Multi-Objective Optimization for Overlapping Community Detection

Jingfei Du, Jianyang Lai, and Chuan Shi*

Beijing University of Posts and Telecommunications, Beijing, China
du1342157416@gmail.com, 330939204@qq.com, shichuan@bupt.edu.cn

Abstract. Recently, community detection in complex networks has attracted more and more attentions. However, real networks usually have number of overlapping communities. Many overlapping community detection algorithms have been developed. These methods usually consider the overlapping community detection as a single-objective optimization problem. This paper regards it as a multi-objective optimization problem and proposes a Multi-Objective evolutionary algorithm for Overlapping Community Detection (MOOCD). This algorithm simultaneously optimize two objective functions to get proper community partitions. Experiments on artificial and real networks illustrate the effectiveness of MOOCD.

Keywords: Complex network, overlapping community detection, multi-objective optimization, evolutionary algorithm.

1 Introduction

In recent years, there is a surge on community detection in complex networks. The main reason lies in that communities play special roles in the structure-function relationship, and thus detecting communities (or modules) can be a way to identify substructures which could correspond to important functions. Generally, communities are groups of nodes that are densely interconnected but only sparely connected with the rest of the network [1][2]. For example, on an online shopping site, users in the same community usually have the same taste in choosing similar goods. However, recent study shows that real networks usually have number of overlapping communities [21]. That is, some nodes in networks exist in multiple communities. It is reasonable in real world, since objects often have multiple roles. For example, a professor collaborates with researchers in different fields; a person has his family group as well as friends group at the same time, etc. So, in overlapping community detection, these objects should be divided into multiple groups.

Up till now, many overlapping community detection algorithms have been developed [11][13][14][20], which can be roughly classified as "node-based" or "link-based" methods. The node-based methods classify nodes of the network directly

* Corresponding author.

H. Motoda et al. (Eds.): ADMA 2013, Part II, LNAI 8347, pp. 489–500, 2013.

[20]. The link-based methods cluster the edges of network, and then map the final link communities to node communities by simply gathering nodes incident to all edges within each link communities [11]. The contemporary methods all consider the overlapping community detection as a single-objective optimization problem. That is, the overlapping community detection corresponds to discover a community structure that is optimal on one single-objective function. However, these single-objective algorithms may confine the solution to a particular community structure property because of only considering one objective function. When the optimization objective is inappropriate, these algorithms may fail. Moreover, the overlapping community structure can be evaluated from multiple criteria, which can comprehensively measure the quality of overlapping communities. Although multi-objective optimization has been applied for community detection [17][19], it has not been exploited for overlapping community detection.

In this paper, we first study the multi-objective optimization for overlapping community detection and propose a Multi-Objective evolutionary algorithm for Overlapping Community Detection (MOOCD). The algorithm employs a well-known multi-objective optimization framework for numerical optimization (PESA-II) [22], and uses two conflict objective functions. In addition, the effective genetic representation, operators and model selection strategies are designed. Experiments on typical artificial networks show MOOCD not only accurately detects overlapping communities but also comprehensively reveals community structures. Moreover, experiments on three real networks illustrate that MOOCD discovers more balanceable overlapping communities compared to other well-established algorithms.

2 Related Work

In this section, we will introduce the most related work, including community detection, overlapping community detection , and multi-objective optimization for community detection.

Community detection is crucial for analyzing structures of social networks. There are lots of algorithms aiming at finding proper community partition.One of the most known algorithms proposed so far is the Girvan-Newman (GN) algorithm that introduces a divisive method by iteratively cutting the edge with the greatest betweennes value [3]. Some improved algorithms have been proposed [23][24]. These algorithm are based on a foundational measure criterion of community, modularity, proposed by Newman [3].

Recently,some studies show that real networks usually have number of overlapping communities [21]. Many algorithms have been proposed to detect overlapping communities in complex networks, such as CPM [11], GA-NET+ [13], GaoCD [14], etc. CPM is the most widely used, but its coverage largely depends on the feature of network. GA-NET+, developed by Pizzuti, is the first algorithm that adopts genetic algorithm to detect overlapping communities. However, GA-NET+ costs so much computation in transformation between line graph and node. GaoCD is also a genetic algorithm. But the difference is that GaoCD is

a link-based algorithm. Besides these algorithms, some people extend conventional disjointed community detection criterions to overlapping ones. For example, Shen[15] introduced a practical extended modularity for finding overlapping communities. And Wang[16] also extended modularity Q and proposed an efficient method for adjusting classical algorithms to match the requirement for discovering overlapping communities.

