Community Detection Using Cooperative Co-evolutionary Differential Evolution

Qiang Huang[1], Guanbo Jia[2], Thomas White[2], Mirco Musolesi[2], Nil Turan[3], Ke Tang[4], Shan He[2,3*], John K. Heath[3], and Xin Yao[2,4]

[1] School of Software, Sun Yat-sen University, Guangzhou, China
[2] Cercia, School of Computer Science
[3] Center for Systems Biology, School of Biological Sciences, The University of Birmingham, Birmingham, B15 2TT, UK
[4] Nature Inspired Computation and Application Laboratory (NICAL), Department of Computer Science, University of Science and Technology of China, Hefei, Anhui 230027, China

Abstract. In many scientific fields, from biology to sociology, community detection in complex networks has become increasingly important. This paper, for the first time, introduces Cooperative Co-evolution framework for detecting communities in complex networks. A Bias Grouping scheme is proposed to dynamically decompose a complex network into smaller subnetworks to handle large-scale networks. We adopted Differential Evolution (DE) to optimize network modularity to search for an optimal partition of a network. We also designed a novel mutation operator specifically for community detection. The resulting algorithm, Cooperative Co-evolutionary DE based Community Detection (CCDECD) is evaluated on 5 small to large scale real-world social and biological networks. Experimental results show that CCDECD has very competitive performance compared with other state-of-the-art community detection algorithms.

1 Introduction

Many complex systems, such as social [11] and biological networks [3], can be naturally represented as complex networks. A complex network consists of nodes (or vertices) and edges (or links) which respectively represent the individual members and their relationships in systems. By representing complex systems as complex networks, many theories and methods in graph theory can be applied to enable us to gain insights into complex systems. Therefore, in recent years, the study of complex networks has attracted more and more attention.

Unlike simple networks such as lattices or random graphes, complex networks possess many distinctive properties, of which community structure [1] is one of the most studied. The community structure is usually considered as the division of networks into subsets of vertices within which intra-connections are dense, while between which inter-connections are sparse [1]. The identification of the community structure provides

* Correspondence and requests for materials should be addressed to Dr. S. He (email: s.he@cs.bham.ac.uk)

important information about the relationship and interaction among nodes in the complex network. Such information ultimately leads to insights into how network function and topology affect each other.

In the past few years, many algorithms have been proposed to detect the underlying community structure in complex networks [1]. These algorithms can roughly be grouped as traditional methods, such as graph partitioning, spectral methods, modularity maximization methods, and methods based on statistical inference. Among them, the most popular group is *modularity maximization* methods, because of its superior performance on real-world complex networks. For modularity maximization methods, many deterministic optimization algorithms such as greedy algorithms have been employed [1]. However, according to [4], we should also treat results from deterministic algorithms such as greedy optimization or spectral methods with "particular caution" because they only return one unique solution, which might "obscure the magnitude of the degeneracy problem and the wide range alternative solutions".

To address the above problem, we previously proposed a stochastic network community detection algorithm, Differential Evolution based Community Detection (DECD) [5], in which Differential Evolution (DE) was used to evolve a population of potential solutions for network partitions, to maximize the network modularity [8]. The results show that DECD can achieve competitive community detection results on several benchmark and real-world complex networks. However, our further investigation showed that DECD is not satisfactory on large-scale networks.

In order to achieve better scalability to handle large-scale networks, this paper proposed CCDECD (Cooperative Co-evolutionary Differential Evolution based Community Detection) by incorporating a Cooperative Co-evolution (CC) framework into our DECD algorithm. To the best of our knowledge, this is the first time CC framework has been introduced for community detection. A CC framework employs a divide and conquer strategy, which divides a large-scale problem into subcomponents and evolves those subcomponents independently and co-adaptively. Compared with traditional Evolutionary Computation, the advantages of a CC framework are: 1) it is capable of handling large scale optimization problems; and 2) it can better deal with problems with complex structure. Such a framework is very natural and attractive to community detection because of two distinctive properties of complex networks: 1) large scale, e.g., consists of thousands or even millions of nodes; and 2) highly structured, e.g., hierarchical.

