A Multi-objective Multi-agent Optimization Algorithm for the Community Detection Problem

Amir Hossein Hosseinian
Department of Industrial Engineering, University of Islamic Azad University, Tehran North branch, Tehran, Iran
ah_hosseinian@iau-tnb.ac.ir
Vahid Baradaran*
Department of Industrial Engineering, University of Islamic Azad University, Tehran North branch, Tehran, Iran
V_Baradaran@iau-tnb.ac.ir

Received: 24/Feb/2018 Revised: 22/Aug/2018 Accepted: 16/Sep/2018

Abstract

This paper addresses the community detection problem as one of the significant problems in the field of social network analysis. The goal of the community detection problem is to find sub-graphs of a network where they have high density of within-group connections, while they have a lower density of between-group connections. Due to high practical usage of community detection in scientific fields, many researchers developed different algorithms to meet various scientific requirements. However, single-objective optimization algorithms may fail to detect high quality communities of complex networks. In this paper, a novel multi-objective Multi-agent Optimization Algorithm, named the MAOA is proposed to detect communities of complex networks. The MAOA aims to optimize modularity and community score as objective functions, simultaneously. In the proposed algorithm, each feasible solution is considered as an agent and the MAOA organizes agents in multiple groups. The MAOA uses new search operators based on social, autonomous and self-learning behaviors of agents. Moreover, the MAOA uses the weighted sum method (WSM) in finding the global best agent and leader agent of each group. The Pareto solutions obtained by the MAOA is evaluated in terms of several performance measures. The results of the proposed method are compared with the outputs of three meta-heuristics. Experiments results based on five real-world networks show that the MAOA is more efficient in finding better communities than other methods.

Keywords: Community Detection Problem; Complex Networks; Multi-agent Systems; Social Networks.

1. Introduction

Networks are usually used to model complex systems in various fields, such as computer science, physics, biology and sociology [1]. Many complex systems can be structured as networks, such as computer networks, technological networks, collaboration networks, e-mail networks, biological networks, political election networks, etc. A network is defined as a graph where the nodes represent network objects and edges show the relations between them. Each node represents a network member, while an edge between two nodes indicates that there is a relation between two members of a network. For each complex system, there is a structural property called community structure. A community is defined as a group of nodes within a network that have a high density of within-group connections, while they have a lower density of between-group connections [2]. Discovering informative and hidden structures of networks is known as community detection problem. The real number of communities is not known in real-world networks. Therefore, an automatic clustering method is needed to identify the real number of communities in a network. Since the network objects in the same community may usually have similarities, the identified communities can be used in product recommendations, dimensionality reduction, information spreading, link prediction, knowledge sharing and other beneficial applications [3]. Many community detection methods have been developed in the literature which can be used in many practical cases such as product recommendations, reduction of dimensionality in pattern recognition, prediction of links and detection of cancers [4].

Detecting communities of a network can be modeled as an optimization problem. The aim of an optimization-based algorithm is to find an optimal solution with respect to a predefined objective function. Community detection problem is an NP-hard optimization problem [2]. The solutions obtained by single-objective approaches are limited to a particular community structure property. Therefore, if an improper objective function is chosen, these algorithms may fail to find high-quality communities. Besides, in hierarchical networks where there are multiple potential structures, a fixed community structure detected by single-objective approaches may not be appropriate. In this respect, it is desirable to optimize multiple objectives simultaneously so as to explore different potential network structures [4].

To solve the community detection problem which belongs to the set of NP-hard problems, we propose a multi-objective multi-agent optimization algorithm (MAOA) based on multi-agent systems (MAS). In the

* Corresponding Author

proposed algorithm, each agent is treated as a feasible solution for the problem. Agents work together in a grouped environment. The MAOA uses the Pareto dominance concept to find non-dominated agents and to approximate Pareto optimal front. This algorithm maximizes modularity and community score as objective functions. To find promising solutions for the community detection problem, new search operators based on social, autonomous and self-learning behaviors of agents have been designed for the proposed method. As another contribution, the weighted sum method (WSM), as a multi-attribute decision making (MADM) approach, has been utilized in various stages of the proposed algorithm such as finding the best global agent and the leader agent of each group.

To evaluate the performance of the MAOA, several numerical experiments are conducted on five real-world networks. The results demonstrate that the MAOA has been more successful in terms of several performance measures compared to three other meta-heuristics.

The rest of the paper is organized as follows: Section 2 reviews the literature of the community detection problem. Section 3 explains the community detection problem. Section 4 describes the proposed algorithm. Section 5 provides experimental results over five real-world networks. Ultimately, Section 6 concludes the paper and gives some suggestions for future studies.

2. Literature Review

There are many real-world applications for the community detection problem. Detecting fraud movements in telecommunication networks, prediction of connections in dynamic social networks, discovering terrorist groups in social networks and recommending products to customers in online shopping websites are some of the examples.

Many algorithms have been developed for the community detection problem. These algorithms have different strategies to find the most homogenous communities. One of the most important strategies is to treat a community detection problem as a combinatorial optimization problem. In this respect, the community structure is identified by optimizing a predefined criterion such as modularity, modularity density, community score, etc. Pizzuti [5] developed a genetic algorithm (GA) known as GA-Net to detect communities in social networks. Gong et al. [6] proposed a memetic algorithm called the Meme-Net to optimize modularity density as a quality function for estimating the quality of detected communities. Pizzuti [7] proposed a multi-objective community detection algorithm (MOGA-Net) for complex networks. The proposed algorithm uses the Non-dominated Sorting Genetic Algorithm II (NSGA-II) as the optimization procedure for maximizing community score and minimizing community fitness, simultaneously. Community score and community fitness indicate the intra-connections within communities and inter-connections between communities, respectively. Shi et al., [8] developed a multi-objective evolutionary algorithm known as the MOCD to detect community structure. The proposed method optimizes two terms of negatively correlated modularity, concurrently. Gong et al., [9] proposed an evolutionary algorithm with decomposition to optimize ratio cut and negative ratio association, simultaneously. An extended compact genetic algorithm was developed by Li and Song [10] for community detection problem. Amiri et al., [11] proposed a firefly algorithm to discover communities by using fuzzy-based grouping and mutation operators. Cai et al., [12] developed a discrete particle swarm optimization (PSO) algorithm for detecting communities in signed social networks. Gong et al., [13] developed a multi-objective particle swarm optimization (PSO) algorithm that utilizes search strategies of the PSO so as to discover communities of complex networks. The proposed algorithm minimizes two objective functions known as Kernel K-Means and the ratio cut. Cai et al., [14] developed a greedy discrete particle swarm optimization (PSO) method to tackle large-scale social networks. A multi-objective community detection method called the MOLS-Net has been proposed by Zhou et al., [15] that aims to optimize the Kernel K-means and Ratio Cut, concurrently. To optimize each objective function, a local search method has been embedded in the MOLS-Net. A meta-heuristic based on affinity propagation has been proposed by Shang et al., [16] to decompose networks. The network is decomposed by optimizing the Ratio Association and Ratio Cut.

