2020 28th Iranian Conference on Electrical Engineering (ICEE)

Overlapping Community Detection in Social Networks Using Cellular Learning Automata

Mohammad Mehdi Daliri Khomami
Department of Computer Engineering
Amirkabir University of Technology
Tehran, Iran
m.daliri@aut.ac.ir

Alireza Rezvanian
Department of Computer Engineering
University of Science and Culture
Tehran, Iran
rezvanian@usc.ac.ir

Ali Mohammad Saghiri
Department of Computer Engineering
Amirkabir University of Technology
Tehran, Iran
a_m_saghiri@aut.ac.ir

Mohammad Reza Meybodi
Department of Computer Engineering
Amirkabir University of Technology
Tehran, Iran
mmeybodi@aut.ac.ir

Abstract— **The community detection problem in social networks has been extensively investigated from different aspects. In this problem, it is attempted to divide a network into a group-set of similar nodes (communities), in which nodes inside a community structure reflect more similar functional characteristics or than that of other nodes outside the community. However, a typical characteristic of some real networks is that the presence of overlapping communities, where nodes can be a member of quite one community at an equivalent time. In this paper, an algorithm for overlapping community detection is proposed using cellular learning automata (CLA-OCD). In the CLA-OCD, a group of learning automata collaborates to find overlapping communities. The performance of the CLA-OCD is evaluated by several experiments, which is the design of both synthetic and real networks. Through experiments, the CLA-OCD achieves significant improvement in performance measures, including modularity, F-score, and normalized mutual information (NMI).**

Keywords— Social Network Analysis, Community Detection, Overlapping Community Detection, Cellular Learning Automaton, Learning Automaton

I. INTRODUCTION

Community Detection (CD) has been the topic that many researchers have been studied in the domain of Social Network Analysis (SNA). It is known as an important problem with applications in different domains, including Sociology, Biology, and Computer Science, whose data can be easily modeled by networks or graphs [1], [2]. One of the unsupervised methods in computer science for identifying the fundamental structure was classifying the most similar parts and called clustering. Hence, clustering can be interpreted as the process of classifying the vertices into clusters by considering the underlying structure [2]. The community detection goal is to find optimal groups of vertices, which are called communities based on the topology of the network. Since there is not a unique definition for presenting the concept of community formally, there are numerous definitions depends on the application are used in the literature. Therefore, different graph clustering algorithms have been applied to find communities such as Random Walks, Spectral Clustering, Modularity Maximization, and Statistical Mechanics, among others [1]–[4].

In the literature, communities are defined as dense sub-graph induced by a set of nodes, with higher edge density within the sub-graph and lower density of edges between them. Community detection is a valuable and significant function in the analysis of network structure and predicting the behaviors of networks. In order to deal with community detection, most previous studies for community detection have focused on finding disjoint communities. In disjoint community detection, the nodes are grouped into the disjoint sets with any intersection, and each set is densely connected. Unlike, in many domains, a node at the same time can belong to different communities. This characteristic is generally known as overlapping community detection. For example, people in many real social networks are usually members of different groups as their interests. Therefore, the concept of overlap indeed is known as a crucial characteristic of many real-world networks [5], [6].

In literature, there are several classifications for potential overlapping community detection algorithms. Based on [7], overlapping community detection algorithms are categorized into including clique percolation (CPM), eigenvector methods, ego network analysis, low-rank models, and local expansion algorithm. The CPM algorithm works based on finding overlap nodes with the fixed clique -sizes in the network [8]. Ego network analysis methods employed by structural holes [9] and using Ego network analysis are involved in calculating and combining. The eigenvector is extending to spectral analysis and utilize eigenvectors of the normalized Laplacian or modularity matrix to determine communities [10]. Optimization objectives and expansion methods are investigated in a local expansion algorithm. In OSLOM method [11], it evaluated the statistical significance of communities for a random configuration during community enlargement. The OSLOM method first selects a node randomly and in a greedy procedure, then develop the partition by checking either the developed partition is statically proper or not and finally reveals overlapping partitions and outliers in a network.

A CLA-based algorithm called CLA-Net is investigated for disjoint CD problem [12]. CLA-Net treats the resolution limit of modularity optimization as properly. As mention before, CLA-net is applied for disjoint CD and not able to reveal the overlapping communities.