However, because the definition of community is multi-objective, the community detection problem is multi-objective. And the conventional single-objective community detection methods have several crucial disadvantages. Therefore, there are some researchers who have been aware of the multi-objective community detection. For instance, Gong [17] solves the community detection by maximizing the density of internal degrees, and minimizing the density of external degrees simultaneously. Besides, Gong [18] provides a novel multi-objective immune algorithm to solve the community detection problem in dynamic networks. And Shi[19] formulated a multi-objective framework for community detection and proposes a multi-objective evolutionary algorithm for finding efficient solutions under the framework. However, there is few work applies multi-objective community detection methods to find overlapping community partitions.

3 Multi-Objective Evolutionary Algorithm for Overlapping Community Detection

In this section, we will describe the Multi-Objective algorithm for Overlapping Community Detection (MOOCD) in detail, which includes the algorithm framework, objective function, genetic representation, genetic operators and multi-objective model selection method.

3.1 Framework of the Algorithm

This paper applies the evolutionary algorithm (EA) to solve the multi-objective optimization problem. It can simultaneously generate a set of candidate solutions. The framework of MOOCD is described in Algorithm 1.

The framework of MOOCD is based on an existing multi-objective evolutionary algorithm: PESA-II [22]. Different from standard evolutionary algorithms, PESA-II follows standard principles of an EA with the difference that it maintains two populations of solutions: internal population and external population. External population contains non-dominated set, or Pareto front for each updating. A solution dominates other solutions if all objective functions of this solution are superior to other solutions. A solution is said to be Pareto optimal if and only if there is no other solution dominating it. Selection occurs at the interface between the two.

Algorithm 1 randomly generates genes and updates external population at first. At every iteration, internal population is filled with genes selected from external population. New genes are generated based on internal population through genetic operators. And external population is updated by the new genes. After

several iterations, model selection method is used to select a single solution from external population.

3.2 Objective Functions

As we said in Section 2, it is a good choice to use multiple objective functions to solve the drawbacks of the single-objective community detection algorithms. However, it is also a challenge to choose the objective functions. Different objective functions can reflect different characters of partitions. So ideal objective functions had better contain intrinsic conflicts and thus the optimal community partitions can be obtained through the trade-off of multiple objectives. Therefore, the following two objective functions are selected in this paper.

One of the two objective functions is partition density D, which is raised by Ahn [20]. The partition density D is a kind of link community evaluate function whose mathematical definition is as following.

$$D = \frac{2}{M} \sum_{c} m_c \frac{m_c - (n_c - 1)}{(n_c - 2)(n_c - 1)} \tag{1}$$

Define $P = \{P_1, \ldots, P_c\}$ as a partition of the network's links into C subsets. $m_c = |P_c|$ is the number of links in subset c. $n_c = |U_{e_{ij}} \in P_c\{i, j\}|$ represents the number of nodes incident to links in subset c. D_c refers to the link density of subset c. The intuitive meaning of D is the link density within the community. As we said above, partition density is a link based function and it is also appropriate when it is applied to evaluation overlapping community partition.

The other objective function is extended modularity which is proposed by Shen [15]. This objective function is extended from modularity which is used by many community detection methods. Traditional modularity measures the number of within-community edges, relative to a null model of a random graph with the same degree distribution. But we can say that traditional modularity definition cannot be applied to overlapping community detection directly. To adopt Q to overlapping community detection problem, Shen modified the traditional modularity we mentioned above as follow.

$$Q_{OL} = \frac{1}{2m} \sum_{k=1}^{c} \sum_{i,j \in C_k} \frac{1}{O_i O_j} (A_{ij} - \frac{k_i k_j}{2m}) \tag{2}$$

where m is the total number of links in the network, k_i and k_j are the degrees of nodes i and j respectively, A_{ij} are the terms of the adjacency matrix of the network. O_i and O_j are the numbers of communities to which nodes i and j belong respectively.

The two objective functions chosen in this paper are described above. Besides, we can find out through our experiments that partition density tends to find small communities. On the other hand, the modularity optimization may come across the resolution limit problem[25]. From this problem, we can find that modularity can lead the optimization algorithms to large community partitions.

