ChinaRxiv

AI translation · View original & related papers at
chinarxiv.org/items/chinaxiv-201805.00280

Postprint: Cuckoo Community Detection Algorithm Combining Genetic and Greedy Search

Authors: Wang Xiaogang, Yan Guanghui, Zhou Ning

Date: 2018-05-20T00:00:00+00:00

Abstract

To improve the accuracy of community structure detection in complex networks, a cuckoo community detection algorithm for modularity optimization (GGCSCA) is proposed by combining genetic inheritance and greedy search. The cuckoo population performs dimension-by-dimension random walks on ordered neighbor lists and adopts a high-quality gene inheritance strategy to enable efficient population optimization, while simultaneously applying a greedy preference search algorithm for maximizing local modularity increments to rapidly enhance population quality, thereby obtaining high-quality community partitioning results. GGCSCA was tested on benchmark and classic networks, and compared with several representative algorithms. The results demonstrate the effectiveness, accuracy, and fast convergence of this community detection algorithm, which possesses strong community identification capability and can detect network community structures with high precision.

Full Text

Preamble

Cuckoo Search Algorithm Combining Gene Inheritance and Greedy Search for Community Detection

Wang Xiaogang, Yan Guanghui, Zhou Ning
(School of Electronic & Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China)

Abstract: To improve the accuracy of community structure mining in complex networks, this paper proposes a cuckoo community detection algorithm (GGCSCA) that combines gene inheritance and greedy search for modularity optimization. The cuckoo population performs dimension-by-dimension random walks on ordered neighbor lists and employs a high-quality gene inheritance

strategy for efficient population optimization. Simultaneously, a greedy preference search algorithm that maximizes local modularity increment is applied to rapidly improve population quality and achieve good community partition results. GGCSCA was tested on both benchmark and classical networks and compared with several typical algorithms. The results demonstrate the effectiveness, accuracy, and fast convergence of the proposed community discovery algorithm, which exhibits strong community identification capability and can finely detect network community structures.

Keywords: complex network; network community; cuckoo search algorithm; greedy search; gene inheritance

0 Introduction

Complex networks abstract and generalize important features of complex systems. The core idea is to transform all entities and their relationships in real systems into network nodes and edges, describing the relationships between system components in network form to facilitate in-depth analysis of topological characteristics and reveal the essential laws of real systems. Many real-world systems can be represented as or converted into complex networks, such as social networks, biological networks, computer networks, and transportation networks. Complex networks have become an important interdisciplinary research field.

Complex networks consist of several communities where internal connections are relatively dense while connections between communities are relatively sparse. Community discovery detects inherent community structures in complex networks and is currently a research hotspot in the field. Community discovery helps analyze complex network functions, intrinsic laws, and topological structure characteristics, providing a mesoscopic perspective for complex network evolution research. Its findings have been successfully applied to protein function prediction, terrorist organization identification, public opinion analysis, customer relationship clustering, search engines, and many other areas.

Complex network community mining has been extensively studied, yielding numerous community detection algorithms. Typical algorithms mainly include graph decomposition and division-based methods such as the Kernighan-Lin algorithm, spectral partitioning, and the GN algorithm; agglomerative methods such as the Fast Newman algorithm and CNM algorithm; random walk-based methods such as the Walk Trap algorithm; and optimization-based methods that optimize certain community evaluation functions to obtain partition results, most commonly modularity-based optimization algorithms. Newman and Girvan defined a modularity function to evaluate community partitions. Modularity optimization algorithms include simulated annealing, BGLL algorithm, and extremal optimization. Although many community detection algorithms exist, improvements are still needed in community identification accuracy, efficiency, usability, and even universality.

1 Related Work

In recent years, community detection using intelligent evolutionary algorithms has become a research hotspot. By selecting appropriate community partition evaluation functions, the complex network community discovery problem can be transformed into an optimization problem, though achieving optimal objectives is often NP-hard. Using intelligent evolutionary algorithms with appropriate heuristic rules can yield good approximate optimization results.

