Population Cross Learning Algorithm Combining Greedy Search for Community Detection

Xiaogang Wang¹, Guanghui Yan*, Shikui Li¹, Ning Zhou¹

¹School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China
*Corresponding author’s e-mail address: yanghacademic@163.com

Abstract. In order to improve the precision of modularity optimization and community detection, this paper presented a complex network community detection algorithm based on cross learning among individuals of population combining greedy search. Individuals’ codes indicated community partition. Individuals comparatively studied with each other to spread good genes and optimize modularity fast. Besides, aiming at improving the algorithm, the best communities, where some randomly selected nodes will move in, would be found by using greedy search maximizing the local modularity increment. The algorithm was tested on artificial networks and some typical real networks, compared with some typical algorithms. The results show that algorithm can get convergence quickly, achieve better modularity value, and finely detect and identify community structures.

1. Introduction

Discovery of complex network communities has very important theoretical significance for analyzing the topology, understanding the functions, discovering hidden rules in complex networks and predicting the behavior of complex networks [1]. Its application fields include commercial customer network clustering analysis, public opinion analysis, search engines, partnership research, task scheduling allocation, etc.

The community discovery methods can be divided into the following categories [1]: 1) Hierarchical clustering, including agglomerative algorithms and divisive algorithms. The former is bottom-up cluster merging algorithms, such as CNM [2]; the latter method, splits clusters from top to bottom, such as GN [3]. 2) Spectral clustering [4], using quadratic form optimization techniques to minimize the predefined "Cut" function. "Cut" is the connection density between subnets. 3) Optimization methods, which is based on the technology of search (global search or local search), contain three basic parts: the objective function, search strategy for candidate solutions and search strategy for the best solutions. Such as BGLL Algorithm [5], LGA [6], MDCL [7]. 4) Heuristic methods, based on some intuitive hypothesis. For example, Chopade proposed the large network community discovery framework based on the game theory model [8]. Jin proposed an adaptive community detection algorithm of density peak clustering (KDED) [9]. 5) Dynamic methods, the characteristics of information propagation inside and outside of a community are different. Dynamics propagation law of complex networks can be used to discover communities, like LPA [10]. 6) similar aggregation, for example, Qiu proposed a community detection algorithm based on similar aggregation of multiple attribute nodes [11].

Optimization of the modularity [12] (or other objective functions) is a kind of important method commonly used in community detection. Existing algorithms need to be further improved in the...
following aspects of accuracy and efficiency of optimization, quality and accuracy of community discovery, and ease of use and generality of methods. Referenced to the idea of Firefly Algorithm [13], this paper proposed a population cross learning algorithm for complex network community detection combing greedy search (CLGSCDA). Individuals of population representing the community partition, learned from each other and disseminated high-quality genes. The degree of learning was related to the modularity difference. The algorithm can converge quickly because the individuals absorb high-quality genes from each other. In addition, greedy search maximized the local modularity increment and optimized the community partition. Without any special initialization algorithm and preconditions, CLGSCDA can achieve the high-quality community detection result.

2. Algorithm of modularity optimization and community detection

2.1. Encoding Mode
A network is $G=(V,E)$, $V$ is node set, and $E$ is edge set. Individuals of population is coded on the basis of neighbor notation to avoid generating the illegal solution. Given individual $i$, $X_i = \{x_{i1}, x_{i2}, \cdots, x_{im}\}$, if $x_{im} = k$ $(m=1 \cdots n)$, it means that nodes $m$ and $k$ belong to the same community, while $m$ is adjacent to $k$. Figure 1(a) is a simple network, figure 1(b) is two communities of this network, and figure 1(c) shows the individual code.

