A New Adaptive Genetic Algorithm for Community Structure Detection

Yilmaz Atay and Halife Kodaz

Abstract Community structures exist in networks which has complex biological, social, technological and so on structures and contain important information. Networks and community structures in computer systems are presented by graphs and subgraphs respectively. Community structure detection problem is *NP-hard* problem and especially final results of the best community structures for large-complex networks are unknown. In this paper, to solve community structure detection problem a genetic algorithm-based algorithm, *AGA-net,* which is one of evolutionary techniques has been proposed. This algorithm which has the property of fast convergence to global best value without being trapped to local optimum has been supported by new parameters. Real-world network which are frequently used in literature has been used as test data and obtained results have been compared with 10 different algorithms. After analyzing the test results it has been observed that the proposed algorithm gives successful results for determination of meaningful communities from complex networks.

Keywords Combinatorial optimization · Community structure detection · Complex networks · Evolutionary computation · Genetic algorithm · Modularity

1 Introduction

Understanding networks provides us very important information about the extraction of meaningful information from complex systems. In eliciting meaningful information from these networks the importance of structures which are named as community structures is huge. The graph structures are used to present the real-world networks. Community structures or clusters can be considered as subgraphs which are partially or completely independent from each other in graph structures.

Y. Atay · H. Kodaz(✉)
Department of Computer Engineering, Selcuk University, Konya, Turkey
e-mail: {yilmazatay,hkodaz}@selcuk.edu.tr

K. Lavangnananda et al. (eds.), *Intelligent and Evolutionary Systems*,
Proceedings in Adaptation, Learning and Optimization 5,
DOI: 10.1007/978-3-319-27000-5_4

As an example, tissues or organs which have the same role in the human body can be considered as clusters [1]. Community structure detection (CSD) is important to an understanding of the biological, economic, social, technological and so on networks. These networks can be synthetic or real-world networks. For real-world networks we can give some example like economic structure networks [2], food networks [3], networks of chemical interaction between proteins and molecules in cells [4-6] and social networks like networks of determination of friendship in groups, relation analysis networks and networks of detection of terrorist attacks [7].

Objects and connections in networks are presented with nodes and edges respectively. Graph structures which are used to represent the above given networks are referred to as simplest form of undirected networks [8].

Community mining problem (CMP) refers to discovery of meaningful subgraph in many complex networks data [9]. In this paper, many real-world data were analyzed by CMP and the obtained results are given in the experimental results section.

Many methods have been developed for detection of community structures in complex networks. These methods give successful results according to many properties yet generalizations about obtaining the best result cannot be made. Performances of the algorithms in literature are very low on large networks. In addition, for detection of community, many algorithms need prior knowledge like community number. Optimal grouping in network is a very difficult problem. Therefore, CSD problem is a *nondeterministic polynomial time - hard* problem [10, 11].

Usually to solve complex problems like CSD two different methods are proposed. These are exact and (meta-)heuristic methods [12]. From these two methods (meta-)heuristic method can offer more convenient solutions for difficult and complex problems than exact methods. Algorithms like memetic and genetic are covered by (meta-)heuristic methods and are also known as bio-inspired algorithms [12]. These algorithms use various community calculating measures according to their own methods in the CSD problems. The most common calculation measure used recently which is recommended by Girman and Newman is *modularity Q* measure [13, 14].

So far the most well-known community detection algorithm is Girman-Newman (*GN*) algorithm [13, 15]. Fast Newman (*FN*) algorithm is an algorithm based on the maximum *modularity Q* [16]. Similarly, another algorithm based on maximum *modularity Q* is called Fast Unfolding algorithm [17]. In addition to these, algorithms like Random Walks [18], Eigenvectors [19], Label Propagation (*LP*) [20] with Spin Glass Type Potts method [21] and *LTE* (Local Tightness Expansion) algorithm [22] are used in the literature. *FN* [16], community detection algorithm for large networks which is proposed by Clauset et al. [23], Extremal Optimization [24] and other algorithms like this have $O(e^3)$ complexity in terms of time complexity. Here the e refers to the number of edges [25].