This paper has focused on the overlapping community detection with the aid of cellular learning automata (CLA) called CLA-OCD. Briefly, The structure of the CLA-OCD consists of two main steps. In first, the network is treated as an ICLA by a one to one function from nodes to cells. Through interactions LAs by the environment, and an initial solution is formed, then CLA-OCD evolves the solution and gradually moves to find the near-optimal communities. In the second step, a belonging co-efficient is specified as the degree of vertices that belong to a community. This measure indicates which nodes are overlapping among communities. By this policy, the algorithm assigns nodes to more than a community.

The remainder of the paper is organized as follows: the next section theory of automata and its variations are introduced briefly. In section 3, the CLA-based algorithm for overlapping community detection (CLA-OCD) is presented. Section 4 gives details of the experimental analysis and their results. Finally, section 5 gives the concluding remarks.

II. THEORY OF AUTOMATA

In this section, we describe cellular automata (CA), learning automata (LA), cellular learning automaton (CLA), and irregular CLA.

A. CA

Cellular Automata models are interesting models in mathematics [13]. Theses model consists of a set of cells and also a structure for organizing the cells. In these models, several states are defined for the cells. Every cell selects a state of the other states. In each cell, the operation of state selection is performed locally. During this operation, each cell decides its states base on its local observation, which is the state of its neighboring cells.

B. LA

The theory of learning automata belongs to the primary models of reinforcement learning [14]. These model can be considered as an agent which operate in unknown stochastic environments. In these models, a set of actions is defined for the agent and also a set of feedback that is defined to represent the response of the environment to the actions of the agents. The agent using learning automata theory tries to find an appropriate action. The appropriate action is one of the actions from an action set that triggers the environment to generate reward signals asymptotically. Figure 1 shows the relationship between LA and its environment.

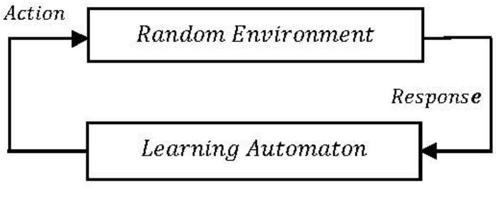

Figure 1. The schematic structure of the environment and LA

The structure of the learning automata used in this paper may be defined with 4-tuple $\{\alpha, \beta, P, T\}$ such that:

- $\alpha = \{\alpha_1, \dots, \alpha_r\}$ is the limited number of actions
- $\beta = \{\beta_1, \dots, \beta_m\}$ denotes the reinforcement signal
- $P = \{p_1, \dots, p_r\}$ is the probability vector
- $T[\alpha(n), \beta(n), p(n)]$ is the learning algorithm.

Formula (1) and (2) show an update in the vector of probability action in each iteration. Learning algorithm moderates the vector of the probability of action using formula (1) for promising responses as:

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1 - p_i(n)] \\ p_j(n+1) &= (1-a)p_j(n) \end{aligned} \tag{1}$$

and equation (2) for unfavorable ones:

$$\begin{aligned} p_i(n+1) &= (1-a)p_i(n) \\ p_j(n+1) &= \left(\frac{b}{r-1}\right) + (1-b)p_j(n) \end{aligned} \tag{2}$$

where $P(n)$ is the action probability vector at instant n. r is the number of actions that can be taken by the LA. The learning rates a and b denotes the reward and penalty parameters and determines the number of increases and decreases of the action probabilities, respectively [24, 25].

C. CLA

The theory of CLA *[15]* brings together the learning capabilities of LA and the distributed computation characteristics of the CA. The core operation of the learning algorithm in CLA can be expressed as follows: A d-dimensional CLA is a structure $CLA = (Z^d, \Phi, A, N, F)$, where $Z^d$is a matrix of d-tuples, Φ is a limited number of states, A is the number of learning automata, where every LA is placed the corresponding cell of cellular automata, $N = \{x_1, \dots, x_m\}$ is a limited number of of Z^d called a neighboring hood vector, where $x_i \in Z^d$ in the CA, and $F: \Phi_m \to \beta$ is the local rule of the CLA, in which β denotes the values of reinforcement.

D. ICLA

The Irregular CLA (ICLA) [16] is a new version of CLA, where the limitation on the regular structure in conventional CLA is eliminated. Many problems cannot be modeled by a regular grid structure. The promising results for the application of ICLA for network analysis and the detection of the community are given in [1], [17], [18].

