Postprint: Overlapping Community Detection Using an Improved Ant Colony Algorithm

Authors: Xu Ying, Jin Xuwen

Date: 2019-04-01T00:00:00+00:00

Abstract

To address the problem of improving accuracy in overlapping community detection, a novel overlapping community detection algorithm based on an improved ant colony algorithm is proposed. The algorithm comprises three stages: position initialization, movement, and post-processing. Through approaches including initial position identification and label list storage, heuristic information redefinition based on inter-node similarity, and cooperative label list preservation, the algorithm achieves enhanced performance in detecting overlapping communities and nodes in both synthetic and real-world datasets. Experimental results demonstrate that on synthetic and real-world network platforms using various detection algorithms, the proposed method attains higher detection accuracy for overlapping communities and overlapping nodes compared to traditional detection methods, thus providing valuable reference and guidance for solving overlapping community detection problems and understanding network functional structures.

Full Text

Overlapping Community Detection and Analysis Method Using Improved Ant Colony Algorithm

Xu Ying
School of Applied Mathematics, Xinjiang University of Finance & Economics, Urumqi 830012, China

Abstract: This paper proposes a novel overlapping community detection algorithm based on an improved ant colony algorithm to enhance detection accuracy. The algorithm comprises three stages: position initialization, movement, and post-processing. Through initial position identification and label list storage, heuristic information redefinition based on inter-node similarity, and cooperative label list preservation, the algorithm achieves superior performance in detecting overlapping communities and nodes in both synthetic and real-world datasets. Experimental results demonstrate that when compared with different detection algorithms on synthetic and real-world network platforms, the proposed method exhibits higher detection accuracy for overlapping communities and nodes compared to traditional approaches, providing valuable insights for solving overlapping community detection problems and understanding network functional structures.

Keywords: overlapping community and node detection; improved ant colony algorithm; heuristic information redefinition; tag list iteratively update

0 Introduction

A community is defined as a group of vertices in a network structure that share common attributes [?]. Similar to individuals playing different roles in social networks, in standardized network partitioning, each vertex is assigned to a community [?] and may be shared among multiple communities. Moreover, since different nodes play distinct roles in network structure and function, detecting overlapping communities enables deeper understanding of network functionality and architecture. Consequently, overlapping community detection has attracted significant academic attention in recent years, with numerous algorithms demonstrating strong analytical performance and finding applications in delay-tolerant networks [?, ?], recommendation systems [?], and other domains.

Current community detection algorithms can be broadly categorized into five types [?]: clique percolation method (CPM), local expansion and optimization algorithms, link partitioning algorithms, fuzzy detection algorithms, and agent-based algorithms. Reference [?] introduced an indexed local adjacency representation into individual representation for community detection, transforming community structure analysis into an integer optimization problem and proposing a differential evolution-based community detection algorithm. Reference [?] proposed a self-organizing overlapping community structure analysis algorithm based on swarm intelligence principles, though its detection performance is significantly influenced by convergence precision among agents. Reference [?] developed a social network community detection method based on interactive behavior and connection analysis through effective fusion of two different types of heterogeneous information, yet its performance heavily depends on data attributes. Reference [?] presented a locally prioritized dynamic network overlapping community evolution analysis method, though its global optimality remains to be validated. Reference [?] introduced a spectral bisection community detection method using optimal eigenvectors. While these overlapping community detection algorithms can effectively address community analysis to some extent, they all neglect the impact of node label changes during iterative processes on detection performance.

Drawing inspiration from pheromone updating and path selection strategies in ant colony algorithms, this paper proposes an overlapping community detection algorithm based on an improved ant colony algorithm (AntCBO). Building upon an ant colony algorithm-based overlapping community detection framework, the algorithm achieves overlapping community detection through two key components: heuristic information redefinition based on inter-node similarity and ant path selection based on label list updating and storage. Performance analysis is conducted on both synthetic and real-world datasets.

