Application of State Transition Simulated Annealing Algorithm in Community Detection

Xiaowei Qin¹, Xiaoxia Han*¹
¹ College of electrical and power engineering, Taiyuan university of technology, Taiyuan, Shanxi, 030024, China
*Corresponding author’s e-mail: hanxiaoxia@tyut.edu.cn

Abstract. Community detection is one of the most important attributes to reveal the hidden structure of complex networks. The way of community detection based on the intelligent optimization algorithm has been widely used. Aiming at the initialization problem of the solution, the paper uses the density peaks clustering (DPC) algorithm to obtain a stable and high-quality solution. Aiming at the search strategy, state transition simulated annealing algorithm (STASA) is proposed. On the basis, three types of operators are designed: vertex replacement operator, community fusion operator and cross mutation are designed. In the initial optimization stage, the vertex substitution operator is used to generate diverse solution. The community fusion operator speeds up the optimization in a certain sense. The introduction of the cross mutation operator in the later optimization stage is only suitable for high-quality solution and enhances the local search ability of the algorithm. Finally, the experimental results on the GN benchmark and the real-world networks show that the algorithm is superior to other classic community detection algorithms in terms of stability and accuracy.

1. Introduction

Many systems in nature and society can be abstracted into complex networks, such as social networks, power transmission networks, and the World Wide Web. The study of network structure is of great significance to efficiently manage these large-scale networks. In fact, the network structure often has such a rule: a certain cluster has close internal links and sparse external links. This structural feature of the network is called community [1]. Community detection is the basis of complex network research. It not only has important theoretical value, but also has been widely used in various fields of society. At present, community detection has been used in unfamiliar protein function analysis [2], semantic community detection [3], computer network analysis [4,5], etc.

Due to the significant practical significance of community detection, many community detection algorithms have emerged. The method based on hierarchical clustering can be further divided into agglomeration and split method [6,7]. These methods will form a tree diagram during the division process, and any layer in the tree can be a kind of community division results. The method based on random walk uses random walk to redefine the distance between two nodes [8]. The community detection method based on evolutionary algorithm takes the modularity function as the fitness function, and uses the intelligent optimization algorithm to maximize the fitness function to achieve the optimal partitioning. Tasgin [9] applied genetic algorithm to community detection for the first time. With the rise of evolutionary algorithm, more scholars use evolutionary algorithm in the field of community detection. Gong [10] proposed a multi-objective discrete particle swarm algorithm, and applied it to community detection. In addition, cloning algorithm [11], bat [12] algorithm, multi-agent genetic
algorithm [13] were all used to solve community detection problems. However, these algorithms usually
cannot obtain a stable modularity value, and due to the mechanism of the greedy algorithm, the algorithm
is easy to fall into a local optimal solution.

Therefore, the paper combines the density peaks algorithm [14] and the state transition simulated
annealing algorithm [15] and applies them to the community detection. Fundamentally speaking,
community detection can be regarded as a graph-based clustering process. Therefore, firstly, we use the
improved density peaks clustering algorithm (IDPC) to complete the initialization process of the solution.
Then for the optimization of the search strategy, the STASA is introduced. Based on the algorithm, three
state transition operators are proposed: vertex replacement operator, community fusion operator and
cross-transform operator, taking into account the global search and the local search capability.

2. State Initialization Based on IDPC

2.1. Density peaks clustering
The DPC algorithm is mainly based on two assumptions about the cluster center:
1) The density of the center is higher than its neighbours;
2) The distance between centers is distant.
Firstly, the local density and the distance are given as follows:

\[
\rho_i = \sum_j \chi(d_{ij} - d_c), \chi(x) = \begin{cases} 
1, & x < 0 \\
0, & x \geq 0 
\end{cases}
\]

(1)

where \(d_{ij}\) represents the distance between \(i\) and \(j\), \(d_c\) is the cutoff distance which need to be determined manually.

\[
\delta_i = \begin{cases} 
\max_{j \neq i} d_{ij}, & \rho_i = \max(\rho) \\
\min_{j \neq i} d_{ij}, & \text{else}
\end{cases}
\]

(2)

After calculating the local density and \(\delta\) of each point, the user manually selects these points with
larger two indicators as the cluster centers on the decision graph. Then, arrange each remaining point in
descending order of density to the cluster where the closest point with higher density is located.

