A Game-Theoretic Approach for Detection of Overlapping Communities in Dynamic Complex Networks

ELHAM HAVVAEI
University of Central Florida
Department of Computer Science
4000 Central Florida Blvd, Orlando
USA
ehavvaei@cs.ucf.edu

NARSINGH DEO
University of Central Florida
Department of Computer Science
4000 Central Florida Blvd, Orlando
USA
deo@cs.ucf.edu

Abstract: Complex networks tend to display communities which are groups of nodes cohesively connected among themselves in one group and sparsely connected to the remainder of the network. Detecting such communities is an important computational problem, since it provides an insight into the functionality of networks. Further, investigating community structure in a dynamic network, where the network is subject to change, is even more challenging. This paper presents a game-theoretic technique for detecting community structures in dynamic as well as static complex networks. In our method, each node takes a role of a player that attempts to gain a higher payoff by joining one or more communities or switching between them. The goal of the game is to reveal community structure formed by these players by finding a Nash-equilibrium point among them. The main contribution of this paper involves the development of a computationally feasible algorithm for extracting high quality community structure by deployment of existing novel game-theoretic techniques. We present the experimental results illustrating the effectiveness of the proposed method on both synthetic and real-world networks.

Key–Words: Community Structure, Overlapping Community Detection, Modularity, Dynamic Network, Extremal Optimization, Nash Equilibrium

1 Introduction

Community detection problem in complex networks has been the subject of intensive studies throughout the last decade [12]. There is no formal or universally accepted definition for the notion of *community*. Intuitively, a community can be seen as a dense set of nodes with more edges within itself than to the nodes outside the set. Community structure is the set of all such communities in a network. Being able to extract community structure within a network provides a deeper insight through the functionality of systems represented as networks. Most complex networks such as social or biological networks exhibit community structure properties. In social networks a community is interpreted as a set of people who may have common interests, ethnicity or geographic location. Protein-protein interaction network is an example of a biological network displaying community structure in which proteins in a community are likely to have the same functionality[7]. Most techniques for detecting community structures partition networks into disjoint communities. However, most networks inherently have overlapping community structure. In social networks, people may be a member of multiple social communities. In citation networks, researchers may collaborate with different research groups. In the protein-protein interaction network, a large fraction of proteins, simultaneously, reside in multiple protein complexes. Thus, overlapping community detection reveals a deeper feature of real-world networks [34, 40, 42].

Fundamentally, most complex networks are highly dynamic. In each timestep, a dynamic network is subject to a series of changes where new nodes or edges either appear or some existing ones vanish. Tracking and monitoring changes happening in a community structure while the network is experiencing a series of events (including adding or removing nodes or edges), give an insight into the future functionality of the system. Therefore, pursuing the evolution of community structures, over time, in dynamic networks is more informative in comparison to its static counterparts. An online community detection algorithm which is capable of updating the community structure at each timestep, should be able to exploit the history of the community structure instead of starting from scratch.

A major advance in the study of community detection problem was made by Newman and Girvan who introduced a quality measure Q, called *mod-*

ularity [31]. Given a community structure $C = \{c_1, c_2, ..., c_k\}$ in a corresponding network $G = (V, E)$ with node set V and the edge set E ($n = |V|$, $m = |E|$), modularity is computed as

$$Q = \sum_{c \in C} \frac{L(c)}{m} - (\frac{d(c)}{2m})^2, \quad (1)$$

where the sum runs over all communities. $L(c)$ is the number of edges inside community c and $d(c)$ is the sum of degrees of nodes in c.

Modularity was designed to measure the strength of partition of a network into communities. High value of this measure implies a community structure with dense communities which has more number of intra-edges within communities than the expected value of such number in a random partition. Optimizing modularity value is recognized as a NP-hard problem [4]. Therefore, a vast number of works have been devoted to optimizing this measure by employing approximation and heuristic methods.

In [37], modularity is extended to the case of overlapping community structure by involving number of communities to which a node belongs:

$$Q_{ov} = \frac{1}{2m} \sum_{c \in C} \sum_{v,w \in c} \frac{1}{O_v O_w} (A_{vw} - \frac{k_v k_w}{2m}). \quad (2)$$

Here, O_v and O_w are the numbers of communities to which nodes v and w belong, respectively. In the case of disjoint community structure, Q_{ov} is reduced to Newman and Girvan's modularity Q.

Intuitively, the emergence of community structure in real networks is a bottom-up phenomenon and nodes of a network as indivisible objects play a vital role in the formation of communities. A less studied approach for community detection, first introduced by Chen et al. [8] , involves game-theoretic techniques which considers nodes of networks as rational decision makers [8, 23, 26]. In this line of work, nodes are considered as a set of players who decide to join or leave communities. Since, the game-theoretic approach in community detection naturally reflects the community structure formation in real-world networks, it is worth further studies and expansion. Our approach improves upon the best-known game-theoretic algorithms (Game [8]), both in terms of quality of obtained community structure and running time. Our proposed method is capable of resolving resolution limit problem, existing in modularity optimization methods and detecting high quality overlapping as well as non-overlapping community structure from both static and dynamic networks, with an asymptotically fast complexity.

We aim at, maximizing the extended version of modularity by establishing a game among the nodes of the networks as players. Initially, each player resides in a singleton community and their goal is to maximize their payoffs by choosing to reside in one or more communities. The ideal state of the game is to reach a Nash equilibrium point where no player has a tendency to deviate from its situation by taking a unilateral action. In a community formation game, a Nash equilibrium point implies that players are not able to gain a higher payoff by changing their communities, unilaterally. Further, an extremal optimization method is employed in order to perform the search and direct the game to a Nash equilibrium point.