Apart from introducing CC framework for community detection, the other main contributions of this paper include: 1) a Bias Grouping scheme to dynamically decompose the complex network into smaller subcomponents; 2) a novel mutation operator called global network mutation specifically designed for community detection; and 3) a thorough evaluation of the performance of CCDECD on several real-world networks, including a large scale network which consists of 6927 nodes.

The remainder of this paper is organized as follows. Section 2 introduces the details of CCDECD. In Section 3, the performance of CCDECD is tested on biological and real-world social networks and then the experimental results are discussed. Finally, Section 4 concludes this paper.

2 The proposed algorithm

In this paper, a new algorithm based on CCDE called CCDECD is proposed for community detection in complex networks. Similar to the random grouping framework in [15], the main idea behind our CCDECD is also to split a large network into m s-dimensional subcomponents, and then evolve each of them with standard DE. However, we found that the random grouping scheme used in [15] is not suitable for a complex network community detection problem because it will lose connectivity information of the network, which is crucial for the search performance of DE on modularity. Therefore, we introduce a novel bias grouping scheme which utilizes the connectivity information. The key steps of our CCDECD can be summarized as follows:

Step 1) Set $g = 0$ where g denotes the generation number.

Step 2) Randomly initialize population P_g.

Step 3) $g = g + 1$

Step 4) Split the n-dimensional complex network into m sub-components G_i ($i = 1, \ldots, m$), where G_i consists of s indices of nodes ($n = m \times s$) using bias grouping scheme (See Section 2.4 for details).

Step 5) Set $i = 1$.

Step 6) Construct subpopulation SP_i for G_i by extracting s genes as defined by G_i from P.

Step 7) For subpopulation SP_i, optimize the network division using a standard DE by maximizing network modularity of G_i with g_s generations (See Section 2.2 for details).

Step 8) Select the best individual SI_{best} from SP_i.

Step 9 Update population P_g by replacing the s genes as defined by G_i with SI_{best}.

Step 10 $g = g + g_s$

Step 11) If $i < m$ then $i + +$, and go to **Step 4**.

Step 12) Optimize the network division of the whole network represented by P_g using a modified DE with the global network mutation operator for g_g generations (See Section 2.5 for details).

Step 13 $g = g + g_g$

Step 14) Stop if $g > g_{\max}$ where $g_{\max}$ is the maximum number of generations and output the best individual I_{best}; otherwise go to **Step 4**.

2.1 Individual representation

CCDECD uses the community identifier-based representation proposed in [14] to represent individuals in the population for the community detection problem. For a graph $G = (V, E)$ with n nodes modelling a network, the kth individual in the population is a vector that consists of n genes $\boldsymbol{x}_k = \{x_1, x_2, \ldots, x_n\}$ in which each gene x_i can be assigned an allele value j in the range $\{1, 2, \ldots, n\}$. The gene and allele represent the node and the community identifier (commID) of communities in G respectively. Thus, $x_i = j$ denotes that the node i belongs to the community whose commID is j, and nodes i and d belong to the same community if $x_i = x_d$. Since at most n communities exist in G and then the maximum value of commID is n.

2.2 Fitness function

Newman and Girvan [8] proposed the network modularity to measure the strength of the community structure found by algorithms. The network modularity is a very efficient quality metric for estimating the partitioning of a network into communities and has been used by many community detection algorithms recently [1, 14, 7].

CCDECD also employs the network modularity which is maximized as the fitness function to evaluate individuals in the population. The network modularity is defined as follows [14].

$$Q = \sum_{j=1}^{m} \left[\frac{l_j}{L} - \left(\frac{d_j}{2L} \right)^2 \right], \tag{1}$$

where j is the commID, m is the the total number of communities, l_j is the number of links in module j, L is the total number of edges in the network and d_j is the degree of all nodes in module j.

2.3 Initialization

At the beginning of the initialization process, CCDECD places each node into a random community by assigning a random commID and generates individuals in the initial population. However, such random generation of individuals is likely to cause some unfavorable individuals that consist of some nodes having no connectivity with each other in the original graph. Considering that nodes in the same community should connect with each other and in the simple case are neighbors, the initialization process proposed in [14] is used to overcome the above drawbacks. The process works as follows: once an individual is generated, some nodes in an individual are randomly selected and their commIDs are assigned to all of their neighbors. By this process, the space of the possible solutions is restricted and the convergence of CCDECD is improved.