2.2. The framework of IDPC for community detection
Since the clustering is not completely the same as the community detection, some non-common
problems need to be solved. The input of the density peaks clustering algorithm is the distance matrix.
But in complex networks, the given information is only the adjacency matrix. Then the calculation of
the distance between nodes in the network becomes an urgent problem.

Firstly, the connection weight between two points is calculated as follows:

\[
w_{ij} = a_{ij} + \sum_{j \neq i} a_{ij} a_{ij}
\]

(3)

\(a_{ij}\) is the element in adjacency matrix, the right item represents the number of paths from point \(i\) to \(j\)
through two steps.

Furthermore, the weight of each edge of the node is introduced, and the updated link strength is
defined as follows:

\[
str_{ij} = \frac{n \cdot w_{ij}}{\sqrt{\sum_{p \neq i} w_{pj} \sum_{q \neq j} w_{iq}}}
\]

(4)

The local density is defined as follows:

\[
\rho_i = \sum_{j \in \text{neigh}(i)} \exp(-\text{dist}(i, j))
\]

The framework for IDPC on community detection is as follows:
Step 1: generate adjacency matrix;
Step 2: calculate the distance matrix according to Eqs.3,4;
Step 3: calculate the local density and the distance according to Eq.2;
Step 4: select cluster centers on the decision graph;
Step 5: assign the remaining nodes to the cluster of the closest point with higher density.

3. Community Detection based on STASA

STASA is a new global optimization algorithm. STASA uses the state transition algorithm (STA) to enrich the solution generation method of the simulated annealing algorithm (SA). STASA shows its powerful search ability in both traveling salesman problem (TSP) and continuous function optimization problems.

In STASA, each state represents a way of community division. More intuitively, each state contains the community information to which each point belongs.

3.1. Three state transition operators

The original three state transition operators are designed for the TSP, but they are not suitable for community detection, so the state transition operator needs to be redesigned.

1) Vertex replacement operator

The idea of vertex replacement operator comes from label propagation. The label of a node is easily affected by its neighbor nodes. The process of the vertex replacement is shown in Figure 1. The neighbors of node 6 include [1, 7, 11, 17], where the label value of node 1 is 1 (shown in light orange), the label value of other neighbors is 2 (shown in green), and the label of node 6 is random replaced with the label value 2 of the neighbor node 17. Figure 1 shows that the vertex replacement operator optimizes the division effect.

![Figure 1. The process of vertex replacement](image1)

2) Community fusion operator

In the vertex replacement operator, the label of only one node changes each time, so the efficiency of finding the global optimal solution is undoubtedly extremely low. In order to speed up the optimization speed, the label value of the entire community is replaced.

Firstly, we randomly selects a point $i$, the community it belongs to is recorded as $c_i$, and replaces the labels of all nodes in the $c_i$ with the labels of adjacent communities. If there is an edge between two communities, then the two communities are adjacent communities.

![Figure 2. The schematic diagram of consolidating community](image2)
The process of community fusion is shown in Figure 2. The node 24 is randomly selected, since node 24 is connected to node 26, the community label that node 26 belongs to is assigned to all nodes in the entire community where node 24 is located. The modularity is increased from 0.3833 to 0.3991, and the result of community division is optimized.

3) Cross mutation operator

The purpose of the cross mutation operator is to cross over the excellent solution sets generated by the previous two operators in an attempt to produce better solutions.

The process of the cross transform operator is shown in Figure 3. Firstly, node 6 is randomly selected, and the points with the same label as node 6 in the current state 1 are {4, 5, 7}, and then the labels of {4, 5, 7} in the next state 2 will be replaced with the label value of node 6 in state 1. Similarly, {3, 4, 5} in next state 1 is replaced with the label value of node 6 in current state 2. The cross mutation operator enhances the local optimization ability of the algorithm.