![Figure 1. Example of network, community, and coding.](image)

2.2. Evaluation Function
Modularity [5,12] is a mathematical measurement to evaluate the quality of complex network community partition, which is expressed as follows:

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

$m$ is total number of edges of a network. $i$ and $j$ are two nodes in the network. $a_{ij}$ is the element of the network adjacency matrix, $k_i$ and $k_j$ are degree of node $i$ and node $j$. $C_i$ and $C_j$ represent communities to which $i$ and $j$ belong. When $i$ and $j$ belong to the same community $\delta(C_i,C_j)=1$, otherwise, it is equal to 0. Generally, the higher value of the modularity, the more accurate the community division.

2.3. Algorithm and Its Process

2.3.1. Ideas of Crossing Learning. $Q(X_i)$ and $Q(X_j)$ are the modularity value of individuals $i$ and $j$. If individual $i$ has better modularity than individual $j$, i.e. $Q(X_i) \geq Q(X_j)$, individual $j$ learn part
genes from individual $i$ according to the probability threshold $p^{c \rightarrow i}$ (A random value greater than $p^{c \rightarrow i}$). The learning probability threshold is determined by the difference of modularity:

$$p^{c \rightarrow i} = \begin{cases} 1 - \frac{Q(X_i) - Q(X_j)}{Q(X_i) - Q(X_j)} & Q(X_i) - Q(X_j) \leq 1 \\ 0.1 & Q(X_i) - Q(X_j) > 1 \end{cases}$$

(2)

The above formula shows that the closer the individual $i$ modularity is to the individual $j$ modularity, the larger the probability threshold is and the smaller the probability of $j$ learning from $i$ is. Conversely, the greater the modularity difference between individual $i$ and $j$, the smaller $p^{c \rightarrow i}$ is and the greater the probability of $j$ learns from $i$ is. $\Delta Q_{ij}$ indicates that the value of individual $j$ is very poor, basically, it should absorb the individual $i$’s genes completely, but still need to avoid the exact same. Therefore, set $p^{c \rightarrow i}$ to 0.1.

Let $R = \{r_1, r_2, \ldots, r_n\}$ be an n-dimensional random probability vector, and each random element takes a value between [0-1]. The elements of vector $S^{c \rightarrow i} = \{s_1, s_2, \ldots, s_n\}$ are the various gene learning factors that individual $j$ learn from $i$. According to the probability vector $R$, which is generated randomly, and the probability threshold $p^{c \rightarrow i}$, $S^{c \rightarrow i}$ is calculated as:

$$s_i = \begin{cases} 1 & r_i \geq p^{c \rightarrow i} \\ 0 & r_i < p^{c \rightarrow i} \end{cases} \quad i = 1 \cdots n$$

(3)

the updated formula of $j$ learns from $i$ is

$$X'_j = S^{c \rightarrow i} \cdot X_j + (E - S^{c \rightarrow i}) \cdot X_j = \{s_1 x_{i1} + (1 - s_1) x_{j1}, s_2 x_{i2} + (1 - s_2) x_{j2}, \ldots, s_n x_{in} + (1 - s_n) x_{jn}\}$$

(4)

$E = \{1, 1, \ldots, 1\}$ is the n-dimension vector. The update of individuals is determined by the modularity difference and the random probability, which ensures the transmission of high-quality genes, avoids premature homogeneity convergence, and keeps diversity in the optimization process.

2.3.2. Local Greedy Search

Local search algorithm uses greedy search operator: calculates the modularity gain $\Delta Q$ when a node $i$ leaves its current community to enter a neighbouring community and finds the community that makes $\Delta Q$ the largest one as the target community of the node moving. There are two mainly steps: 1) Let node $i$ leave the original community and become the independent node, generate $\Delta Q_1$, see formula (5); 2) Increment $\Delta Q_2$ generated by moving the independent node $i$ into the new community, see formula (6). $\Delta Q = \Delta Q_1 + \Delta Q_2$.

$$\Delta Q_1 = \left[ \sum_{in} - \frac{k_{i,n}}{2m} \right] \left( \sum_{out} - \frac{k_i}{2m} \right)^2 \left[ \sum_{in} - \frac{\sum_{out}}{2m} \right]$$

