scientific reports

OPEN

Detecting overlapping communities in complex networks using non-cooperative games

Farhad Ferdowsi✉ & Keivan Aghababaei Samani

Detecting communities in complex networks is of paramount importance, and its wide range of real-life applications in various areas has caused a lot of attention to be paid to it, and many efforts have been made to have efficient and accurate algorithms for this purpose. In this paper, we proposed a non-cooperative game theoretic-based algorithm that is able to detect overlapping communities. In this algorithm, nodes are regarded as players, and communities are assumed to be groups of players with similar strategies. Our two-phase algorithm detects communities and the overlapping nodes in separate phases that, while increasing the accuracy, especially in detecting overlapping nodes, brings about higher algorithm speed. Moreover, there is no need for setting parameters regarding the size or number of communities, and the absence of any stochastic process caused this algorithm to be stable. By appropriately adjusting stop criteria, our algorithm can be categorized among those with linear time complexity, making it highly scalable for large networks. Experiments on synthetic and real-world networks demonstrate our algorithm's good performance compared to similar algorithms in terms of detected overlapping nodes, detected communities size distribution, modularity, and normalized mutual information.

Complex relationships between components existing in society, technology, biology, economy, and other various fields, in many cases, can be modeled as complex networks by regarding components as nodes and relationships as edges[1]. As a consequence, all of the tools available for complex networks analysis could be applied to extract valuable information about the under investigation system. An important consideration of network structures is the possibility of classifying nodes into groups or communities[2]. Indeed, it is observed that many real-world networks have a community structure[3]. In a network, It is a crucial issue how to define communities, and its definition has to be problem-driven. Defining communities in complex networks is a multi-faceted issue that has been addressed and discussed in many studies[4]. However, according to its general definition, In a network, community refers to a group of nodes that are densely connected internally and have a sparser connection with the rest of the network[3]. Detecting communities is of great importance since nodes in a community usually have similarities in function, property, and characteristics[5]. For instance, community detection in the network of protein-protein interaction could reveal groups of closely connected proteins that possess an identical function in the body[6]. The discovery of community structure can be constructive in many fields, such as drug discovery[7], precision marketing[8], brain neural network[9], online social interaction analysis[10], and public opinion analysis[11]. Network communities typically can be categorized into two types. Disjoint communities with no shared members (also called non-overlapping communities or partitions) and overlapping communities with shared members (also called covers). Examples of overlapping communities are widely seen in the real world. Researchers, based on their various research interests or multiple affiliations, can be a member of more than one research group, or a gene can be involved in causing various diseases[12]. As a result, it is crucial to design community detection algorithms that be able to identify overlapping nodes. In recent years, a variety of approaches, including greedy algorithms based on modularity optimization[3,13], label propagation algorithms (LPA)[14], Stochastic block models[15], and Edge betweenness algorithms, have been employed for this purpose[13].

The use of game theory in this context was initialized by Athey and Jha in 2006 to model an organization's workers interaction[16] and followed by a game theoretic-based algorithm proposed by Chen in 2010[17,18]. A comprehensive discussion of game theoretic-based methods for detecting community structure in networks is provided in a survey done by Jonnalagadda and Kuppusamy[18]. However, the number of game theoretic based algorithms proposed in the last decay is not very large, and most of them are not scalable for large networks[12]. Community detection algorithms using the game theory are typically based on cooperative or non-cooperative

Department of Physics, Isfahan University of Technology, Isfahan 84156-83111, Iran. ✉email: f.ferdowsi@alumni.iut.ac.ir

games. Our proposed algorithm is based on the non-cooperative game in which nodes are assumed as rational selfish players who decide to be part of the communities which bring them the most profit. Although our algorithm is designed to detect overlapping communities, in contrast with similar algorithms, nodes are not allowed to be part of multiple communities before the exact boundaries of communities are determined (phase one), and overlapping nodes are identified in phase two. Such two phases algorithm not only increase accuracy but also, along with the appropriate stop criterion used in current work, speed up convergence. Moreover, in the present work, players have only local interactions, which leads the algorithm to be more effective than some other game theoretic-based algorithms in which interaction with all nodes is considered in the utility function[17,19,20]. The remainder of this paper is organized as follows. In the next section, the framework of the proposed algorithm and related definitions are given, and it is followed by a discussion on the time complexity of the algorithm. Afterward, the experimental results of our algorithm and its comparison with some other state-of-the-art algorithms are given. Finally, the concluding remark is stated.

Proposed algorithm

The proposed algorithm consists of two phases. The non-cooperative game is the basis of the first phase leading to non-overlapping community detection, while in the second phase, the overlap of the communities is determined. The game-theoretic framework is based on considering each node as a selfish agent trying to maximize its payoff by choosing different strategies, and each agent's choice can influence the other ones'. Strategy is a term in game theory that in the current context, refers to the communities in which the agent wants to participate. Based on this, each agent's strategy s_i is actually a list of community labels it is a member of, and the strategy profile of all agents is defined as $S = (s_1, s_2, \ldots, s_n)$. As stated, each agent aims to maximize its payoff, which for the agent v_i is represented through a utility function defined as follows.

$$U(S_{-i}, s_i) = \sum_{a_{ij} \neq 1} (1 + sim_{ij}) \frac{|s_i \cap s_j|}{\sqrt{|s_j|}} \tag{1}$$

Where a_{ij} is the adjacency matrix element; s_i is the strategy of agent v_i, and S_{-i} is the strategy profile of all agents but her; $|s_i \cap s_j|$ is the number of common labels between agent v_i and v_j; and $|s_j|$ is the number of communities agent v_j belongs to. Unlike some other game-theoretic overlapping community detection algorithms, in phase one agents are not allowed to acquire multiple labels and consequently, expression $\frac{(|s_i \cap s_j|)}{(|s_j|)}$ can only have two values of 0 or 1. Also, in this phase agents are only allowed to do switch operation among different community labels. In utility function, sim_{ij} is the similarity between agents v_i and v_j, which can be calculated through different available metrics as follows.