Algorithm 1. Framework of MOOCD

Require:
The set of the internal population, ip_{size};
The set of the external population, ep_{size};
The probability of mutation, p_m;
The probability of crossover, p_c;
The running generation, $gens$;
Ensure:
The final population, P;
$P_{in} = \phi$, $P_{ex} = \phi$
for each i in 1 to ep_{size} do **do**
  g_i = generate_gene()
  calculate_functions(g_i)
  $P_{ex} = P_{ex} \cup \{g_i\}$
end for
for each t in 1 to $gens$ **do**
  $P_{in} = \phi$, $i = 0$, $i = \frac{ip_{size}}{2}$
  in_select(P_{ex}, P_{in}, i)
  while $i < ip_{size}$ **do**
    randomly select two individuals (g_j and g_k) from P_{in}
    generate random value $r \in [0, 1]$
    if $r < p_c$ **then**
      g'_j, g'_k =crossover(g_j, g_k)
    else
      g'_j = mutate(g_j)
      g'_k = mutate(g_k)
    end if
    $i = i + 2$
    calculate_functions(g'_j);$P_{in} = P_{in} \cup \{g'_j\}$
    calculate_functions(g'_k);$P_{in} = P_{in} \cup \{g'_k\}$
  end while
  ex_select($P_{ex}, P_{in}, ip_{size}, ep_{size}$)
end for
P = model_selection(P_{ex})
return P
generate() //initialize individual i according to the genetic representation.
calculate_function(g_i) //evaluate the objective functions of g_i.
ex_select($P_{ex}, P_{in}, ip_{size}, ep_{size}$) //update EP(maximum size is epsize).
crossover(g_i, g_j), *mutate*(g_i) //crossover and mutation genetic operator

These findings reflect the intrinsic conflict between the two. And the experiments in section 4 shows that the algorithm using these functions can find community partitions with different characters.

3.3 Genetic Representation and Operators

In this section, we describe two parts of the algorithm which are encoding and decoding as well as mutation and crossover in detail.

Genetic Encoding and Decoding. To apply genetic algorithm to our problem, we need to transfer the community partitions into some forms which can execute genetic operations. To satisfy the requirement of overlapping community detection, we choose link-based genotype to represent solutions. In this representation method, links are clustered into different partitions. It is possible for nodes that belong to two or more communities. As for the implement of this method, we use the strategy provided by Cai [14].

In this link-based representation, an individual g of the population consists of m genes $\{g_0, g_1, \ldots, g_i, \ldots, g_{m-1}\}$, where $i \in \{0, \ldots, m-1\}$ is the identifier of edges, m is the number of edges, and each g_i can take one of the adjacent edges of edge i. As Fig. 1 shows

Position	0	1	2	3	4	5	6	7	8	9	10
Genotype	1	2	0	4	5	6	7	8	9	10	3

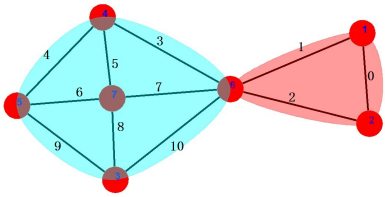


Fig. 1. Genetic Encoding and Decoding

The decoding phase transfers genotype to partition, which consists of link communities. Gene g_i of the genotype and it's value j is interpreted that edge i and edge j have one node in common, and should be classified to same component. In the decoding phase, all components of edges are found, and all edges with in the component constitute a link community.

Genetic Mutation and Crossover. To implement genetic algorithm, we need to confirm some necessary operators such as mutation and crossover. To describe our mutation and crossover strategies, we suppose there are two solutions which are represented through the method above as $g1$ and $g2$. In the crossover operation, we randomly generate a value i. And then, the two genotypes exchange their genes whose positions are i. As for the mutation operation, one random value j is generated. And the jth gene of a certain genotype g is replaced by another value we generate randomly. Fig. 2 show these operations in detail.

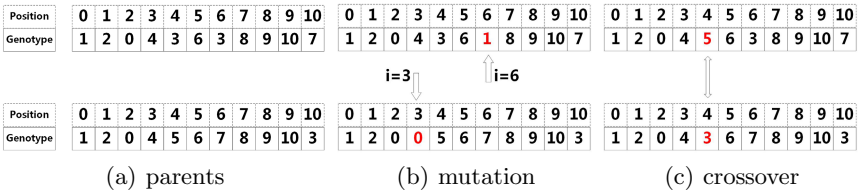


Fig. 2. Genetic Mutation and Crossover

3.4 Model Selection

When the genetic iterations finish, MOOCD returns its external population which is a set of Pareto optimal solutions. And it's time for the decision makers to choose one solution from them. However, we provide an automated strategy to select a more reasonable result. There are many methods that can identify one promising solution in the candidate set. And the principle of some of these model selection methods is to make use of new objective functions to find out a proper solution. In this paper, we use another strategy called Max-Min Distance strategy. The principle of this method is to find a solution which deviates from the random solutions most. And the concrete procedure of it is as follows.