### Figure 3. The schematic diagram of cross mutation operator

| nodes | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-------|---|---|---|---|---|---|---|---|---|----|
| Next state 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 4 | 4 | 4 |
| Current state 1 | 1 | 1 | 2 | 3 | 3 | 3 | 3 | 4 | 4 | 4 |
| Current state 2 | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 4 | 4 |
| Next state 2 | 1 | 1 | 2 | 3 | 3 | 3 | 3 | 3 | 4 | 4 |

3.2. The framework of STASA on community detection

Step 1: Initialize parameters;
Step 2: Compute the fitness function of the initial solution $f(\text{Best})$;
Step 3: Perform vertex replacement/community fusion transformation to generate $S_E$ new solutions, select the solution with the largest fitness function value $\text{new}_\text{Best}$, if $f(\text{new}_\text{Best}) > f(\text{Best})$, or $e^{-\eta} \cdot f(\text{new}_\text{Best}) > e^{-\eta} \cdot f(\text{Best})$, $\eta \in (0, 1)$, $\text{Best}_\text{history} = [\text{Best}_\text{history}, \text{new}_\text{Best}]$;
Step 4: Determine whether the number of inner loop is reached. If yes, go to the next step, otherwise return to step 3;
Step 5: Updated the temperature: $T = \alpha \times T, \alpha < 1$;
Step 6: Check if the temperature reaches the terminal temperature $t$. If yes, go to step 7, otherwise return to step 3;
Step 7: Sort the solutions in $\text{Best}_\text{history}$ in descending order according to fitness value, and take the first $\text{elite}_\text{size}$ elements for cross-transformation.
Step 8: Check if the number of iterations is satisfied, if yes, print $\text{Best}$ and $f(\text{Best})$; otherwise return to step 7.

4. Experimental results and analysis

4.1. Argument setting

To demonstrate the effectiveness of the IDPC-STASA algorithm, the GN benchmark and real-world networks are used to test its performance. Moreover, we take Louvain [16], LPA [17], Danon [18], GA [19] as the control group. The GN benchmark [20] is widely used in community detection. In the benchmark, the parameter $\mu$ controls the structural complexity of the network.
The parameters involved in optimization for STASA are mainly SA internal termination iteration number $\text{IterIn}$, starting temperature $T$ and ending temperature $t$, cooling rate and the elite population used for crossover operator transformation, the number of iterations $\text{Iter}$ in the cross transformation, the search enforcement $\text{SE}$. Through experiments, $\text{elite\_size} = 20$, and $\text{IterIn} = 40$, $\text{Iter} = 100$, $T = 1000$, $t = 1$, $\alpha = 0.97$.

The experimental environment is as follows: win7 64bit, Intel(R) Core(TM) i7-6700 CPU@3.4GHz, MATLAB R2019b.

![Figure 4. The results of five algorithm GN benchmark](image)

4.2. **GN benchmark**

In order to compare the performance of the algorithm on networks with different complexity levels more comprehensively, $\mu$ is set to 0.1-0.5 to generate networks with different structural complexity. It should be noted that the structure of the GN benchmark is known, so it is more reasonable to use normalized mutual information (NMI) to measure the quality of community detection. All algorithms run 20 times, and the average value of NMI is taken as the experimental result. It can be seen intuitively from Figure 4 that the proposed algorithm can achieve the highest NMI value at each $\mu$, which shows the effectiveness of IDPC-STASA. As mentioned before, $\mu$ determines the complexity of the network structure. When $\mu$ is between 0.1 and 0.15, that is, when the network structure is relatively simple, each algorithm can eventually reach a high precision value, which is equal to 1 or close to 1. When $\mu$ increases to 0.2, the NMI of the GA algorithm decreases significantly, the remaining four algorithms can still achieve good results. When the value increases from 0.2 to 0.4, only the Louvain and IDPC-STASA can maintain the accuracy close to 1, and the NMI for Danon has decreased, but it is not as sharp as GA and LPA. When the value is between 0.45 and 0.5, the network structure is already very complicated, and the NMI of LPA is almost close to 0. In conclusion, IDPC-STASA can achieve the best results, and the Louvain can also achieve relatively good results. In summary, the results on GN benchmark fully demonstrate the effectiveness of the proposed algorithm.