$$Jaccard\ coefficient(JC) : sim_{ij}^{JC} = \frac{|\Gamma_i \cap \Gamma_j|}{|\Gamma_i \cup \Gamma_j|} \tag{2}$$

$$Saltin\ index\ (SI) : sim_{ij}^{SI} = \frac{|\Gamma_i \cap \Gamma_j|}{\sqrt{|\Gamma_i||\Gamma_j|}} \tag{3}$$

$$Sorensen\ index\ (SO) : sim_{ij}^{SO} = \frac{2|\Gamma_i \cap \Gamma_j|}{|\Gamma_i| + |\Gamma_j|} \tag{4}$$

$$Hub\ promoted\ index\ (HP) : sim_{ij}^{HP} = \frac{|\Gamma_i \cap \Gamma_j|}{min(|\Gamma_i|, |\Gamma_j|)} \tag{5}$$

$$Hub\ depressed\ index\ (HD) : sim_{ij}^{HD} = \frac{|\Gamma_i \cap \Gamma_j|}{max(|\Gamma_i|, |\Gamma_j|)} \tag{6}$$

Where Γ_i and Γ_j denote the neighbors of agents v_i and v_j , respectively. The proposed algorithm results do not significantly depend on different similarity metrics except for a few special cases. However, represented results have been obtained using HP similarity, which slightly performs better than other similarity metrics. The algorithm starts with an initial condition in which each agent v_i is assigned to a singleton community c_i. Next, in each iteration, all agents, by order of their degrees, update their strategy by imitating their neighbors with the aim of maximizing payoff. For more clarity, the phase one framework is given in Algorithm 1 in Fig. 1.

Lines 4 to 23 repeat until the stop criterion is met and finally agents with the same label belong to the same community. The stop criterion should be defined in a way that satisfies accuracy and efficiency at the same time. In the proposed algorithm, there are some cases in which some agents' strategy fluctuates permanently, and some other agents need too many iterations to reach their stable one. Since a minimal number of agents often fall in such category, defining a stop criterion that ignores such agents' stability could speed up the algorithm without significant loss of accuracy. For this reason, instead of waiting for all agents' strategy to be fixed, the stop criterion is satisfied as soon as the number of agents with a fixed strategy does not increase more than a specific value. This value in each iteration Δ_{stop} is defined as a fraction of fixed agents number n_{fixed} in the previous iteration.

$$\Delta_{stop} = \epsilon . n_{fixed} \tag{7}$$

Algorithm 1 phase one

```
Input : A Network G = ( V , E )
Output : single strategy profile of all agents S
making initial partition : assign different label to each agent resulting singleton partitions.
iter ← 0
Δ ← 0
while $\Delta \le \Delta_{stop}$ and iter < 2 do
    iter ← iter + 1
    $n_{fixed}$ [iter] ← n
    for agent $v_i$ in G do
        U0 ← Utility($v_i, s_i$)
        for each label $l$ available in the vertex $v_i$ neighborhood do
            if Utility($v_i, \{l\}$) > U0 then
                $s_{new} \leftarrow \{l\}$
            end if
        end for
        if $s_{new} \ne s_i$ then
            $s_i \leftarrow s_{new}$
            $n_{fixed}$ [iter] ← $n_{fixed}$ [iter] - 1
        end if
    end for
    if iter > 2 then
        $\Delta \leftarrow | n_{fixed}[iter] - n_{fixed}[iter-1] |$
        $\Delta_{stop} \leftarrow \alpha \,.\, n_{fixed}[iter-1]$
    end if
end while
```

Algorithm 2 phase two

```
Input : A Network G = ( V , E ) and non overlapping strategy profile S
Output : overlapping strategy profile $\bar{S}$
for agent $v_i$ in G do
    for each label $l$ available in the vertex $v_i$ neighborhood do
        add Utility($v_i, \{l\}$) to U_list[$v_i$]
    end for
    max_U[$v_i$] = max(U_list[$v_i$] )
    for each label $l$ available in the vertex $v_i$ neighborhood do
        add Utility($v_i$, {l}) / max_U[$v_i$] to Normalized_U_list[$v_i$]
    end for
    Thresh_list [$v_i$] = RMS(Normalized_U_list[$v_i$])
end for
for agent $v_i$ in G do
    for each label $l$ available in the vertex $v_i$ neighborhood do
        if > Thresh_list [$v_i$] then
        if Utility($v_i$, {l}) / max_U[$v_i$] > Thresh_list [$v_i$] then
            add $l$ to $\bar{S}[v_i]$
        end if
    end for
end for
```

Figure 1. Phase one and Phase two framework algorithms.

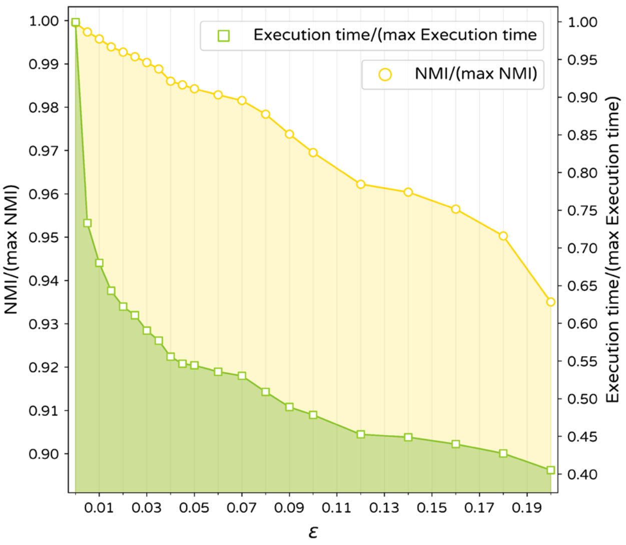

Figure 2. Effect of ϵ value on algorithm accuracy and elapsed time.

By adjusting ϵ value, a balance between accuracy and efficiency can be obtained. Variation of relative phase one execution time (execution time divided by longest execution time) and relative NMI (Obtained NMI divided by the best achievable NMI) obtained for LFR synthetic networks is represented in Fig. 2. According to the results, nonzero but small values of ϵ such as 0.005 and 0.01 can reduce phase one elapsed time while giving acceptable accuracy. The effect of ϵ value on the scalability of the algorithm will be discussed more in the algorithm complexity section.