(5)

$$\Delta Q_2 = \left[ \sum_{in} + \frac{k_{i,n}}{2m} \right] \left[ \sum_{out} + \frac{k_i}{2m} \right] \left[ \sum_{in} - \frac{\sum_{out}}{2m} \right]$$

(6)

Let $C$ be the community which node $i$ is to join (or leave), $\sum_{in}$ represents the weight sum of the $C$ internal connection. $\sum_{out}$ represents the sum of the weights from the node in $C$, including the connections within the community and the connections outside the community. $k_i$ represents the degree of node $i$, $k_{i,n}$ represents the sum of the weight of the node $i$ with nodes in $C$. The weight within the community is calculated twice per edge.
2.4. Algorithm Framework
According to the above algorithm ideas, the algorithm implementation framework is given as followed:

```
Community detection algorithm
Input: Complex network $G = (V,E)$, population $s$, local search probability threshold $pa$, number of iterations $T$
Output: Community division $C = \{C_1, C_2, \cdots, C_k\}$
Begin:
1. Build neighbor ordered table $AdjTb$;
2. Define the modularity function $Q$ as the objective function;
3. Initialize the population. Each dimension in individual vector $X_i (i = 1,\ldots,s)$ gets the value in the neighbor ordered table randomly. Calculate the initial $Q$ value; $f_{max}$ is the current maximum $Q$ value, $best$ is the current optimal solution, $t = 0$.
4. Generate a random probability vector $R$; Compare individuals pairwise and calculate the probability threshold $p_{i-ni}$ according to formula (2); calculate $S_{i-ni}$ according to formula (3).
5. Update the individuals according to formula (4) and get $X_i'$.
6. According to $X_i'$, calculate the value of objective function $Q$, if $Q(X_i') > Q(X_i)$, then $X_i \leftarrow X_i'$.
7. Replace the current population global maximum $Q$ value $f_{max}$ with the largest of $Q(X_i)$ ($i = 1,2,\ldots,s$), replace $best$ with the corresponding individual solution.
8. Perform local search algorithm LocalSearch($X,pa$)
1) According to the probability threshold $pa$, the candidate individuals applying the local greedy search operator are selected. Each candidate individual $X_i$ is decoded for obtaining the corresponding community division.
2) Calculate $\Delta Q$ for each dimension $x_j (j = 1,\ldots,n)$ of candidate individual $X_i$, When node $j$ leaves its community to join other neighboring communities, $\Delta Q = \Delta Q_1 + \Delta Q_2$ can be calculated from formulas (5) and (6).
3) Find the community that makes $\Delta Q$ the biggest, $j$ joins this community.
4) Obtain a new population code $X' = \{X'_1, X'_2, \cdots, X'_i\}$, and $X \leftarrow X'$.
9. $t = t + 1$, if the iteration times $t < T$, go to step 4; Otherwise, go to step 10.
10. Exit the iteration, decode the optimal solution $best$, then get the community partition $C = \{C_1, C_2, \cdots, C_k\}$.
```

3. Experiment and analysis
The numerical test is operated on a computer with an Intel i5 processor, 4G memory and Win7 operating system, and is programmed in MATLAB environment. In CLGSCDA, population scale is set to 30.

The networks used in the experiment are real networks[14] including Karate, Dolphin, Hamster, PolBook, Football, Benchmark test network [15], synthetic networks Lfr500 with 500 nodes and Lfr1000 with 1000 nodes.
3.1. Local Search Probability Threshold \( p_a \)

Table 1. The effect of \( p_a \) on modularity in CLGSCDA.