Cheraghchi and Zakerolhosseini [17] proposed a novel dynamic community detection algorithm inspired by social theories. Bilal and Abdelouahab [18] developed an evolutionary algorithm to find community structures by maximizing modularity. Li et al., [19] developed two algorithms for the community detection problem. One of the proposed algorithms is a quantum-mechanism-based PSO algorithm, which is a parallel method. The other algorithm uses the non-dominated sorting procedure instead of the quantum mechanism. Based on the studies reviewed in this section, none of the previous researches have developed a multi-objective multi-agent algorithm for the multi-objective community detection problem to optimize modularity and community score, simultaneously.

3. Problem Description

3.1 Multi-objective Optimization Problem

The aim of a multi-objective optimization problem (MOP) is to find a vector of decision variables that meets constraints and optimizes a vector function. A vector function is a mathematical description of performance criteria formed by objective functions which are usually in conflict with each other. The MOP tries to optimize (minimize or maximize) conflicting objective functions $(f_1(X), \dots, f_m(X))$ when the decision variables $(X = x_1, \dots, x_n)$ can take their values within a feasible region. Typically, there is not a single solution that concurrently optimizes all objectives.

Considering a minimization problem, the MOP can be modeled as follows [4]:

$$\operatorname*{Min}_{\mathbf{x} \in \mathbf{S}} \mathbf{F}(\mathbf{x}) = \operatorname*{Min}_{\mathbf{x} \in \mathbf{S}} (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})) \tag{1}$$

Where, $F(X): S \rightarrow R^m$ consists of m real-valued continuous functions that should be minimized, concurrently. $f_i(X)$ is the i^{th} objective function and $x = (x_1, \ldots, x_k) \in S$ is the decision vector. S denotes the set of feasible solutions. In a minimization problem, a decision vector $x_a \in S$ can dominate another decision vector $x_b \in S$ if and only if [4]:

$$f_i(x_a) \leq f_i(x_b) \quad \wedge \quad f_j(x_a) < f_j(x_b) \quad \begin{array}{l} \forall i = 1, \ldots, n \\ \exists j = 1, \ldots, n \end{array} \tag{2}$$

The solutions of a MOP is a set of Pareto points. A solution $x^* \in S$ is a Pareto optimal if there is no solution (x) in the feasible solution space such that x dominates x^*. A set of solutions that dominate other solutions, while they cannot dominate themselves are called non-dominated solutions.

3.2 Community Definition

A network can be represented as an undirected graph denoted as G (**V**, **E**), where **V** and **E** are the **sets** of nodes and edges, respectively. A community is defined as a partition of nodes in the network that have more intra-links than inter-links. It is possible to represent a graph by the adjacency matrix. Let assume that **A** is the adjacency matrix of the graph G. Considering the adjacency matrix, if the element in row i and column j is equal to 1, there is an edge between nodes i and j in the graph. The degree of node i is computed as $k_i = \sum_j A_{ij}$. Suppose that node i belongs to a sub-graph S ($S \subset G$). In this respect, the degree of node i is defined as $k_i(s) = k_i^{in}(s) + k_i^{out}(s)$. $k_i^{in}(s)$ denotes the number of edges connecting node i to other nodes in sub-graph S, while $k_i^{out}(s)$ represents the number of edges connecting node i to the rest of the graph ($G \backslash S$). A sub-graph like S is considered as a strong community if $k_i^{in}(s) > k_i^{out} (\forall i \in S)$. To be more specific, a strong community is defined as a group of nodes which have higher intra connections comparing to the rest of the graph [4].

3.3 Fitness Functions

Modularity and community scores are two of the most important objective functions considered in the literature. Both objective functions need to be maximized. The proposed algorithm optimizes these objective functions to detect community structures of networks. These criteria are described as follows:

Modularity:

Modularity (Q) a quantitative criterion that measures the quality of network partitions. Modularity has been designed to quantify the strength of partitioning a network into modules. Modularity takes a value in the range of 0 and 1. The modularity value close to 1 implies that a community has the best possible strength, while the modularity value close to 0 indicates that the fraction of edges connecting nodes within a community is not better than the fraction of edges connecting a random gathering of nodes. Modularity is computed as follows [2]:

$$Q = \frac{1}{2L} \sum_{ij} \left(A_{ij} - \frac{k_i . k_j}{2L} \right) \delta(C_i, C_j) \tag{3}$$

Where, **A** is the adjacency matrix and L represents the number of edges in the network. A_{ij} is equal to 1 if nodes i and j are connected to each other. Otherwise, A_{ij} is equal to 0. k_i and k_j denote the degree of nodes i and j, respectively. $\delta(C_i, C_j)$ is equal to 1 if the nodes i and j belong to the same community.

Community score:

Suppose that node i belongs to a sub-graph S ($S \subset G$). μ_i represents the fraction of edges connecting node i to other nodes in community S. μ_i is calculated as follows [4]:

$$\mu_i = \frac{1}{|S|} k_i^{in}(S) \tag{4}$$

Where, $|S|$ is the cardinality of community S. $PM(S)$ is the power mean of S in order of p that can be defined as:

$$PM(S) = \frac{\sum_{i \in S} (\mu_i)^p}{|S|} \tag{5}$$

The volume of community S denoted as v_S is defined as the number of edges connecting nodes within the community S. v_S is computed by Eq. (6) and the score of community S ($SC(S)$) is obtained by Eq. (7):

$$v_S = \sum_{i,j \in S} A_{ij} \tag{6}$$

$$SC(S) = PM(S) \times v_S \tag{7}$$

The community score (CS) of a clustering $\{S_1, S_2, \ldots, S_k\}$ of a network is computed as follows:

$$CS = \sum_{i=1}^{k} SC(S_i) \tag{8}$$

Community score sums up the local scores of detected communities so as to provide a global measure of the network division.

4. Proposed Algorithm

4.1 Solution Representation

In this paper, the proposed algorithm employs the locus-based adjacency representation (LAR) [20]. According to the locus-based adjacency representation, each solution is considered as an array of N genes. Each gene represents a node in the graph and it is randomly connected to one of its neighbors. Therefore, each gene takes a value on the interval $[1, N]$. Each solution is decoded as a graph in which the value of j assigned to the gene i is interpreted as a link between node i and node j.

Thus, connected nodes of each solution are recognized as communities. In this representation, there is no need to know the number of communities in advance. This implies that the number of communities is determined during the decoding procedure. Moreover, the decoding process of a solution is performed in a linear time. The graph structure of a network with 16 nodes is illustrated in Figure 1. A feasible solution and its translation to a graph is depicted in Figure 2. As shown in Figure 2, each gene takes a value on the interval $[1, 16]$ that is randomly chosen from one of its neighbors. According to Figure 2, for instance, the third node has taken the value of “4”. This means that there is a link between node 3 and node 4 in the corresponding graph. Therefore, these two nodes are placed in a same community. Communities are depicted by dashed circles in Figure 2.