Before the procedure, the method executes MOOCD on some random networks with the same scale. The Pareto front of their solutions are called random Pareto front compared with the real Pareto front.

Firstly, the distance between a solution in the real Pareto front and one in the random Pareto front is defined in

$$dis(C, C') = \sqrt{((intra(C) - intra(C'))^2 + ((intra(C) - intra(C'))^2)}$$

where C and C' represent the solutions in the real and random Pareto fronts, respectively. Then the deviation of a solution in the real Pareto front from the whole random Pareto front is quantified by the minimum distance between this solution and any solutions in the random Pareto front. the deviation is defined in

$$dev(C, CF) = \min\{dis(C, C')|C' \in CF\}$$

where CF represents the random Pareto front. Finally we select the solution in the real Pareto front with the maximum deviation. The model selection process is formulated in

$$S_{max-min} = \arg\max_{C \in SF}\{dev(C, CF)\}$$

where SF represents the real Pareto front.

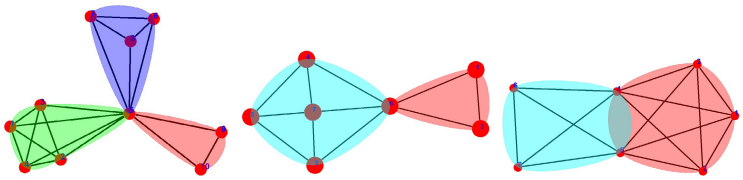
In this section, we provided the framework of MOOCD and described some crucial aspects of our algorithms. And we will evaluate the effectiveness of this method based on artificial networks as well as real networks in the next section. Besides, some other overlapping community detection algorithms are chosen to compare with MOOCD in the experiments.

4 Experiments

This section will validate the effectiveness of MOOCD through experiments on artificial and real networks. The artificial netwrk experiments will illustrate the advantages of multiple solutions returned by MOOCD, and the real network experiments will validate the quality of the solution provided by the model selection method. The experiments are carried out on a 2.2GHz and 2G RAM computer running Windows 7.

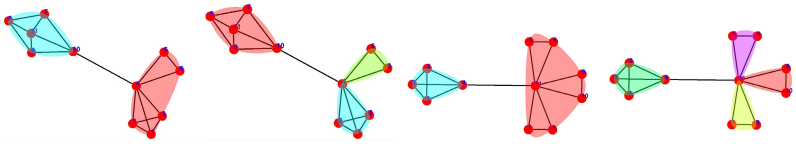
4.1 Experiments on Artificial Networks

To explore the character and ability of MOOCD, we create 5 small typical artificial networks. In the artificial network experiment, we won't use the model selection methods like max-min distance method to choose a single result. Rather, we will represent all the candidate results in Pareto set provided by MOOCD. And these results are shown in Fig. 3 and Fig.4.



(a) N1 (b) N2 (c) N3

Fig. 3. The Community Partition Results of the Artificial Network N1, N2 and N3



(a) N4-1 (b) N4-2 (c) N5-1 (d) N5-2

Fig. 4. The Community Partition Results of the Artificial Network N4 and N5

As we can see in the results, for the networks with single correct community partition (N1,N2,N3), MOOCD can find the correct ovelapping community partitions. More importantly, all the candidate results of MOOCD are meaningful for the networks with multiple community partitions(N4,N5). At the meantime, MOOCD can simultaneously find community partitions in different sizes. This matches our purpose to choose the two objective functions. And we can say that

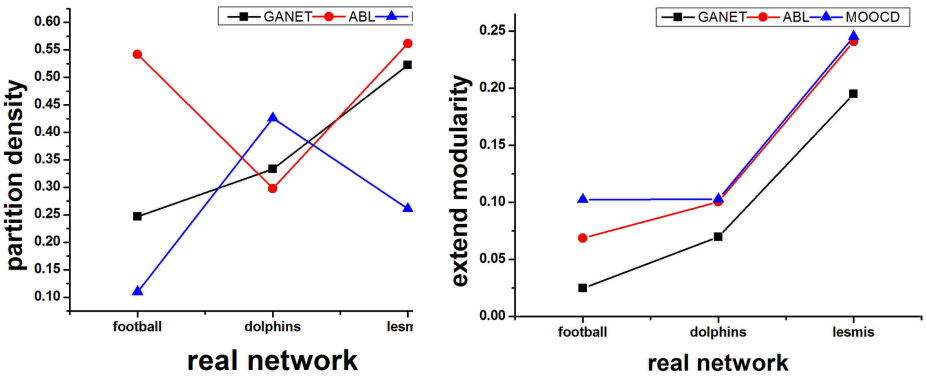
MOOCD has the ability to find different types of communities like overlapping communities and disjoint communities. This conclusion matches our analysis and assumption in Section 3.