4.3. **Real-world datasets**

In this section, 8 real-world networks are used to test the effect of IDPC-STASA. Table 1 shows the modularity and NMI value of each algorithm on different datasets. The optimal modularity value and NMI value are highlighted in bold. It should be noted that not all real-world datasets have known structure. In this case, only the modularity is used to evaluate the division result of the algorithm. The larger modularity indicates better division effect.
Table 1. The results of five algorithms in unknown real-world datasets

| Algorithm       | Maximum Modularity | Minimum Modularity | Average Modularity | Maximum NMI | Minimum NMI | Average NMI | Average NMI |
|-----------------|--------------------|--------------------|--------------------|-------------|-------------|-------------|-------------|
| Email Karate    |                    |                    |                    |             |             |             |             |
| LPA             | 0.236              | 1.13E-16           | 0.0079             | 0.402       | 6.41E-18    | 0.287       | 0.3279      |
| GA-net          | 0.261              | 0.2041             | 0.2439             | 0.4024      | 0.3356      | 0.3703      | 0.3392      |
| Louvain         | 0.5412             | 0.5412             | 0.5412             | 0.4188      | 0.4188      | 0.4188      | 0.5866      |
| Danon           | 0.5471             | 0.536              | 0.5399             | 0.4087      | 0.4033      | 0.4065      | 0.5377      |
| IDPC-STASA      | **0.5637**         | **0.5637**         | **0.5637**         | **0.4198**  | **0.4198**  | **0.4198**  | **0.6955**  |
| Netscience      |                    |                    |                    |             |             |             |             |
| LPA             | 0.9255             | 0.9114             | 0.9197             | 0.5265      | 0.3735      | 0.4867      | 0.2644      |
| GA-net          | 0.9211             | 0.8396             | 0.8523             | 0.4678      | 0.3643      | 0.4138      | 0.1862      |
| Louvain         | 0.9543             | 0.9543             | 0.9543             | 0.5188      | 0.5188      | 0.5188      | 0.5162      |
| Danon           | 0.9588             | 0.9588             | 0.9588             | 0.5136      | 0.5136      | 0.5136      | 0.5743      |
| IDPC-STASA      | **0.9598**         | **0.9598**         | **0.9598**         | **0.5267**  | **0.5267**  | **0.5267**  | **0.5715**  |
| Dolphin         |                    |                    |                    |             |             |             |             |
| LPA             | 0.4428             | 0.282              | 0.4344             | 0.603       | 0.4997      | 0.5779      | 0.6823      |
| GA-net          | 0.3652             | 0.2373             | 0.2997             | 0.5588      | 0.3759      | 0.457       | 0.3705      |
| Louvain         | 0.4431             | 0.4431             | 0.4431             | 0.6046      | 0.6046      | 0.6046      | 0.8903      |
| Danon           | 0.4401             | 0.4387             | 0.4394             | 0.5773      | 0.5633      | 0.5725      | 0.7541      |
| IDPC-STASA      | **0.451**          | **0.451**          | **0.451**          | **0.6646**  | **0.6646**  | **0.6646**  | **0.892**   |
| Jazz Football   |                    |                    |                    |             |             |             |             |
| LPA             | 0.6763             | 0.6691             | 0.6607             | 0.4986      | 1.79E-17    | 0.4106      | 0.3479      |
| GA-net          | 0.6515             | 0.6443             | 0.6353             | 0.497       | 0.3242      | 0.3883      | 0.1542      |
| Louvain         | 0.9335             | 0.9335             | 0.9335             | 0.4986      | 0.4986      | 0.4986      | 0.5745      |
| Danon           | **0.9364**         | 0.934             | **0.9354**         | 0.5269      | 0.5237      | 0.5245      | 0.5343      |
| IDPC-STASA      | 0.9345             | **0.9345**         | 0.9345             | **0.5312**  | **0.5312**  | **0.5312**  | **0.5821**  |