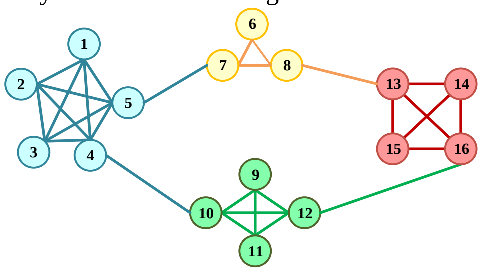

Fig. 1. A sample network

Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Solution	5	5	4	2	7	7	6	6	11	4	12	10	14	15	14	13

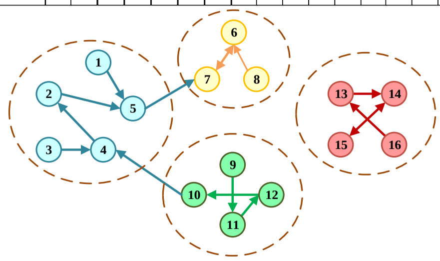

Fig. 2. A locus-based adjacency representation

4.2 Multi-agent System

An agent is defined as a computer system placed in a particular environment. Agents are able to receive information from the environment by means of sensors. An agent analyzes the information and takes consequent actions to affect the environment [21]. Agents have social behavior which makes it possible for them to interact with each other. A group of independent agents can form a multi-agent system (MAS). In a MAS, agents interact with each other and perform their tasks in an environment to achieve common goals. Each multi-agent system has three elements: (1) a set of independent agents $A = \{A_1, A_2, \dots, A_n\}$, (2) an environment where the agents carry out their duties and communicate with each other, and (3) a set of reactive rules that control the interactions between agents and environment [22]. In this paper, the agents are arranged as multiple groups. Figure 3 illustrates the group organization of agents.

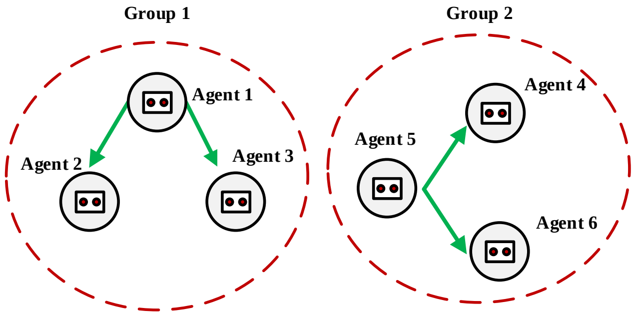

Fig. 3. Group-organized agents

Each multi-agent system has three main features: (1) behaviors of agents, (2) environment, and (3) interactions between agents. These features are explained as follows [23]:

Adjustment of Environment:

In this paper, each agent is considered as a solution. An environment is formed by multiple agents and their corresponding interactions. As mentioned earlier, we have used the grouped structure introduced by Zheng and Wang [23]. In this structure, there are G $(g = 1, \dots, G)$ groups in the environment. Each group contains N_g agents, where N_g represents the number of agents in the group g. The best agent in each group is chosen as the “leader”. The group that has the best leader agent is known as the elite group. The second best agent in each group is called the “active” agent. Figure 4 shows a leader-group organization.

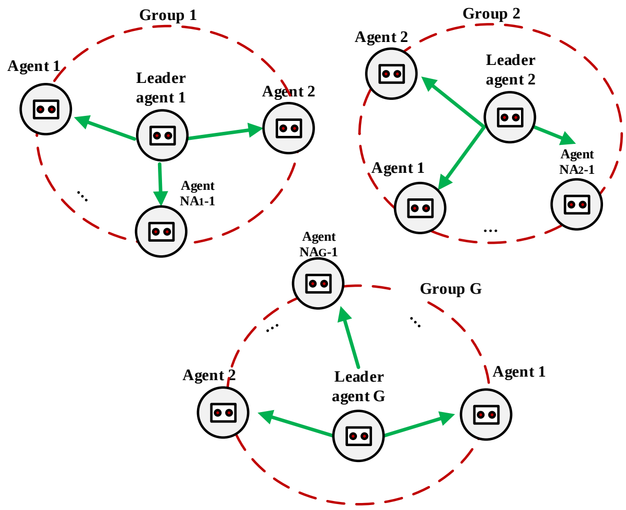

Fig. 4. Leader-group organization

Agents are re-grouped for adjustment of environment. By adjusting the environment, the agents are able to search the solution space, accurately. To adjust the environment, the active agent of each group is replaced with the worst agent of the elite group [22].

Behaviors of Agents:

There are three types of behaviors for each agent: (1) social behavior (local and global behaviors), (2) autonomous behavior, and (3) self-learning behavior. These behaviors are described as follows:

- Social behavior

Social behavior includes local and global behaviors. Local behavior shows the interactions between the leader agent of a group and other agents within the same group. Figure 5 illustrates the social behavior of agents.

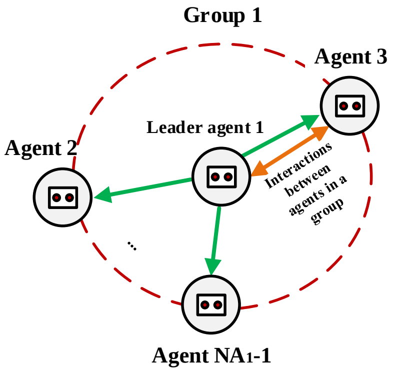

Fig. 5. Local social behavior

Global social behavior indicates the cooperation between the leader agent of the elite group and the leader agents of other groups. Figure 6 illustrates the global social behavior of agents.

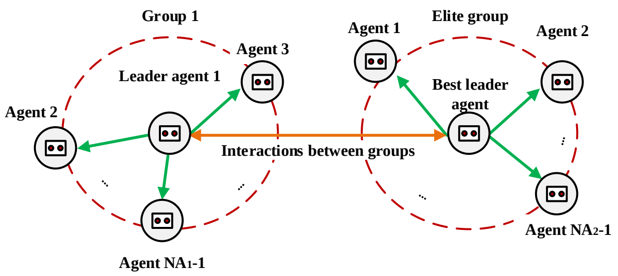

Fig. 6. Global social behavior

- Autonomous Behavior

According to the autonomous behavior, agents can act independently without external interference. Based on this behavior, each agent searches its neighborhood so as to find better solutions [23].

- Self-learning Behavior

It is possible for each agent to improve itself by learning from the obtained knowledge. Self-learning behavior of agents is a problem-dependent local search procedure used in the multi-agent optimization methods [24].

4.3 Multi-objective Multi-agent Optimization Algorithm

Due to the multi-objective essence of the problem tackled in this research, we propose a multi-objective multi-agent optimization algorithm called the MAOA. In the MAOA, the agents are initially divided into multiple groups to form the environment. Agents move among groups to share information in order to adjust the environment. The procedures of the MAOA are described in the following sub-sections.