Intelligent evolutionary community discovery algorithms are mainly divided into multi-objective optimization and single-objective optimization. Multi-objective optimization methods include MOCD-PSO and MDCL. Single-objective methods generally optimize modularity, iteratively evolving to maximize modularity. The earliest intelligent evolutionary algorithm applied to community detection was simulated annealing, with Guimera et al. proposing a complex network community detection algorithm based on simulated annealing that optimizes modularity, which was reported in *Nature* in 2005.

Huang Faliang proposed a particle swarm optimization algorithm for network community discovery (CDPSO), which performs global search based on encoding of ordered node neighbor lists, alleviating the local optimal partition problem caused by binary encoding-based iterative bisection strategies. This node neighbor-ordered approach has been adopted in many evolutionary community detection algorithms. Qiu Xiaohui proposed an improved particle swarm optimization algorithm for social network community discovery, which uses a maximum neighbor affiliation mutation strategy where nodes mutate to the community with the most neighbors with a certain probability. Tasgin et al. designed a single-point crossover operation suitable for string encoding, using a genetic algorithm (GA) to optimize the community modularity Q function for approximate optimal partition. Deng Kun et al. presented a community detection algorithm based on a genetic framework, implementing directed mutation strategies based on node evaluation values to overcome the blindness of random mutation. Jin Di et al. analyzed the local gradient characteristics of the modularity function and combined them with genetic algorithms to propose a fast and effective local search mutation strategy applicable to large-scale network community detection. Gach and Hao' s community detection algorithm combined genetic crossover operators with memetic algorithms, using the BGLL algorithm to generate initial solutions and producing new community results through community crossover from two parent clusters based on priority. Jin Di et al., from a bionics perspective, proposed a community discovery algorithm based on ant colony algorithms and random walks, integrating ant local solutions into global solutions and making community structures emerge by "strengthening intra-cluster connections and weakening inter-cluster connections."

In recent years, several novel evolutionary community detection algorithms have

emerged. Chopade proposed a complex network community discovery method based on game theory, dividing networks into tight communities based on Nash equilibrium. By redefining node similarity for weighted networks, Laplacian matrices, and modularity, the method finds Nash equilibrium points for fitness in evolutionary games. Each node in the network acts as a game player, deciding which community to join to maximize its benefit until no node can increase its benefit, resulting in a community partition. Duan Zhen proposed a multi-layer granulation community discovery method based on quotient space, which performs multi-level granulation operations on complex networks to form multi-granularity quotient spaces with layer-by-layer granulation and abstraction, selecting the optimal granulation layer as the partition result. Jin Zhigang proposed an adaptive community discovery algorithm based on density peak clustering (KDED), which quantifies node relationships into a distance matrix based on trust, performs kernel density estimation and statistical analysis of node influence based on the distance matrix, and improves the calculation process using a heat diffusion model to adapt to datasets of different scales. According to density peak clustering principles and community attributes, community center nodes are determined, then the internal hierarchical structure of communities and natural structures outside communities are obtained based on distances between nodes, and finally remaining nodes are assigned to corresponding communities to obtain the partition result.

---

2 Cuckoo Search Algorithm

The cuckoo search algorithm is a swarm intelligence search algorithm proposed by Yang Xin-she, characterized by fast search speed and high precision, and has been widely applied to optimization problems in scientific research and engineering practice. The algorithm draws inspiration from the brood parasitism behavior of cuckoos. Each nest represents a candidate solution, and new solutions are searched through Levy flight-based random walks, demonstrating strong global search capability.

For an n-dimensional nest with current solution $X_i^t = \{x_{i1}, x_{i2}, ..., x_{in}\}$, the Levy flight random walk formula for generating a new solution at time t+1 is:

$$X_i^{t+1} = X_i^t + \alpha \oplus \text{Levy}$$

where α is the step size and $\oplus$ denotes entry-wise multiplication. According to Mantegna' s algorithm, the Levy distribution is simulated as:

$$\text{Levy} \sim u = \frac{\phi \times \mu}{|\nu|^{1/\beta}}$$

where μ and ν follow normal distributions: $\mu \sim N(0, \sigma_\mu^2)$ and $\nu \sim N(0, \sigma_\nu^2)$, with $\sigma_\nu = 1$ and

$$\sigma_\mu = \left\{ \frac{\Gamma(1+\beta) \times \sin(\pi\beta/2)}{\Gamma[(1+\beta)/2] \times \beta \times 2^{(\beta-1)/2}} \right\}^{1/\beta}$$

The basic cuckoo algorithm operates as follows: based on the current solution, new solutions are generated through Levy flight random walks, evaluated, and better solutions are retained; then some solutions are discarded with probability p_a and replaced with the same number of new solutions through preference search, which are again evaluated and retained.