| Network    | 0.1       | 0.2       | 0.3       | 0.4       | 0.5       | 0.6       | 0.7       | 0.8       | 0.9       |
|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Karate     | 0.4198    | 0.4198    | 0.4198    | 0.4198    | 0.4198    | 0.4198    | 0.4198    | 0.4198    | 0.4198    |
| Dophin     | 0.5278    | 0.5281    | 0.5282    | 0.5277    | 0.5279    | 0.5280    | 0.5276    | 0.5276    | 0.5272    |
| PolBook    | 0.5272    | 0.5272    | 0.5272    | 0.5272    | 0.5272    | 0.5271    | 0.5266    | 0.5271    |           |
| Football   | 0.6043    | 0.6040    | 0.6046    | 0.6041    | 0.6044    | 0.6042    | 0.6043    | 0.6038    |           |
| Lfr500     | 0.6314    | 0.6314    | 0.6314    | 0.6314    | 0.6312    | 0.6314    | 0.6314    | 0.6306    |           |
| Lfr1000    | 0.6624    | 0.6623    | 0.6624    | 0.6624    | 0.6624    | 0.6624    | 0.6624    | 0.6623    | 0.6622    |

It is apparently seen from Table 1 that when \( p_a = 0.3 \) each network obtains the maximum modularity (underlined number).

3.2. Artificial Benchmark Test Network

Benchmark test network proposed by Lancichinetti can be used to examine the ability of algorithms to identify communities. There are 128 nodes and 1024 edges in the network, which is divided into four communities. Each community has 32 nodes. Nodes average degree is 16.

The hybrid parameter for generating the network significantly affects the community structure. The role of the hybrid parameter \( \mu \) is to control the ratio between degree of intra-community of a node and its total degree. Each node has a fraction \( 1 - \mu \) of its links with the nodes in the same community and a fraction \( \mu \) with the rest of the network. The probability \( \mu \) is from 0.1 to 0.5 and community structure changed from clear to vague. When \( \mu < 0.5 \), a node and its neighbors are more likely to belong to the same community than to belong to different communities, and community structure is easy to identify; When \( \mu \) taking 0.5, averagely there are half of the connections of each node point to the nodes outside the community and the community structure is rather vague.

This experiment uses the standard mutual information (NMI) [4] to evaluate the community identification effect, see formula (8):

\[
\text{NMI}(A,B) = \frac{-2 \sum_{i=1}^{C_A} \sum_{j=1}^{C_B} C_{ij} \log \left( \frac{nC_{ij}}{C_iC_j} \right)}{\sum_{i=1}^{C_A} C_i \log \left( \frac{C_i}{n} \right) + \sum_{j=1}^{C_B} C_j \log \left( \frac{C_j}{n} \right)}
\]  

(8)

\( A \) and \( B \) are two community partitions, \( C \) is a hybrid matrix, \( C_A \) (or \( C_B \)) is the number of communities in \( A \) (or \( B \)). \( C_{ij} \) is the number of nodes belonging to the community \( i \) in division \( A \) and the community \( j \) in the division \( B \) at the same time. \( C_i \) represents the number of \( i \) th community nodes in \( A \) and \( C_j \) represents the number of \( j \) th community nodes in \( B \). The NMI value ranges from [0,1]. If \( A = B \), \( \text{NMI}(A,B) = 1 \); If \( A \) and \( B \) are completely different, \( \text{NMI}(A,B) = 0 \). the larger the NMI is, the more similar two communities are. Communities detected by CLGSCDA are compared with communities detected by classical algorithms such as Walktrap, BGLL, Edge Betweenness (EB), Infomap, Leading Eigenvector (LE), Fastgreedy, and LPA according to NMI with real community partitions. As shown in Table 2.
Table 2. NMI results of different algorithms on artificial networks.