Finding the Leader Agents of Groups:

To determine the leader agent of a group, the non-dominated sorting method proposed by Deb et al. [25] is used to find the non-dominated solutions (agents). The non-dominated sorting method makes it possible to approximate the Pareto optimal front. In case of having a single non-dominated agent, this agent is selected as the leader agent of the group. On the other side, if there are more than one non-dominated agent in a group, the weighted sum method (WSM) [26] as a multi-attribute decision making (MADM) technique is used to rank agents. The WSM provides the overall scores of non-dominated agents by computing the weighted sum average of all the criteria values. To utilize the WSM, a decision matrix (*DM*) is created, where its rows and columns represent non-dominated agents and criteria, respectively. These criteria are modularity and community score with the same importance. According to the WSM, the relative importance weights are multiplied with the normalized value of the criteria for each agent. Then, the obtained product value is summed up. The non-dominated agent with the highest score is selected as the leader agent of the group. Eq. (9) is used to calculate the overall score of agent i (OS_i) $(i = 1, \dots, N)$ with respect to M criteria [26]:

$$OS_i = \sum_{j=1}^{M} w_j . n_{ij} \qquad \forall \; i = 1, \dots, N \qquad (9)$$

Where, n_{ij} is the normalized rating of i^{th} non-dominated agent with respect to j^{th} criterion. w_j denotes the importance of j^{th} criterion. Normalized elements (n_{ij}) $(i = 1, \dots, N, j = 1, \dots, M)$ for benefit and cost criteria are calculated by the Eqs. (10) and (11), respectively [26].

$$n_{ij} = \frac{r_{ij}}{\max_i (r_{ij})} \qquad \forall \; i = 1, \dots, N \quad \forall \; j = 1, \dots, M \qquad (10)$$

$$n_{ij} = \frac{\min_i (r_{ij})}{r_{ij}} \qquad \forall \; i = 1, \dots, N \quad \forall \; j = 1, \dots, M \qquad (11)$$

Where, r_{ij} is the original rating of i^{th} non-dominated agent with respect to j^{th} criterion.

Finding the Global Best Agent:

Once again the non-dominated sorting method is used to detect the non-dominated agents among leader agents of all groups. In case of having a single non-dominated agent, this agent is selected as the global best leader agent. Otherwise, the WSM is employed to rank the leader agents in order to find the global best agent.

Social Behaviors in the MAOA:

In this paper, an operator is developed for the MAOA to generate new offspring agents from the global best agent and the leader agent. The proposed operator is considered as the global social behavior since it performs the interactions between the global best leader agent and the leader agents of each group. The best offspring agent is substituted with the leader agent if the best offspring agent dominates the leader agent. This operator initiates by generating a random binary $(1 \times N)$ string (*RBS*), where N is the number nodes in the network. If the $RBS_i = 1$ $(i = 1, \ldots, N)$, the value on the i^{th} gene of the global best agent is assigned to the i^{th} gene on the offspring agent. Otherwise, if the $RBS_i = 0$, the value on the i^{th} gene of the leader agent is assigned to the i^{th} gene of the offspring agent. This procedure always generates feasible solutions. Based on the network depicted in Figure 1, an example of generating a new offspring agent by the proposed operator is illustrated in Figure 7.

Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Best global agent	2	4	2	5	7	8	6	7	10	12	12	10	14	16	16	13
Leader agent	4	3	5	2	1	7	5	6	12	11	10	16	16	15	13	12
Random binary string (RBS)	1	0	0	0	1	1	0	1	0	1	1	0	1	0	1	1
Offspring agent	2	3	5	2	7	8	5	7	12	12	12	16	14	15	16	13

Fig. 7. Generating an offspring agent

In the following of this section, another procedure is proposed as the local social behavior to improve each agent with guidance of the leader agent within its own group. Therefore, for each agent, a random binary $(1 \times N)$ string (*RBS*) is generated. If the $RBS_i = 1$, the value on the i^{th} gene of the leader agent of the group will replace the value on the i^{th} gene of the agent. Otherwise, if the $RBS_i = 0$, the value on the i^{th} gene of the agent remains without any change. Figure 8 shows an example of changing an agent by the proposed procedure.

Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Leader agent	4	3	5	2	1	7	5	6	12	11	10	16	16	15	13	12
An agent	3	1	5	10	7	8	5	13	11	4	12	10	8	16	13	14
Random binary string (RBS)	0	0	1	1	1	0	0	0	1	0	1	0	0	1	0	1
Modified agent	3	1	5	2	1	8	5	13	12	4	10	10	8	15	13	12

Fig. 8. An operator based on local social behavior of agents

Autonomous Behavior in the MAOA:

A simple procedure is presented for the autonomous behavior of each agent. In this procedure, a random integer number (*RIN*) is generated in range of 1 to *N*. The *RIN* determines the number of genes that should be changed. In this procedure, the *RIN* number of genes are randomly selected and these genes take values based on the network.

Self-learning in the MAOA:

In this paper, a self-learning process has been designed for the MAOA to adjust a solution, iteratively. In each iteration, some genes of the best leader agent are changed to optimize modularity and community score.

Adjustment of Environment in the MAOA:

As mentioned earlier, the adjustment of environment is required to share information among groups of agents. Therefore, the environment is adjusted every 20 generations. Let assume that the global best agent belongs to group *l*. For each group g ($g \neq l$), the WSM is used to find the active agent (AG_g). If the AG_g dominates the worst agent of group l (WG_l), the AG_g transfers to group l and the WG_l moves to group g.

Elitism in the MAOA:

We consider an archive of non-dominated agents for the MAOA. In each iteration, the non-dominated agents produced by social behavior, autonomous behavior, self-learning and adjustment of environment are combined. Each newly generated agent is compared with the agents existing in the archive. If a new agent dominates any of the agents in the archive, the new agent is substituted with the dominated agent. The maximum number of iterations (*MaxIt*) has been considered as the stopping criterion for the proposed algorithm. Figure 9 shows the flowchart of the MAOA.

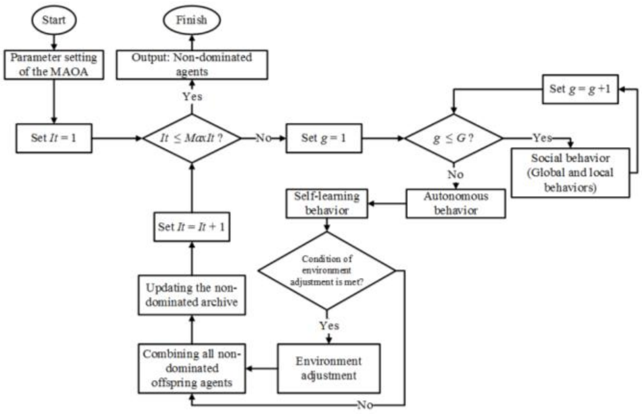

Fig. 9. Flowchart of the MAOA

5. Computational Experiments

In this section, the performance of the proposed algorithm is evaluated by comparing its results with the outputs of three multi-objective evolutionary algorithms, i.e. Multi-objective particle swarm optimization (MOPSO) algorithm [27], Non-dominated Sorting Genetic Algorithm II [25], and Strength Pareto Evolutionary Algorithm II [28]. All algorithms have been coded in the Matlab R2017b software. The codes have been run on a personal computer (PC) with Intel Core 2 Quad processor Q8200 (4M Cache, 2.33 GHz, 1333 MHz FSB) and 4GB memory.

5.1 Real-life Datasets

All algorithms are applied to five real-life networks, which can be downloaded from http://www-personal.umich.edu/~mejn/netdata. These networks are described in Table 1.

Table 1. Description of real-life networks

Network	*Description*
Karate Club	The Karate Club network consists of 34 nodes and 78 edges. The nodes of this network are divided into two groups [29].
Les Misérables	This network shows the relationships between characters of the Victor Hugo's novel "Les Misérables". The graph consists of 77 nodes and 257 edges. The network is divided into five groups [30].
Bernard	The Bernard technical dataset contains five sets of data on human interactions. There are 34

Network	***Description***
	nodes and 350 edges in this network [31].
Grevy's zebra	This network indicates the tendency of Grevy's zebras to appear together. The Grevy's zebra network consists of 28 nodes that are divided into three groups. There are 111 edges between nodes in this network [32].
Facebook	This network shows the interactions between users of the famous social network known as Facebook. This network includes 4039 nodes and 88234 edges [2].