4.2 Experiment on Real Network

In this section, we compare MOOCD with ABL and GA_NET on real networks. We execute these 3 algorithm on 3 real networks and calculate D as well as extend Q_{OL} of the partition results. After that, we calculate the number of communities, the size distribution of communities and the average size of communities of the partition results. At last, an intuitive view of partitions on Dolphin found by MOOCD will be posted. Here we choose 3 networks as described in the Table 1 to execute the algorithms. The density and extended modularity are shown in Fig.5.

Table 1. Real Networks Attributes

	dolphins	football	lemis
Nodes	62	115	77
Edges	159	613	254

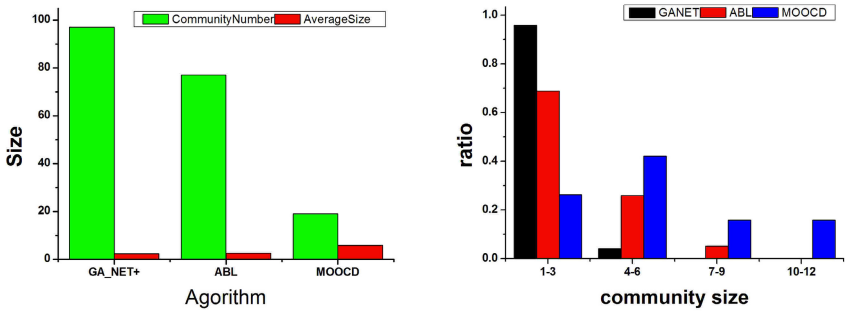


(a) Partition density D (b) Extend modularity Q_{OL}

Fig. 5. The Experiment Result of 3 Methods in Real Networks

The modularity Q_{OL} of our method is much better than that of other methods. Though the partition density of our method is lower than the partition density of other methods in some network, we find a more appropriate community partition through the trade-off between partition density D and extend modularity Q_{OL}. However, these results are not enough to demonstrate the ability and superiority

of MOOCD. Additionally, to evaluate the rationality of the result of MOOCD, in the results of the three methods, we calculate the number of communities, the size distribution of communities and the average size of communities. The results of dolphin network are shown if Fig. 6.



(a) Community Number and Average Size

(b) Community Size Distribution

Fig. 6. Statistic Information for the Partition Results of Dolphin Network

As we can see in this figure, ABL and GA_NET both tend to find small communities and they find too many communities. Noticing that the objective function of GA_NET is community density, we can find community density D can lead the algorithms to find tiny communities. This conclusion demonstrates what we described in Section 3. This doesn't match the real situation. On the other hand, MOOCD can find bigger communities and the size distribution of communities is more balanced.

And then, we show an intuitive view of partition on Dolphin found by MOOCD in Fig.7 and analyze this partition. This figure shows the partitions found by MOOCD in dolphin network. Dolphin network is a social network of frequent associations between 62 dolphins in a community living off Doubtful Sound,

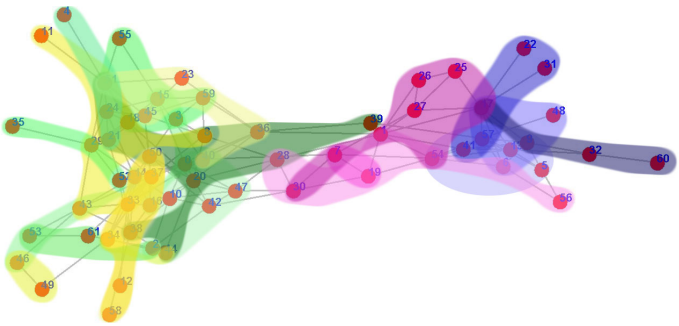


Fig. 7. The Partition of Dolphin Network

New Zealand. The network fell into two parts because of SN100(node 36). In the partition result of MOOCD, node 36 is belonging to many communities. In other word, many communities overlap with each other on the node 36. Removing it makes many communities disjoint with each other, which then splits the networks.