III. CLA-OCD

In this section, we provide a new algorithm for the detection of overlapping communities based on the CLA called (CLA-OCD) in a social network. In the following, the different steps of the CLA-OCD algorithm are described.

Initially, the input network $G = \langle V, E \rangle$ is an undirected network where $V = \{v_1, v_2, \dots, v_n\}$ indicates a vertex set, and $E \subseteq V \times V$ is the link set that is given in the network. The proposed CLA-OCD algorithm consists of two general categories, including the Configuration Phase and the Problem-Solving Phase. We describe each category of the CLA-OCD algorithm in the following in detail.

A. Configuration phase

Configuration phase consists of two main steps, including solution representation and solution construction which is described as follows:

1) Solution Representation

In the network $G = (V, E)$, a community can be depicted by association vector $C = (c_1, c_2, \dots, c_n)$, where c_i pointed out to the index of the community. However, the main drawback of this association is that it demands former knowledge related to the size of communities. To overcome this issue, in CLA-OCD, we have used the locus-based adjacency representation [17]. In this representation, a solution is shown by identical vector $S = (s_1, \dots, s_n)$ which can be interpreted as s_i Furthermore, node i are in the identical community. We note that the identical vector-only represents the edges instead of the communities. Since we are looking for representation for communities, therefore, a process is needed to transform the identical vector into an association vector. To achieve this goal, a decoding mechanism is presented to show the communities, which works by running a linear-time search algorithm, either breadth first search or depth first search. At the termination of this process, the vector of identical is converted to the association vector that represents communities.

2) Solution construction

This phase, an asynchronous ICLA, is created by a one to one function from ICLA to the input graph. This function mapped each vertex to each cell of ICLA such that each cell is equipped with an LA. In ICLA, each cell has different states which determine the current action taken by the LA residing in it. Formally, LA_i exists in node i is defined by a 3-tuple (α, β, p), where:

- $\alpha_i = \{\alpha_{i1}, \dots, \alpha_{ir}\} = N(i)$ indicates the action set, and $N(i)$ corresponds to the vertex-set, which is neighboring to i in the network.
- $\beta_i = \{0,1\}$ indicates the response value corresponds to reward and a penalty from the environment.
- $P_i = (p_{i1}, p_{i2}, \dots, p_{ir})$ refer to the probability vector of actions where p_{ij} is the action selection probability for α_{ij} by LA_i.

According to the status of the cells, an identical vector S is constructed, in which current action is chosen by all LAs. Therefore, at cycle t, the solution can be described by equation (3).

$$S(t) = \{\alpha_1(t), \alpha_2(t), \dots, \alpha_n(t)\} \quad (3)$$

where $\alpha_i(t)$ indicates the action selected by LA_i at iteration t. By the evaluation of the CLA, the identical vector is updated. Figure 2 depicts the pseudo-code of the CLA-OCD.

The problem-solving phase consist of two main steps:

B. Learning phase and post-processing phase:

1) Learning Phase

In the learning phase, a solution vector has been constructed, and each LA selects its actions independently. After action selection is made, the process of decoding solution vector to association vector is started. We note that, in the learning phase, we obtain non-overlapping communities. In order to evaluate communities properly, the selected action by each LA should be evaluated. More precisely, if the chosen action by an LA is favorable, then the selected action is rewarded otherwise is penalized. To do this, we define new criteria called the coefficient of belonging, which compute the degree of a vertex that belongs to a community. More precisely, the belonging coefficient of a node determines the dependency of a node to a community, which defined as follows. A belonging coefficient for node i is indicated as $[b_{i1}, b_{i2}, \dots, b_{ik}]$, where b_{ic} satisfies two following conditions:

$$\sum_{c=1}^{|C|} b_{ic} = 1 \quad (4)$$

$$0 \le b_{ic} \le 1, \forall i \in V, \forall c \in C \quad (5)$$

in which $|C|$ is the number of communities and the belonging coefficient is b_{ic} is specified as:

$$b_{ic} = \frac{\sum_{j \in c} A_{ij}}{\sum_{j \in c} \sum_{j \in c} A_{ij}} \quad (6)$$

where A_{ij} is the matrix of adjacency that if two nodes i and j are incident then $A_{i,j} = 1$ otherwise $A_{i,j} = 0$. To reward or penalize of each automaton, we compute b_{ic} for each node in the networks. If b_{ic} is higher than $1/v$; the selected action by an automaton is rewarded otherwise is penalize. Moreover, we have used community definition by *Raghavan et al.* in [19] for the reward and penalize of LA. The learning phase is continued until the selected communities are fixed after some iteration. After learning phase is done the post processing phase is starts.