Phase two is responsible for finding overlapping nodes. In some non-cooperative game-theoretic algorithms, a *loss function* is used as a method for controlling multiple memberships of agents[17,19–21]. In such a method multiple membership criteria usually are defined in a way that is similar for all nodes in spite of different conditions they may have. Moreover, in some other algorithms like[22,23], the manually defined threshold is responsible for determining multiple memberships of nodes. Nevertheless, in our algorithm, this criterion is defined uniquely for each agent based on payoffs it acquires from membership in different communities. Accordingly, phase two

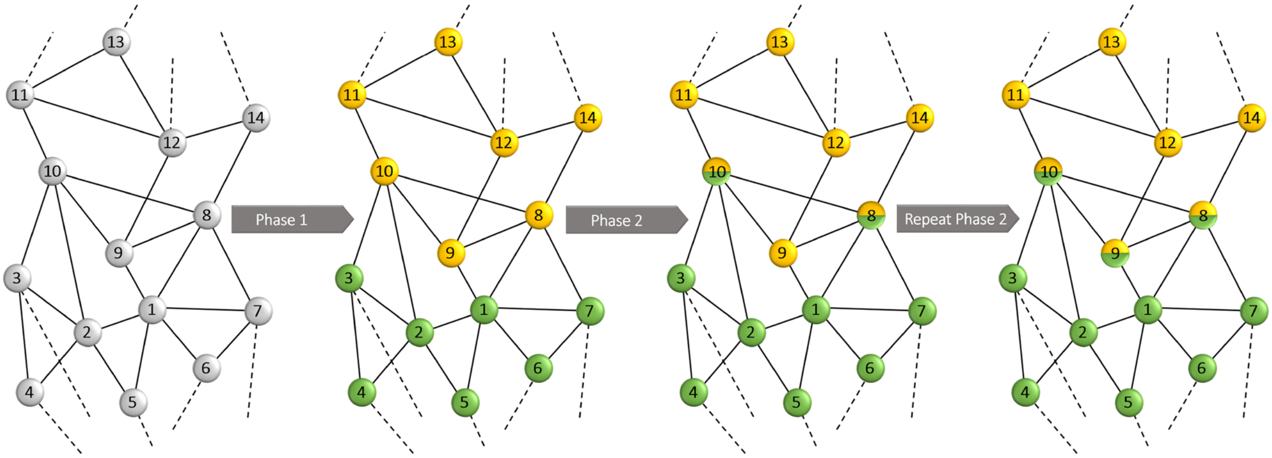

Figure 3. The toy example to illustrate the performance of each phase.

contains two stages. In the first stage in which payoff thresholds are calculated, the following operations should be done for each agent:

1. Calculating payoffs that agent acquires by Adopting any of community label available in its neighborhood.
2. Normalizing all obtained payoffs with respect to maximum payoff the agent has obtained.
3. Finding payoff threshold for the agent by calculating root mean square of normalized payoff values obtained for that agent.

In the following stage, each agent adds community labels that have a payoff above her payoff threshold value to her strategy. The framework of phase two is given in Algorithm 2 in Fig. 1.

Finally, each agent belongs to all communities which those labels exist in its strategy list. In networks with a high degree of overlap, it is very probable for overlapping nodes to be connected with other overlapping nodes. In such cases, repetition of phase two can help discover overlapping nodes more reliably. For more illustration, a toy model representing community structure before and after applying each phase is shown in Fig. 3.

It should be noted that described phase two returns crisp communities with binary membership coefficient of nodes in different communities. Although often it is the desirable form, sometimes the fuzzy communities are more suitable for the intended use. In such cases, the normalized payoff values of each agent are representative of that agent's fuzzy membership coefficients.

Time complexity of algorithm

The proposed algorithm consists of three parts. The first one is initialization which requires $O(n)$, where n is the total number of nodes. In phase one, the outer loop continues until stop criterion satisfaction. In inner loops, for each agent, the payoff should be calculated for all labels in the neighborhood, which is maximally equal to the number of the agent's neighbors. Therefore, phase one requires $O(T.n.K)$ on average, where K is the average degree and T is the maximum iteration. In some other algorithms, T is defined manually. In the proposed algorithm, although the maximum iteration number is determined dynamically based on stop criterion satisfaction, it does not depend on n or the total number of edges m if the network topology is kept the same and if the ϵ value is selected appropriately. For LFR synthetic networks, the variation of the maximum iteration number for three small values of ϵ were calculated as a function of n and m (Fig. 4a,b). As it can be seen, especially for small nonzero values of ϵ the maximum iteration number does not depend considerably on n or m. Phase two has a similar calculation structure as the inner part of the phase one algorithm. Considering the second phase repeats two times, it requires $O(2n)$. Therefore, the time complexity of the entire algorithm is $O(n)$ in sparse networks and $O(m)$ in arbitrary ones. For a naive implementation of the algorithm, Fig. 4c,d shows the execution time for LFR synthetic networks. As it can be seen, for ϵ value of 0.01, the execution time is just slightly slower than linear growth.

Experimental results and comparison

With the aim of evaluating our proposed algorithm performance, we compare it with some other algorithms named GAME1[17], GAME2[24], GAME3[25], SLPA[22], OSLOM[26], CPM[27], GCE[28] and LFM[29]. GAME1 is based on non-cooperative game theory with the time complexity of $O(m^2)$. GAME 2 and GAME3 are based on cooperative game theory with the time complexity of $O(n^2)$ and $O(n.log(n)) + O(n.k_{max})$, respectively ($k_{max}$ is graph maximum degree). Our algorithm results in this section are obtained by set ϵ value to 0.01. Other algorithms' results are extracted from those algorithms' original papers or comparative study papers[30]. In these papers for algorithms with tunable parameters, it is stated that the results with the best setting are reported.