2 Related Work

There exist various techniques in literature which try to extract good quality community structures from complex networks. Some methods reduce the community detection problem to graph partitioning, in which a graph is divided into a predefined number of communities such that the number of inter-edges crossing the network is minimal[19]. One drawback of graph partitioning is that, generally, the number of communities is unknown. Hierarchical clustering is another technique which considers a hierarchical structure for the underlying networks and applies an agglomerative [29] or divisive [35] method to build a hierarchy of clusters. Hierarchical clustering technique requires a definition for similarity measure to group similar nodes together. One advantage of this technique is that there is no need for preliminary knowledge about the size or number of clusters. However, since each level of hierarchy corresponds to one partition of network, choosing the one as a true community structure without any quality function is problematic.

In principle, most techniques try to define a quality function and by maximizing it, obtain a high quality community structure [22, 15]. Modularity is the best known quality function used as the quality index of extracted community structure [31]. Modularity maximization is NP-hard since the number of possible divisions of a network into communities exceeds any power of the network size (number of nodes). Therefore, many works have been proposed to approximate Q and obtain a fair value of this measure by using greedy [29, 9], extremal optimization [11], simulated annealing [17, 18, 36] or spectral optimization methods [39, 30].

In most techniques in literature, the extracted community structure is a set of disjoint communities. However, it is more reasonable to have overlapping community structure where a node is not restricted to merely belonging to one single community.

A very popular method to extract overlapping community structure is the clique percolation method (CPM) introduced in [10]. CPM builds a community structure by percolating a k-clique, i.e., a complete graph of k nodes, over a network. In this context, a community structure is the union of all k-cliques that can be reached from each other through the percolation process. Trivially, since the k-cliques can share nodes, the resulting community structure is overlapping. One limitation of CPM is to make an assumption about the value k. Besides, another critical problem with CPM is its failure on networks with fewer number of cliques.

Despite the difficulty of static community detection where the nodes and edges are fixed, this problem has also been studied on dynamic networks [33, 16, 38, 1, 23]. Several evolutionary clustering methods [6] have been proposed to extract evolving clusters, in a way that changes occurring in each timestep are not dramatic and in a user point of view, transitions seem smooth. In another attempt, AFOCS [32] categorizes changes happening in a network, and tries to take an appropriate action, accordingly, to update the community structure. As an example, adding a new edge inside an existing community is considered as one category.

A more recent and less studied approach for community detection is to formulate the problem as a game where nodes of a network are a population of decision makers who decide to join or leave communities with respect to their payoffs. Typically, at the start of community detection games, players reside in singleton communities [8, 26] or are distributed randomly to some number of communities [23]. The choice of payoff function plays an important role in the quality of extracted community structures. One possible payoff function is the contribution of each individual to the global modularity value. Therefore, by establishing a game among the nodes of networks as the players, instead of directly optimizing a global objective function, the players are appointed to take actions and maximize it as they seek to maximize their own payoffs. Since, generally, achieving the maximum possible payoff does not exist, Nash equilibrium is often employed as the solution concept in games with no cooperation among the players. Nash equilibrium is a situation where no player has the tendency to deviate from its strategy while the strategy of others are kept unchanged.

3 Overlapping Community Detection Game

We refer to our method as Overlapping Communities Extremal Optimization (OCEO). In this method, each node takes a role as a player. A pair of players are considered neighbors if there is an edge between them. The neighboring set of player i is defined as

$$\mathcal{N}_i = \{j \in V \mid \exists\, (i,j) \in E\}. \tag{3}$$

The number of players is finite and the payoff of each player is a function of chosen action from the player's strategy space. The players can only act on their turn. The elements of the game are as follows:

- **Players**: $\{1, 2, ..., n\}$ denotes the set of players. Each player corresponds to a node in the set V.
- **Strategies**: The strategy space available to each player i is to join an adjacent community or leave one of its current communities. A community c is considered adjacent to a player if at least one of the player's neighbors resides in c. The set of all adjacent communities for the player i is defined as
$$AC(i) = \{c \in C \mid (\exists\, j \in \mathcal{N}_i) \wedge (c \in C_j)\}, \tag{4}$$
where C_j is the set of all communities that player j resides in. The strategy spaces of players (the set of adjacent communities) are finite which leads to a finite game.
- **Payoffs**: As stated, modularity Q_{ov} is the global variable to optimize. The idea is to appoint players in the game to maximize it for us by defining the payoff function of players proportional to their contributions to the modularity Q_{ov}. The contribution of each individual *i* to modularity Q_{ov}, by residing in community c, is computed as
$$q_i(c) = \sum_{j \in c \,\cap\, j \in \mathcal{N}_i} \frac{1}{O_i O_j} - \frac{k_i}{2mO_i} \sum_{j \in c} \frac{k_j}{O_j} \tag{5}$$
where $L_i(c)$ is the number of edges player i has inside community c, k_i is the degree of the player (degree of the associated node) and $d(c)$ is the sum of degrees of all players residing in c. The total contribution of player *i* to the modularity is the sum of $q_i(c)$ over all communities that i belongs to.
More specifically,
$$q_i = \sum_{c \in C_i} q_i(c). \tag{6}$$

Note that $Q_{ov} = \frac{1}{2m}\sum_i q_i$ where $i \in V$. In OCEO, we define the payoff function of each player i in community c as the contribution of the mentioned player to the global modularity by residing in c, rescaled by the degree of the player:

$$u_i(c) = \frac{q_i(c)}{k_i} \tag{7}$$

Due to rescaling, the payoff of each associated player would be relative to their own degree and also normalized in the interval [-1,1] with 1 being the highest payoff a player can gain. Therefore during the game as long as players attempt to improve their payoffs, modularity Q_{ov} is optimized.