This basic cuckoo algorithm is designed for continuous space function optimization and cannot be directly applied to community detection in complex network spaces. This paper leverages the algorithm' s fundamental ideas and framework to design and implement a cuckoo community detection algorithm suitable for discrete complex network spaces.

---

3 Cuckoo Community Detection Algorithm

3.1 Evaluation Function

The evaluation function uses Newman' s modularity function:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(C_i, C_j)$$

where m is the total number of edges in the network, A_{ij} is the adjacency matrix, k_i is the degree of node i, and C_i represents the community to which node i belongs. The term $\delta(C_i, C_j)$ equals 1 when i and j belong to the same community and 0 otherwise. The formula' s meaning is: the ratio of total edges within communities to total edges in the network minus an expected value, which represents the ratio that would exist if the network were random with the same community assignment. This mathematically characterizes community partition quality, with a value range of -0.5 to 1.

Many community discovery algorithms are based on modularity optimization, aiming to maximize modularity. However, optimizing Q is NP-hard, so approximate optimization is used. A limitation of modularity-based methods is the difficulty in detecting small communities below a certain scale, which is related to specific network size. Despite this resolution limit, modularity optimization remains an effective and general method. The modularity function in Equation (5) is for undirected, unweighted networks; corresponding modularity functions exist for directed and weighted networks. This paper focuses on undirected, unweighted networks without loss of generality.

3.2 Information Encoding

Cuckoo nest encoding adopts a node label-based encoding scheme. For a network $G = (V, E)$ with n nodes, a nest solution $X_i = \{x_{i1}, x_{i2}, ..., x_{in}\}$ indicates that nodes i and j are in the same community when $x_{im} = x_{jm}$. The final community partition is obtained through merging. This paper uses ordered neighbor lists as the foundation, with nests performing random walks dimension-by-dimension on the network, meaning the value of x_{ij} is obtained through random walks on node i' s neighbor node sequence, ensuring no illegal solutions are generated. During initialization, x_{ij} randomly takes a node label from node i' s neighbor node sequence.

Figure 1(a) shows a simple network example, Figure 1(b) shows two communities partitioned from this network, and Table 1 shows the network' s neighbor ordered list. The vector in Figure 1(c) represents the cuckoo nest encoding and dimension-by-dimension random walks.

Table 1 Neighbor Ordered List

Node ID	Neighbor Node Sequence
1	2, 3, 4
2	1, 5, 6
3	1, 6, 7
4	1, 5, 7
5	2, 4, 6
6	2, 3, 5
7	3, 4, 6

3.3.1 Dimension-by-Dimension Random Walk on Neighbor Lists and Gene Inheritance

The global search of the cuckoo search algorithm is based on random walks. The random walk designed in this paper is a dimension-by-dimension random walk based on ordered adjacency lists. A nest solution at time t is $X_i^t = \{x_{i1}^t, x_{i2}^t, ..., x_{in}^t\}$, and at the next moment becomes $X_i^{t+1} = \{x_{i1}^{t+1}, x_{i2}^{t+1}, ..., x_{in}^{t+1}\}$, where x_{ij}^{t+1} is obtained by random walks on the adjacency list with probability p, as shown in Figure 1(c).