2.4 Bias Grouping scheme

Similar to the random group scheme proposed in [15], we proposed a bias grouping scheme for handling large scale networks. The idea behind this bias grouping scheme is to dynamically decompose the whole networks into smaller subcomponents which each consist of nodes that are more likely connected to each other. Therefore, the search algorithm can optimize these tightly interacting variables together, which will ultimately lead to better results than splitting variables into subcomponents with unconnected nodes. The bias grouping scheme works as follows: we randomly select s nodes in the network, where s is the size of a subcomponent. Then we find all the first neighbors of the s nodes and concatenate them to form a set **G**. Finally, we select the first s nodes from **G** to form a subcomponent. If all the s nodes have no first neighbors, the s nodes will be selected to form a subcomponent.

2.5 Mutation

There are two different mutation operators in our CCDECD. For the standard DE used in Step 3 for optimizing the division of subcomponents, the most popular "rand/1" mutation strategy is used [6] since it has no bias to any special search directions.

In Step 5, in order to optimize the division of the global network, we design a novel global network mutation operator: for each population, we randomly select one node i and find all its neighbors. For each node in its neighbors, we randomly assign a probability in the range $[0, 1]$. If the probability of nodes is larger than the mutation rate we predefined, their commIDs will be mutated to the commID of the selected node i. Otherwise, nothing will be changed. This mutation can make use of connectivity information of the network, and thus improve the search ability.

2.6 Clean-up step

CCDECD also adopts the clean-up operation proposed by Tasgin and Bingol [14] to correct the mistakes of putting nodes into wrong communities in both mutant and trial vectors and improves the search ability. The clean-up operation is based on the community variance $CV(i)$, which is defined as the fraction of the number of different communities among the node i and its neighbors to the degree of the node i as follows:

$$CV(i) = \frac{\sum_{(i,j)\in E} \mathrm{neq}(i,j)}{\deg(i)}, \tag{2}$$

where $\mathrm{neq}(i,j) = \begin{cases} 1, & \text{if } \mathrm{commID}(i) \neq \mathrm{commID}(j) \\ 0, & \text{otherwise} \end{cases}$, $\deg(i)$ is the degree of the ith node, E is the set of edges, and commID is the community containing ith node.

The clean-up step works as follows: Firstly some nodes are randomly selected. Then for each of these nodes i, $CV(i)$ is computed and compared with a threshold which is a predefined constant obtained by experience. If $CV(i)$ is larger than the threshold, the community ID of this node will be assigned to the one which is the most common community ID among the neighbors. Otherwise, no operation is performed on this node.

3 Experiments and results

In this section, the performance of CCDECD is evaluated on 4 well known real-world social and biological networks. CCDECD is implemented in MATLAB 7.0 and all the experiments are performed on Windows XP SP2 with a Pentium Dual-Core 2.5GHz processor and 2.0GB RAM. The parameters in CCDECD are set as follows: the population size is 30; the maximum number of cycles is $c_{\max} = 100$ and $m = 30$; the mutation rate for the global network mutation operator is set to be 0.2; for the standard DE, the maximum of generations was 30 and for the "rand/1" mutation operator, the scaling factor is $F = 0.9$ and the threshold value is $\eta = 0.32$. The threshold for clean step is set to be 0.35 as used in [14].

For comparison, we implement DECD and another community detection algorithm based on a Genetic Algorithm (GA), named GACD. We adopt the MATLAB Genetic

Algorithm Optimization Toolbox (GAOT) to optimize the network modularity to detect communities in networks. The GA we use is real encoded GA with heuristic crossover and uniform mutation. The values of all the parameters use in the experiments are the default parameters in GAOT. Moreover, for the sake of fairness, the same initialization process and the clean-up operation in CCDECD are employed in the DECD and GACD algorithms. The number of function evaluations of DECD and GACD is set to be the same as CCDECD. We also adopt MATLAB implementations of Girvan-Newman (GN) algorithm [7] from Matlab Tools for Network Analysis (http://www.mit.edu/˜gerganaa) for comparison.

3.1 Datasets

In this paper, we selected the following 5 well known real-world social and biological networks to further verify the performance of CCDECD: 1) the Zachary's Karate Club network; 2) Dolphins network; 3) the American College Football network; 4) Protein and Protein Interaction (PPI) network and 5) Erdös collaboration network.

