X_i) < f(X_j)\ \ i \neq j \\ X_i + rand(.) \times (X_j - X_i) & else \end{cases} \quad (4)$$

If the new solution X_{new} is better than X_i, it will be accepted in the population and replaced X_i otherwise the X_i remain unchanged. The output of this phase is considered as the input of the next iteration. The algorithm will continue until the termination condition is met.

B. Particle Swarm Optimization (PSO)

PSO is a nature-inspired and population-based optimization algorithm which was proposed by Eberhart and Kennedy in 1995. Since then, this algorithm is widely used to solve a broad range of optimization problems. PSO is based on the natural swarm behavior of birds and fish. In fact, this algorithm is a simulation of group social behavior of the birds which are searching an environment to find food. None of the birds has information about the position of food but in each stage they know how far they are from the food. Based on this, the best way to find food is following the bird nearest to the food. In PSO, population is called swarm and each member of the swarm is called a particle. Each particle's location in the problem space represents a possible solution of that problem which is defined by its cost function.

In PSO algorithm, first the particles are created and distributed randomly over the problem space and then search for the optimal answer begins. In the search process, each particle follows the particle with best fitness function, while not forgetting its own experience. Therefore, at each algorithm's iteration the position of every particle is updated according to two values. One of the values is the best personal position which is known as pbest and the other one is the best position in the whole population called gbest. In fact gbest is the general knowledge of the population and when particles change their position according to gbest it means they are trying to upgrade their own knowledge to the level of the population knowledge. PSO algorithm updates the particles' velocities and positions iteratively until a stopping criterion is met. Each particle's new velocity is updated by equation (5) based on the previous velocity of that particle and the distances of current position from personal best position and global best position.

$$V_i(t+1) = \omega V_i(t) + C_1 \times rand_1(pbest_i(t) - X_i(t)) + C_2 \times rand_2(gbest_i(t) - X_i(t)) \quad (5)$$

Where:

$V_i(t+1)$: New velocity of a particle.

$V_i(t)$: Current velocity of particle i.

$pbest_i(t)$: Best solution found by particle i.

$gbest_i(t)$: Best solution found in a population.

$X_i(t)$: Current position of particle i.

ω: Random inertia weight between 0.5 and 1.

C_1 and C_2: Priority factors.

$rand_1$ and $rand_2$: are random numbers between 0 and 1 [15].

The new position of a particle is then given by equation (6), where $X_i(t+1)$ is the new position:

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (6)$$

C. Imperialist Competitive Algorithm (ICA)

Imperialist Competitive Algorithm (ICA) is a meta-heuristic optimization technique which mimics the human's socio-political evolution and it was presented by Atashpaz-Gargari and Lucas in 2007. This algorithm simulates the competition among the imperialists and starts with a random initial population called countries which are basically similar to the chromosomes in genetic algorithm and particles in particle swarm optimization algorithm. The population (countries) is separated into two types: colony and imperialist. The Stronger countries are chosen as imperialist and rest of them are considered as the colonies of these imperialists. The colonies are distributed among the mentioned imperialists depending on their relative power [16]. The ICA

includes three main operators which are called assimilation, revolution, and competition.

Assimilation: In this step, an imperialist tries to improve its colonies, so all colonies are forced to move toward their relevant imperialists. In this movement, d is the Euclidean distance between an imperialist and its colony, β denotes a positive number greater than 1, and x are random numbers with uniform distribution as shown in equation (7).

$$x \epsilon U(0, \beta \times d) \quad (7)$$

Revolution: This process is similar to mutation process in genetic algorithm. The revolution increases the exploration of the algorithm and it helps optimization process to escape local optima. It randomly selects some countries and replaces them with new random position.

Exchange positions of the imperialist and a colony: While the colonies are moving towards the imperialist, there is a possibility that a colony reaches to a better position (lower cost) than of the respective imperialist. In this case, the imperialists and the colony exchange their positions and the algorithm will be continued using this new country as the imperialist and then the colonies start to move toward the imperialist in its new position.