1 AntCBO Overlapping Community Detection Algorithm Architecture

A network can be abstractly modeled as an undirected graph $G = (V, E)$, where V and E represent the sets of nodes and edges, respectively. Each node v possesses a unique label (identification, ID) l_v, and $N(v)$ denotes the neighborhood of node v. The objective of overlapping community detection is to find a method to partition the abstracted undirected graph G into a series of small clusters $(C_1, C_2, \dots, C_m)$ such that nodes within each cluster share common characteristics.

Received: October 16, 2018; Revised: December 12, 2018
Funding: National Natural Science Foundation of China (11461063)
Author bio: Xu Ying (1981-), female, from Urumqi, Xinjiang, associate professor, Ph.D., main research direction: complex network theory and applications (lu77320jiang68@163.com)

1.1 Algorithm Framework

The proposed AntCBO algorithm framework consists of four modules: parameter initialization, ant position initialization, movement path decision, and post-processing. The basic workflow is as follows: (a) initialize parameters, treat each node as an ant individual, and calculate the transition probability matrix based on pheromones and the proposed heuristic information; (b) initialize ant positions and determine corresponding ID lists; (c) all ants move based on the transition probability model, during which each ant transfers node labels from one node to another and updates pheromones; (d) when termination conditions are met, save the current label list of each ant individual to obtain label sequences for each node; (e) employ a post-processing mechanism to partition the network into overlapping communities.

Algorithm 1: AntCBO Algorithm Framework
Input: Complex network $G = (V, E)$.
Output: A set of overlapping communities C.

The algorithm proceeds through parameter initialization, ant position initialization, ant movement, and post-processing stages. During parameter initialization, the maximum iteration count T, number of ants n, initial pheromone value τ, pheromone evaporation rate ρ, threshold θ, pheromone increment x, and pheromone threshold b are set. The transition probability matrix is calculated according to Equations (1)-(3). In the ant movement stage, for each iteration t from 1 to T and each ant i from 1 to n, ant i moves according to the transition probability matrix while pheromone diffusion and updating occur.

1.2 Improvements to the AntCBO Algorithm

For AntCBO, the relative magnitude of $\tau_{ij}(t)$ (where $i, j = 1, 2, \dots, n$ and $i \neq j$) directly affects the transition probability between ant positions, thereby influencing solution quality. During the search process, initially dispersed pheromones gradually concentrate on certain edges, continuously reinforcing the search direction. When certain edges exhibit significantly higher pheromone intensity than others, the selected edges during solution construction become too similar, causing the algorithm to fall into local optima. The fundamental approach to avoiding local optima is to increase solution diversity by preventing excessive concentration of pheromones on edges, enabling more edges to participate in feasible solution construction with higher probabilities. This means fully utilizing the positive feedback mechanism of ant colony algorithms to accelerate search while maximizing the search region for feasible solutions using more edges to form new solutions.

Following this principle, this paper proposes directly exchanging pheromones on partial edges to alter the distribution of $\tau_{ij}(t)$ across different edges, thereby avoiding local optima in later algorithm stages. Specifically, each node is assigned an exchange probability p_i $(i = 1, 2, \dots, n)$, and a random number $r_i \sim U(0, 1)$ is generated. If $r_i \leq p_i$, a certain number of edges among the $n - 1$ edges that node i can form with other nodes are randomly selected, and their corresponding pheromone values are swapped pairwise. If $r_i > p_i$, no such operation is performed.