The total payoff of Player i is the sum of all i' payoffs in all communities i resides in, which is computed as

$$u_i = \sum_{c \in C_i} u_i(c). \tag{8}$$

In order to obtain a high quality community structure which corresponds to a high value of modularity, a heuristic search, based on extremal optimization is employed to perform a local search. Extremal optimization algorithm, proposed by Boettcher and Percus [2], finds high quality solutions for hard optimization problems by successively improving the undesirable components of the sub-solution. In other words, extremal optimization works on a single candidate solution and performs a local modification to the worst components by treating each of the components as species evolving locally through the process [3]. In the context of our game, extremal optimization keeps two solutions, that each of which corresponds to a community structure. One community structure preserves the best community structure found so far, and the other evolves through some number of iterations by performing a modification in the strategy of worst-payoff players. OCEO is a non-cooperative game since the players' decisions are made independently and there is no collaboration or coalition among the players.

One of the most fundamental ideas for non-cooperative games is the concept of Nash equilibrium introduced by Nash [28]. Nash equilibrium point of a game is a situation in which no player can gain a higher payoff by taking a unilateral action. Existence of Nash equilibrium depends on the choice of the payoff function in a game with finite number of players.

3.1 Existence of Nash Equilibrium in OCEO

In this section, we prove the existence of Nash equilibrium in OCEO by characterizing it as a potential game, introduced by Monderer and Shapley [27]. In every finite potential game, the existence of Nash equilibrium is guaranteed [5]. First, we define the notions required for the definition of potential games. Consider strategy profile $\chi = (\chi_1, \chi_2, ..., \chi_n)$ as a set of players' strategies in the game in which χ_i denotes the strategy of player i. In our game χ_i is interpreted as a set of communities that Player i resides in and χ is capable of revealing the community structure of the underlying network. A reduced strategy profile χ_{-i} is the strategy profile χ for all players excluding the strategy of player i, i.e., $\chi_{-i} = (\chi_1, \chi_2, ..., \chi_{i-1}, \chi_{i+1}, ..., \chi_n)$. We also use (χ_{-i}, χ_i') to denote a strategy profile where the strategy of player i is replaced with strategy χ_i'.

Definition 1 *A game $\Gamma(V, \chi, \{u_i\}_{i\in V})$ with $u_i : \chi \to R$, is called an exact potential game, if it possesses a potential function $\phi : \chi \to R$ such that $\forall i \in V$, $\forall x_{-i} \in \chi_{-i}$, $\forall x_i' \in \chi_i$:*

$$\phi(x_{-i}, x_i') - \phi(x) = u_i(x_{-i}, x_i') - u_i(x). \tag{9}$$

Specifically, the change in the payoff of Player i by taking a unilateral action x_i' is equal to the change in the global potential function. There is a weaker class of potential games, called weighted potential which admits a $\omega-$potential function.

Definition 2 *Let $\omega = \{\omega_i\}_{i\in V}$ be a vector of positive weights. A game Γ is weighted potential if it admits a $\omega-$potential function $\phi : \chi \to R$:*

$$\phi(x_{-i}, x_i') - \phi(x) = \omega_i(u_i(x_{-i}, x_i') - u_i(x)). \tag{10}$$

Note that the exact potential games are subset of weighted potential games with $\omega_i = 1$ for each player $i \in V$. it is proved that in every finite exact or weighted potential games, the best response dynamics, i.e., dynamics in which each player chooses a strategy with highest payoff, given the strategies of the other players, converges to a Nash equilibrium in a finite number of iterations [5].

Lemma 3 *OCEO is a weighted potential game.*

Proof: By considering the choice of payoff function u, defined in Equation 8, and assigning weight vector $\omega = (\omega_i)_{i\in V}$ where $\omega_i = \frac{1}{k_i}$ for each player $i \in V$, it can be verified that OCEO accepts the following

$\omega-$potential function:

$$\phi(\chi) = \sum_i (\sum_{\substack{j \in \mathcal{N}_i \wedge \\ c_i = c_j \in \chi}} \frac{1}{O_i O_j} - \frac{1}{2m} \sum_{\substack{i <= j \wedge \\ c_i = c_j \in \chi}} \frac{k_i k_j}{O_i O_j}) \tag{11}$$

Therefore, OCEO possesses at least one Nash equilibrium point and we aim at finding such a point which corresponds to the highest value of modularity Q_{ov}. In order to incorporate the Nash equilibrium concept into OCEO, we use a domination relation, Nash ascendancy, introduced by Lung, *et al*., [24] to direct the search performed by extremal optimization towards a Nash Equilibrium point. This relation enables the comparison of two solutions in Nash sense and determines which solution is closer to an equilibrium. The use of this relation reduces the computational complexity of the search in comparison with the deterministic Nash equilibrium relation [24]. In [23], Nash ascendancy relation is defined for two non-overlapping community structures. By extending this relation to overlapping community structures, community structure D precedes P in Nash sense, if there are more number of players who prefer D over P:

$$\zeta(D, P) > \zeta(P, D), \tag{12}$$

where $\zeta(D, P)$ is the number of players who prefer their communities assignments in community structure D over P. More formally, $\zeta(D, P)$ is defined as

$$\zeta(D, P) = |\{i \mid u_i(D) > u_i(P)\}|, \tag{13}$$

where $|S|$ denotes cardinality of set S. In other words, if D Nash ascends P, there are fewer number of players i who can increase their payoffs by switching from $C_i(D)$ to $C_i(P)$ than vice-versa and as a result, D seems more stable than P, in Nash-sense.