For node j, the probability $p = 1/d_j$, where d_j is the degree of node j. For each nest X_i, besides using random walks to ensure solution flexibility and globality, each generation must largely evolve from the previous generation to ensure solution efficiency and convergence. Therefore, in addition to the global optimal solution, the top s better nest solutions are preserved in an elite library $\{h_1, h_2, ..., h_s\}$ each generation. Each dimension x_{ij} can be viewed as a gene, and a gene inheritance strategy is introduced: elite genes (h_m) and global optimal solution genes (g_{best}) are retained with probabilities p_c and p_g, respectively. The formula for obtaining x_{ij}^{t+1} in each iteration is:

$$x_{ij}^{t+1} = \begin{cases} h_m(j) & \text{if rand}() < p_c \\ g_{best}(j) & \text{if } p_c \leq \text{rand}() < p_g \\ \text{random walk on adjacency table} & \text{if rand}() \geq p_g \end{cases}$$

where h_m is randomly selected from the elite library, with $m = \lceil \text{random}() \times s \rceil$.

The framework of the cuckoo community search algorithm is given below:

Algorithm 1: Cuckoo Community Search Algorithm

Input: Complex network $G = (V, E)$, nest population size l, preference search probability p_a, iteration count $iter$.

Output: Community partition $\{C_1, C_2, ..., C_k\}$

Begin: 1. Construct neighbor ordered list $AdTable$ from the network; construct elite library $\{h_1, h_2, ..., h_s\}$ of size s with initial empty elements; 2. Define objective function as modularity function Q; 3. Initialize cuckoo nest population, with each nest X_i randomly taking values from neighbor ordered lists in each dimension; calculate objective function values to obtain initial Q values; 4. Update elite library $\{h_1, h_2, ..., h_s\}$; 5. Update nests according to Equation (6): each nest X_i partially inherits excellent genes and partially performs one random walk on the adjacency table to obtain new nest X_i'; 6. Calculate objective function values based on X_i'; if $Q(X_i') > Q(X_i)$, then $X_i \leftarrow X_i'$; 7. Find the maximum Q value in $\{X_i\}$ as the current overall optimal g_{best}; if $Q(g_{best}) > Q(best)$, then $best \leftarrow g_{best}$; 8. Execute local greedy search algorithm (see Algorithm 2) with probability p_a to obtain new nests $\{X_1', X_2', ..., X_l'\}$, then execute steps 6 and 7 sequentially; 9. $t = t + 1$; if iteration count $t < iter$, go to step 4; otherwise go to step 10; 10. Exit iteration, decode the optimal solution $best$ to obtain community partition $\{C_1, C_2, ..., C_k\}$.

3.3.2 Local Preference Search

The local preference search aims to improve population quality and accelerate the search process. This local search method uses a greedy algorithm to calculate the modularity increment ΔQ when node j leaves its current community and joins other adjacent communities, finding the community that maximizes ΔQ as the node' s new community. This mainly involves two steps: (1) the increment ΔQ_1 when node j leaves its original community to become an independent node, see Equation (8); (2) the increment ΔQ_2 when this independent node joins a new community, see Equation (9).

In Equations (8) and (9), $\sum_{in}$ represents the sum of internal connection weights in the community that node j is leaving/joining, $\sum_{tot}$ represents the sum of weights of edges connecting node j to points in the community (including intra-community and inter-community connections), k_j represents the degree of node j, and $k_{j,in}$ represents the sum of edge weights between node j and nodes within the community.

$$\Delta Q_1 = \left[\frac{\sum_{in}+k_{j,in}}{2m} - \left(\frac{\sum_{tot}+k_j}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_j}{2m}\right)^2\right]$$

$$\Delta Q_2 = \left[\frac{\sum_{in}+k_{j,in}}{2m} - \left(\frac{\sum_{tot}+k_j}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_j}{2m}\right)^2\right]$$

The local preference search steps are given below:

Algorithm 2: Local Greedy Search Algorithm

Input: Nest $\{X_1, X_2, ..., X_l\}$, neighbor ordered list $AdTable$, decode and merge each candidate nest X_i to obtain corresponding community partition.

Process: For each nest X_i with probability p_a, determine partial candidate genes ($x_{ij}, j = 1, 2, ..., n$); calculate the modularity increment when node j leaves its community and joins other neighbor communities using Equations (8) and (9); find the community that maximizes ΔQ and move node j to that community; obtain new nest $\{X'_1, X'_2, ..., X'_l\}$.