1) Post Processing Phase

This phase constitutes to compute the belonging coefficient based on the final community detection, which is assigned to the neighbors of the nodes in the network. After computing the belonging coefficient, we are ready to decide based on the largest value of the belonging coefficient. If the belonging coefficient for a specific node has the largest value and at the same time is greater than a pre-defined threshold, it indicates that the vertex belongs to only one community. Otherwise, if the belonging coefficient for a specific node has not the largest value and is higher $1/v$, then the node at the same time is a member of the communities that the belonging coefficient is higher than $1/v$. As such, each vertex may be a member of at last v communities in the networks.

Algorithm CLA-OCD;
Input: the network $G = (V, E)$
Step 1: Initialization
Repeat
Step 2: Each learning automaton LA_i chooses an action $\alpha_i(t)$ according to its action probability vector
Step 3: The solution vector $S(t)=(\alpha_1(t), \alpha_2(t), \dots, \alpha_n(t))$ is transferred into the association vector $C(t)=(c_1(t), c_2(t), \dots, c_n(t))$ to represent the obtained community structure through the decoding process.
Step 4: The belonging co-efficient of the community structure represented by the association vector C(t).
Step 5:
For Each learning automaton L_i **do**
  If $k_i(c_i(t)) \ge k_i(c'), \forall c' \ne c_i(t)$ & $W_i^{in}(C) > W_i^{out}(C), \forall i \in C$
    The response from the environments $\beta_i(t)=0$
  Else
    The response from the environments $\beta_i(t)=1$
  End if
End for
Step 6: For each learning automaton LA_i $(1 \le i \le n)$, given the chosen reward or penalize
Step 7: Update the action probability vector p_i of each learning automaton LA_i
Step 8: **Until** The obtained community structure remains fixed in some consecutive cycles

Figure 2. Pseudo-code of the CLA-OCD algorithm in the social network.

IV. SIMULATION RESULTS

To investigate the performance of the CLA-OCD, we have designed some experiments on the popular network called LFR [20] in addition to real networks. Table 1 shows the information on networks that are utilized for experiments. Also, CLA-OCD is compared with several algorithms, such as CPM [8], COPRA [21], ANTCBO [22], and GANET+ [23]. Besides, the performance of the CLA-OCD, we applied standard measures, namely Modularity and Normalized Mutual Information (NMI) [4].

Table I. Test networks employed for the experiments

Graphs	$\|N\|$	$\|E\|$	Type
Dolphins	62	159	Real
Email	1,133	5,451	Real
Football	115	613	Real
Jazz	198	2743	Real
Karate	34	78	Real
LFR 1	1000	-	Synthetic
Political Books	105	441	Real

A. Modularity

Modularity Q is a well-known measure for assessing a set of communities which achieves by the algorithms. Since Q is proposed for a non-overlapping algorithm, therefor several extensions are proposed to apply for overlapping community detection algorithms. One of the well-known extensions for this measure is proposed by *Nicosia et al.* [24] that makes a node be able to member of different communities at the same time. Since a node may belong to different communities, therefore it can use a belonging degree for vertices that define as follows

$$0 \le b_{i,c} \le 1 \ \forall_i \in V, \forall_c \in C \tag{7}$$

$$\sum_{c=1}^{|C|} b_{i,c} = 1 \tag{8}$$

where $b_{i,c}$ indicates the belonging degree of node i to community c. Besides, F is a combination of the two nodes belonging to community c and compute as:

$$\beta_{l,c} = F(\alpha_{i,c}, \alpha_{j,c}) \tag{9}$$

hence, overlapping Q is defined as follows:

$$Q_{ov} = \frac{1}{m} \sum_{c \in C} \sum_{i,j \in V} [A_{ij} * r_{ijc} - \frac{k_i k_j}{m} s_{ijc}] \tag{10}$$

where r_{ijc} and s_{ijc} compute as equation (11) and (12), respectively.