Total power of an empire: The power of an empire is mainly affected by the power of imperialist country. However power of the colonies of an empire has an effect, albeit negligible, on the total power of that empire. The total cost of an empire is defined by equation (8):

$$T.C._n = Cost(imperialist_n + \xi mean\{Cost(Colonies\ of\ empire_n)\} \quad (8)$$

Where $T.C._n$ represents the total cost of the nth empire and zeta (ξ) is a positive number which is considered to be less than one. The small value for ξ causes the cost of the entire empire to mainly depend on the cost of imperialist. As the value of ξ increases, the importance of the role of the colony will be incremental.

Imperialist Competition: It is the most important process of this algorithm in which all empires make an attempt to take over the colonies of other empires and control them. Gradually, weaker empires lose their colonies to the stronger ones. This event is modeled by picking the weakest colony of the weakest empire and assigning it to the appropriate empire which is selected based on a competition among all empires. Finally, the power of stronger empires will be increased and consequently the power of weaker ones will be decreased or they will be eliminated.

Eliminating the powerless empires:

When the weak empire loses all of colonies, this empire will collapse and its imperialist is considered as a colony and is assigned to other empires.

Convergence: The competition process could be continued until there is just one imperialist in the search space. However, different conditions can be considered as termination criteria such as reaching the predefined maximum number of iterations or having little improvement in solution.

IV. The Proposed Model

For modeling decision variables in the problem, a binary matrix (n×n) is used. In this matrix, elements are in accordance with the following concepts, definitions, and constraints:

- x_{ii} indicates the value of the elements on main diagonal of the matrix. If $x_{ii} = 0$, it means i-th node in the network is a non-hub node, and does not service to any other node. Otherwise$x_{ii} = 1$, and it means i-th node is a hub node and it can serve other nodes.
- If $x_{ii} = 0$, i-th node is a non-hub node and does not service to any other nodes, so the value of all x_{ij} in i-th row is zero.
- If $x_{ii} = 1$, then i-th node is a hub node and can serve the other nodes, so in i-th row of the matrix a column like j can be found where $x_{ij} = 1$. It means node j receives service from node i.
- In each row of the matrix, the equation (9) is always true.

  $$X_{ij} \leq X_{ii} \quad (9)$$

  If x_{ii} is zero($x_{ii} = 0$), then x_{ij} is zero($x_{ij} = 0$). Because the node i is a non-hub node and cannot give services to node j.

  If ($x_{ii} = 1$) then there are two cases. If ($x_{ij} = 0$) it means node j is a non-hub node and it does not get any service from hub node i. If ($x_{ij} = 1$) it means node j gets services from node i.
- The total number of nodes is considered as n nodes, and the number of hub nodes is considered between 0.15n and 0.3n.

The greater distance from the hub node imposes more cost to connect the node to the hub. Thus the cost of creating a network increases. So connecting the nodes to a hub node, which is proportional to the distance, is to be determined with the aim of connecting a node to the nearest hub node in order to reduce the cost. According to equation (10), Euclidean distance is used to determine the distance of a node of the hub.

$$d(hub(i), i) = \sqrt{(x_{hub(i)} - x_i)^2 + (y_{hub(i)} - y_i)^2} \quad (10)$$

($x_{hub(i)}$,$y_{hub(i)}$) is the coordinates of hub i, (x_i,y_i) is the coordinates of node i, and d(hub(i),i) is the distance between node i and hub i.

The cost of node connection to the hub node is obtained from the following equation.

$$c(hub(i), i) = \beta d(hub(i), i) \quad (11)$$

Where β the cost per meter of wire and d is is the distance. Also the cost of creating hubs is determined in accordance with the requests, nodes, and distance of

nodes. The following equation presents the objective function.

$$f(x) = Min(\sum_{i=1}^{n}\sum_{j=1}^{n}(c_{i,hub(i)} + \alpha c_{hub(i),hub(j)} + c_{hub(j),j})x_{i,hub(i)}x_{hub(j),j} + \sum_{i=1}^{nhub} x_{ii}f_i) \quad (12)$$

In equation (12), if $x_{i,hub(i)}=1$, it means node i receives service from hub(i) and if $x_{hub(j),j}=1$, it means node j receives service from hub(j). $c_{i,hub(i)}$ is the cost of connecting node i to hub i. There are two purposes; the first one is to reduce the cost of creating network by selecting the nearest nodes to hub nodes. The second one is to reduce the cost of creating hub nodes. In equation(12), f_i is the cost of creating i-th hub if $x_{ii}=1$.