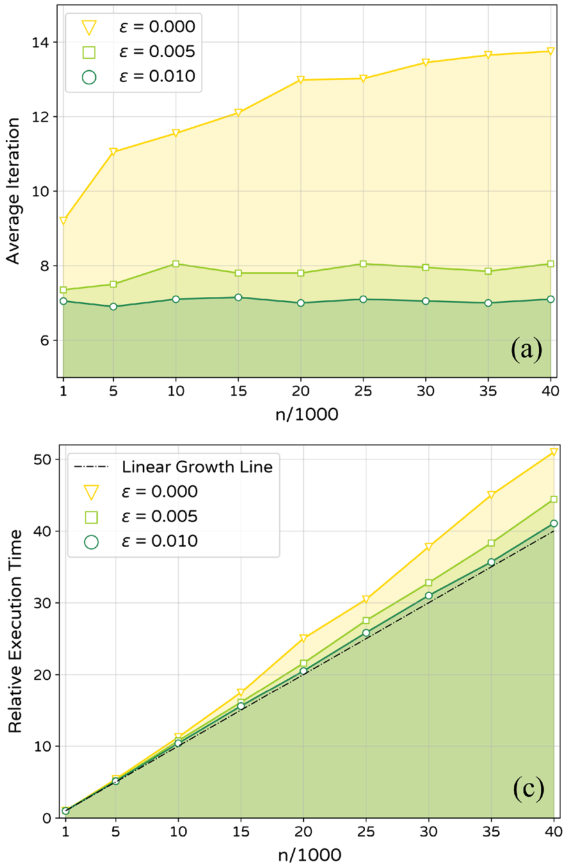

Figure 4. (**a**, **b**) Phase one average max iteration as a function of n and m. (**c**, **d**) Entire relative algorithm execution time as a function of n and m. Results of a and c is obtained for LFR networks with $\overline{k} = 10$. Results of b and d is obtained for LFR networks with $n = 5000$.

Evaluation criteria. There are various metrics in order to evaluate obtained results of algorithms, and it is often challenging since no canonical solutions are available[31]. A comprehensive discussion about the relationship between the topological properties of the community structure and the alternative evaluation measures and reliability of different evaluation criteria has been addressed in many studies[32]. In the first place, choosing appropriate evaluation criteria depends on whether there is known ground truth for the examined network. In the cases with known ground truth, different evaluation measures, including Average F1 score (AvgF1)[33], Adjusted Rand Index (ARI), which ensures that the value of random clustering is close to zero, Omega Index[34], which is the overlapping version of ARI[30] and adopts the number of clusters that each pair of nodes shares, to compare the detected communities versus ground truth communities, and Normalized Mutual Information (NMI)[35], derived from information theory, are widely used. In the current work, we used AvgF1 and an extended version of NMI, which is appropriate for comparison of two overlapping community structures[29]. The closer value of NMI or AvgF1 to 1, the more similar the detected community structure to ground truth; and the 0 value indicates the least similarity.

When it comes to testing the performance of overlapping community detection algorithms, especially when the ground truth of communities is unknown, the Q_{ov} is a well-known and frequently used metric[36]. It is an extension of the classical modularity, and the higher value of this means the better-detected communities. For directed networks this metric is defined as follows:

$$Q_{ov} = \frac{1}{m}\sum_{c\in C}\sum_{i,j\in V}\left[\beta_{l(i,j),c}A_{j,j} - \frac{\beta^{out}_{l(i,j)}k^{out}_i\beta^{in}_{l(i,j)}k^{in}_j}{m}\right] \tag{8}$$

By applying minor changes as follows, it can be used for undirected networks:

$$Q_{ov} = \frac{1}{2m}\sum_{c\in C}\sum_{i,j\in V}\left[\beta_{l(i,j),c}A_{j,j} - \frac{\beta'_{l(i,j)}k_i\beta'_{l(i,j)}k_j}{2m}\right] \tag{9}$$

The components of this equation is given by:

	n	$\bar{k}$	k_{max}	μ	$\tau 1$	$\tau 2$	Min Community Size (minC)	Max Community Size (maxC)	Om	On
LFR1	1000	10	50	0.1	2	1	20	100	2-6	0.1
LFR2	1000	10	50	0.3	2	1	20	100	2-6	0.1
LFR3	5000	10	50	0.3	2	1	20	100	2-6	0.1
LFR4	5000	10	50	0.3	2	1	20	100	2-6	0.5
LFR5	5000	10	50	0.3	2	1	10	50	2-6	0.1
LFR6	5000	10	50	0.3	2	1	10	50	2-6	0.5

Figure 5. LFR synthetic networks used for performance tests.

$$\beta'_{l(i,j)} = \beta^{out}_{l(i,j)} = \beta^{in}_{l(i,j)} = \frac{\sum_{i\in V} F(\alpha_{i,c}, \alpha_{j,c})}{|V|} \tag{10}$$

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

$$k_i^{out} = k_i^{in} = k_i \tag{12}$$

$$F(\alpha_{i,c}, \alpha_{j,c}) = \frac{1}{(1 + e^{f(\alpha_{i,c})})(1 + e^{f(\alpha_{j,c})})} \tag{13}$$

$$f(x) = 2px - p, p \in \mathbb{R} \tag{14}$$

where $\alpha_{(i,c)}$ is the belonging coefficient of node i to community c and p in $f(x)$ is an arbitrary value that in the current study is set to 30.

Synthetic networks. One of the most famous benchmark networks is synthetic networks called LFR which can be generated by the method proposed by Lancichinetti and Fortunato[37]. While in real-world networks, degree correlation among nodes is clearly nonzero, and the transitivity is relatively high, networks generated by LFR method have near-zero degree correlation and low transitivity[38–40]. Despite this drawback and some other limitations of LFR method, these networks still exhibit relatively very high realistic properties, and considering a large amount of experimental data available from the test of other algorithms on them, LFR networks are among the most proper choices for community detection algorithms performance test. In the networks made by this method 10 parameters are adjustable. By setting these parameters, we generated 6 groups of LFR networks for the performance tests, as shown in Fig. 5. The mixing parameter μ refers to the fraction of links through which a node connects to other nodes in other communities; $k_i^{in} = (1 - \mu)k_i$. τ_1 and τ_2 are exponents of power-law distribution of node degrees and community sizes, respectively. Furthermore, overlapping features of LFR network are controlled by *Om* (the number of communities to which each overlapping node belongs) and *On* (the fraction of nodes that belongs to more than one community). It should be noted that for our algorithm performance test on LFR networks, we have reported averaged results of runs over at least 10 instantiations of these networks for each parameter set.

The NMI values for results obtained using our proposed and other algorithms are represented in Fig. 6. As expected, by increasing *Om,* the NMI values gradually decrease. However, it is observed that in most cases, our algorithm outperforms others, especially in synthetic networks with smaller community sizes and more overlapping nodes.