$$r_{ijc} = \beta_{l,c} = F(\alpha_{i,c}, \alpha_{j,c}) \tag{11}$$

$$s_{ijc} = \frac{\sum_{i \in V} F(\alpha_{i,c}, \alpha_{j,c})}{|V|} * \frac{\sum_{j \in V} F(\alpha_{i,c}, \alpha_{j,c})}{|V|} \tag{12}$$

B. Normalized Mutual Information

Normalized mutual information (NMI) is a metric which is evaluating how similar are the detected communities by the methods to the known communities. In NMI, A and B are two divisions of the graph. NMI returns a value in the range of [0, 1]. The value is 0, which means that the similarity between the actual community and the detected community does not equal at all. On the contrary, if the value is 1, it indicates that the actual community and detected community are matched entirely. The NMI formula is provided in Equation (13).

$$NMI(A,B) = \frac{-2 \sum_{a \in A} \sum_{b \in B} |a \cap b| \log(\frac{|a \cap b| n}{|a||b|})}{\sum_{a \in A} |a| \log\left(\frac{|a|}{n}\right) + \sum_{b \in B} |b| \log\left(\frac{|b|}{n}\right)} \tag{13}$$

The NMI is helpful when prior knowledge about the structure of communities exists. We note that the additional information is provided in [1].

C. Experiments

1) Performance evaluation

The experiment aims to evaluate the performance of different learning models of CLA, which is designed for this problem and select a proper model based on the obtained results for experiments. In order to do this experiment, we applied four different learning scheme including L_{R-I}, L_{R-p}, $L_{R-\epsilon p}$ and CP_{RP} [21]. We note that for simplicity Algorithm 1 for L_{R-I}, Algorithm 2 for L_{R-p}, Algorithm 3 for $L_{R-\epsilon p}$ and Algorithm 4 for CP_{RP} are applied. These algorithms are applied for real networks, and the result is demonstrated in Figure 3.

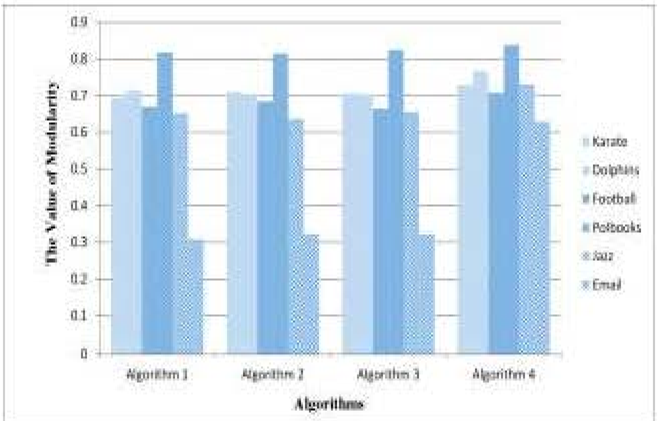

Figure 3. The results of different learning algorithms on CLA-OCD w.r.t. Modularity.

According to the obtained results, one can say that the 4th algorithm is outperforming from algorithm 1, algorithm 2, and algorithm three regarding the modularity measure.

2) Impact of the learning parameter

The policy behind this experiment shows the effect of the learning rate of the LAs in the CLA-OCD. To do this, we examine the different learning rates, which alters from 0 to 0.1 with step length 0.01, and the NMI result is reported for LFR synthetic network in figure 4.

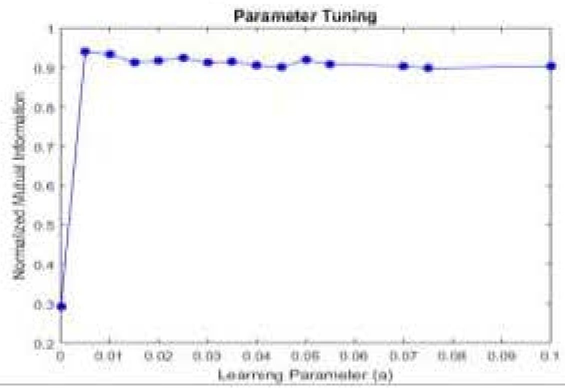

Figure 4. NMI for different rates of learning parameter

As depicted in figure 4, the most favorable outcome is achieved when the learning rate is 0.05, and the value near 0.05 leads to proper results. Since the best result is obtained by 0.05, we then, for the further experiment, we set the rate of learning value to 0.05.