3.4 Algorithm Time Complexity Analysis

Let n be the number of nodes, m the number of edges, l the number of nests, $iter$ the number of iterations, and d the average degree.

The main steps and their complexities are: (a) constructing neighbor ordered list with time complexity $O(m)$; (b) initializing nests with complexity $O(l \times n)$; (c) dimension-by-dimension random walk selecting a neighbor node or gene retention operation with complexity $O(iter \times l \times n)$; (d) decoding and merging communities with complexity $O(iter \times l \times c)$, where c is the number of communities; (e) calculating modularity with complexity $O(iter \times l \times c^2)$; (f) local greedy search only requires local information, so its complexity is $O(iter \times l \times d)$. The highest complexity is $O(iter \times l \times c^2)$. Generally, $c \ll n$, so the complexity is $O(iter \times l \times n^2)$. Since $iter \times l$ is a constant, the complexity is $O(n^2)$.

---

4 Experiments and Analysis

Numerical experiments were conducted on a computer with an Intel i5 processor, 4GB memory, and Windows 7 operating system, programmed in MATLAB 2011. Parameter settings: in the cuckoo algorithm, population size l was set to 25, preference search probability p_a to 0.25, and elite size s to 4.

p_c and p_g are probability thresholds for inheriting elite genes (h_m) and global optimal solution (g_{best}) genes, respectively. Taking p_c as an example, assuming

the population only inherits optimal genes, the influence of p_g on modularity values was determined through the average of 20 independent runs, as shown in Figure 2 [Figure 2: see original paper]. The four networks in the figure are Karate (karate club network), Dolphin (dolphin network), Football (football league network), and an artificially generated network Lfr500 with 500 nodes. It can be seen that without considering inheritance of other elite genes, the best effect occurs when p_g is 0.1, indicating that inheriting optimal (g_{best}) genes works best while maintaining certain flexibility and randomness. Therefore, according to Equation (6), p_c and p_g should be between 0.1-0.3, and p_c should not exceed p_g. Based on this analysis, in these experiments p_g was set to 0.1 and p_c to 0.2.

4.1 Artificial Networks

The algorithm' s community identification capability was examined using benchmark test networks proposed by Lancichinetti. The network has 128 nodes divided into 4 communities with 32 nodes each, with an average degree of 16. The mixing parameter μ significantly affects community structure when generating networks: it connects external community nodes with probability μ and internal community nodes with probability $1-\mu$. As μ increases from 0.1 to 0.5, community structure changes from clear to fuzzy. When $\mu < 0.5$, a node' s neighbors belong to the same community with higher probability than outside the community, so community structure should be identifiable. When $\mu = 0.5$, each node has on average half of its connections pointing outside its community, making community structure relatively ambiguous.

This experiment uses Normalized Mutual Information (NMI) to evaluate community identification effectiveness, as shown in Equation (10):

$$\mathrm{NMI}(A,B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} C_{ij}\log\left(\frac{C_{ij}\times n}{C_{i.}\times C_{.j}}\right)}{\sum_{i=1}^{C_A} C_{i.}\log\left(\frac{C_{i.}}{n}\right)+\sum_{j=1}^{C_B} C_{.j}\log\left(\frac{C_{.j}}{n}\right)}$$

where A and B are two community partitions, C is the confusion matrix, C_A and C_B are the numbers of communities in A and B, respectively. C_{ij} is the number of overlapping nodes between community i in partition A and community j in partition B, $C_{i.}$ represents the sum of elements in row i, and $C_{.j}$ represents the sum of elements in column j. NMI ranges in [0,1], where 1 indicates identical community partitions and 0 indicates completely different partitions.

Five networks were generated with mixing parameters from 0.1 to 0.5, with each algorithm run independently 50 times to obtain average NMI values. Figure 3 [Figure 3: see original paper] shows that for networks with $\mu = 0.1$ and $\mu = 0.2$, all algorithms can correctly or basically correctly identify communities. At $\mu = 0.3$, results diverge, with LPA and LeadingEigenvector dropping to 0.9 and 0.86, respectively. For the fourth network, only BGLL and GGC-SCA can completely correctly identify communities. For the first four networks,

both BGLL and GGCSCA accurately identify communities, significantly outperforming other algorithms. For the fifth network, GGCSCA achieves NMI = 0.41, better than BGLL' s 0.32. This fully demonstrates the community detection and identification capability of the proposed algorithm.