Therefore the main goal is creating a hub network and allocating the nodes to the nearest hubs in order to reduce the overall cost in computer networks.

Improve Optimization Algorithms

The swap and reversal operators, using which improves the optimization algorithms and increase their capability for finding more optimum solutions, can be utilized. By using these operators the characteristics that do not exist in the population are created. Because it changes the amount of one or more elements, this means if value of an element is one, it would be changed to zero vice versa if it is zero, it would be changed to one. Therefore it is suitable for the population which does not coverage prematurely, because one of the causes of premature convergence is the population members' similarity. Hence these operators reduced the probability of the members' similarity of new populations. The implementation method of these operators is described as follow.

Swap: In order to create diversity in responses of optimization algorithms, the swap operator was used on members of the population. In this method two elements are selected randomly, and then they are swapped. In this study, members of the population are considered as a matrix. Therefore, several methods can be used for swapping as below:

1) Swap rows: Two rows are selected randomly, and then they are swapped.
2) Swap columns: Two columns are selected randomly, and then they are swapped.
3) Swap elements: Two elements of the matrix elements are selected randomly, and then they are swapped.
4) Swap the elements of the main diagonal: Two elements of the main diagonal are selected randomly, and then they are swapped.

In this section, the results of simulation of hub location-allocation in the network using optimization algorithms will be discussed. Teaching and learning based optimization algorithm, particle swarm optimization, and imperialist competitive algorithm which are used to solve the hub location problem in the computer networks.

A. Results of using TLBO

In using TLBO for solving the problem, was considered 200 students, 400 iteration of algorithm, for 40 nodes. Table 1 represents the results of TLBO implementation, best cost, mean cost, runtime of algorithm, number of hub nodes, number of non-hub nodes, the number of truly allocated nodes to the hub nodes, the number of false allocated nodes, accuracy and convergence.

Table 1.Evaluation Results of using TLBO

Position	40
Best Cost	7014066
Mean Cost	7014066
Time	626
Hub	4
Node	36
True Positive	35
False Negative	1
Accuracy	97.22
Convergence	253

The figure 1 represents the convergence of cost function and average cost function at each iteration in TLBO.

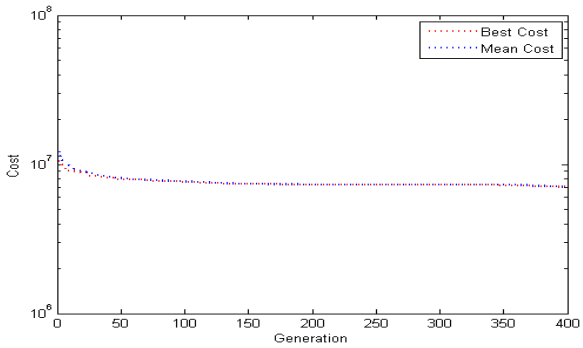

Figure 1.Convergence of objective function in TLBO

The Figure 2 represents the optimal arrangement of nodes and hubs in a network using TLBO for 40 nodes. As shown, the nodes are blue and hubs are purple.

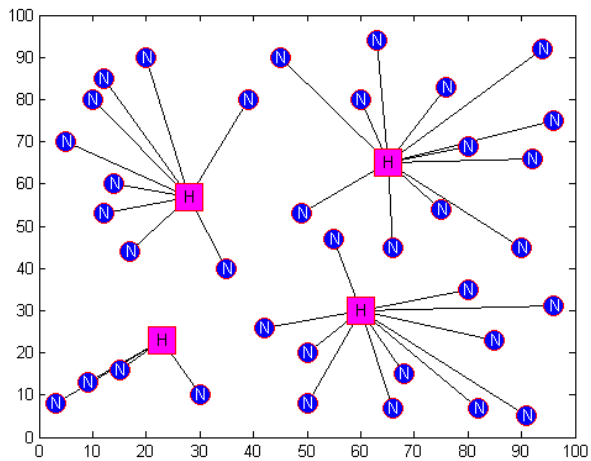

Figure 2.Optimum hubs location in the network using TLBO for 40 nodes.