Time complexity increases in a huge amount as the size of the network increases. In small or regular size networks community detection can be done very easily with algorithms given above but as the network size get larger existing algorithms are inadequate in terms of both performance and success. Also when inclusion of prior knowledge to these algorithms become mandatory, discovery of new and efficient algorithms are inevitable. Due to this need, CSD problems are tried to solve with algorithms like genetic algorithms, particle swarm optimization algorithm, ant colony optimization algorithm, memetic algorithms, and differential algorithm. In this paper genetic algorithm which is one of the above algorithms constitutes the basic structure of the proposed algorithm. Genetic algorithm is already very successful in terms of computation, time complexity and solution convergence in *NP-hard* problems and it is almost used in most problems in literature. For the first time Tasgin et al. [26] used genetic algorithms in CSD problem. The method that they developed was named as *GATB* (or *GATHB* [27]) [25]. After that both genetic and other algorithms started to be used widely in CSD problems. In particular, to find the optimal Q value in the most economical way, many methods have been developed by making several changes on methods of genetic algorithm like mutation, crossover, selection and so on.

Shi et al. tried to solve the CSD problem by using genetic algorithm based *GACD* [28] algorithm. They tested their own method with real-world networks which are used quite a lot in the literature and compare the results with *GN* [13], *GN Fast* [23] and *GATB* [25] results. When the obtained results were analyzed, it was stated that genetic algorithm-based algorithms such as *GACD* and *GATB* were quite effective to solve CSD problems. There are many advantages of the developed methods based on genetic algorithm. For example *GATB* [25] has a time complexity of $O(e)$ and does not require any prior knowledge for CSD. In this paper, a new approach based on genetic algorithms has been proposed and has been named as *AGA-net*. *AGA-net* has a time complexity of $O(e)$ and does not require any prior knowledge. The proposed algorithm is based on *adaptive* design of genetic algorithm to reach the most appropriate solution in less time for CSD problem. *AGA-net* was tested in networks given in section 3 and the obtained results were compared with some existing algorithms in literature (see section 3).

1.1 Community Structure Detection

When any given network presented by graph structure, obtained community structures can be considered as subgraphs which have quality or quantity like maximum common feature in itself, number of interactions, positional similarities and so on. Nodes which are the elements of these structures should have maximum interaction and common properties with its own community nodes and less interaction and common properties with other community nodes. Group of people who have strong relationship in social environment, colony of living creatures in environmental networks who feed on each other and cluster of computers having maximum data exchange cooperation can be examples related with CSD.

Let the given $G(V,E)$ graph structure represent undirected and unweighted network. Here the graph G has V set of nodes (vertices) and E set of edges (links).

$$V = \{v_i \mid i = 1, 2, 3, \ldots, n\} \text{ and } E = \{e_j \mid j = 1, 2, 3, \ldots, m\}$$

Here; i, j, n and m represent the node index, edge index, number of node and total edge number respectively. Let define adjacency matrix as *Adj* with *nxn* size. And let *Adj* matrix show the relationship of the elements of set V by the elements of the set E. *Adj* adjacency matrix is generated by Equation (1) [23].

$$Adj = \begin{cases} 1 & \text{if } i. \text{ and } j. \text{ nodes are connected,} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Modularity Q for graph G is given in Equation (2). This fitness function has been proposed by Newman and Girvan in their work by the name of *Finding and Evaluating Community Structure in Networks* [15].

$$Q = \frac{1}{2\times m} \sum_{ij} \left(Adj_{(i,j)} - \frac{k_i \times k_j}{2\times m} \right) \times \delta(C_i, C_j) \tag{2}$$