3) Impact of the overlapping parameter

We examine the impact of v on the value of overlapping community detection. For this aim, we test different values for different values of v, and the best and average results are reported for real networks. We note that these results are reported 30 times for average. These results are given in figure 4 and figure 5.

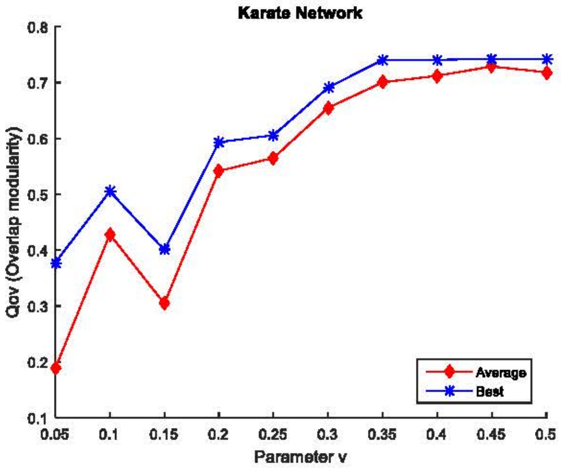

Figure 5. The modularity for different value of v for karate

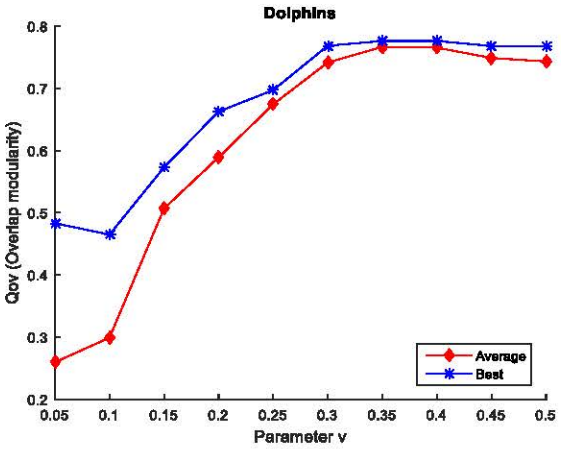

Figure 6. The modularity for different value of v for dolphins

As can be seen from figure 5, for the graph of the best value for v when between $0.35 \leq v \leq 0.5$. The highest value of v is 0.45, which gains 0.7286 for the overlapping modularity, and for other networks, the best value of v is 0.35, which leads to 0.7666 for modularity. For other networks, the same result may conclude.

4) Comparing algorithms

This section provides the comparison of the CLA-OCD with CPM [8], COPRA [21], ANTCBO [22], and GANET+ [23] in terms of NMI. The result of this experiment is depicted in figure 7, where the vertical axis indicates the NMI and horizontal axis indicate overlapping modularity (OM).

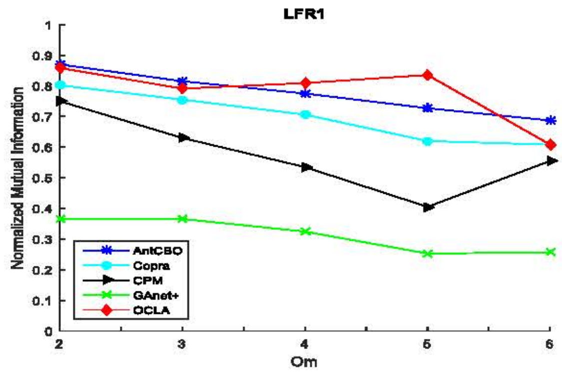

Figure 7 NMI comparison for different OM values.

The result, which is demonstrated in figure 7, shows the CLA-OCD is a more significant performance from other algorithms and gain higher NMI concerning other algorithms.

V. Conclusion

This paper presents a novel algorithm using CLA called CLA-OCD for overlapping community detection. The CLA-OCD comprises two phases, such as the configuration and the problem-solving phase, respectively. In the configuration phase, primarily initialize network configuration by LA's and solution representation is constructed. In the problem-solving phase, the algorithm reveals communities with overlapping structures by learning mechanisms. The CLA-OCD was simulation experiments evaluated against the popular algorithms for the overlapping community. The results of CLA-OCD confirm the superiorities of the CLA-OCD regarding the normalized mutual information and modularity.