We selected the above 5 datasets because for small to medium scale datasets 1) to 4), their true community structures are known, which provide gold-standards, e.g., normalized mutual information, for the evaluation of our CCDECD algorithm. We also selected the Erdös collaboration network which is the largest network tested in [9]. The characteristics of the five networks are summarized in Table 1.

Table 1. The characteristics of the five networks tested in the paper. N and M stand for nodes and edges of the network, respectively. Q_{opt} is the known global optimal modularity value.

Dataset	N	M	Q_{opt}
Karate	34	78	0.41979
Dolphins	62	159	0.52852
Football	115	613	0.60457
PPI	1430	6531	–
Erdös	6927	11850	–

3.2 Small real-world social networks

We first validate our algorithm on the small-scale real-wold social networks with true community structure: 1) the Zachary's Karate Club network; 2) Dolphins network; and 3) the American College Football network. As pointed out in [13], performance metrics based on network modularity Q are not always reliable. Therefore, apart from Q, we also adopt normalized mutual information (NMI) as proposed in [2] for performance evaluation.

Since CCDECD, DECD and GACD are stochastic optimization algorithms, we perform the experiments 30 times on these three networks. The average values of Q and

NMI, e.g., Q_{avg} and NMI_{avg} and their best values, e.g., Q_{bst} and NMI_{bst}, are compared with that obtained by GN (a deterministic algorithm) from one run of an experiment. We also perform two sample student's t-test between the results obtained from CCDECD and those from other algorithms. The results are presented in Table 2

Table 2. Experimental results of the Zachary's Karate Club, Dolphins and the American College Football networks. N_{pr} is the average predicted number of communities; Q_{avg} and NMI_{avg} are the average values of modularity Q and NMI, respectively. Q_{bst} and NMI_{bst} are the best values of modularity Q and NMI, respectively. The results with asterisks indicate the results are significantly difference from the results obtained from CCDECD.

Network	Algorithm	N_{pr}	Q_{avg}	Q_{bst}	NMI_{avg}	NMI_{bst}
Karate	CCDECD	4.0 ± 0.0	$\mathbf{0.41979 \pm 0.00000}$	$\mathbf{0.41979}$	0.69 ± 0.00	0.69
	DECD	4.1 ± 0.3	$0.41341 \pm 0.00446^*$	$\mathbf{0.41979}$	$0.65 \pm 0.06^*$	0.71
	GACD	3.3 ± 0.9	$0.39552 \pm 0.01492^*$	0.41724	0.69 ± 0.10	$\mathbf{0.84}$
	GN	2	0.35996	0.35996	$\mathbf{0.84}$	0.84
Dolphins	CCDECD	4.1 ± 0.3	$\mathbf{0.52078 \pm 0.00026}$	$\mathbf{0.52162}$	0.80 ± 0.04	0.93
	DECD	4.7 ± 0.8	$0.51557 \pm 0.00374^*$	0.52069	$0.83 \pm 0.05^*$	0.95
	GACD	4.9 ± 0.8	$0.50987 \pm 0.01499^*$	0.51986	$\mathbf{0.87 \pm 0.07}^*$	$\mathbf{1.00}$
	GN	4	0.50823	0.50823	0.84	0.84
Football	CCDECD	10.1 ± 0.7	$\mathbf{0.60382 \pm 0.00089}$	$\mathbf{0.60457}$	0.89 ± 0.02	$\mathbf{0.93}$
	DECD	10.1 ± 0.8	0.60363 ± 0.00071	$\mathbf{0.60457}$	0.90 ± 0.02	0.92
	GACD	8.7 ± 1.4	$0.59044 \pm 0.01239^*$	$\mathbf{0.60457}$	0.85 ± 0.05	0.93
	GN	12	0.59726	0.59726	$\mathbf{0.93}$	0.93

From Table 2, it can be seen that CCDECD performed better than the other three competitors, i.e., DECD, GACD and GN on the three networks. In [9], the author proposed a novel multi-objective genetic algorithm (MOGA-Net) for community detection. The objective is not to maximize modularity but to maximizes the number of connections inside each community and minimizes the number of links between the modules. The average best Q values obtained by MOGA-Net are 0.416, 0.505 and 0.515 for Karate, Dolphin and Football networks, respectively; and the corresponding average NMI values are 0.602, 0.506 and 0.775. The best NMI obtained by MOGA-Net on the Football network is 0.795, even worse than NMI_{avg} obtained by CCDECD. Such results show that maximizing Q with our CCDECD can also achieve better NMI, a gold standard for evaluating CD algorithms, than MOGA-Net.