B. *Results of using PSO*

In using PSO for solving the problem, was considered 200 populations, 400 iteration of algorithm, 0.4 inertia weigh, 1.4962 personal learning coefficient, and 1.3 global learning coefficient for 40 nodes. Table 2 represents the results of PSO implementation, best cost, mean cost, runtime of algorithm, number of hub nodes, number of non-hub nodes, the number of truly allocated nodes to the hub nodes, the number of false allocated nodes, accuracy and convergence.

Table 2.Evaluation Results of using PSO

Position	40
Best Cost	7048559
Mean Cost	7073384
Time	649
Hub	5
Node	35
True Positive	31
False Negative	4
Accuracy	88.57
Convergence	266

The figure 3 represents the convergence of cost function and average cost function at each iteration in PSO.

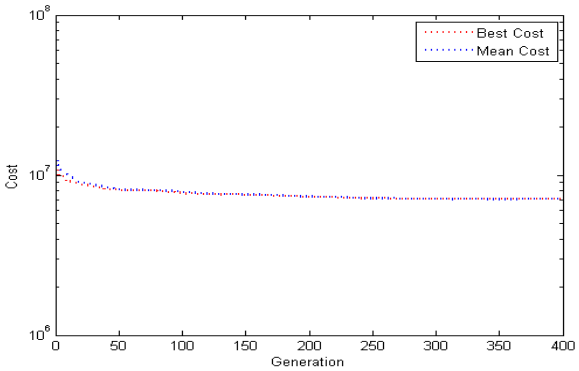

Figure 3.Convergence of objective function in PSO

The Figure 4 represents the optimal arrangement of nodes and hubs in a network using PSO for 40 nodes. As shown, the nodes are blue and hubs are purple.

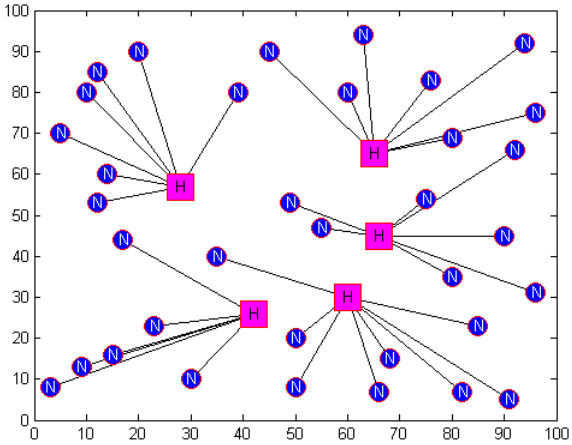

Figure 4.Optimum hubs location in the network using PSO for 40 nodes.

C. *Results of using ICA*

In using ICA for solving the problem, was considered 200 countries which consist of 150 colonies and 50 imperialists, 400 iteration of algorithm, assimilation coefficient (β) 2, revolution probability 0.6 for 40 nodes. Table 3 represents the results of ICA implementation, best cost, mean cost, runtime of algorithm, number of hub nodes, number of non-hub nodes, the number of truly allocated nodes to the hub nodes, the number of false allocated nodes, accuracy and convergence.

Table 3.Evaluation Results of using ICA

Position	40
Best Cost	7047195
Mean Cost	7047195
Time	780
Hub	5
Node	35
True Positive	33
False Negative	2
Accuracy	94.28
Convergence	274

The figure 5 represents the convergence of cost function and average cost function at each iteration in ICA.