Where Q is named as *modularity Q* and expresses the objective function to be maximized. $Adj_{(i,j)}$, represents the adjacency matrix of given G graph. m demonstrates the total number of edge in network and calculated by Equation (3). k_i demonstrates the degree of i^{th} node, k_j demonstrates the degree of j^{th} node and as an example k_i can be calculated by Equation (4). C_i and C_j demonstrate the i^{th} and j^{th} node community respectively. $\delta(C_i, C_j)$ is a function which demonstrates the i^{th} and j^{th} node whether exist in the same community. $\delta(C_i, C_j)$ function is calculated by Equation (5).

$$\mathrm{m} = \frac{1}{2} \sum_{ij} Adj_{(i,j)} \tag{3}$$

$$k_i = \sum_j Adj_{(i,j)} \tag{4}$$

$$\delta = \begin{cases} 1 & \text{if } C_i = C_j \\ 0 & \text{if } C_i \neq C_j \end{cases} \tag{5}$$

Detection of community structure according to fitness value was done by Tasgin et al. in 2007 by the name of *Community Detection in Complex Networks using Genetic Algorithms* [26]. In the specified paper, proposed algorithm was named as *GATHB* [26, 27]. After publication of this paper, many evolutionary algorithms were applied to CSD problems. The *AGA-net* algorithm that we proposed has also used the same objective function as *GATHB* algorithm which is given in Equation (2).

1.2 Genetic Algorithm

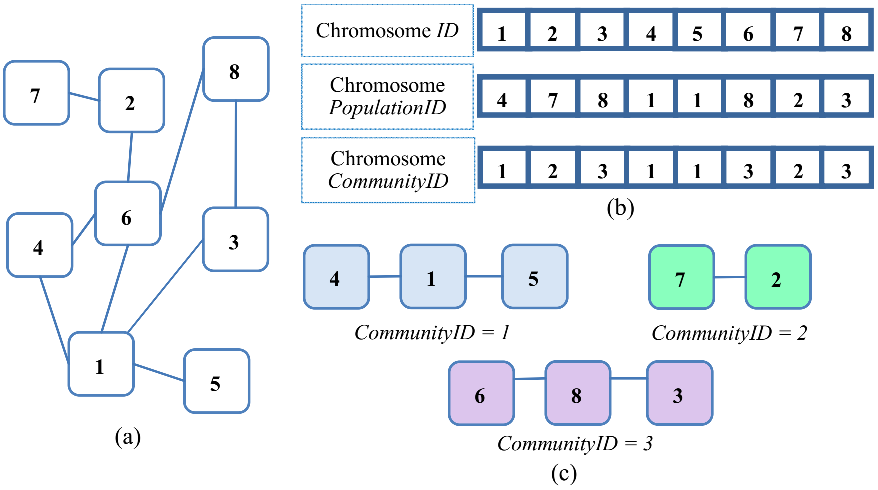
GAs were first described by John Holland in the 1960s and further developed by Holland and his students and colleagues at the University of Michigan in the 1960s and 1970s. Holland's goal was to understand the phenomenon of "adaptation" as it occurs in nature and to develop ways in which the mechanisms of natural adaptation might be imported into computer systems. Holland's 1975 book Adaptation in Natural and Artificial Systems (Holland, 1975) presented the GA as an abstraction of biological evolution and gave a theoretical framework for adaptation under the GA [29]. GA is a population based algorithm and can be modal without requiring any prior knowledge or assumptions. Thus this algorithm can be adapted to many problems and has a general-purpose structure property.

2 The Proposed Algorithm

In this paper to solve CSD problem, *AGA-net* algorithm has been proposed which is based on genetic algorithm. As every node of network in CSD problem has a limited number of neighbors therefore the probability of selected neighbor to be selected again is very high. This situation, searching for the best solution, it may cause to be entered to a vicious cycle in the various networks. This problem has been solved by the help of genetic operators given in section 2.4. Thus for rapid convergence to best solution, better solutions has been selected by elitism while entering in to vicious circle is also prevented by crossover and mutation mechanism. The proposed algorithm has the property of convergence of best global *modularity Q* without being trapped in local best solution. *AGA-net* also has a linear time complexity. In addition to basic parameters and operators of standard genetic algorithm, specific changes for CSD problems and new parameters have been included. The proposed algorithm has been named as *Adaptive Genetic Algorithm* (*AGA-net*). The adaptive phrase used herein indicates that every mechanism of algorithm can be adapted to all networks. The proposed algorithm can be operated for all networks on CSD problem without being depended to any internal or external data, with its new specific parameters. Proposed algorithm's steps are given below in detail under separate headings.