4.2 Classic Networks

Experiments were conducted on three classic datasets: Karate (karate club network), Dolphin (dolphin network), and Football (football league network).

4.2.1 Convergence As shown in Figure 4 [Figure 4: see original paper], the algorithm converges quickly. The karate club network converges to the global optimal value of 0.4198 in only 2 iterations (one iteration refers to one global search and one local preference search of the population), and the Football network, which requires the most iterations, converges within 25 iterations.

4.2.2 Community Partition and Evaluation Table 2 shows the average modularity values obtained from 50 runs of various algorithms on the three networks. The algorithms include two typical intelligent evolutionary algorithms: Particle Swarm Optimization (PSO) and Genetic Algorithm CCGA.

Table 2 Comparison of Modularity Values of Various Algorithms on Classic Networks

Algorithm	Karate	Dolphin	Football
GGCSCA	0.4198	0.5278	0.6037
InfoMap	···	···	···
FastGreedy	···	···	···
LeadingEigenvector	···	···	···
BGLL	···	···	0.6037
LPA	···	···	···

For the karate network, GGCSCA achieves modularity Q = 0.4198, superior to other algorithms (only CCGA also achieves 0.4198), and higher than the Q value 0.3715 corresponding to the real community structure. From the modularity optimization perspective, the proposed method achieves excellent results. Similar to the karate network, the dolphin network' s real network corresponds to a Q value of 0.3722, also converging to a local optimum. GGCSCA obtains an average value of 0.5278, better than other algorithms, demonstrating good modularity optimization results.

For the Football network, GGCSCA achieves an average Q value of 0.6037, identical to BGLL and superior to other algorithms. The real Football network is divided into 12 communities. GGCSCA' s best partition result contains 11 communities, with 6 communities perfectly matched and 5 communities partially

matched. Nodes are basically correctly identified, with only 1 real community node assigned to other communities, as shown in Figure 7 [Figure 7: see original paper] and Table 5 (superscripts represent node numbers in the counterpart community).

5 Conclusion

This paper proposes an algorithm that uses modularity Q as the evaluation function, cuckoo search algorithm as the framework, combined with random walks on adjacency tables and gene inheritance strategies, and applies local preference search for maximum modularity increment. This approach ensures global flexibility while accelerating population convergence. Experiments on benchmark and real networks demonstrate that the algorithm possesses strong community identification and detection capabilities. The algorithm does not require prior knowledge or assumptions about communities and can effectively reveal the intrinsic community structure of networks. Future research will combine it with other algorithms to improve search efficiency, implement parallelization mechanisms, and apply it to community mining research on large-scale real-world complex networks such as bank transaction networks.

References

[1] Girvan M, Newman M E J. Community Structure in Social And Biological Networks [J]. Proceedings of National Academy of Sciences of the United States of America, 2002, 99 (12): 7821-7826.

[2] Fortunato S. Community Detection in Graphs [J]. Physics Reports, 2010, 486 (3-5): 75-174.

[3] 杨博, 刘大有, 金弟, 等. 复杂网络聚类方法 [J]. 软件学报, 2009, 20 (1): 54-58.

[4] White S, Smyth P. A Spectral Clustering Approach to Finding Communities in Graphs [C]// Proc of SIAM International Conference on Data Mining. 2005: 76—84.

[5] Newman M E J. Fast Algorithm for Detecting Community Structure in Networks [J]. Physical Review E. 2004, 69 (6): 066133.

[6] Clauset A, Newman M E J, Moore C. Finding Community Structure in Very Large Networks [J]. Physical Review E Statistical Nonlinear & Soft Matter Physics [J]. 2004, 70 (2): 066111.

[7] Pons P, Latapy M. Computing Communities in Large Networks Using Random Walks [J]. Journal of Graph Algorithms and Applications . 2006, 10 (2): 191-218.