3.2 Extremal Optimization Over Overlapping Community Detection Game

Assume initially each player resides in a singleton community. Community structure D is initialized as the set of all these singleton communities. Our purpose is to evolving D through some number of iterations and favoring the one which is more stable in Nash-sense by employing extremal optimization.

In contrast with OCEO which the focus of each iteration is to evolve the obtained community structure, Game [8] aims to improve the current strategy of a random node, in each iteration. This approach for reaching a local equilibrium, is the main reason behind Game's long running time. This process repeats until no node can improve itself for a long number of iterations. The uperbound for this recurrence is of the order of $O(m^2)$ which explains the slow convergence of Game.

Intuitively by giving the turn to a Player i with the least payoff in particular community $c_i \in C_i(D)$ to perform some actions, that player is provided a chance to improve its total payoff by leaving c_i or joining a new community available on its strategy space. We define pairs of $<i, c>$, in which i is a player who belongs to community c. In case of overlapping community structure, there are multiple pairs of $<i, \cdot>$. In our method, we rank such pairs $<i, \cdot>$ according to their corresponding $u_i(\cdot)$ and keep all the sorted pairs in a pairs-pool. Thus, the first ranked pair corresponds to a player who has the least payoff in the associated community. In each iteration, γ pairs are chosen based on a selection mechanism from the pool.

For each selected pair $<i, c>$, player i tries to improve its payoff by joining a new community or leaving c. In our selection mechanism, we naturally favor pairs with the lower payoff. We apply truncation selection in which the γ top-ranked pairs are selected from the pool. Within the selected set of pairs there is no further selection and all the corresponding players have a chance to take action.

To avoid early convergence, when the best solution has not been updated for η iterations, we switch from truncation to tournament selection method for the next iteration which we call that an *impulse iteration*. A tournament consists of picking δ (known as tournament size) pairs from the pool using a uniform distribution probability. The winner of the tournament is the one with the lower value of payoff. Choosing γ pairs requires running γ tournaments. The selection pressure is adjusted by the tournament size. Small assignment of δ brings in more randomness in the selection process and the pure randomness occurs when $\delta = 1$. On the other hand, larger value of δ results in more selection of worst-payoff pairs which contradicts with our initial purpose of employing tournament selection. In our method, $\eta = 5$ and $\delta = 3$ work the best.

OCEO proceeds as follows to evolve the single community structure D:

1. Community structures D and P are initialized as each player resides in a singleton community. Community structure P is used to store the best D found in each iteration.

2. The payoffs of players, in each community they reside, belonging to community structure D are computed.

3. Pairs of $<i, c>$ are sorted according to the payoff of player i in c and are maintained in the pairs-pool.

4. γ pairs are chosen from the pairs-pool based on truncation selection. The selection method is switched to tournament selection if it is an impulse iteration. For each selected pair $<i, c>$ Player i, performs the following actions:

   - Considering both cases of staying in or leaving c, calculates its possible total payoff for each community in its strategy space. It joins the one which leads to the maximum payoff higher than the current u_i.
   - Leaves c, if this actions results in a higher total payoff.

   The selected players, by performing actions, modify the community structure D. Note that the proposed algorithm is also capable of detecting non-overlapping communities. In this case, Player i, joins community $c' \in AC(i)$ and leaves c, if $u_i(c') > \max(u_i(c), u_i(c''))$, for $\forall c'' \in AC(i) \setminus \{c'\}$

5. If the resulting D Nash-ascends P, P is replaced by D. This replacement implies that the resulting D is the best solution found so far. Otherwise, D keeps performing the search without being stored.

6. Steps 4-5 are repeated until either the maximum number of iterations is reached or P has not been updated for ψ iterations after the last update. Upon termination, highly overlapped communities in the community structure P will be merged together and it is returned as the solution. In our method, two communities are considered highly overlapped if the division of common nodes to the size of smaller community exceeds 70%.

In step 4, after players' movements, payoffs of all involved players need to be recalculated. When a Player i joins/leaves community c, the payoffs of all players residing in c should be updated according to Equation 7, and the pair $<i, c>$ should be inserted/deleted to/from the pairs-pool. In each iteration, the average number of players whose payoffs are imposed to change is $\gamma|c|$, where $|c|$ is the average number of players in one community. Payoffs of all other players remain unchanged. In our method, we limit the players to reside in at most κ communities at the same time. Consequently, the maximum size of the pairs-pool is $\kappa|V|$ and the complexity of keeping the pairs-pool updated, in each iteration, is of the order of O($\gamma|c| \log^{\kappa|V|}$) where in the worst case, $\gamma|c|$ is of the order of $O(|V|)$. The pseudocode of the explained process is presented in Algorithm 1.