5.2 Performance Measures

In this paper, five performance measures are used to evaluate the performances of algorithms. These performance measures are described as follows:

Normalized Mutual Information (*NMI*):

The *NMI* is a metric that measures the similarity between the real community structure of a network and the community structure found by an algorithm. Let assume that real partitioning of a network is $\alpha = \{\alpha_1, \dots, \alpha_R\}$ and the partitioning identified by an algorithm is $\beta = \{\beta_1, \dots, \beta_K\}$. R and K denote the number of communities in the partitioning α and β, respectively. Confusion matrix (C) is created at the first step to compute the *NMI*. Each element C_{ij} is the number of common nodes in communities $\alpha_i \in \alpha$ and $\beta_j \in \beta$. Then, $NMI(\alpha, \beta)$ is calculated using Eq. (12) [4]:

$$NMI(\alpha,\beta) = \frac{-2\sum_{i=1}^{R}\sum_{j=1}^{K} C_{ij}\log\left(\frac{C_{ij}N}{C_{i.}C_{.j}}\right)}{\sum_{i=1}^{R} C_{i.}\log\left(\frac{C_{i.}}{N}\right)+\sum_{j=1}^{K} C_{.j}\log\left(\frac{C_{.j}}{N}\right)} \tag{12}$$

Where, $C_{i.}$ and $C_{.j}$ are the sums of the elements in the confusion matrix over row i and column j, respectively. N is the number of nodes existing in the network. The *NMI* takes a value in the range of $[0,1]$. $NMI = 1$ implies that α and β are exactly equal. On the other hand, $NMI = 0$ indicates that α and β are completely different.

Mean ideal Distance (*MID*):

This metric measures the closeness between the solutions of the approximation front and the ideal point. The mean ideal distance metric is computed by Eq. (13) [33]:

$$MID = \frac{\sum_{i=1}^{NDS}\sqrt{\left(O_{i1}-O_1^*\right)^2+\left(O_{i2}-O_2^*\right)^2}}{NDS} \tag{13}$$

Where, *NDS* represents the number of non-dominated solutions found by an algorithm. O_{i1} and O_{i2} denote the first and the second objective function values of i^{th} solution on the approximation front, respectively. O_1^* and O_2^* are the ideal solutions regarding the first and the second objective functions, respectively. Lower values of *MID* metric show better performance of an algorithm.

Diversification Metric (*DM*):

Diversification metric estimates the extension of the approximation front. Higher values of this metric mean higher diversity of solutions obtained by an algorithm. Diversification metric is computed as follows [33]:

$$DM = \sqrt{\left(\max_{i=1:NDS} O_{i1}-\min_{i=1:NDS} O_{i1}\right)^2+\left(\max_{i=1:NDS} O_{i2}-\min_{i=1:NDS} O_{i2}\right)^2} \tag{14}$$

Set Coverage (*C*-metric):

Suppose that there are two Pareto fronts denoted as F_1 and F_2 obtained by two different algorithms. $C(F_1, F_2)$ shows the percentage of solutions on the F_2 dominated by at least a single solution of F_1. $C(F_1, F_2)$ is calculated using the Eq. (15) [33]:

$$C(F_1,F_2) = \frac{\left|\{j' \in F_2 \mid \exists j \in F_1 : j \quad Dom \quad j'\}\right|}{\left|F_2\right|} \tag{15}$$

Where, j and j' are the solutions on the F_1 and F_2, respectively. $|F_2|$ represents the number of solutions on the F_2.

Computation Time (CPU Time):

Another criterion that differentiates algorithms is their required computation time (CPU time) to find optimal or near optimal solutions [34].

5.3 Parameter Setting

In this study, the Taguchi method has been used to set the parameters of algorithms. To obtain optimal values for parameters, the Taguchi method provides a statistic known as signal to noise (*S*/*N*) ratio [35]. Three levels have been considered for parameters of algorithms. Table 2 shows the levels defined for each parameter.

Table 2. Parameters of algorithms

Algorithm	***Parameter***	***Symbol***	***Levels***		
			Level 1	***Level 2***	***Level 3***
MAOA	Number of groups	*G*	3	4	5
	Number of agents in each group	*NA*	30	40	50
	Maximum number of iterations	*MaxIt*	50	100	200
MOPSO	Population size	*Npop*	50	100	200
	Maximum number of iterations	*MaxIt*	50	100	200
	Social factor	C_1	1	1.25	1.5
	Cognitive factor	C_2	1	1.25	1.5
	Inertia weight	*INW*	0.70	0.75	0.80
NSGA-II	Population size	*Npop*	50	100	200
	Maximum number of iterations	*MaxIt*	50	100	200
	Crossover rate	p_c	0.75	0.80	0.85
	Mutation rate	p_m	0.05	0.10	0.15
SPEA-II	Population size	*Npop*	50	100	200
	Maximum number of iterations	*MaxIt*	50	100	200
	Crossover rate	p_c	0.75	0.80	0.85
	Mutation rate	p_m	0.05	0.10	0.15
	Archive size	*ARS*	5	10	15

A response variable known as the multi-objective coefficient of variation (*MOCV*) was proposed by Rahmati et al. [36] for the Pareto-based algorithms to provide diverse solutions with proper convergence. The *MOCV* incorporates the *MID* and *DM* metrics, simultaneously. The MID metric estimates the convergence rates of algorithms, while the *DM* metric evaluates the diversity of solutions. The *MOCV* can be computed as follows [36]:

$$MOCV = \frac{MID}{DM} \tag{16}$$

Three real-world networks have been used to conduct the Taguchi method. Each algorithm has been run for 10 times to obtain reliable results. To calculate the *MOCV*, the values of the *MID* and *DM* metrics are converted to the relative percentage difference (*RPD*). The *RPD* is obtained as follows [37]:

$$RPD = \frac{\left|\lambda - \lambda^*\right|}{\lambda^*} \times 100 \tag{17}$$

Where, λ denotes the value of performance measure acquired by an algorithm, while λ^* is the best value of performance measure among all values. The average of *RPD*s ($\overline{RPD}$) are calculated for all experiments. Afterwards, $MOCV = \overline{RPD}(MID)/\overline{RPD}(DM)$ is obtained for all experiments. Figures 10 to 13 show the *S/N* ratio plots for the parameters of the MAOA, MOPSO, NSGA-II and SPEA-II, respectively. Table 3 reports optimal values of parameters.