Furthermore, the pheromone evaporation rate also affects pheromone distribution. In traditional ant colony algorithms, the pheromone evaporation rate ρ is often set as a constant with identical values for all edges. This approach leads to excessive probability for some edges to be used in constructing feasible solutions, hindering other edges from participating in new solution construction. Therefore, this paper employs a variable parameter evaporation rate to prevent certain edges on optimal paths from losing selection opportunities due to excessively low pheromone intensity. The calculation formula is as follows:

$$\rho_{ij}(t) = \begin{cases} k_1 & \text{if } t \leq t_{early} \\ k_2 \cdot \frac{C}{\tau_{ij}(t)} & \text{otherwise} \end{cases}$$

where $\rho_{ij}(t)$ represents the pheromone evaporation rate on edge ij at time t; t_{early} is the early search time; $k_1, k_2 \in (0, 1)$ and $k_1 > k_2$. Additionally, the values of k_1 and k_2 should not differ excessively, otherwise the exchanged pheromones would evaporate too quickly or too slowly, affecting the aforementioned edge exchange effectiveness. C is a constant between the mean and maximum values of $\tau_{ij}(t)$.

Since the algorithm core involves designing reasonable heuristic information and pheromone updating strategies to calculate the transition probability matrix for ant movement path decisions, we first adopt similarity measures between any nodes i and j to update heuristic information, as shown in Equations (4)-(5).

Let $N_{comm}(i,j)$ denote the common neighbors of nodes i and j, $N(i)$ denote the neighbors of node i, $N(j)$ denote the neighbors of node j, and $E(N_{comm}(i,j))$ denote the number of edges connecting each common neighbor node. The heuristic information update strategy consists of two components controlled by weight coefficient γ, as shown in Equation (3):

$$\eta_{ij} = \gamma \times \frac{|N_{comm}(i,j)|}{|N(i) \cup N(j)|} + (1-\gamma) \times \frac{2|E(N_{comm}(i,j))|}{|N_{comm}(i,j)| \times (|N_{comm}(i,j)| - 1)}$$

The first term measures the average number of common neighbor nodes between vertices i and j, calculated as the number of common neighbors divided by the union size of both vertices' neighbor sets. The second term quantifies the connection distance among common neighbors, calculated as the number of edges between common neighbors divided by the maximum possible edges between them. The physical significance is that if two neighboring nodes share many common neighbors with close connections, they have a higher probability of belonging to the same community. The state transition probability matrix incorporating both pheromone and heuristic information is given by Equation (4):

$$p_{ij} = \begin{cases} \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{j \in N(i)} \tau_{ij}^{\alpha}\eta_{ij}^{\beta}} & \text{if } j \in N(i) \\ 0 & \text{otherwise} \end{cases}$$

where α and β represent the intensification degrees of pheromone and heuristic information during iteration.

2.1 Ant Position Initialization

This step initializes ant positions (i.e., nodes) and their corresponding label lists. Network G is first sorted in descending order by node degree [**?**], after which common neighbors between nodes are computed and partitioned into initial communities sg. Label lists are generated based on node IDs within initial communities, and ant individuals are placed in nodes belonging to initial clusters.

Algorithm 2: Step 1 - Ant Position Initialization
Input: A complex network $G(V, E)$.
Output: Ant positions C_{node} and label lists $l(v.l)$ stored in each node v.

The algorithm proceeds by sorting all nodes by degree in descending order to obtain list L. For each node $v_i \in L$, if v_i is available and has degree ≥ 3, it forms the seed of an initial community sg. The algorithm then iteratively adds nodes that share common neighbors with existing members of sg until no more such nodes can be found. Finally, C_{node} consists of nodes in CN, while nodes not in C_{node} are initialized with their own IDs.

2.2 Ant Movement

Each ant transfers IDs between nodes based on transition probabilities. To increase the probability of ants traversing the entire abstract graph G, the movement rule is designed as follows: if a random number $r < 0.1$, the ant selects the neighbor with maximum probability as the next node to visit; otherwise, it randomly selects a neighbor other than the one with maximum transition probability. When multiple IDs share the same transition probability, one is randomly chosen.

At each iteration’ s end, ants move from their current node to the next node. Additionally, each ant selects one label from the current node’ s label list and stores it in the next node’ s label list. Pheromones are deposited on paths traversed by ants between different nodes, while pheromone evaporation occurs after each iteration. The ant movement strategy is outlined in Algorithm 3.