When it comes to networks with overlapping communities, evaluation of a community detection algorithm performance must include checking the number of identified overlapping nodes, which is one of the important parameters determining the algorithm's accuracy. Overlapping nodes play a crucial role in real-world social networks considering the fact they usually act as bridges or messengers between communities[30]. Identified On detected by proposed and other algorithms for two groups of LFRs with ground truth On of 0.1 and 0.5 are shown in Fig. 7. Overlapping nodes identified by our algorithm increase gradually by the increase of *Om*. This trend is in contrast with other algorithms except SLPA in LFR3 network.

Aiming to find more comprehensive insight into algorithms performance, it would be beneficial to investigate the distribution of detected community sizes (*CS*). For this purpose, we used algorithms results on LFR3 averaging on all values *Om* and 10 instantiations of these networks. In the histogram of community sizes which is shown in Fig. 8, small fluctuations were omitted by representing fitted curves instead of raw data. For comparison, the ground truth power law distribution is visible in each histogram. Except for ours and SLPA algorithms, other algorithms have remarkable weaknesses in detecting larger size communities. Besides, some algorithms tend to break communities into smaller parts that cause distribution concentration in the range of small communities which do not exist in real distribution. Although such miss clustering occurs to some extent by our algorithm, it is not as much as some other algorithms such as GAME1, LFM, and especially CPM and OSLOM. Particularly, results demonstrate the relatively better performance of our algorithm in detecting larger communities.

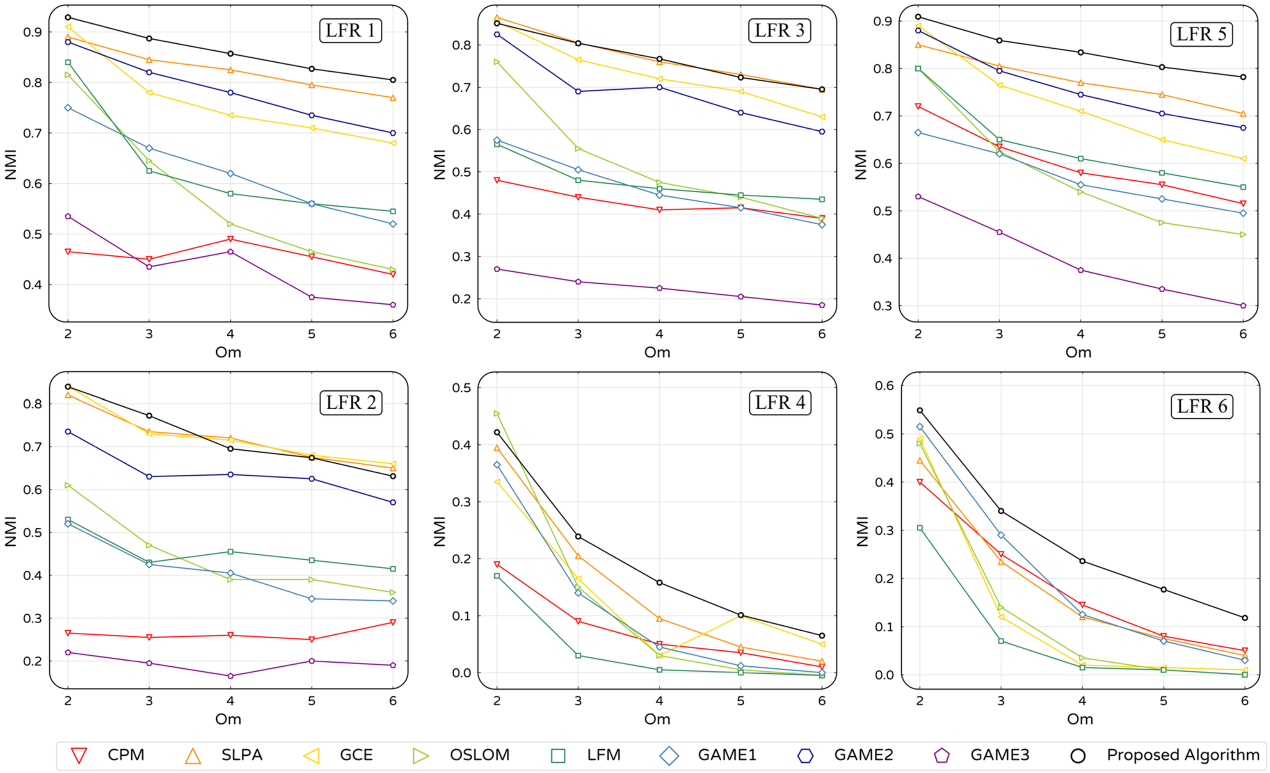

Figure 6. Comparative NMI value for proposed and other algorithms on LFR synthetic networks listed in Fig. 5.

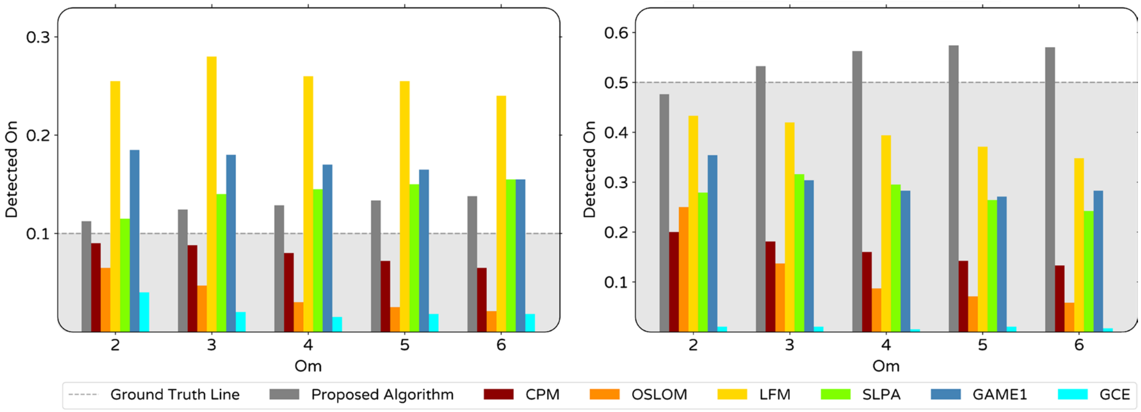

Figure 7. Overlapping nodes fraction detected by proposed and other algorithms in LFR3 and LFR4.