3.3 Parameter Settings

There are a number of parameters whose values can have a considerable influence in the performance of the algorithm.

- Number of selected players (γ): This parameter determines the number of players who are provided the chance to play simultaneously and change their strategies, in each iteration. The higher the size of a network, the greater number of players are selected to play. Thus, parameter γ should be proportional to the size of network. In an ideal situation, no pair of selected players should be adjacent. For clarification, assume two selected players i and j are adjacent who reside in communities c_1 and c_2, respectively. Obviously, c_2 appears in the strategy space of the player i and may persuade i to join c_2. Further, assume upon the joining of i to c_2, in the meantime, j leaves c_2. In this case, however rare, i's payoff estimate would be impaired, since i relies on the current strategy of its adjacent players. More nodes are adjacent to each other when the network possesses a high average degree. Thus, the number of selected players should be related to both size of network and also the average degree of nodes:

$$\gamma = \frac{N}{\hat{k}}. \tag{14}$$

- Max-Iteration: As shown in Algorithm 1, the game is repeated until either the maximum number of iterations reaches or the best solution obtained, does not get updated for ψ iterations. We set the Max-Iteration and ψ to 2000 and 100, respectively.

3.4 OCEO Complexity

Let $\hat{k}$ be the average degree of players in a network. Then, the time complexity O($\gamma\hat{k}$) is required for γ players to search through their strategy spaces and find the best communities to join. As stated above, the complexity of efficiently updating the pairs-pool is of the order of O($|V| \log^{\kappa|V|}$). Further, in each iteration, for determining the Nash-ascendency relation, payoff of each involved player in the dummy and parent communities need to be compared which results in O($|V|$) comparisons. Therefore, in overall, the complexity of OCEO is of the order of O(MAX-Iteration $\times|V| \log^{\kappa|V|}$).

Algorithm 1 Overlapping Community Detection Game

Initialize P and D as each player resides in a singleton community
for each player $i \in V$ **do**
  for each community $c \in C_i(D)$ **do**
    $u_i(c) \leftarrow \frac{q_i(c)}{k_i}$
  end for
end for
Sort pairs $<i, c>$ with respect to $u_i(c)$ and keep them in the pairs-pool
$temp \leftarrow 0$, $iteration \leftarrow 0$
while $iteration < MAX\text{-}Iteration$ **do**
  if $temp \geqslant 5$ **then**
    Pick and evict γ pairs from the pairs-pool based on tournament selection
  else
    Pick and evict γ pairs from the pairs-pool based on truncation selection
  end if
  for each selected pair $<i, c>$ **do**
    Player i, Considering both cases of staying in or leaving c, joins $c' \in AC(i)$, if this action leads to the maximum total payoff, higher than the current u_i.
    Leaves c, if this actions results in a higher total payoff.
  end for
  Recalculate payoffs of involved pairs and update the pairs-pool
  if D Nash-ascends P **then**
    $D \leftarrow P$
    $temp \leftarrow 0$
  else
    $temp \leftarrow temp + 1$
  end if
  if $temp \equiv \psi$ **then**
    break
  end if
  $iteration \leftarrow iteration + 1$
end while
Merge highly overlapped communities
return P

3.5 Resolution Limit and the solution mechanism

Modularity optimization is one of the most popular methods for community detection. However it fails to detect smaller communities than some scale and tends to merge such communities to gain a higher modularity value. This phenomenon, known as resolution limit, merges two connected communities c_1 and c_2 when $d(c_1) \times d(c_2) < 2m$ [26]. In fact, modularity is a sum of terms and it establishes a trade-off between number of terms (which corresponds to the number of communities) and magnitude of each term.

Fortunato, *et al.* [13] designed a network, known as ring network, which consists of identical cliques as modules connected by single edges to two other modules to manifest the drawback of modularity optimization. Ring network, shown in Fig.1a, has a clear modular structure where each community corresponds to a module/clique. The expectation is that any community detection algorithm, including a modularity optimization method extracts the true community structure. However, in this type of network, if the number of modules exceeds $\sqrt{m}$, the modularity optimization fails to extract each module as one community and merges two or more modules together to increase the modularity value.

For illustration purposes, we consider a ring network with 50 modules where each module is a complete graph K_4. As can be computed, this network has 200 nodes and 350 edges and the number of modules exceeds $\sqrt{m}$. Fig.1b exhibits a closer view of two arbitrary connected modules of this network. A simplistic modularity optimization mechanism suffers from resolution limit and fails to detect the true modular structure. It finds a partition in which modules are merged together into groups of two, leading to the maximum modularity value (represented with dotted lines).

In our method, this issue is addressed and the algorithm is capable of identifying each module as one community. In this network, combining two or more modules together into one community would increase the modularity value. Therefore, there should exist some number of players who make more contribution to the modularity value and also gain a higher payoff upon the merger. Remember that the payoff of each player is the contribution of that individual rescaled by the degree of that player. Considering the two modules as separate communities, in Fig.1b, contribution of each player $\{2, 3, 6, 7\}$ to modularity is all the same and equal to $q_{\{2,3,6,7\}} = 3 - \frac{3\times14}{2\times350} = 2.94$. Accordingly, for the players $\{1, 4, 5, 8\}$, it is equivalent to $q_{\{1,4,5,8\}} = 3 - \frac{4\times14}{2\times350} = 2.92$. By merging these two modules together, the first set of players including Players 1 and 8 experience a drop in their contribution to the modularity (and also their payoffs) since non of them have an edge to another module while the sum of degree of the players in the resulted community has increased (an increase in the second term of Equation 5). In this case $q_{\{2,3,6,7\}} = 3 - \frac{3\times28}{2\times350} = 2.88$ and