Algorithm 3: Step 2 - Ant Movement

```
While termination conditions are not met do
  For each ant do
    The ant moves from one node to the next, obtaining a label from the current node and preserving it in the next node’ s label list.
    The pheromone on this edge increases by $x$.
    If the pheromone value exceeds threshold $b$ then
      Set the pheromone value to $b$.
    End if
    After all ants have moved, evaporate pheromones on all edges by rate $\rho$.
  End for
  Recalculate the transition probability matrix.
End while
```

To prevent premature convergence to non-global optimal solutions, when pheromone values exceed the predetermined threshold b, edge pheromones are capped at this maximum value. This effectively prevents information on one path from becoming overwhelmingly dominant and eliminates scenarios where all ants concentrate on the same path.

2.3 Post-processing

After iteration termination, the frequency of each affiliation value in nodes’ label lists is counted. If a particular value appears with high frequency across the label list, the node has a higher probability of belonging to that label. To improve overlapping community detection accuracy, the algorithm can perform multiple scans on the target network to determine final community partitions. The post-processing steps are shown in Algorithm 4, where $N_v(l)$ represents the number of neighbor nodes of node v with label l, and sum denotes the total number of possible labels among node v' s neighbors.

Algorithm 4: Step 3 - Post-processing
Input: Nodes and their label lists.
Output: A set of overlapping communities C.

```
For each node $v$ do
  Obtain the node' s label from the most frequent label in its label list.
  Find node $v$' s neighbor nodes and record their label set $L$.
  For each label $l$ in $L$ do
    If $|N_v(l)|/sum > \theta$ then
      $l$ is a community to which node $v$ belongs.
    End if
  End for
End for
```

3 Experimental Results and Analysis

To validate the accuracy and analytical performance of the AntCBO algorithm for overlapping community detection, comparative experiments were conducted on both synthetic and real-world networks. The clique percolation method (CPM) and community overlap propagation algorithm (COPRA) were used as baseline algorithms for comparison with AntCBO, evaluating the superiority of each algorithm in overlapping community detection accuracy.

Parameter settings for the three algorithms are as follows: For CPM, $k = 3$ on synthetic datasets, while k varies from 3 to 10 on real-world datasets. For COPRA, $v = 8$ on synthetic datasets, while v varies between 1 and 9 on real-world datasets. AntCBO parameters are listed in Table 1 .

Table 1 Parameters of AntCBO Algorithm

Parameter	Value
Maximum iterations	100
Initial pheromone	1.0
Pheromone decay	0.1
Pheromone increment	0.05
Pheromone threshold	5.0

3.1 Synthetic Network Experiments

Synthetic network experiments employ the widely-used LFR (Lancichinetti-Fortunato-Radicchi) benchmark, which better implements power-law distributions for node degrees and community sizes, making it more representative of real social networks. This study generated four datasets using the LFR benchmark, with parameters detailed in Table 2 .

Table 2 Parameters of Synthetic Network

Dataset	Nodes	Community Size Range
LFR1	1000	10-50
LFR2	5000	10-50
LFR3	10000	10-50
LFR4	50000	10-50

3.2 Overlapping Community Identification Accuracy Analysis

Synthetic network experiments employ normalized mutual information (NMI) to compare the accuracy of the three methods in identifying overlapping communities. NMI characterizes similarity between detected partitions and ground-truth partitions, with values ranging from 0 to 1, where higher NMI indicates better partitioning results. The best NMI values for the three algorithms across three synthetic datasets are shown in Figure 1 [Figure 1: see original paper]. For partitions C' and C'', NMI values are determined by Equations (7) and (8):

$$NMI = 1 - \frac{1}{2}[H(X|Y) + H(Y|X)]$$

where X and Y represent random variables associated with partitions C' and C'', and $H(X|Y)$ is the normalized conditional entropy of nodes belonging to X given they belong to Y.