3.3 Biological network: Yeast Protein-Protein Interaction Network

We apply our CCDECD algorithm to a biological network, e.g., Yeast Protein-Protein Interaction (PPI) Network [3], which contains 1430 nodes (proteins) and 6531 edges (interactions). We use CYC2008 [10], a complete and up-to-date set of yeast protein complexes (or communities) as a reference set to evaluate the predicted modules by CCDECD. We compute precision, recall and F-measure to measure the performance of

CCDECD. The performance of CCDECD is compared with DECD, GACD and GN. We also adopt results from recent literature, e.g., [12] for comparison.

Similar to the experiments in [12], we use the affinity score to decide whether a predicted module is matched with a reference complex:

$$\text{Affinity}(A, B) = \frac{|A \bigcup B|^2}{|A| \times |B|}, \tag{3}$$

where A and B are two modules of proteins, e.g., one of predicted module or reference complexes. We assume a module A matches module B if and only if $\text{Affinity}(A, B)$ is above a predefined threshold ω. Then we can define $Hit(\mathcal{A}, \mathcal{B})$ which contains all the matched modules:

$$Hit(\mathcal{A}, \mathcal{B}) = \{A_i \in \mathcal{A} | \text{Affinity}(A_i, B_j) > \omega, \exists B_j \in \mathcal{B}\}. \tag{4}$$

We define precision, recall and F-measure as follows:

$$\text{Recall} = \frac{|Hit(\mathcal{R}, \mathcal{P})|}{|\mathcal{R}|}, \tag{5}$$

$$\text{Percision} = \frac{|Hit(\mathcal{P}, \mathcal{R})|}{|\mathcal{P}|}, \tag{6}$$

$$\text{F-measure} = \frac{2 \times \text{Recall} \times \text{Percision}}{\text{Recall} + \text{Percision}}, \tag{7}$$

where $\mathcal{P}$ is the predicted module set and $\mathcal{R}$ is the reference complex set.

Following the experimental settings in [12], we set $\omega = 0.4$ and 0.5 and select the best results from 30 runs of experiments in order to compare with their algorithms fairly. We compared the results from Critical Module (CM) algorithm proposed in [12]. It is worth mentioning that, due to the large size of the PPI network, the GN algorithm in the Matlab Tools for Network Analysis failed to produce results in reasonable time. Therefore, we adopt the results of the GN algorithm from [12] for comparison.

Table 3. The best results from 30 runs of experiments of the Yeast Protein-Protein Interaction Network.

ω	Algorithm	#. pred. complex	Precision	Recall	F-measure
0.4	CCDECD	108	0.5093	**0.3**	**0.3776**
	DECD	143	0.5083	0.2927	0.3715
	GACD	109	0.5046	0.2902	0.3685
	CM	65	**0.5745**	0.0667	0.1195
	GN	65	0.383	0.042	0.0757
0.5	CCDECD	94	0.4681	**0.2683**	**0.3411**
	DECD	115	0.4696	0.2390	0.3168
	GACD	106	0.4340	0.2220	0.2937
	CM	65	**0.6154**	0.0691	0.1241
	GN	65	0.5231	0.0568	0.1025

From Table 3, we can see that compared with other algorithms, CCDECD has better performance. It is interesting to see that, the difference of performance among CCDECD, DECD and GACD is not as significant as those between CCDECD and other non-population-based algorithms, e.g., CM. Such results indicate that, at least for medium size networks, which are commonly seen in biology, population-based algorithms are preferred because of their better search performance.

3.4 Large-scale network: Erdös collaboration network

In this section, we further evaluate the performance of CCDECD with a large-scale network: Erdös collaboration network. We report the results, e.g., average number of communities and average values of modularity obtained by our CCDECD in comparison with those of DECD, GACD, GN and MOGA-Net [9] in Table 4.

Table 4. Experimental results of Erdös collaboration network. N_{pr} is the average number of communities; Q_{avg} are the average values of modularity Q. The results with asterisks indicate the results are significantly difference from the results obtained by CCDECD.