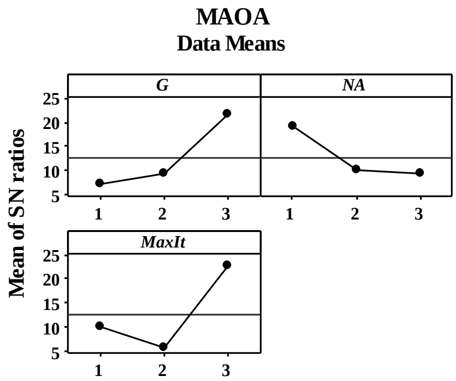

Fig. 10. The mean S/N ratio plot for the MAOA

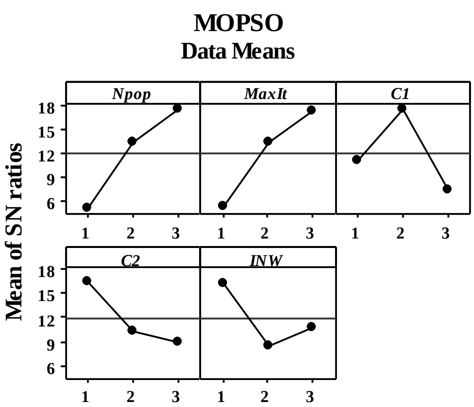

Fig. 11. The mean S/N ratio plot for the MOPSO

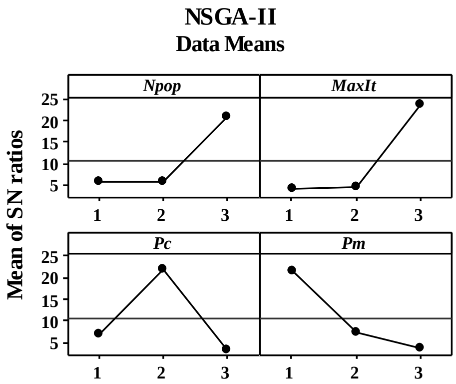

Fig. 12. The mean S/N ratio plot for the NSGA-II

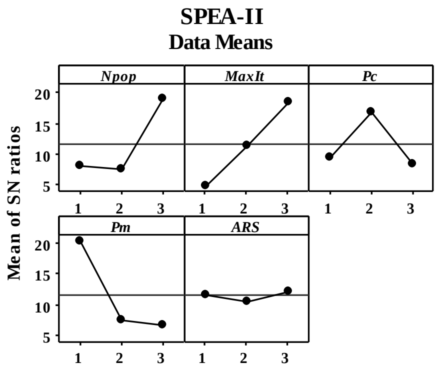

Fig. 13. The mean S/N ratio plot for the SPEA-II

Table 3. Optimal values of parameters

Algorithm	*Parameter*	*Optimal value*
MAOA	*G*	5
	NA	30
	MaxIt	200
MOPSO	*Npop*	200
	MaxIt	200
	C_1	1.25
	C_2	1
	INW	0.70
NSGA-II	*Npop*	200
	MaxIt	200
	p_c	0.80
	p_m	0.05
SPEA-II	*Npop*	200
	MaxIt	200
	p_c	0.80
	p_m	0.05
	ARS	15

5.4 Results and Discussion

The performance of the MAOA is compared with the MOPSO, NSGA-II and SPEA-II in terms of metrics described in Section 5.2. Each method has been run for twenty time and Table 4 reports the average values of performance measures. According to the results summarized in Table 4, the following outlines have been achieved:

1. The MAOA has the best *NMI* values in all networks. This means that the communities found by the MAOA have the most similarity to the real communities of networks.
2. The MAOA has been more successful in providing lower

values of the *MID* metric. This implies that the MAOA has better convergence in comparison with other methods.

3. The MAOA provides more diverse solutions than other algorithms.
4. The MOPSO is the fastest algorithm in detecting communities of all complex networks. Hence, one of the main disadvantages of the MAOA is that it requires more computation time comparing to other methods used in this study.
5. As the number of nodes and edges in networks increase, the values of most of the metrics increase.
6. In terms of the *NMI* metric, the improvement which has been made by the MAOA is nearly 59.4%.
7. For the *MID* metric, the outputs of the MAOA are 62.8% better than other algorithms.
8. In terms of the *DM*, the superiority of the results obtained by the MAOA is approximately 28%.

Table 4. Comparison of algorithms

Network	***Algorithms***	***Performance measures***			
		NMI	***MID***	***DM***	***CPU time***
Karate Club	MAOA	1.00	6.34	6759.61	5.06
	MOPSO	0.95	13.92	5815.48	3.82
	NSGA-II	0.91	24.26	5390.93	5.97
	SPEA-II	0.87	32.78	5015.48	4.94
Les Misérables	MAOA	1.00	7.09	5968.64	13.17
	MOPSO	0.87	21.08	4875.37	11.75
	NSGA-II	0.84	42.45	4322.47	15.25
	SPEA-II	0.80	46.69	4035.76	12.90
Bernard	MAOA	1.00	1.78	6790.40	5.77
	MOPSO	0.74	9.34	5671.26	3.04
	NSGA-II	0.63	13.80	5138.87	6.13
	SPEA-II	0.38	17.01	5019.57	4.48
Grevy's zebra	MAOA	1.00	3.79	8566.41	4.02
	MOPSO	0.71	5.53	7989.95	2.28
	NSGA-II	0.65	6.16	7398.58	4.91
	SPEA-II	0.49	9.19	7016.98	3.19
Facebook	MAOA	0.73	26.42	4763.95	18.57
	MOPSO	0.52	37.40	4468.02	15.09
	NSGA-II	0.36	41.29	3359.63	19.81
	SPEA-II	0.24	49.80	3101.66	17.93

Tables 5 to 9 report the average values of *C*-metric obtained by comparing the non-dominated solutions of algorithms. It is obvious from these tables that the MAOA obtained higher *C* (MAOA, MOPSO), *C* (MAOA, NSGA-II) and *C* (MAOA, SPEA-II) values in all networks. This means that the solutions of the MAOA prevailed the solutions obtained by other algorithms.

Table 5. C-metric values for Karate Club network

C(MAOA, MOPSO)	*C*(MAOA, NSGA-II)	*C*(MAOA, SPEA-II)
0.83	0.97	1.00
C(MOPSO, MAOA)	*C*(MOPSO, NSGA-II)	*C*(MOPSO, SPEA-II)
0.24	0.61	0.77
C(NSGA-II, MAOA)	*C*(NSGA-II, MOPSO)	*C*(NSGA-II, SPEA-II)
0.13	0.25	0.28
C(SPEA-II, MAOA)	*C*(SPEA-II, MOPSO)	*C*(SPEA-II, NSGA-II)
0.07	0.11	0.18

Table 6. C-metric values for Les Misérables network

C(MAOA, MOPSO)	*C*(MAOA, NSGA-II)	*C*(MAOA, SPEA-II)
0.82	1.00	1.00
C(MOPSO, MAOA)	*C*(MOPSO, NSGA-II)	*C*(MOPSO, SPEA-II)
0.26	0.73	0.81
C(NSGA-II, MAOA)	*C*(NSGA-II, MOPSO)	*C*(NSGA-II, SPEA-II)
0.11	0.49	0.76
C(SPEA-II, MAOA)	*C*(SPEA-II, MOPSO)	*C*(SPEA-II, NSGA-II)
0.08	0.14	0.18

Table 7. C-metric values for Bernard network

C(MAOA, MOPSO)	*C*(MAOA, NSGA-II)	*C*(MAOA, SPEA-II)
0.90	0.96	1.00
C(MOPSO, MAOA)	*C*(MOPSO, NSGA-II)	*C*(MOPSO, SPEA-II)
0.35	0.54	0.84
C(NSGA-II, MAOA)	*C*(NSGA-II, MOPSO)	*C*(NSGA-II, SPEA-II)
0.19	0.40	0.88
C(SPEA-II, MAOA)	*C*(SPEA-II, MOPSO)	*C*(SPEA-II, NSGA-II)
0.02	0.17	0.22