2.1 Genetic Representation

The proposed algorithm uses locus-based adjacency representation (*LAR*) structure for graph based representation [30]. Each gene in chromosome holds two different information (*communityID* and *populationID*). Information about these is given in Fig. 1. The first information stores randomly selected neighbor node from i^{th} node neighbors. The second information keeps community knowledge (*communityID*) of i^{th} node for communities generated by the first information. An example of 8-node network has been given in Fig. 1(a), Fig. 1(b) shows an example of chromosomes generated according to the given network and Fig. 1(c) provides community structures generated from given chromosome information. Obtained community structures have been given in different colors.



(a) (b) (c)

Fig. 1 Examples of a network with 8 node, a chromosome and obtained community structures

The chromosome given in Fig. 1(b) is represented by 3 different array which keeps information about *ID, populationID and communityID.* The first array keeps node sequence number, the second keeps selected neighbor node and the third array keeps information about the node's community.

2.2 Population Initialization

The proposed algorithm produces chromosomes as the size of population in initialization process. Each gene in the chromosome represents a node. The second array from the arrays given in Fig. 1(b) selects randomly neighboring node according to *ID*. After all population formed in this way the 3rd array is formed according to 2nd array which is given in Fig. 1(b). The 2nd array which is given in Fig. 1(b) provides forming of necessary community's list during *modularity Q* calculation.

While determining *CommunityID* of a gene inside chromosomes there should be neighborhood of neighbor gen with existing gen. According to this principle solution space becomes restricted and it saves time.

2.3 Fitness Function

In this paper, *modularity Q* has been used as fitness function. This measure has been first used by Newman and Girvan [15] in 2004. The function has been given in Equation (2). CSD problem can be considered as combinatorial optimization problem according to given objective function. The objective function in the best graph clustering reaches maximum Q value. Q value varies in the range of -1 to +1.

2.4 Genetic Operators

Elitism, selection, crossover and mutation operations have been used in proposed algorithm. Each operator parameters used in the process is adapted to CSD problem to achieve the most suitable solution. Unlike standard genetic algorithm new parameters have been included in elitism, crossover and mutation operators. Operators and parameters proposed by *AGA-net* algorithm are presented in detail below.

Elitism. This operator is used at two stage of the algorithm. In the first stage, it is selected to transfer chromosomes at the rate of *elitismRate (%)* which has the best *Q* value in the population to the next generation. At the second stage, new chromosomes with the better *Q* values change place with bad chromosomes at the same rate. Here *elitismRate* ensure the elimination of the worst chromosome from solution cluster. This parameter has been used in small rates to not reduce the chromosome diversity.

Selection. The process of the production of a new generation individual selection process was carried out with the roulette wheel selection (RWS) [31]. In the proposed algorithm selection process according to RWS method are done as follows.

- The fitness value of each chromosome is calculated and sum of all chromosomes fitness value in the population is calculated by Equation (6).

$$TfitnessQ = \sum_{t=1}^{popSize} fitnessQ_t \tag{6}$$

- The selection probability of each chromosome is calculated by Equation (7).

$$P_t = fitnessQ_t / TfitnessQ \tag{7}$$

- The cumulative total is calculated for each chromosome and cumulative probability is determined by Equation (8).

$$Q_t = \sum_{k=1}^{t} P_k \tag{8}$$

- A random number between 0 and 1 is generated. A chromosome is selected according to the generated number's Q_t range. So, chromosomes to be transferred to the next generation are selected.