Real networks. In order to further evaluation of the proposed algorithm, we tested its performance on some real-world networks. Eight real networks have been chosen for this test, and their description can be observed in Fig. 9 (Data for the last three larger networks are available at http://snap.stanford.edu). As an evaluation measure, for the first six networks and for the last two ones, the overlapping modularity and AvgF1 score were used, respectively.

Stack bar chart of Q_{ov} for obtained community structure of first six networks by ours and other algorithms are shown in Fig. 10. Such illustration makes us able to compare the overall performance of algorithms on all six networks. Our algorithm gets Q_{ov} value for Dolphins, Football, Polbooks, and PGP, which is slightly higher than other algorithms. Moreover, the sum of Q_{ov} obtained by our algorithm is higher than the others. As an example, the community structure of the karate network, which is obtained by our algorithm, is shown in Fig. 10. This network is of traditional importance and was studied by Wayne W. Zachary for three years, from 1970 to 1972[41]. The ground truth of this network that was observed by Zachary contains two communities represented in Fig. 10. As it can be seen, the detected community structure is exactly fitted to ground truth if excluding node 10. However, locating node 10 in the overlapping of two communities is sensible, considering its equal connection with both.

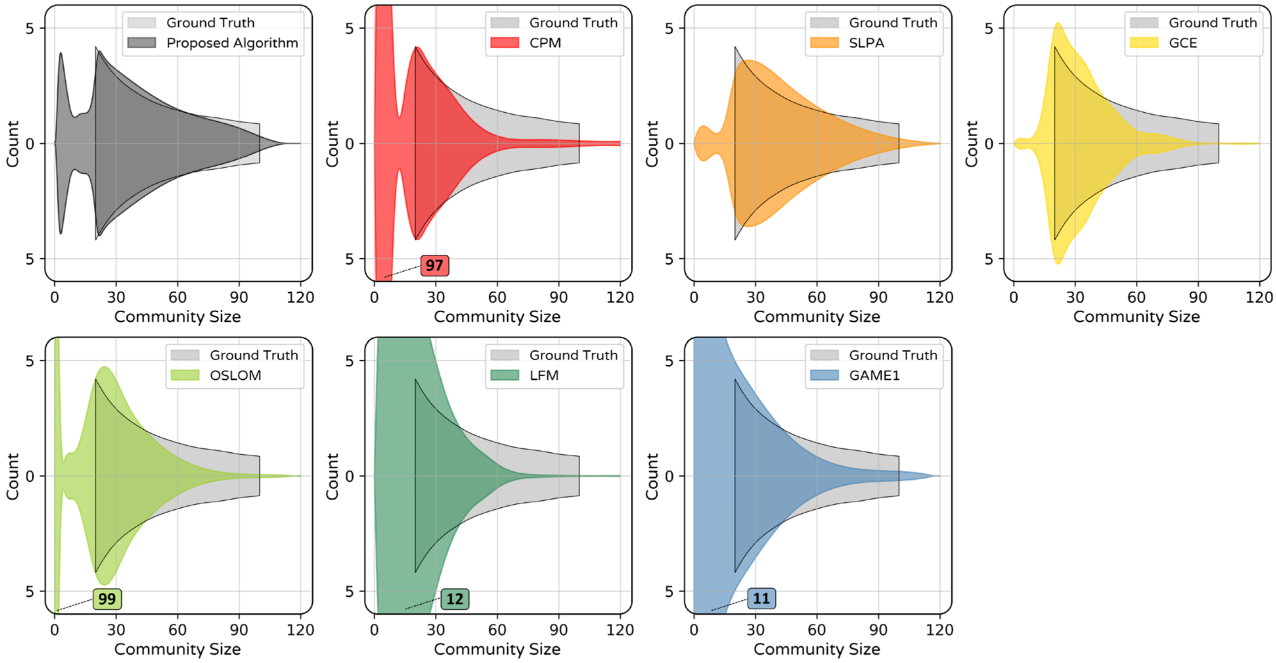

Figure 8. Histogram of detected community sizes for LFR3 (averaged on all *Om*). In each plot, the height of peaks is written next to them if they locate out of the frame.

	n	$\bar{k}$
Karate	34	4.5
Dolphins	62	5.1
Lesmis	77	6.6
Polbooks	105	8.4
Football	115	10.6
PGP	10680	4.5
Amazon	334863	5.5
DBLP	317080	6.6

Figure 9. Real networks in test.

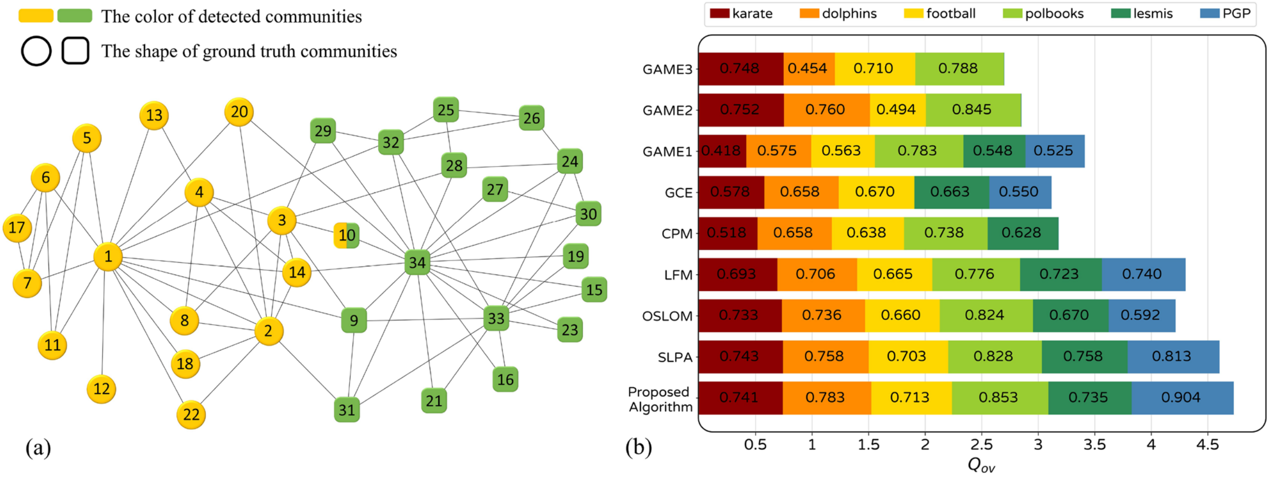

Figure 10. (**a**) Ground truth and detected community structure of karate network. (**b**) The Q_{ov} value obtained by proposed and other algorithms on first six real-world networks.