| Hybrid Parameter | CLGSCDA | CDA | Walktrap | BGLL | EB | Infomap | LE | Fast greedy | LPA |
|------------------|---------|-----|----------|------|----|---------|----|-------------|-----|
| 0.05             | 1       | 1   | 1        | 1    | 1  | 1       | 1  | 1           | 1   |
| 0.1              | 1       | 1   | 1        | 1    | 1  | 1       | 1  | 1           | 1   |
| 0.15             | 1       | 1   | 1        | 1    | 1  | 1       | 1  | 1           | 0.9971 |
| 0.2              | 1       | 1   | 1        | 1    | 1  | 1       | 1  | 1           | 0.9575 0.9914 |
| 0.25             | 1       | 1   | 1        | 1    | 1  | 1       | 1  | 0.9748      | 0.8895 0.9616 |
| 0.3              | 1       | 1   | 1        | 1    | 1  | 1       | 1  | 0.8422      | 0.9748 0.8537 |
| 0.35             | 1       | 1   | 1        | 1    | 1  | 1       | 1  | 0.9748      | 0.9748 0.6570 |
| 0.4              | 1       | 1   | 1        | 1    | 1  | 0.34    | 0.34 | 0.8753      | 0.7638 0.2417 |
| 0.45             | 1       | 1   | 1        | 0.5730 | 0.5387 | 0 | 0.8491 | 0.4142 | 0.0337 |
| 0.5              | 0.6022  | 0.5506 | 0.3166  | 0.4445 | 0   | 0.3884 | 0.2896 | 0           | 0    |

It can be seen from Table 2 that the CLGSCDA algorithm fully identified the community structure of the nine benchmark networks with $\mu$ from 0.05 to 0.45. For the last network with $\mu=0.5$, the community structure, the NMI reaches 0.6022. That is the best performance of all algorithms. The best of the remaining algorithms is Walktrap, which also accurately identifies the 9 networks with $\mu$ from 0.05 to 0.45. But the effect of identifying the last network community is slightly worse than CLGSCDA. Additionally, BGLL and Edge Betweenness performed well and 8 networks $\mu$ from 0.05 to 0.4 were correctly identified.

3.3. Real Network

From Table 3, compared with other algorithms, CLGSCDA has achieved superior values on several classic real network datasets. For the Karate network, the CLGSCDA got the optimal value of 0.4198 and only the LGA algorithm got 0.4198 also. In the Dophin network, the CLGSCDA achieved the optimal value of 0.5282, which is better than all the other algorithms. CLGSCDA in the PolBook network achieved 0.5272, which is as same as LGA and better than other algorithms. In the Football network, the algorithm achieved 0.6046, which is the same as the LGA and BGLL and is better than other algorithms. In the Hamster network, the CLGSCDA algorithm got the optimal value of 0.5527, which is better than all the other algorithms.

Table 3. Modularity results of different algorithms on real networks.

| Algorithm  | Karate | Dophin | PolBook | Football | Hamster |
|------------|--------|--------|---------|----------|---------|
| BGLL       | 0.4188 | 0.5185 | 0.5205  | 0.6046   | 0.5467  |
| CNM        | 0.417  | 0.485  | 0.502   | 0.576    | 0.5160  |
| GN         | 0.4013 | 0.4706 | 0.5168  | 0.5996   | 0.5278  |
| LPA        | 0.3380 | 0.4782 | 0.4872  | 0.5857   | 0.3274  |
| LGA        | 0.4198 | 0.5280 | 0.5272  | 0.6046   | 0.5450  |
| Infomap    | 0.4020 | 0.5240 | 0.5228  | 0.6005   | 0.5039  |
| KDED       | 0.402  | 0.447  | 0.506   | 0.585    | 0.5246  |
| MDCL       | 0.3715 | 0.3735 | 0.461   | 0.59     | 0.5138  |
| CLGSCDA    | 0.4198 | 0.5282 | 0.5272  | 0.6046   | 0.5527  |
As examples, the specific community partitions of Karate and Football obtained by CLGSCDA are illustrated as followed. The Karate real community structure modularity value is 0.37. The CLGSCDA algorithm got optimal modularity value of 0.4198 and has achieved best result from the perspective of optimization. Karate real community structure is divided into two communities, and by this algorithm is divided into four communities, which reflects the real network community division is deviated from the community structure based on mathematical division. From another perspective, it is reasonable: four communities are the subdivision of two real communities, as shown in Table 4.