Crossover. Two different parameters associated with this operator by the names of *crossover rate* (*CR*) and *crossover choice* (*CC*) has been defined. Of these the *CR* parameters will be subjected to the individuals crossover process in the population and determine the number of subjected process. The *CC* parameter will then provide the production of change control sequence for pairs of chromosome subjected to crossover process. The sequence is generated such that the *CC* value would be 0 if it is smaller than the generated random number and 1 if it is bigger. The occurred crossover process has been given in Fig. 2.

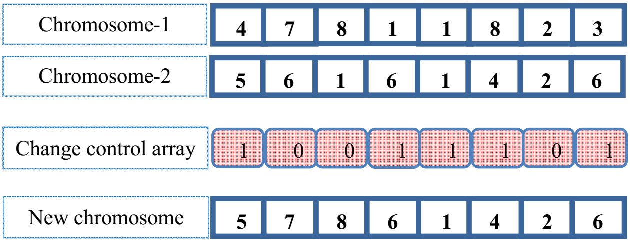


Fig. 2 Crossover operation of *AGA-net* algorithm

Mutation. In the proposed algorithm, mutation process is carried out in two cases, *one-point* and *multi-point*. The first one is single point mutation and the second one is multiple point mutations. Also two parameters have been used in mutation process. The first parameter is *mutation rate* (*MR*) and the second parameter is *multi-point mutation rate* (*multiP*). *MR* parameter is selected in a small ratio and it will determine whether the incoming chromosome mutates or not. And the *multiP* parameter allows the selection of one of the single or multiple mutations option. If the value of this parameter is less than randomly generated number, single point mutation but if it is equal or greater than the randomly generated number multi-point mutation applies. Representative examples showing this process has been given in Fig. 3. Here, each selected genes are mutated by the neighborhood condition (refer to Fig. 1(a)).

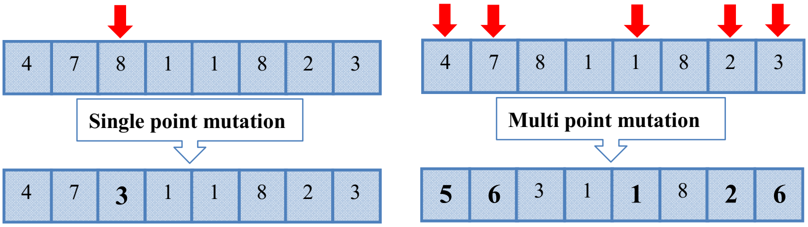


Fig. 3 Single-point and multi-point mutations

3 Experimental Results

In this section the *AGA-net* algorithm has been tested on 5 real networks which are (*Z*) Zachary's Karate Club [32], (*D*) Dolphins Social Network [33], (*A*) American College Football [13], (*B*) Books about US Politics [14] and (*C*) Cattle Protein Interactions (*IntAct*) [34]. These networks are organized as undirected and unweighted. Each node in the network is identified by an *ID*. For example the first node of Cattle Protein Interactions (*IntAct*) network which is *AATM_BOVIN* [35] has been identified by *ID* number 1. Networks and their properties used in the experiments are given in Table 1. All experiments have done on a computer which has the following specifics:

Microsoft Windows 7 (*x64*) OS environment, Intel (R) Core (TM) *i7-3632QM* CPU @ *2.20* GHz processor and *4* GB of RAM.

Table 1 Networks and their properties used in the experiments

Networks***	***Number of Nodes (V)***	***Number of Edges (E)***
[*Z*] Zachary's Karate Club	34	78
[*D*] Dolphins Social Network	62	159
[*A*] American College Football	115	613
[*B*] Books about US Politics	105	441
[*C*] Cattle Protein Interactions (IntAct)	268	303

* The given node and edge numbers have been obtained after turning networks to undirected and unweighted by subtracting self-loops featured nodes.