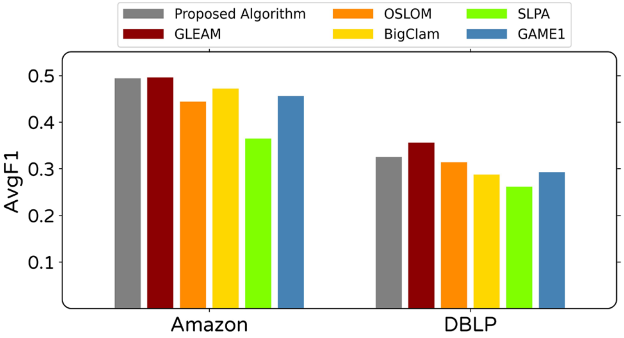

Figure 11. The AvgF1 score value obtained by the proposed and other algorithms on last two real-world networks with known community structure (ground truth).

For the last two larger networks, which have know community structure, the bar chart of AvgF1 scores for obtained community structure by ours and other algorithms are shown in Fig. 11. For these networks, in addition to previously used algorithms, the result of BigClam[33] and GLEAM[5] algorithms are represented for comparison. Data related to other algorithms' performance on these two networks are extracted from GLEAM algorithm's original paper[5]. Based on the results represented in 11, it can be seen that the proposed algorithm, along with the GLEMAo algorithm, has the best performance in the detecting community structure of these two networks.

Conclusion

In this paper, we proposed a novel game theoretic-based algorithm for community detection in networks. The algorithm performance test on synthetic and real-world networks indicates our algorithm has a relatively better performance compared with similar algorithms presented in the literature. Our proposed algorithm has a time complexity of $O(m)$, making it a good choice for applying on ultra-large networks. Besides, no stochastic factors are influencing the process of community detection, which eliminates the need for multiple executions and averaging of results and causes our algorithm to be categorized among stable ones. In addition, this framework can be straightforwardly applied to weighted networks by making minor changes.

Data availability

All data generated or analyzed during this study are included in this published article. The proposed algorithm python code is available in the Supplementary Material.

Received: 8 February 2022; Accepted: 17 June 2022
Published online: 30 June 2022

References

1. Wang, Y., Bu, Z., Yang, H., Li, H.-J. & Cao, J. An effective and scalable overlapping community detection approach: Integrating social identity model and game theory. *Appl. Math. Comput.* **390**, 125601. https://doi.org/10.1016/j.amc.2020.125601 (2021).
2. Chen, Y., Cao, X. & Liu, K. J. R. Community detection in networks: A game-theoretic framework. *EURASIP J. Adv. Signal Process.* **2019**, 60. https://doi.org/10.1186/s13634-019-0655-z (2019).
3. Girvan, M. & Newman, M. E. J. Community structure in social and biological networks. *Proc. Natl. Acad. Sci.* **99**, 7821–7826. https://doi.org/10.1073/pnas.122653799 (2002).
4. Schaub, M. T., Delvenne, J.-C., Rosvall, M. & Lambiotte, R. The many facets of community detection in complex networks. *Appl. Netw. Sci.* **2**, 4. https://doi.org/10.1007/s41109-017-0023-6 (2017).
5. Bu, Z., Cao, J., Li, H.-J., Gao, G. & Tao, H. GLEAM: A graph clustering framework based on potential game optimization for large-scale social networks. *Knowl. Inf. Syst.* **55**, 741–770. https://doi.org/10.1007/s10115-017-1105-6 (2018).
6. Guimerà, R. & NunesAmaral, L. A. Functional cartography of complex metabolic networks. *Nature* **433**, 895–900. https://doi.org/10.1038/nature03288 (2005).
7. Udrescu, L. *et al.* Clustering drug-drug interaction networks with energy model layouts: Community analysis and drug repurposing. *Sci. Rep.* **6**, 32745. https://doi.org/10.1038/srep32745 (2016).
8. Costa, H., Merschmann, L. H., Barth, F. & Benevenuto, F. Pollution, bad-mouthing, and local marketing: The underground of location-based social networks. *Inf. Sci.* **279**, 123–137. https://doi.org/10.1016/j.ins.2014.03.108 (2014).
9. Vidaurre, D., Smith, S. M. & Woolrich, M. W. Brain network dynamics are hierarchically organized in time. *Proc. Natl. Acad. Sci.* **114**, 12827–12832. https://doi.org/10.1073/pnas.1705120114 (2017).
10. Caplan, S. E. Preference for online social interaction. *Commun. Res.* **30**, 625–648. https://doi.org/10.1177/0093650203257842 (2003).
11. Gong, M., Cai, Q., Chen, X. & Ma, L. Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition. *IEEE Trans. Evol. Comput.* **18**, 82–97. https://doi.org/10.1109/TEVC.2013.2260862 (2014).
12. Arava, R. Community detection using coordination games. *Soc. Netw. Anal. Mining* **8**, 65. https://doi.org/10.1007/s13278-018-0543-9 (2018).
13. Cherifi, H., Palla, G., Szymanski, B. K. & Lu, X. On community structure in complex networks: Challenges and opportunities. *Appl. Netw. Sci.* **4**, 117. https://doi.org/10.1007/s41109-019-0238-9 (2019).
14. Fiscarelli, A. M., Brust, M. R., Danoy, G. & Bouvry, P. Local memory boosts label propagation for community detection. *Appl. Netw. Sci.* **4**, 95. https://doi.org/10.1007/s41109-019-0210-8 (2019).
15. Lee, C. & Wilkinson, D. J. A review of stochastic block models and extensions for graph clustering. *Appl. Netw. Sci.* **4**, 122. https://doi.org/10.1007/s41109-019-0232-2 (2019).