References

[1] A. Rezvanian, B. Moradabadi, M. Ghavipour, M. M. Daliri Khomami, and M. R. Meybodi, "Social Community Detection," in *Learning Automata Approach for Social Networks*, Cham: Springer International Publishing, 2019, pp. 151–168.

[2] S. E. Schaeffer, "Graph clustering," *Computer science review*, vol. 1, no. 1, pp. 27–64, 2007.

[3] R. Hosseini and A. Rezvanian, "AntLP: ant-based label propagation algorithm for community detection in social networks," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 1, pp. 34–41, 2020.

[4] A. Rezvanian, B. Moradabadi, M. Ghavipour, M. M. D. Khomami, and M. R. Meybodi, *Learning Automata Approach for Social Networks*. Springer, 2019.

[5] O. R. B. Speily, A. Rezvanian, A. Ghasemzadeh, A. M. Saghiri, and S. M. Vahidipour, "Lurkers Versus Posters: Investigation of the Participation Behaviors in Online Learning Communities," in *Educational Networking*, Springer, 2020, pp. 269–298.

[6] L. Huang, F. Liu, and Y. Zhang, "Overlapping Community Discovery for Identifying Key Research Themes," *IEEE Transactions on Engineering Management*, vol. doi: 10.1109/TEM.2020.2972639, no. in-press, 2020.

[7] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: The state-of-the-art and comparative study," *ACM Computing Surveys (csur)*, vol. 45, no. 4, p. 43, 2013.

[8] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, pp. 9–18, 2005.

[9] J. Leskovec and J. J. Mcauley, "Learning to discover social circles in ego networks," in *Advances in neural information processing systems*, 2012, pp. 539–547, Accessed: Sep. 20, 2014. [Online].

[10] S. Zhang, R.-S. Wang, and X.-S. Zhang, "Identification of overlapping community structure in complex networks using fuzzy c-means clustering," *Physica A: Statistical Mechanics and its Applications*, vol. 374, no. 1, pp. 483–490, 2007.

[11] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato, "Finding statistically significant communities in networks," *PloS one*, vol. 6, no. 4, 2011.

[12] Y. Zhao, W. Jiang, S. Li, Y. Ma, G. Su, and X. Lin, "A cellular learning automata based algorithm for detecting community structure in complex networks," *Neurocomputing*, vol. 151, pp. 1216–1226, 2015.

[13] S. Wolfram, *Cellular automata and complexity: collected papers*, vol. 152. Addison-Wesley Reading, 1994.

[14] A. Rezvanian, A. M. Saghiri, S. M. Vahidipour, M. Esnaashari, and M. R. Meybodi, *Recent Advances in Learning Automata*. Springer, 2018.

[15] A. Rezvanian, A. M. Saghiri, S. M. Vahidipour, M. Esnaashari, and M. R. Meybodi, "Cellular Learning Automata," in *Recent Advances in Learning Automata*, Springer, 2018, pp. 21–88.

[16] M. Esnaashari and M. R. Meybodi, "Irregular cellular learning automata," *IEEE transactions on cybernetics*, vol. 45, no. 8, pp. 1622–1632, 2015.

[17] M. M. D. Khomami, A. Rezvanian, and M. R. Meybodi, "A New Cellular Learning Automata-based Algorithm for Community Detection in Complex Social Networks," *Journal of Computational Science*, vol. 24, pp. 413–426, 2018.

[18] M. R. Mirsaleh and M. R. Meybodi, "A Michigan memetic algorithm for solving the community detection problem in complex network," *Neurocomputing*, vol. 214, pp. 535–545, 2016.

[19] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, p. 036106, 2007.

[20] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E*, vol. 78, no. 4, p. 046110, 2008.

[21] S. Gregory, "Finding overlapping communities in networks by label propagation," *New Journal of Physics*, vol. 12, no. 10, p. 103018, 2010.

[22] X. Zhou, Y. Liu, J. Zhang, T. Liu, and D. Zhang, "An ant colony based algorithm for overlapping community detection in complex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 427, pp. 289–301, 2015.

[23] C. Pizzuti, "Overlapped community detection in complex networks," in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, 2009, pp. 859–866.

[24] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri, "Extending the definition of modularity to directed graphs with overlapping communities," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2009, no. 03, p. P03024, 2009.