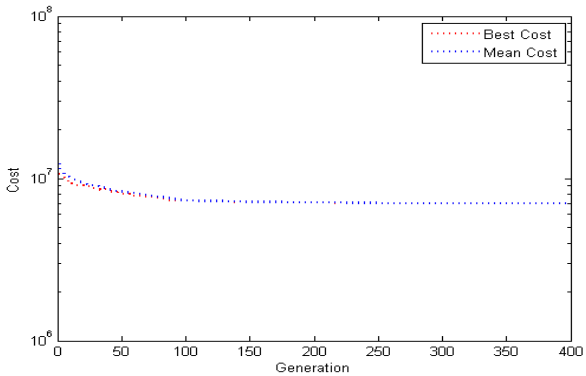

Figure 5.Convergence of objective function in ICA

The Figure 6 represents the optimal arrangement of nodes and hubs in a network using ICA for 40 nodes. As shown, the nodes are blue and hubs are purple.

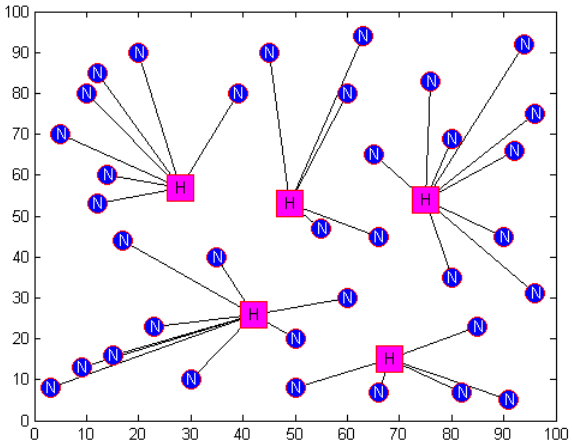

Figure 6.Optimum hubs location in the network using ICA for 40 nodes.

D. Overal Results

The results of using different optimization algorithms for hub location-allocation problem are shown in table 4.

Table 4. Evaluating the results of intelligent optimization algorithms for 40 nods

Algorithm	TLBO	ICA	PSO
Best Cost	7014066	7047195	7048559
Time	626	780	649
Hub	4	5	5
Node	36	35	35
True Positive	35	33	31
False Negative	1	2	4
Accuracy	97.22	94.28	88.57
Convergence	253	374	266

Figure 7 shows the cost function in different optimization algorithms for 40 nodes. The lowest cost function was achieved by teaching and learning based optimization algorithm.

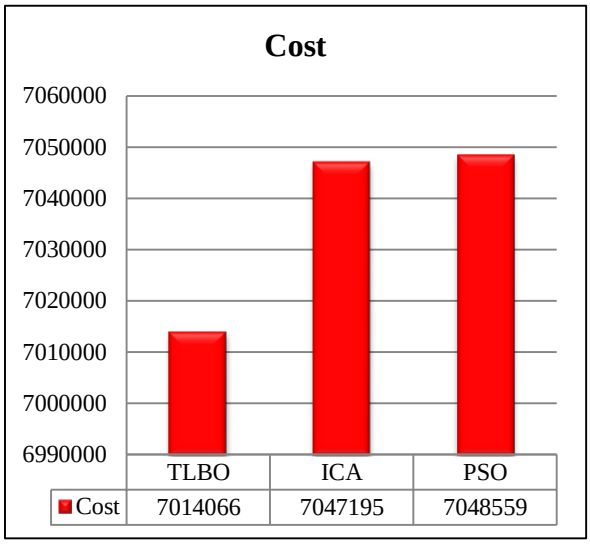

Figure 7.The Cost function in different optimization algorithms

Figure 8 shows the accuracy in different optimization algorithms for 40 nodes. The best accuracy in node allocation to hubs belongs to teaching and learning based optimization algorithm (%97.22).

Figure 9 shows the number of hub nodes and non-hub nodes (Overall 40 nodes) in different optimization algorithms. 36 nodes as non-hub nodes and 4 nods as hub nodes are determined by teaching and learning based optimization algorithm.

Figure 10 shows the runtime and convergence in different optimization algorithms. The best runtime (626 seconds) and the best convergence (iteration 253) belong to teaching and learning based optimization algorithm.