The algorithm was run 50 times for networks in Table 1. The maximum *Q*, average *Q* and standard deviation was recorded as test results. The number of population used in the experiments varies according to the size of the network. Population numbers used for networks according to Table 1 order are 20, 30, 100, 80 and 50 respectively. Also for other parameters the following values have been used; for *elitismRate* 0.05, for *crossover rate* 0.8, for *crossover choice* 0.5, for *mutation rate* 0.2 and *multi point mutation* 0.4. These parameters have been tested with experiments inside their borders and the effect of these values to the algorithm has been tested through trial and error. The best results were achieved in this parameter value for all networks.

The proposed algorithm's experimental results were compared with algorithms which are given in Table 2. The results obtained are given in Table 2. In the table the best Q values are indicated in bold and for ease of reading the decimals have been rounded to three digits. For example, the best *Q* value of *Z* network was approximately 0.419789612097304 and it was rounded to 0.420. Average *modularity Q* values and standard deviation values for *AGA-net* algorithm are given in Table 3.

Table 2 Comparison of the results according to *modularity Q* values

Networks ***Algorithms***	***Z***	***D***	***A***	***B***
DECD [10]	0.416	-	**0.605**	-
Eigenvector [19]	0.393	0.491	0.488	0.467
GACD [37]	**0.420**	**0.529**	0.604	**0.527**
GATHB [8]	0.402	0.522	0.551	0.518
GN [13]	0.401	0.519	0.599	0.510
FN [16]	0.381	0.510	0.550	0.502
MA-COM [39]	**0.420**	**0.529**	**0.605**	**0.527**
MA-Net [8]	**0.420**	**0.529**	**0.605**	**0.527**
MENSGA [38]	**0.420**	0.527	0.604	0.526
MOGA-Net [36]	0.416	0.505	0.515	0.518
Proposed method (AGA-net)	**0.420**	**0.529**	**0.605**	**0.527**

Apart from the networks given in Table 2, maximum *modularity Q* value obtained for network named as Cattle Protein Interactions (IntAct) is 0.720. After analyzing present studies and as for as it is known this network (*C*) has not been used in CSD problem before. So, this network was not included in the list of comparison given in Table 2.

Table 3 The average *modularity Q* and standard deviation values

Networks	*Z*	*D*	*A*	*B*	*C*
Average Q values	0.420	0.528	0.601	0.527	0.719
Standard deviations	± 0	± 0.000790	± 0.002986	± 0.000137	± 0.000017

Best global and local *modularity Q* values were recorded in each run for all networks. According to obtained results, the average and best *Q* values are almost the same. Also a low level of standard deviation values indicates that the proposed algorithm gave consistent results. For example the standard deviation for *Z* network is zero therefore both the average and the best *Q* values are the same which is 0.420. There are not much different between the obtained results for other networks.

While observing the community numbers generated according to the best *Q* values for *Z*, *D*, *A*, *B* and *C* networks the following community numbers 4, 5, 10, 5 and 40 have been obtained respectively.

When table 2 is examined, till now the *AGA-net* algorithm has reached the best *Q* modularity value for four networks (*Z*, *D*, *A* and *B*) which have been used in literature before. When genetic algorithm-based method proposed by us is compared with other genetic algorithm-based methods given in Table 2 (*GACD* and *GATHB*), it is seen that our proposed method has given better results than other two methods. The most important reason for this is the other two methods mentioned above have limited genotype production mechanism while the crossover and mutation mechanisms used in our proposed method narrows down the solution space to converge to optimal solution.

4 Conclusions

In this paper, CSD problem which is often used in complex networks analysis is discussed and the meaningful information from real-world network has been determined. To test the accuracy of the recommended algorithm the obtained results have been compared with state-of-the-art methods in the literature. In the experiments, four social networks and a biological network have been used. In addition to existing operators and parameters of standard genetic algorithm the *AGA-net* algorithm has been supported by the proposed genetic operators and parameters which are given in section 2.4. These operators and parameters have provided fast convergence of proposed algorithm to global best *Q* value. When analyzing the experimental results it is observed that the *AGA-net* algorithm has obtained the best *modularity Q* values known so far for all networks. These results indicate that *MA-Net* [8] and *MA-COM* [39] algorithms yielded similar results with the proposed algorithm. While the proposed algorithm has obtained the same results with *GACD* [37] algorithm in 3 networks (*Z*, *D* and *B*), it has obtained better result in *A* network. Analogously the proposed algorithm has obtained the same results in *A* network with *DECD* [10] algorithm and better result in *Z* network. And also the proposed algorithm with compare to *MENSGA* algorithm [38] has obtained the same result in *Z* network and better results in other networks.