[8] Newman M E J, Girvan M. Finding and Evaluating Community Structure in Networks [J]. Physical Review E. 2004, 69 (2): 026113.

[9] Guimera R, Amaral L A N. Functional Cartography of Complex Metabolic Networks [J]. Natrue, 2005, 433 (7028): 895-900.

[10] Blondel V D, Guillaume J L, Lambiotte R, et al. Fast Unfolding of Community Hierarchies in Large Networks [J]. Journal of Statistical Mechanics: Theory and Experiment, 2008, 10: 10008.

[11] Duch J, Arenas A. Community Detection in Complex Networks Using Extreme Optimization [J]. Physical Review E, 2005, 72 (2): 027104.

[12] 黄发良, 张师超, 朱晓峰. 基于多目标优化的网络社区发现方法 [J]. 软件学报, 2013, 24 (9): 2062-2077.

[13] Zhou X, Liu Y H, Li B. A Multi-Objective Discrete Cuckoo Search Algorithm with Local Search for Community Detection In Complex Networks [J], Modern Physics Letters B, 2016, 30 (7): 1-20.

[14] 黄发良, 肖南峰. 网络社区发现的粒子群优化算法 [J]. 控制理论与应用, 2011, 28 (9): 1135-1139.

[15] 邱晓辉, 陈羽中. 一种面向社会网络社区发现的改进粒子群优化算法 [J]. 小型微型计算机系统, 2014, 35 (6): 1422-1425.

[16] Tasgin M, Herdagdelen A, Bingol H. Community Detection in Complex Networks Using Genetic Algorithms [EB/OL]. (2007-11-04) [2017-05-20]. https://arxiv. org/pdf/0711. 0491. pdf.

[17] 邓琨, 张健沛, 杨静. 利用改进遗传算法进行复杂网络社团发现 [J]. 哈尔滨工程大学学报, 2013, 34 (11): 1438-1443.

[18] 金弟, 刘杰, 杨博, 等. 局部搜索与遗传算法结合的大规模复杂网络社区探测 [J]. 自动化学报, 2011, 37 (7): 873-879.

[19] Gach O, Hao J K. A Memetic Algorithm for Community Detection in Complex Networks [C]// Proc of International Conference on Parallel Problem Solving From Nature. [S. l.] : Springer-Verlag, 2012: 327-336.

[20] 金弟, 杨博, 刘杰, 等. 复杂网络簇结构探测——基于随机游走的蚁群算法 [J]. 软件学报, 2012, 23 (3): 451-456.

[21] Chopade P. A Framework for community detection in large networks using game-theoretic modeling [J]. Ieee Transactions On Big Data, 2017, 3 (3): 337-348.

[22] 段震, 闵星, 王倩倩, 等. 基于商空间的多层粒化社区发现方法 [J]. 南京大学学报: 自然科学版, 2017, 53 (4): 764-772.

[23] 金志刚, 徐珮轩. 密度峰值聚类的自适应社区发现算法 [J/OL]. 哈尔滨工业大学学报, 2018, 50 (5). [2017-08-24]. http://kns. cnki. net/kcms/detail/23. 1235. T. 20170824. 1116. 002. html.

[24] Yang X S. Nature Inspired Meta heuristic Algorithms [M]. 2nd ed. [S. l.] : Luniver Press, 2010.

[25] Fortunato S, Barthelemy M. Resolution Limit In Community Detection [J]. Proceedings of National Academy of Sciences of the United States of America, 2007, 104 (1): 36-41.

[26] Lancichinetti A, Fortunato S, Radicchi F. Benchmark Graphs for Testing Community Detection Algorithms [J]. Physical Review E, 2008, 78 (4): 046110.

[27] Danon L, Duch J, Diaz-Guilera A, et al. Comparing Community Structure Identication [J]. Journal of Statistical Mechanics: Theory and Experiment, 2005, 78 (9): 09008.

[28] 何东晓, 周栩, 王佐, 等. 复杂网络社区挖掘——基于聚类融合的遗传算法 [J]. 自动化学报, 2010, 36 (8): 1160-1170.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.