5 Conclusion and Future Work

In this paper, we propose an evolutionary algorithm for multi-objective overlapping community detection. This algorithm uses two classical community partition evaluation functions as objective functions. These objective functions reflect different characters of community structures and make our algorithms have some interesting abilities. The experiments show that our method works well on finding overlapping communities. Besides, MOOCD can simultaneously find different types of community partitions.

In the future, we will try some other interesting objective functions to extend MOOCD. At the meantime, we will apply more than two objective functions to this algorithm. Furthermore, we are going to use the ideas and strategies of this method to solve other problems like dynamic community detection.

Acknowledgement. It is supported by the National Natural Science Foundation of China (No. 61375058, 60905025, 61074128, 71231002). This work is also supported by the National Basic Research Program of China (2013CB329603) and the Fundamental Research Funds for the Central Universities.

References

1. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.U.: Complex networks: Structure and dynamics. Physics Report 424(4-5), 175–308 (2006)
2. Guimera, R., Amaral, L.A.N.: Functional cartography of complex metabolic networks. Nature 433, 895–900 (2005)
3. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physics Review E 69(026113) (2004)
4. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. Proceedings of the National Academy of Sciences 101(9), 2658–2663 (2004)
5. Pothen, A., Sinmon, H., Liou, K.P.: Partitioning sparse matrices with eigenvectors of graphs. SIAM Journal on Matrix Analysis and Applications (11), 430–452 (1990)
6. Kannan, R., Vempala, S., Vetta, A.: On clusterings: good, bad and spectral. Journal of the ACM 51(3), 497–515 (2004)
7. Pizzuti, C.: GA-net: A genetic algorithm for community detection in social networks. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 1081–1090. Springer, Heidelberg (2008)
8. Shi, C., Yan, Z.Y., Wang, Y., Cai, Y.N., Wu, B.: A genetic algorithm for detecting communities in large-scale complex networks. Advance in Complex System 13(1), 3–17 (2010)

9. Tasgin, M., Bingol, H.: Community detection in complex networks using genetic algorithm. arXiv:cond-mat/0604419 (2006)
10. Duch, J., Arenas, A.: Community detection in complex networks using extremal optimization. Physical Review E 72(2), 027104 (2005)
11. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435, 814–818 (2005)
12. Gregory, S.: An Algorithm to Find Overlapping Community Structure in Networks. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 91–102. Springer, Heidelberg (2007)
13. Pizzuti, C.: Overlapping Community Detection in Complex Networks. ACM (2009)
14. Cai, Y., Shi, C., Dong, Y., Ke, Q., Wu, B.: A Novel Genetic Algorithm for Overlapping Community Detection. In: Tang, J., King, I., Chen, L., Wang, J. (eds.) ADMA 2011, Part I. LNCS (LNAI), vol. 7120, pp. 97–108. Springer, Heidelberg (2011)
15. Shen, H., Cheng, X., Cai, K., Hu, M.B.: Detect overlapping and hierarchical community structure in networks. Physica A 388, 1706–1712 (2009)
16. Xiaohua, W., Licheng, J., Jianshe, W.: Adjusting from disjoint to overlapping community detection of complex networks. Physica A 388, 5045–5056 (2009)
17. Maoguo, G., Lijia, M., Qingfu, Z., Licheng, J.: Community detection in networks by using multiobjective evolutionary algorithm with decomposition. Physica A 391, 4050–4060 (2012)
18. Maoguo, G.: Complex Network Clustering by Multiobjective Discrete Particle Swarm Optimization Based on Decomposition. IEEE Transactions on Evolutionary Computation (2013), doi:10.1109/TEVC.2013.2260862
19. Shi, C., Yan, Z., Cai, Y., Wu, B.: Multi-objective community detection in complex networks. Applied Soft Computing 12(2), 850–859 (2012)
20. Ahn, Y.Y., Bagrow, J.P., Lehmann, S.: Link communities reveal multiscale complexity in networks. Nature 466, 761–764 (2010)
21. Palla, G., Dernyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community strcture of complex networks in nature and society. Nature 435, 814 (2005)
22. Corne, D.W., Jerram, N.R., Knwoles, J.D., Oates, M.J.: PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001) (2001)
23. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. PNAS 101(9), 2658–2663 (2004)
24. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Physical Review E 70(6), 06611 (2004)
25. Fortunato, S., Barthelemy, M.: Resolution limit in community detection. Proceedings of the National Academy of Sciences 104(1), 36–41 (2007)