$q_{\{1,8\}} = 3 - \frac{4\times28}{2\times350} = 2.84$. On the other hand, two Players 4 and 5 make a higher contribution due to the existence of an edge connecting them, which is equal to $q_{\{4,5\}} = 4 - \frac{4\times28}{2\times350} = 3.84$. The whole point is that the increase in the contribution of Players 4 and 5 are greater than the loss of other players which leads to a higher global modularity and as a result, the merger of the two modules. However, OCEO is capable of detecting all modules as communities, despite the fact that true community structure has the lower modularity value. The mechanism to resolve this issue is embedded in Nash ascendancy relation where the preferences of all individuals are taken into account, regardless of magnitude of the change in their payoffs or contributions to the modularity value. According to Equation 13, the true community structure Nash ascends the one with the pairs of adjacent modules as communities since there are fewer number of players who prefer the false community structure. Fundamentally, the true partition is inherently more stable in Nash sense. Hence, the Nash ascendancy relation, in addition to directing the game to the Nash equilibrium point, resolves the resolution limit in OCEO by considering the preferences of majority of players.

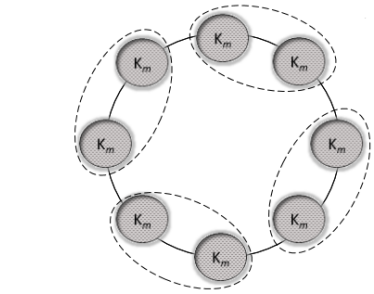
3.6 OCEO on Dynamic Networks

The dynamic network $G_d = \{G_1, G_2, ..., G_t\}$ is defined as a set of network snapshots evolving over time. Each G_i is a snapshot of the network G_d at timestep i. The problem of community detection in a dynamic network is to detect the community structure at each timestep by using the extracted one of the previous snapshot.

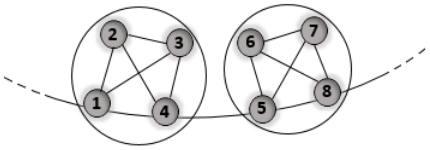
In [23], when detecting changes, NEO-CDD reinitializes community structure P while keeps D unchanged. By keeping D unchanged, the information from the community structure of the previous snapshot is utilized. However, a wiser approach is to swap D and P and then perform the above-mentioned process. In this way, we resume from the best found community structure of the previous snapshot and then adapt it to recent changes. Further, in contrast with NEO-CDD which initializes P by randomly distributing nodes to a predefined number of communities, OCEO assigns each node to a singleton community.

4 Experimental Results

In this section, we compare the effectiveness of OCEO on both synthetic and real-world networks with other community detection algorithms. In the following, we describe datasets, metrics and analysis.



(a)



(b)

Figure 1: (a) A network of identical cliques as communities connected by single edges, known as ring network. Each clique is a complete graph of size m, i.e., K_m. If the number of cliques exceeds $\sqrt{m}$, modularity optimization merges each pair of connected cliques and introduce it as a community (represented by dash lines) (b) A closer view of a ring network with m = 4.

4.1 Datasets

Synthetic Networks: Lancichinetti, *et al*., present a benchmark (LFR benchmark) for community detection algorithms [21]. The LFR benchmark generates static networks with built-in community structure. The configuration of generated networks depends on various user-specified parameters. Number of nodes is N, k specifies the average degree of nodes and k_{max} is the upper bound on degrees of nodes. The mixing parameter μ is the fraction of edges that a node has to the nodes outside of its community. Therefore, as we decrease μ we obtain a clear set of communities with fewer number of inter-edges. Later, the authors adapted the LFR benchmark to generate overlapping communities [20]. Parameter O_n specifies the number of overlapping nodes and O_m controls the number of membership of such nodes.

Real-world Networks: We present the performance of OCEO on several real-world complex networks with the absent of ground-truth communities: Zachary's karate club [41], Jazz musician network [14] and C. elegans metabolic network [11].

Metrics: Investigating the effectiveness of community detection algorithms involves defining a similarity measure between the extracted community structure and the partition one wishes to discover.