Table 8. C-metric values for Grevy's zebra network

C(MAOA, MOPSO)	*C*(MAOA, NSGA-II)	*C*(MAOA, SPEA-II)
0.85	0.92	0.98
C(MOPSO, MAOA)	*C*(MOPSO, NSGA-II)	*C*(MOPSO, SPEA-II)
0.30	0.58	0.69
C(NSGA-II, MAOA)	*C*(NSGA-II, MOPSO)	*C*(NSGA-II, SPEA-II)
0.26	0.52	0.73
C(SPEA-II, MAOA)	*C*(SPEA-II, MOPSO)	*C*(SPEA-II, NSGA-II)
0.03	0.09	0.12

Table 9. C-metric values for Facebook network

C(MAOA, MOPSO)	*C*(MAOA, NSGA-II)	*C*(MAOA, SPEA-II)
0.78	0.81	0.94
C(MOPSO, MAOA)	*C*(MOPSO, NSGA-II)	*C*(MOPSO, SPEA-II)
0.15	0.33	0.37
C(NSGA-II, MAOA)	*C*(NSGA-II, MOPSO)	*C*(NSGA-II, SPEA-II)
0.17	0.25	0.64
C(SPEA-II, MAOA)	*C*(SPEA-II, MOPSO)	*C*(SPEA-II, NSGA-II)
0.00	0.05	0.10

Figures 14 and 15 show the comparisons between algorithms in terms of modularity and community score, respectively. These figures show the average values of objective functions obtained by ten runs. As shown in Figures 14 and 15, the MAOA has achieved better results comparing to other methods.

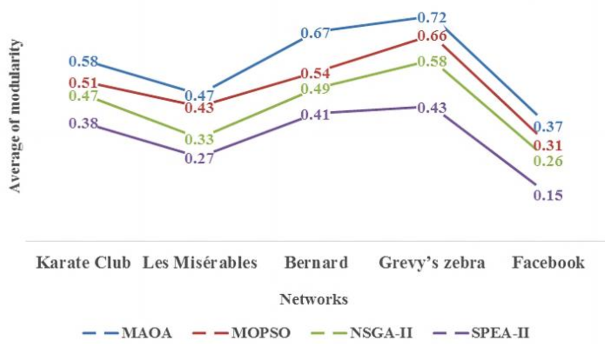

Fig. 14. Comparison of algorithms in terms of modularity

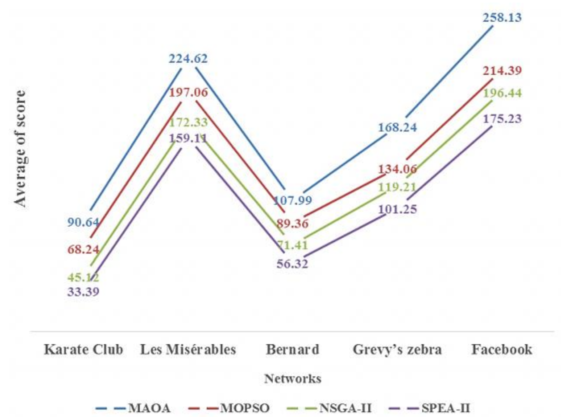

Fig. 15. Comparison of algorithms in terms of community score

6. Conclusions

This paper studied the community detection problem as a multi-objective optimization problem. A multi-objective multi-agent optimization algorithm called the MAOA was proposed to find appropriate partitions of networks by optimizing modularity and community score, concurrently. The MAOA has been inspired by the multi-agent system and swarm intelligence. For the proposed algorithm, new search operators based on social, autonomous and self-learning behaviors of agents were designed. Moreover, a new procedure was developed for adjusting the environment containing agents. Besides, the MAOA uses the weighted sum method (WSM) in finding the global best agent and leader agent of each group. The performance of the proposed algorithm was evaluated and validated by comparing its results to the outputs of three other meta-heuristics. All algorithms were tuned by the Taguchi method. Comparisons were made based on five real-world networks in terms of several metrics including normalized mutual information (*NMI*), mean ideal distance (*MID*), diversification metric (*DM*), computation time (CPU time) and *C*-metric. Results demonstrate that the MAOA has been more successful in providing better results in terms of most of the performance measures. To extend the current study, it is possible to test the effectiveness of the proposed algorithm in larger networks. For another study, the agents can have the ability to choose their social or autonomous behaviors. The agents can also have the right to refuse the role of leadership in groups. Furthermore, other multi-criteria decision making methods can be embedded in multi-objective multi-agent algorithms. Development of other procedures for finding the global best agent and the leader agent of each group is another interesting topic for further studies. In another study, the proposed algorithm can be compared with more recent algorithms to evaluate its performance comparing to state-of-the-art methods.

References

[1] K.R. Zalik, and B. Zalik, “Multi-objective evolutionary algorithm using problem-specific genetic operators for community detection in networks”, Neural Computing and Applications, Vol. 1, No. 9, 2017, pp. 1-14.

[2] S. Fortunato, “Community detection in graphs”, Physics Reports, Vol. 486, No. 3, 2010, pp. 1-100.

[3] Z. Zhao, S. Feng, Q. Wang, J.Z. Huang, G.J. Williams, and J. Fan, “Topic oriented community detection through social objects and link analysis in social networks”, Knowledge-Based Systems, Vol. 26, 2012, pp. 164-173, DOI: https://doi.org/10.1016/j.knosys.2011.07.017.

[4] S. Rahimi, A. Abdollahpouri, and P. Moradi, “A multi-objective particle swarm optimization algorithm for community detection in complex networks”, Swarm and Evolutionary Computation, Vol. 39, 2018, pp. 297-309, DOI: https://doi.org/10.1016/j.swevo.2017.10.009.

[5] C. Pizzuti, “GA-Net: A genetic algorithm for community detection in social networks”, In Proc. Parallel Problem Solving from Nature–PPSN X, Springer, 2008, pp. 1081-1090, DOI: https://doi.org/10.1007/978-3-540-87700-4_107.

[6] M. Gong, B. Fu, L. Jiao, and H. Du, “Memetic algorithm for community detection in networks”, Physical Review E, Vol. 84, 2011, pp. 1-9, DOI: 10.1103/PhysRevE.84.056101.

[7] C. Pizzuti, “A multi-objective genetic algorithm to find communities in complex networks”, IEEE Transactions on Evolutionary Computation, Vol. 16, 2012, pp. 418-430, DOI: 10.1109/TEVC.2011.2161090.

[8] C. Shi, Z. Yan, Y. Cai, and B. Wu, “Multi-objective community detection in complex networks”, Applied Soft Computing, Vol. 12, No. 2, 2012, pp. 850-859.

[9] M. Gong, L. Ma, Q. Zhang, and L. Jiao, “Community detection in networks by using multi-objective evolutionary algorithm with decomposition”, Physica A: Statistical Mechanics and its Applications, Vol. 391, No. 15, 2012, pp. 4050-4060.

[10] J. Li, and Y. Song, “Community detection in complex networks using extended compact genetic algorithm”, Soft computing, Vol. 17, No.6, 2013, pp. 925-937.