### Table 4. Community partition of karate.

| Community | Node | Community (Subdivision) | Node |
|-----------|------|-------------------------|------|
| 0         | 1 2 3 4 5 6 7 8 | 0 1 2 3 4 8 12 13 14 |
|           | 11 12 13 14 17 | 18 20 22 | 18 20 22 |
|           | 11 18 20 22    | 5 6 7 11 17 |
| 1         | 9 10 15 16 19  | 2 9 10 15 16 19 21 23 |
|           | 21 23 24 25 26 | 27 30 31 33 34 |
|           | 27 28 29 30 31 | 24 25 26 28 29 32 |
|           | 32 33 34       | 24 25 26 28 29 32 |

There are 12 real football communities. Five nodes (36, 42, 80, 82, 90) which is in one community represent relatively independent teams, and they play more games with other federations (community) teams than they do internally. In this sense, the topology should be divided into 11 communities. During the running of the algorithm, the most complete community partition is the 11-cluster network, as shown in Figure 2.

![Figure 2. 11-clusters community partition of Football.](image)

Comparing 11-cluster community partition with 12-cluster real community partition, the nodes \{36, 42, 80, 82, 90\} are classified in error, and the error rate is \(11/115=9.6\%\). Among them, \{36, 42, 80, 82, 90\} belong to a real community that is quite loose and not recognized by the algorithm.

In general, CLGSCDA optimizes the modularity and detects the community structure of the network more delicately.

### 4. Conclusions

Community mining is a hot topic in the field of complex networks. This paper proposed a community detection algorithm based on population cross learning combining local greedy search. In the implementation, the inferior population members absorb the genes of the better population members with a certain probability, and this probability is related to the modularity difference and random value. At the same time, in order to improve the quality of the population, a local greedy search optimization is performed in each iteration. The algorithm has few parameters, no prior knowledge and no special initialization method are required. Probability-based individuals cross learning algorithms and local
search can ensure the spread of high-quality individual genes and avoid premature convergence. The experimental results show that the proposed algorithm can optimize the modularity well and obtain high quality community partitioning result.

Acknowledgments
This work was supported by the National Social Science Foundation of China under grant NO. 17BXW071

References
[1] Yang B, Liu D Y and Liu J M 2009 J. Software 20 54.
[2] Clauset A, Newman M E J and Moore C 2004 Phys. Rev. E: Stat. Nonlinear, Soft Matter Phys 70 066111.
[3] Girvan M and Newman M E J 2002 Proc. Natl. Acad. Sci. U. S. A 99 7821.
[4] Hu F, Liu J, Li L H and Liang J 2019 Preprint J.phys.A(2019)123633 (https://doi.org/10.1016/j.physa.2019.123633).
[5] Blondel V D, Guillaume J L, Lambiotte R and Lefebvre E 2008 J. Stat. Mech.: Theory Exp 2008 p10008.
[6] Jin D, Liu J and Yang B 2011 Acta Automatica Sinica 37 873.
[7] Zhou X, Liu Y H and Li B 2016 Mod. Phys. Lett. B 30 1650080.
[8] Chopade P and Zhan J 2017 IEEE Trans. big. Data 3 276.
[9] Jin Z G and Xu P X 2018 J. Harbin. Inst. Technol 50 44.
[10] Raghavan U N, Albert R and Kumara S 2007 Phys. Rev. E: Stat. Nonlinear, Soft Matter Phys 76 036106.
[11] Qiu S M, Yu T, Du X L and Chen B 2019 Preprint Comput. Eng( http://kns.cnki.net/kcms/detail/31.1289.tp.20190808.1701.023.html,2019).
[12] Newman M E J and Girvan M 2004 Phys. Rev. E 69 026113.
[13] Yang X S 2010 Nature-Inspired Metaheuristic Algorithms 2nd Edition (Beckington: Luniver Press), p 81.
[14] Florida ecosystem dry network dataset -- KONECT, April 2017. [http] http://konect.uni-koblenz.de/networks/foodweb-baydry.
[15] Lancichinetti A, Fortunato S and Radicchi F 2008 Phys. Rev. E 78 046110.