16. Athey, S., Calvano, E. & Jha, S. A Theory of Community Formation and Social Hierarchy. *SSRN Electron. J.*https://doi.org/10.2139/ssrn.2823777 *(2016).*
17. Chen, W., Liu, Z., Sun, X. & Wang, Y. A game-theoretic framework to identify overlapping communities in social networks. *Data Mining Knowl. Discov.* **21**, 224–240. https://doi.org/10.1007/s10618-010-0186-6 (2010).
18. Jonnalagadda, A. & Kuppusamy, L. A survey on game theoretic models for community detection in social networks. *Soc. Netw. Anal. Mining* **6**, 83. https://doi.org/10.1007/s13278-016-0386-1 (2016).
19. Alvari, H., Hashemi, S. & Hamzeh, A. Detecting overlapping communities in social networks by game theory and structural equivalence concept. *Artif. Intell. Comput. Intell.* **1**, 620–630. https://doi.org/10.1007/978-3-642-23887-1_79 (2011).
20. Zhou, X., Zhao, X., Liu, Y. & Sun, G. A game theoretic algorithm to detect overlapping community structure in networks. *Phys. Lett. A* **382**, 872–879. https://doi.org/10.1016/j.physleta.2018.01.036 (2018).
21. Moscato, V., Picariello, A. & Sperlí, G. Community detection based on game theory. *Eng. Appl. Art. Intell.* **85**, 773–782. https://doi.org/10.1016/j.engappai.2019.08.003 (2019).
22. Xie, J., Szymanski, B. K. & Liu, X. SLPA: Uncovering Overlapping Communities in Social Networks via a Speaker-Listener Interaction Dynamic Process. In *2011 IEEE 11th International Conference on Data Mining Workshops*, 344–349, https://doi.org/10.1109/ICDMW.2011.154 (IEEE, 2011).
23. Psorakis, I., Roberts, S., Ebden, M. & Sheldon, B. Overlapping community detection using Bayesian non-negative matrix factorization. *Phys. Rev. E* **83**, 066114. https://doi.org/10.1103/PhysRevE.83.066114 (2011).
24. Zhou, X., Cheng, S. & Liu, Y. A cooperative game theory-based algorithm for overlapping community detection. *IEEE Access* **8**, 68417–68425. https://doi.org/10.1109/ACCESS.2020.2985397 (2020).
25. Jonnalagadda, A. & Kuppusamy, L. A cooperative game framework for detecting overlapping communities in social networks. *Physica A* **491**, 498–515. https://doi.org/10.1016/j.physa.2017.08.111 (2018).
26. Lancichinetti, A., Radicchi, F., Ramasco, J. J. & Fortunato, S. Finding statistically significant communities in networks. *PLoS ONE* **6**, e18961. https://doi.org/10.1371/journal.pone.0018961 (2011).
27. Palla, G., Derényi, I., Farkas, I. & Vicsek, T. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**, 814–818. https://doi.org/10.1038/nature03607 (2005).
28. Lee, C., Reid, F., McDaid, A. & Hurley, N. Detecting highly overlapping community structure by greedy clique expansion. *Nature* **1002**, 1827 (2010).
29. Lancichinetti, A., Fortunato, S. & Kertész, J. Detecting the overlapping and hierarchical community structure in complex networks. *N. J. Phys.* **11**, 033015. https://doi.org/10.1088/1367-2630/11/3/033015 (2009).
30. Xie, J., Kelley, S. & Szymanski, B. K. Overlapping community detection in networks. *ACM Comput. Surv.* **45**, 1–35. https://doi.org/10.1145/2501654.2501657 (2013).
31. George, R. *et al.* A comparative evaluation of community detection algorithms in social networks. *Procedia Comput. Sci.* **171**, 1157–1165. https://doi.org/10.1016/j.procs.2020.04.124 (2020).
32. Jebabli, M., Cherifi, H., Cherifi, C. & Hamouda, A. Community detection algorithm evaluation with ground-truth data. *Physica A* **492**, 651–706. https://doi.org/10.1016/j.physa.2017.10.018 (2018).
33. Yang, J. & Leskovec, J. Overlapping community detection at scale. In *Proceedings of the sixth ACM international conference on Web search and data mining - WSDM '13*, 587, https://doi.org/10.1145/2433396.2433471 (ACM Press, 2013).
34. Gregory, S. Fuzzy overlapping communities in networks. *J. Stat. Mech. Theory Exp.* **2011**, P02017. https://doi.org/10.1088/1742-5468/2011/02/P02017 (2011).
35. Danon, L., Díaz-Guilera, A., Duch, J. & Arenas, A. Comparing community structure identification. *J. Stat. Mech. Theory Exp.* **2005**, P09008–P09008. https://doi.org/10.1088/1742-5468/2005/09/P09008 (2005).
36. Nicosia, V., Mangioni, G., Carchiolo, V. & Malgeri, M. Extending the definition of modularity to directed graphs with overlapping communities. *J. Stat. Mech. Theory Exp.* **2009**, P03024. https://doi.org/10.1088/1742-5468/2009/03/P03024 (2009).
37. Lancichinetti, A. & Fortunato, S. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E* **80**, 016118. https://doi.org/10.1103/PhysRevE.80.016118 (2009).
38. Newman, M. E. J. The structure and function of complex networks. *SIAM Rev.* **45**, 167–256. https://doi.org/10.1137/S003614450342480 (2003).
39. Orman, G. K. & Labatut, V. A comparison of community detection algorithms on artificial networks. *Networks* **1**, 242–256. https://doi.org/10.1007/978-3-642-04747-3_20 (2009).
40. Orman, G. K., Labatut, V. & Cherifi, H. Towards realistic artificial benchmark for community detection algorithms evaluation. *Int. J. Web Based Commun.* **9**, 349. https://doi.org/10.1504/IJWBC.2013.054908 (2013).
41. Zachary, W. W. An information flow model for conflict and fission in small groups. *J. Anthropol. Res.* **33**, 452–473. https://doi.org/10.1086/jar.33.4.3629752 (1977).

Acknowledgements

We thank Niloufar Najafian for fruitful discussions.

Author contributions

F.F. and K.A.S. contributed in algorithm development and result analysis. F.F. wrote the code and conducted the experiments, generated the figures, and wrote the manuscript. All authors reviewed and edited the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-022-15095-9.

Correspondence and requests for materials should be addressed to F.F.