[11] B. Amiri, L. Hossain, J.W. Crawford, and R.T. Wigand, “Community detection in complex networks: Multi–objective enhanced firefly algorithm”, Knowledge-Based Systems, Vol. 46, 2013, pp. 1-11, DOI: https://doi.org/10.1016/j.knosys.2013.01.004.

[12] Q. Cai, M. Gong, B. Shen, L. Ma, and L. Jiao, “Discrete particle swarm optimization for identifying community structures in signed social networks”, Neural Networks, Vol. 58, 2014, pp. 4-13, DOI: https://doi.org/10.1016/j.neunet.2014.04.006.

[13] M. Gong, Q. Cai, X. Chen, and L. Ma, “Complex network clustering by multi-objective discrete particle swarm optimization based on decomposition”, IEEE Transactions on Evolutionary Computation, Vol. 18, No.1, 2014, pp. 82-97.

[14] Q. Cai, M. Gong, L. Ma, S. Ruan, F. Yuan, and L. Jiao, “Greedy discrete particle swarm optimization for large-scale social network clustering”, Information Sciences, Vol. 316, 2015, pp. 503-516, DOI: https://doi.org/10.1016/j.ins.2014.09.041.

[15] Y. Zhou, J. Wang, N. Luo, and Z. Zhang, “Multi-objective local search for community detection in networks”, Soft Computing, Vol. 20, No.8, 2016, pp. 3273–3282.

[16] R. Shang, S. Luo, W. Zhang, R. Stolkin, and L. Jiao, “A multi-objective evolutionary algorithm to find community structures based on affinity propagation”, Physica A: Statistical Mechanics and its Applications, Vol. 453, 2016, pp. 203-227, DOI: https://doi.org/10.1016/j.physa.2016.02.020.

[17] H.S. Cheraghchi, and A. Zakerolhosseini, “COGNISON: A Novel Dynamic Community Detection Algorithm in Social Network”, Journal of Information Systems and Telecommunication, Vol. 4, No. 2, 2016, pp. 78-84.

[18] S. Bilal, and M. Abdelouahab, “Evolutionary algorithm and modularity for detecting communities in networks”, Physica A: Statistical Mechanics and its Applications, Vol. 473, 2017, pp. 89-96, DOI: https://doi.org/10.1016/j.physa.2017.01.018.

[19] L. Li, L. Jiao, J. Zhao, R. Shang, and M. Gong, “Quantum-behaved discrete multi-objective particle swarm optimization for complex network clustering”, Pattern Recognition, Vol. 63, 2017, pp. 1-14, DOI: https://doi.org/10.1016/j.patcog.2016.09.013.

[20] J. Handl, and J. Knowles, “An evolutionary approach to multi-objective clustering”, IEEE transactions on Evolutionary Computation, Vol. 11, No.1, 2007, pp. 56-76.
[21] N.R. Jennings, K. Sycara, and M. Wooldridge, “A roadmap of agent research and development”, Autonomous Agents and Multi-Agent Systems, Vol. 1, No. 1, 1998, pp. 7–38.
[22] J. Li, H. Jing, and Y.Y. Tang, “Multi-agent oriented constraint satisfaction”, Artificial Intelligence, Vol. 136, No. 1, 2002, pp. 101-144.
[23] X.L. Zheng, and L. Wang, “A multi-agent optimization algorithm for resource constrained project scheduling problem”, Expert Systems with Application, Vol. 42, No. 15-16, 2015; pp. 6039-6049.
[24] W.C. Zhong, J. Liu, M.Z. Xue, and L.C. Jiao, “A multi-agent genetic algorithm for global numerical optimization”, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol. 34, No. 2, 2004, pp. 229-244.
[25] K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan, “A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II”, IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2, 2000, pp. 182-197.
[26] R.T. Marler, and J.S. Arora, “The weighted sum method for multi-objective optimization: new insights”, Structural and Multidisciplinary Optimization, Vol. 41, No. 6, 2010, pp. 853-862.
[27] C. A. Coello Coello, and M. S. Lechuga, "MOPSO: a proposal for multiple objective particle swarm optimization", In Proc. Congress on Evolutionary Computation (CEC'02), Honolulu, HI, USA, Vol. 2, 2002, pp. 1051-1056, DOI: 10.1109/CEC.2002.1004388.
[28] W. Sheng, Y. Liu, X. Meng, and T. Zhang, “An Improved Strength Pareto Evolutionary Algorithm 2 with application to the optimization of distributed generations”, Computers & Mathematics with Applications, Vol. 64, No. 5, 2012, pp. 944-955.
[29] W. W. Zachary, "An information flow model for conflict and fission in small groups", Journal of anthropological research, Vol. 33, No.4, 1977, pp. 452-473.
[30] D.E. Knuth, “The Stanford Graph Base: A Platform for Combinatorial Computing”, Addison-Wesley, Reading, MA (1993), ISBN: 0-201-54275-7.
[31] H. R. Bernard, P. D. Killworth, and L. Sailer, "Informant accuracy in social network data IV: A comparison of clique-level structure in behavioral and cognitive network data", Social Networks, Vol. 2, No.3, 1980, pp. 191-218.
[32] S. R. Sundaresan, I. R. Fischhoff, J. Dushoff, and D. I. Rubenstein, "Network metrics reveal differences in social organization between two fission–fusion species, Grevy’s zebra and onager", Oecologia, Vol. 151, No. 1, 2007, pp. 140-149.
[33] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach", IEEE Transactions on Evolutionary Computation, Vol. 3, No. 4, 1999, pp. 257-271.
[34] V. Hajipour, E. Mehdizadeh, and R. Tavakkoli-Moghaddam, “A novel Pareto-based multi-objective vibration damping optimization algorithm to solve multi-objective optimization problems”, Scientia Iranica, Vol. 21, No. 6, 2014, pp. 2368–2378.
[35] B. Golpalsamy, B. Mondal, and S. Ghosh, “Taguchi method and ANOVA: An approach for process parameters optimization of hard machining while machining hardened steel”, Journal of Scientific & Industrial Research, Vol. 68, 2009, pp. 686-695.
[36] S.H.A. Rahmati, V. Hajipour, and S.T.A. Niaki, “A soft-computing Pareto-based meta-heuristic algorithm for a multi-objective multi-server facility location problem”, Applied Soft Computing, Vol. 13, No. 4, 2013, pp. 1728-1740.
[37] J. Gao, R. Chen, W. Deng, “An efficient tabu search algorithm for the distributed permutation flowshop scheduling problem”, International Journal of Production Research, Vol. 51, No. 3, 2013, pp. 641-651.

Amir Hossein Hosseinian obtained his BSc in industrial engineering from Ershad University, Damavand, Iran, and accomplished MSc in industrial Engineering at Buali-sina University, Hamedan, Iran. Currently, he is a PhD candidate in industrial engineering at Islamic Azad University, Tehran North Branch, Tehran, Iran. His research interests are project scheduling, applied operations research, social network analysis and meta-heuristics.

Vahid Baradaran received his Ph.D. degree from Tarbiat Modares University in 2010. He is assistant professor in Islamic Azad University-Tehran North Branch. He thought some courses such as Multivariate Statistical Analysis, Data Mining and Data Analysis in Master Science and Ph.D. courses.