As a result, the success and consistency of the proposed algorithm can be understood from both comparison and standard deviation tables. In addition, *AGA-net* does not require any prior knowledge and works fast. Thus, with the proposed mechanism almost 20 percent time saving has been provided for each network. This algorithm has been designed in a way which does not consider the size of nodes and edges number therefore it can be applied to all synthetic and real-world networks. After analyzing the test results it has been observed that the proposed algorithm gives successful results for determination of meaningful communities from complex networks.

Acknowledgments This study was supported by Scientific Research Project of Selcuk University and TUBITAK (The Scientific and Technological Research Council of Turkey, 2211-C Domestic Doctoral Scholarship Program Intended for Priority Areas, No. 1649B031402383).

References

1. Fortunato, S.: Community Detection in Graphs. Physics Reports **486**(3), 75–174 (2010)
2. Jackson, M.O.: Social and Economic Networks, vol. 3. Princeton University Press, Princeton (2008)
3. Dunne, J.A., Williams, R.J., Martinez, N.D.: Food-Web Structure and Network Theory: The Role of Connectance and Size. Proceedings of the National Academy of Sciences. **99**(20), 12917–12922 (2002)
4. Gavin, A.-C., et al.: Proteome survey reveals modularity of the yeast cell machinery. Nature **440**, 631–636 (2006)
5. Krogan, N.J., et al.: Global landscape of protein complexes in the yeast saccharomyces cerevisiae. Nature **440**(7084), 637–643 (2006)
6. Lee, J., Hidden, L.J.: information revealed by optimal community structure from a proteincomplex bipartite network improves protein function prediction (2013). http://www.ncbi.nlm.nih.gov/pubmed/23577106
7. Marcus, S., Moy, M., Coffman, T.: Social network analysis. Mining Graph Data, 443–467 (2007)
8. Naeni, L.M., Berretta, R., Moscato, P.: MA-net: a reliable memetic algorithm for community detection by modularity optimization. In: Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, vol. 1, pp. 311–323. Springer International Publishing (2015)