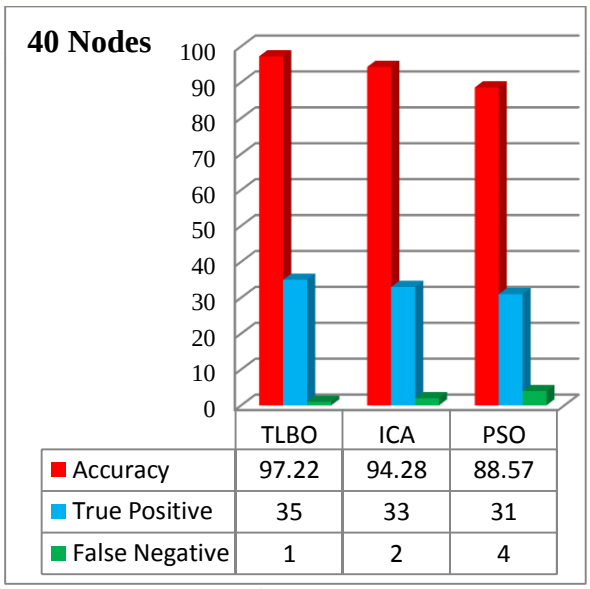

Figure 8.The accuracy of nodes allocation to the hubs in different optimization algorithms

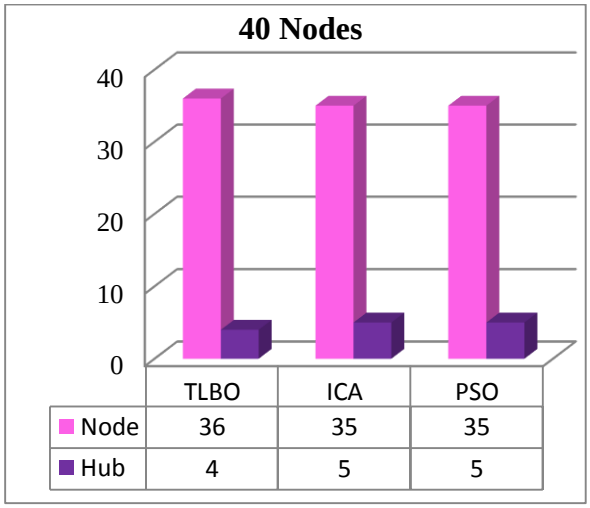

Figure 9.Correct nodes allocation to the hubs

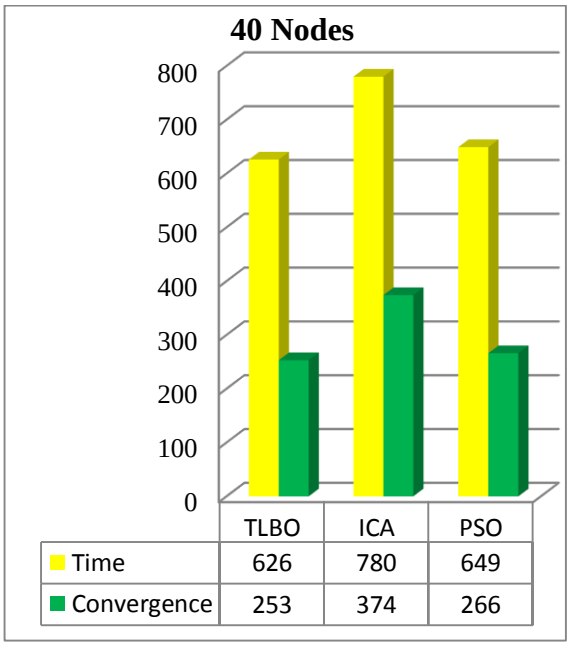

Figure 10.Runtime and convergence

V. Conclusion

This paper discussed on the optimal location of nodes in the network in order to reduce information transfer time through optimized connectivity between nodes and the hub nodes. Optimization algorithms such as TBLO, ICA and PSO were discussed as alternative approaches in this regard. We also tried to model the problem by optimization algorithms such as teaching and learning based optimization, which is improved by swap and reversal operators. According to the results obtained, Teaching and Learning based optimization algorithm (TLBO) has a better performance in comparison with particle swarm optimization and imperialist competitive algorithms. TLBO is therefore more suitable for solving the hub location-allocation problem in computer networks to reduce the transfer time of information through the optimal connection of nodes to the hub nodes.