Algorithm	N_{pr}	Q_{avg}
CCDECD	194.8 ± 17.89	0.6390 ± 0.0042
DECD	407.5 ± 44.92	$0.5598 \pm 0.0095^*$
GACD	277.4 ± 22.47	$0.6070 \pm 0.0108^*$
MOGA-Net	302	0.5502
GN	57	0.6723

Table 4 clearly show that in terms of Q_{avg}, CCDECD performed much better than the other three population-based algorithms. More specifically, in contrast to the results on small scale networks presented in Section 3.2, the performance of CCDECD in terms of Q_{avg} is much better than DECD and GACD, which indicates that CCDECD is more scalable to handle large-scale networks. However, we should notice that, compared with the greedy based GN algorithm, the results of our CCDECD is still not competitive.

4 Conclusion

This paper, for the first time, introduced the Cooperative Co-evolutionary algorithm to detect community structure in complex networks. We have proposed the Bias Grouping scheme to dynamically decompose the complex network into smaller subcomponents for independent and co-adaptive evolution. We have also designed the global network mutation operator specifically for community detection problems which exploits the network connectivity information. We have tested our CCDECD on several benchmark real-world social and biological networks, including the Erdös collaboration network which consists of 6927 nodes, in comparison with DECD, GACD, GN and MOGA-Net algorithms. Apart from the modularity value, for the small scale real-world networks, we have also employed NMI based on true community structure as the performance metric [13]. Compared with other state-of-the-art EACD algorithms, the experimental results have demonstrated that CCDECD is very effective for community detection in complex networks. Compared with greedy based CD algorithms, e.g., GN algorithm, our CCDECD can generate more accurate results on small to medium scale networks. However, although it is a step forward, it is still not competitive to handle large-scale network. It will be our future work to incorporate local search algorithm into our CC framework to further improve CCDECD's scalability.

Acknowledgment

Mr Qiang Huang and Dr Shan He would like to thank Paul and Yuanbi Ramsay for their financial support. Mr Thomas White and Dr Shan He are supported by EPSRC (EP/J01446X/1). Dr Ke Tang, Dr Shan He and Professor Xin Yao are supported by an EU FP7-PEOPLE-2009-IRSES project under Nature Inspired Computation and its Applications (NICaiA) (247619).

References

1. A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70:066111, 2004.
2. L. Danon, A. D. Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *J. Stat. Mech.*, 2005.
3. A. C. Gavin and *et. al.* Proteome survey reveals modularity of the yeast cell machinery. *Nature*, 440:631–636, 2006.
4. B. H. Good, Y. Montjoye, and A. Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81:046106, 2010.
5. G. Jia, Z. Cai, M. Musolesi, Y. Wang, D.A. Tennant, R. Weber, J.K. Heath, and S. He. Community detection in social and biological networks using differential evolution. In *Learing and Intelligent OptimizatioN Conference*, 2012.
6. F. Neri and V. Tirronen. Recent advances in differential evolution: a survey and experimental analysis. *Artificial Intelligence Review*, 33:61–106, 2010.
7. M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69:026113, 2004.
8. M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.
9. C. Pizzuti. A multiobjective genetic algorithm to find communities in complex network. *IEEE Transactions on Evolutionary Computation*, 2011.
10. S. Pu, J. Wong, B. Turner, E. Cho, and S. J. Wodak. Up-to-date catalogues of yeast protein complexes. *Nucleic Acids Res.*, 37:825–831, 2009.
11. J. Scott. *Social network analysis: A Handbook.* Sage Publications, London, 2000.
12. N. Sohaee and C. V. Forst. Modular clustering of protein-protein interaction networks. In *Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2010 IEEE Symposium on*, 2010.
13. K. Steinhaeuser and N. V. Chawla. Identifying and evaluating community structure in complex networks. *Pattern Recognition Letters*, 31:413–421, 2009.
14. M. Tasgin and H. Bingol. Community detection in complex networks using genetic algorithm. In *Proceedings of the European Conference on Complex Systems*, 2006.
15. Z. Yang, K. Tang, and X. Yao. Large scale evolutionary optimization using cooperative coevolution. *Information Sciences*, 178:2985–2999, 2008.