5. Conclusion
The application of IDPC and STASA to community detection is studied in this chapter. On the one hand, the improved density peak clustering algorithm is used to complete the initialization of the solution to obtain a stable initial solution, which provides a good foundation for the subsequent optimization process. On the other hand, according to the search strategy, the STASA is introduced. On this basis, three state transition operators for community detection, vertex replacement operator, community fusion operator and cross mutation operator are designed to ensure the diversity of solution, taking into account the global search and local search ability of the algorithm. Finally, the modularity and NMI value of the proposed algorithm on the GN benchmark and real-world networks are almost better than the other four classic algorithms, verifying the effectiveness of the proposed algorithm. In future work, we are committed to designing search operators with more powerful search capabilities, improving the search efficiency of the algorithm and the universality of dynamic community detection.

References
[1] Newman M. and Girvan M. (2004) Finding and Evaluating Community Structure in Networks. Physical Review E, 69: 026113.
[2] Chen J. Yuan B. (2006) Detecting functional modules in the yeast protein-protein interaction network. Bioinformatics, 22: 2283-2290.
[3] Shen G, Jia C, Jian Y U. (2017) Semantic Community Detection Algorithm for Large Scale Information Network. Journal of Frontiers of Computer Science and Technology.

[4] Dourisboure Y. (2008) Extraction and Classification of Dense Communities in the Web. In: Proceedings of the 16th international conference on World Wide Web. New York, Doi: 10.1145/1242572.1242635

[5] Ino H, Kudo M, Nakamura A. (2005) Partitioning of Web Graphs by Community Topology. In: Proceedings of World Wide Web. Japan

[6] Girvan M and Newman M. (2002) Community Structure in Social and Biological Networks. In: Proceedings of the national academy of sciences. pp: 7821-7826.

[7] Newman M E. (2004) Fast algorithm for detecting community structure in networks. Phys Rev E Stat Nonlin Soft Matter Phys, 69:066133.

[8] Zhou H. (2003) Network landscape from a Brownian particle’s perspective. Physical Review E, 67(4): 041908.

[9] Tasgin M, Herdagdelen A, Bingol H. (2007) Community Detection in Complex Networks using Genetic Algorithm. arXiv:0711.0491.

[10] Gong M, Cai Q, Chen X. (2013) Complex Network Clustering by Multiobjective Discrete Particle Swarm Optimization Based on Decomposition. IEEE Transactions on Evolutionary Computation, 18(1):82-97.

[11] Cai Q, Gong M, Ma L. (2015) A Novel Clonal Selection Algorithm for Community Detection in Complex Networks. Computational Intelligence, 31(3): 442-464.

[12] Song A, Li M, Ding X. (2016) Community Detection Using Discrete Bat Algorithm. International Journal of Computer Science, 43(1): 37-43.

[13] Li Z, Liu J. (2016) A multi-agent genetic algorithm for community detection in complex networks. Physica A: Statistical Mechanics and its Applications, 449: 336-347.

[14] Rodriguez A, Laio A. (2016) Clustering by fast search and find of density peaks Science, 344(6191): 1492-1496.

[15] Han X, Dong Y, Yue L, and Xu Q. (2019) State Transition Simulated Annealing Algorithm for Discrete-Continuous Optimization Problems IEEE Access, 7: 44391-44403.

[16] Blondel V D, Guillaume J L, Lambiotte R. (2008) Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, P10008.

[17] Raghavan U N, Albert R, Kumara S. (2007) Near linear time algorithm to detect community structures in large-scale networks. Physical review E, 76(3): 036106.

[18] Danon L, Diaz-Guilera A, Arenas A. (2006) The effect of size heterogeneity on community identification in complex networks. Journal of Statistical Mechanics: Theory and Experiment, 11: P11010.

[19] Hafez A I, Ghali N I, Hassanien A E, et al. (2012) Genetic algorithms for community detection in social networks. In: 12th International Conference on Intelligent Systems Design and Applications. pp: 460-465.

[20] Lancichinetti A, Fortunato S, Radicchi F. (2008) Benchmark graphs for testing community detection algorithms. Physical review E, 78(4): 046110.