To verify algorithm universality, we first compare overlapping community detection performance across different algorithms on the same dataset. Figures 1(a)-(d) demonstrate that for varying network scales and community size ranges, as the number of community members O_m varies from 2 to 6, the proposed AntCBO algorithm consistently achieves higher NMI values than the other two algorithms on LFR1, LFR2, and LFR4 datasets. For the LFR3 dataset, although NMI values are lower than CPM when member counts are small, they surpass both comparison algorithms as member counts increase. This indicates that AntCBO exhibits strong detection performance for higher member counts, while maintaining relatively high identification capability even at lower member counts.

To verify algorithm robustness and analyze network scale impact on detection performance, we compare each algorithm’ s performance as network scale increases. Figures 2(a)-(c) show NMI value variations for the three algorithms on datasets LFR1-LFR4. As member counts and network scales increase, all three algorithms exhibit varying degrees of NMI decline. While AntCBO and COPRA show similar decline magnitudes (approximately 0.2), AntCBO' s NMI metric remains 30.9% and 83.3% higher than COPRA and CPM, respectively, on large-scale networks (i.e., LFR4 with $O_m = 6$), demonstrating superior accuracy for large-scale, high-member-count overlapping community detection.

3.3 Overlapping Community Detection Quantity Analysis

Further analysis examines AntCBO' s performance in detecting the number of overlapping communities. Figure 3 [Figure 3: see original paper] illustrates the comparison between detected overlapping community counts and ground-truth community counts across different algorithms. Figures 3(a)-(d) reveal that while all three algorithms adequately reflect true overlapping community counts on small networks with few members, AntCBO' s results align more closely with benchmark counts as network scale and member counts increase.

To conveniently analyze algorithm accuracy in identifying network overlapping communities, we define the $Fscore$ function to measure detection accuracy as the harmonic mean of precision and recall:

$$Fscore = \frac{2 \times precision \times recall}{precision + recall}$$

where $precision$ is the number of correctly detected overlapping nodes divided by the total number of detected overlapping nodes, and $recall$ is the number of correctly detected overlapping nodes divided by the true number of overlapping nodes.

Figure 4 [Figure 4: see original paper] shows overlapping community identification accuracy for different detection algorithms on the same dataset. The proposed AntCBO algorithm demonstrates better detection accuracy (higher $Fscore$ values) on LFR1 and LFR2, indicating accurate identification on smaller networks. On larger networks (LFR3 and LFR4), CPM achieves higher $Fscore$ values than AntCBO and COPRA, suggesting more accurate detection. However, AntCBO consistently outperforms COPRA across all datasets. Combined with analyses in Sections 3.1 and 3.2, although CPM shows high accuracy on large-scale, high-member-count networks, AntCBO' s detected community counts are closer to ground truth. Overall, AntCBO demonstrates stronger performance than the two comparison algorithms.

3.5 Real-World Social Networks

This section examines AntCBO' s performance on real-world social networks. Dataset parameters are summarized in Table 3 . Since ground-truth overlapping community structures are unknown for real-world networks, NMI cannot be used for performance measurement. Instead, experiments employ Q_{ov} to evaluate algorithm performance. Q_{ov} is a classical modularity extension for overlapping community detection that considers both the number of communities each vertex belongs to and the degree of community membership. Higher Q_{ov} values indicate better partitioning quality, with main formulas given in Equations (10)-(15).

Table 3 Digest of Real-World Social Networking

Dataset	Nodes	Edges	Average Degree	Communities
Karate[13]	34	78	4.59	2
Dolphins[14]	62	159	5.13	2
Football[15]	115	613	10.66	12

Experimental results comparing Q_{ov} values are presented in Table 4 . Analysis reveals that AntCBO outperforms both COPRA and CPM on Karate, Dolphins, and Football networks. Overall, AntCBO achieves higher Q_{ov} values than COPRA and CPM on most real-world network datasets, demonstrating superior performance.