9. Liu, D., Jin, D., Baquero, C., He, D., Yang, B., Yu, Q.: Genetic Algorithm with a Local Search Strategy for Discovering Communities in Complex Networks. International Journal of Computational Intelligence Systems **6**(2), 354–369 (2013)
10. Jia, G., et al.: Community detection in social and biological networks using differential evolution. In: Learning and Intelligent Optimization, pp. 71–85. Springer Berlin, Heidelberg (2012)
11. Brandes, U., et al.: On Modularity Clustering. IEEE Transactions on Knowledge and Data Engineering **20**(2), 172–188 (2008)
12. Parpinelli, R.S., Lopes, H.S.: A Computational Ecosystem for Optimization: Review and Perspectives for Future Research. Memetic Computing **7**(1), 29–41 (2015)
13. Girvan, M., Newman, M.E.J.: Community Structure in Social and Biological Networks. Proceedings of the National Academy of Sciences **99**(12), 7821–7826 (2002)
14. Newman, M.E.J.: Modularity and Community Structure in Networks. Proceedings of the National Academy of Sciences **103**(23), 8577–8582 (2006)
15. Newman, M.E.J., Girvan, M.: Finding and Evaluating Community Structure in Networks. Physical Review E **69**(2), 026113 (2004)
16. Newman, M.E.J.: Fast Algorithm for Detecting Community Structure in Networks. Physical Review E **69**(6), 066133 (2004)
17. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast Unfolding of Communities in Large Networks. Journal of Statistical Mechanics: Theory and Experiment **2008**(10), P10008 (2008)
18. Pons, P., Latapy, M.: Computing communities in large networks using random walks. In: Computer and Information Sciences-ISCIS 2005, pp. 284–293. Springer Berlin, Heidelberg (2005)
19. Newman, M.E.J.: Finding Community Structure in Networks using The Eigenvectors of Matrices. Physical Review E **74**(3), 036104 (2006)
20. Raghavan, U.N., Albert, R., Kumara, S.: Near Linear Time Algorithm to Detect Community Structures in Large-Scale Networks. Physical Review E **76**(3), 036106 (2007)
21. Ronhovde, P., Nussinov, Z.: Local Resolution-Limit-Free Potts Model for Community Detection. Physical Review E **81**(4), 046114 (2010)
22. Huang, J., Sun, H., Liu, Y., Song, Q., Weninger, T.: Towards Online Multiresolution Community Detection in Large-Scale Networks. PloS one **6**(8), e23829 (2011)
23. Clauset, A., Newman, M.E.J., Moore, C.: Finding Community Structure in Very Large Networks. Physical Review E **70**(6), 066111 (2004)
24. Duch, J., Arenas, A.: Community Detection in Complex Networks using Extremal Optimization. Physical Review E **72**(2), 027104 (2005)
25. Tasgin, M., Bingol, A.: Communities detection in complex networks using genetic algorithm. In: Proc. of the European Conference on Complex Systems (ECSS 2006) (2006)
26. Tasgin, M., Herdagdelen, A., Bingol, H.: Community Detection in Complex Networks using Genetic Algorithms. arXiv preprint arXiv:0711.0491 [physics.soc-ph] (2007)
27. Li, J., Song, Y.: Community Detection in Complex Networks using Extended Compact Genetic Algorithm. Soft Computing **17**(6), 925–937 (2013)
28. Shi, C., Wang, Y., Wu, B., Zhong, C.: A New Genetic Algorithm for Community Detection, pp. 1298–1309. Springer, Heidelberg (2009)
29. Mitchell, M.: Genetic algorithms: An overview. Complexity **1**(1), 31–39 (1995)
30. Park, Y., Song, M.: A genetic algorithm for clustering problems. In: Proceedings of the Third Annual Conference on Genetic Programming, pp. 568–575 (1998)

31. Sastry, K., Goldberg, D., Kendall, G.: Genetic Algorithms: In: Search methodologies, pp. 97–125. Springer US (2005)
32. Zachary, W.W.: An Information Flow Model for Conflict and Fission in Small Groups. Journal of Anthropological Research, 452–473 (1977)
33. Lusseau, D.: The Emergent Properties of a Dolphin Social Network. Proceedings of the Royal Society of London B: Biological Sciences **270**(Suppl 2), S186–S188 (2003)
34. IntAct Protein-Protein Interaction Network of Cattle. http://biit.cs.ut.ee/graphweb/exampleInput/Cattle_protein_interactions_(IntAct).txt (last accessed date June 1, 2015)
35. UniProtKB, Bos taurus (Bovine) - P12344 (AATM_BOVIN). http://www.uniprot.org/uniprot/P12344 (last accessed date May 15, 2015)
36. Pizzuti, C.: A Multiobjective Genetic Algorithm to Find Communities in Complex Networks. IEEE Transactions on Evolutionary Computation **16**(3), 418–430 (2012)
37. Shi, C., Yan, Z., Wang, Y., Cai, Y., Wu, B.: A Genetic Algorithm for Detecting Communities in Large-Scale Complex Networks. Advances in Complex Systems **13**(01), 3–17 (2010)
38. Li, Y., Liu, G., Lao, S.Y.: A Genetic Algorithm for Community Detection in Complex Networks. Journal of Central South University **20**, 1269–1276 (2013)
39. Gach, O., Hao, J.-K.: A memetic algorithm for community detection in complex networks. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN 2012, Part II. LNCS, vol. 7492, pp. 327–336. Springer, Heidelberg (2012)