The most popular measure to compare the simi-

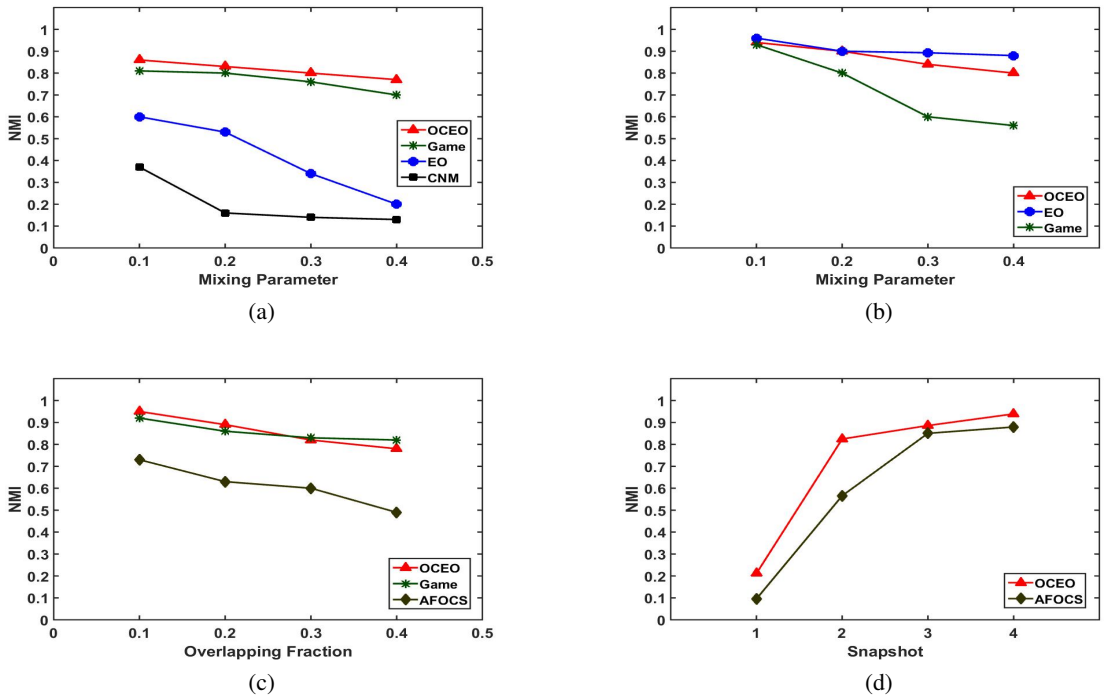


Figure 2: a) NMI of OCEO, Game, EO and CNM in the presence of small built-in communities. b) Comparison among OCEO, EO and Game based on average NMI for detecting non-overlapping communities. Higher NMI value indicates closer proximity to the ground-truth communities. c) Comparison among OCEO, AFOCS and Game for detecting overlapping communities d) NMI of OCEO and AFOCS over a network evolving through four snapshots

larity between the delivered community structure and the ground-truth communities is Normalized Mutual Information (NMI). We have used an implementation [1] of NMI measure made available by McDaid, *et al.*, for sets of overlapping communities [25]. Further, we use modularity value of obtained community structures and number of identified communities as additional measures when the ground-truth community structure is unknown. For each dataset, we run OCEO 25 times due to the non-deterministic aspect of the game and report the average values of NMI or modularity.

4.2 Analysis

In the first experiment we show the effectiveness of our method for resolving the resolution limit problem on the LFR benchmark graphs with implanted non-overlapping communities where mixing parameter varies in the interval of [0.1, 0.4]. The networks consist of 1,000 nodes and 5,000 edges where degree of each node is exactly set to 10, and the size of each built-in community is 10.
These networks meet the resolution limit condition wherein $d(c_1) \times d(c_2) < 2m$ for any pair of adjacent communities, c_1 and c_2. A strict modularity optimization method fails to resolve the small communities which were unambiguously defined. We compare our method with three modularity optimization methods: EO [11], Game [8] and CNM [9]. The results in Fig.2a, implies that EO and CNM fail to detect the important substructure of the network. In the meantime, OCEO achieves a significantly better result and also outperforms Game. Among all these methods EO achieves the highest modularity by merging most communities into groups of two.

For investigating the performance of OCEO in detecting non-overlapping communities, we contrast OCEO with the following methods: Game, AFOCS and CNM. The first two algorithms are also capable of detecting overlapping communities as well as non-overlapping ones. However for the purpose of this experiment, the overlapping aspect is disabled. We compare these methods for the increasing range of mixing parameter values in networks of n=5000 where $\langle k \rangle = 50$. The size of implanted communities lies between 50 and 200. As can be seen in Fig.2b, EO and OCEO are very competitive and both are far better than Game in detecting ground-truth community

[1] https://github.com/aaronmcdaid/Overlapping-NMI

Dataset	Size	$\mathbf{Q_{OCEO}}$	$\mathbf{Q_{EO}}$	$\mathbf{\lvert C_{OCEO}\rvert}$	$\mathbf{\lvert C_{EO}\rvert}$
Zachary	34	0.39 ± 0	0.42 ± 0	5.8 ± 0.44	4 ± 0
Jazz	198	0.41 ± 0	0.44 ± 0	5 ± 0.17	3.8 ± 0.12
C. elegans	453	0.40 ± 0	0.44 ± 0	13 ± 0.90	9.3 ± 0.80

Table 1: Modularity and number of communities obtained by OCEO and EO on different real-world complex networks

structures.

Further, we extend our experiments to examine the efficiency of our method on identifying overlapping communities in static networks. Fig.2c displays the performance of OCEO, Game and AFOCS over networks of n = 1000, m= 7368 and $\langle k \rangle = 15$. The x-axis presents fraction of nodes belonging to multiple communities in the corresponding networks with mixing parameter $\mu = 0.1$. OCEO and Game clearly outperforms AFOCS in detecting overlapping community structures. In this experiment the overlapping threshold is set to 0.6 for OCEO and AFOCS.

We next observe the performance of OCEO on dynamic networks in comparison with AFOCS. The synthesized network used for this experiment is generated by the LFR benchmark with $n = 5000$, $m = 147324$, wherein 10% of nodes are overlapping. The network is evolving through four snapshots where each of which comprises 25% of data. The results are presented in Fig.2d. As expected, both methods obtain increasing values of NMI as the network evolves and perform very well by achieving NMI values being above 85% when received the last snapshot.