Table 4 Values of Q_{ov} with Different Algorithms in Real-World Networks

Dataset	AntCBO	COPRA	CPM
Karate	0.42	0.38	0.35
Dolphins	0.51	0.47	0.45
Football	0.61	0.58	0.59

4 Conclusion

Building upon existing research on overlapping community detection algorithms, this paper proposes an ant colony-based overlapping community detection algorithm comprising parameter initialization, ant position initialization, ant movement, and post-processing stages. Experimental analysis on synthetic and real-world datasets validates the algorithm’ s performance in detail. Results demonstrate that compared to CPM and COPRA, the proposed ant colony-based overlapping community detection algorithm exhibits good performance in community detection quality, providing valuable reference for solving current overlapping community detection problems and understanding functional structures in networks.

References

[1] Wang Xiaofeng, Liu Gongshen, Li Jinhua. Overlapping community detection based on structural centrality in complex networks [J]. IEEE Access, 2017, 10

(5): 25258-25269.
[2] Lyzinski V, Tang M, Athreya A, et al. Community detection and classification in hierarchical stochastic blockmodels [J]. IEEE Trans on Network Science & Engineering, 2017, 4 (1): 13-26.
[3] Huang Hongcheng, Xiong Zhongyang, Hu Min, et al. Routing strategy based on change perception of node mobility characteristic in DTN [J]. Application Research of Computers, 2017, 34 (6): 1825-1829.
[4] Hou Linqing, Cai Ying, Fan Yanfang, et al. Interest community based message transmission scheme in mobile social networks [J]. Computer Science, 2018, 45 (6): 105-110.
[5] Chen Dongming, Yan Yanbing, Huang Xinyu, et al. Recommendation algorithm based on community detection in bipartite networks [J]. Journal of Northeastern University: Natural Science, 2018, 39 (8): 1103-1107.
[6] Xie Jierui, Kelley S, Szymanski B K. Overlapping community detection in networks: the state of the art and comparative study [J]. ACM Computing Surveys, 2011, 45 (4): 1-35.
[7] Sun Hanlin, Ma Sugang, Wang Zhongmin. A community detection algorithm using differential evolution [J]. Software Engineering, 2018, 21 (1): 1-6.
[8] Sun Hanlin, Ma Sugang, Wang Zhongmin. Self-organizing overlapping community structure analysis algorithm based on swarm intelligence [J]. Application Research of Computers, 2019, 36 (5), doi: 10.3969//j.issn.1001-3695.2017.11.0733.
[9] Li Peng, Li Yingle, Wang Kai, et al. Community detection based on user interaction and link analysis in social networks [J]. Computer Science, 2017, 44 (7): 197-202.
[10] Peng Yan, Xi Liya. Local-first detection of overlapping community and its evolution pattern in dynamic networks [J]. Computer Engineering, 2016, 42 (12): 188-195.
[11] Zhou Yang, Chen Xiaoyun, Cheng Jianjun, et al. Bisection spectral community-detection methods using optimal eigenvectors [J]. Journal of Frontiers of Computer Science & Technology, 2017, 11 (12): 1897-1906.
[12] Deng Chenggang, Li Guohui, Yang Hao, et al. Simulation resource allocation strategy based on guidance factor ant algorithms [J]. Journal of Huazhong University of Science and Technology: Natural Science, 2015, 1 (9): 1-6.
[13] Lu Yihong, Zhang Zhenning, Yang Xiong. Community structure detection algorithm based on nodes' eigenvectors [J]. Computer Science, 2017, 44 (6): 419-423.
[14] Allen S J, Pollock K H, Bouchet P J, et al. Preliminary estimates of the abundance and fidelity of dolphins associating with a demersal trawl fishery [J]. Scientific Reports, 2017, 7 (1): 4995-5002.
[15] Wang Yuying, He Wenkun, Shi Jiarong. Community detection algorithm based on community density [J]. Application Research of Computers, 2017, 34 (7): 1975-1977.

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