References

[1] S. A. Alumur, H. Yaman, and B. Y. Kara, "Hierarchical multimodal hub location problem with time-definite deliveries," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, pp. 1107-1120, 2012.

[2] S. Alumur and B. Y. Kara, "Network hub location problems: The state of the art," *European journal of operational research*, vol. 190, pp. 1-21, 2008.

[3] A. Bailey, B. Ornbuki-Berrnan, and S. Asobiela, "Discrete pso for the uncapacitated single allocation hub location problem," in *Computational Intelligence In Production And Logistics Systems (CIPLS), 2013 IEEE Workshop on*, 2013, pp. 92-98.

[4] N. Boland, M. Krishnamoorthy, A. T. Ernst, and J. Ebery, "Preprocessing and cutting for multiple allocation hub location problems," *European Journal of Operational Research*, vol. 155, pp. 638-653, 2004.

[5] T. Aykin, "The hub location and routing problem," *Location Science*, vol. 4, pp. 280-281, 1996.

[6] J. Sender and U. Clausen, "Heuristics for solving a capacitated multiple allocation hub location problem with application in German wagonload traffic," *Electronic Notes in Discrete Mathematics*, vol. 41, pp. 13-20, 2013.

[7] J. F. Campbell, G. Stiehr, A. T. Ernst, and M. Krishnamoorthy, "Solving hub arc location problems on a cluster of workstations," *Parallel Computing*, vol. 29, pp. 555-574, 2003.

[8] T. Aykin, "The hub location and routing problem," *European Journal of Operational Research*, vol. 83, pp. 200-219, 1995.

[9] A. T. Ernst and M. Krishnamoorthy, "Efficient algorithms for the uncapacitated single allocation p-hub median problem," *Location science*, vol. 4, pp. 139-154, 1996.

[10] M. E. O'Kelly, H. P. L. Luna, R. S. De Camargo, and G. de Miranda Jr, "Hub location problems with price sensitive demands," *Networks and Spatial Economics*, vol. 15, pp. 917-945, 2015.

[11] M. Labbé, H. Yaman, and E. Gourdin, "A branch and cut algorithm for hub location problems with single assignment," *Mathematical programming*, vol. 102, pp. 371-405, 2005.

[12] Randall, Marcus. "Solution approaches for the capacitated single allocation hub location problem using ant colony optimisation." Computational Optimization and Applications 39.2 (2008): 239-261.

[13] D. Shilane, J. Martikainen, S. Dudoit, and S. J. Ovaska, "A general framework for statistical performance comparison of evolutionary computation algorithms," *Information Sciences*, vol. 178, pp. 2870-2879, 2008.

[14] R. V. Rao, V. J. Savsani, and D. Vakharia, "Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems," *Information Sciences*, vol. 183, pp. 1-15, 2012.

[15] M. Clerc, "Standard particle swarm optimisation," 2012.

[16] E.Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congresson*, 2007, pp. 4661-4667.

Armond Hartoonian was born in Tehran, Iran on April 22, 1980. He received his B. Sc. degree in Software Engineering from Islamic Azad University Shahre-e-Qods Branch in 2011, and his M. Sc. degree in Software Engineering from Islamic Azad University South Teran Branch in 2016. His research interests include evolutionary computation, computer networks, wirless networks.

Ahmad Khadem-Zadeh was born in Meshed, Iran, in 1943. He received the B.Sc. degree in applied physics from Ferdowsi University Meshed, Iran, in 1969 and the M.Sc., Ph.D. degrees respectively in Digital Communications and Information Theory & Error Control Coding from the University of Kent, Canterbury, UK. He is currently the Head of Education and International Scientific Cooperation Department at Iran Telecom Research Center (ITRC). He was the head of Test Engineering Group and the director of Computer and Communication Department at ITRC. He is also a lecturer at Universities in Tehran and a committee member of the Iranian Electrical Engineering Conference Permanent Committee. Professor Khadem-Zadeh has received four distinguished national and international awards including Kharazmi International Award, and has been selected as the national outstanding researcher of the Iran Ministry of Information and Communication Technology.