In Table 1, we present the performance of OCEO on three real-world complex networks with the absence of ground-truth communities: the Zachary's karate club, Jazz musician network and C. elegans metabolic. We have run OCEO and EO for 25 trials and for the sake of comparison each node resides exactly in one community. According to these results, compared to EO, OCEO is capable of identifying a greater number of communities and at the same time achieving a high modularity value.

5 Conclusion

In this paper, we have proposed a game-theoretic method, OCEO, to detect communities in complex networks. Nodes as players try to maximize their payoffs by choosing one or more communities to join. The payoff of players, in their communities is proportional to their contributions to the modularity value Q. Therefore, global modularity is optimized through the iterations, while players improve their payoffs, and the game is propelled to converge to a Nash-equilibrium point among the players. The choice of individuals' payoff function and also the way the game drags itself to a Nash-equilibrium resolve the major issue of modularity optimization, resolution limit and distinguish our method from strict modularity optimization. Experimental results demonstrate the effectiveness of OCEO, in terms of obtaining high values of NMI and modularity from both synthetic and real-world networks in a reasonable time.

References:

[1] D.S. Bassett, M.A. Porter, N.F. Wymbs, S.T. Grafton, J.M. Carlson, and P.J Mucha, Robust Detection of Dynamic Community Structure in Networks, *Chaos: An Interdisciplinary Journal of Nonlinear Science*, Vol.23, No.1, 2013, p. 013142.

[2] S. Boettcher and A. Percus, Nature's Way of Optimizing, *Artificial Intelligence*, Vol.119, No.1, 2000, pp. 275–286.

[3] S. Boettcher and A.G. Percus, Extremal Optimization: Methods Derived from Co-evolution, *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation*, Vol.1, 1999, pp. 825–832.

[4] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski and D. Wagner, Maximizing Modularity is Hard, *ArXiv Preprint Physics/0608255*, 2006.

[5] O. Candogan, A. Ozdaglar and P.A. Parrilo, Near-potential Games: Geometry and Dynamics, *ACM Transactions on Economics and Computation*, Vol.1, No.2, 2013, p. 11.

[6] D. Chakrabarti, R. Kumar, and A. Tomkins, Evolutionary Clustering, *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, pp. 554–560.

[7] J. Chen and B. Yuan, Detecting Functional Modules in the Yeast Protein–protein Interaction Network, *Bioinformatics*, Vol.22, No.18, 2006, pp. 2283–2290.

[8] W. Chen, Z. Liu, X. Sun and Y. Wang, A Game-theoretic Framework to Identify Overlapping Communities in Social Networks, *Data Mining and Knowledge Discovery*, Vol.21, No.2, 2010, pp. 224–240.

[9] A. Clauset, M.E. Newman and C. Moore, Finding Community Structure in Very Large Networks, *Physical Review E*, Vol.70, No.6, 2004, p. 066111.

[10] I. Derényi, G. Palla, and T. Vicsek, Clique Percolation in Random Networks, *Physical Review Letters*, Vol.94, No.16, 2005, p. 160202.

[11] J. Duch and A. Arenas, Community Detection in Complex Networks Using Extremal Optimization, *Physical Review E*, Vol.72, No.2, 2005, p. 027104.

[12] S. Fortunato, Community Detection in Graphs, *Physics Reports*, Vol.486, No.3, 2010, pp. 75–174.

[13] S. Fortunato and M. Barthélémy, Resolution Limit in Community Detection, *Proceedings of the National Academy of Sciences*, Vol.104, No.1, 2007, pp. 36–41.

[14] P.M. Gleiser and L. Danon, Community Structure in Jazz, *Advances in Complex Systems*, Vol.6, No.4, 2003, pp. 565–573.

[15] M. Gong, B. Fu, L. Jiao and H. Du, Memetic Algorithm for Community Detection in Networks, *Physical Review E*, Vol.84, No.5, 2011, pp. 056101.

[16] D. Greene, D. Doyle and P. Cunningham, Tracking the Evolution of Communities in Dynamic Social Networks, *ASONAM*, 2010, pp. 176–183.

[17] R. Guimera, M. Sales-Pardo, and L.A.N. Amaral, Modularity from Fluctuations in Random Graphs and Complex Networks, *Physical Review E*, Vol.70, No.2, 2004, p. 025101.

[18] D. He, J. Liu, D. Liu, D. Jin and Z. Jia, Ant Colony Optimization for Community Detection in Large-scale Complex Networks, *Proceedings of 7th International Conference on Natural Computation*, Vol.2, 2011, pp. 1151–1155.

[19] B.W. Kernighan and S. Lin, An Efficient Heuristic Procedure for Partitioning Graphs, *Bell System Technical Journal*, Vol.49, No.2, 1970, pp. 291-307.

[20] A. Lancichinetti and S. Fortunato, Benchmarks for Testing Community Detection Algorithms on Directed and Weighted Graphs with Overlapping Communities, *Physical Review E*, Vol.80, No.1, 2009, p. 016118.

[21] A. Lancichinetti, S. Fortunato and F. Radicchi, Benchmark Graphs for Testing Community Detection Algorithms, *Physical Review E*, Vol.78, No.4, 2008, p. 046110.

[22] Z. Li, S. Zhang, R.S. Wang, X.S. Zhang and L. Chen, Quantitative Function for Community Detection, *Physical Review E*, Vol.77, No.3, 2008, p. 036109.

[23] R.I. Lung, C. Chira and A. Andreica, Game Theory and Extremal Optimization for Community Detection in Complex Dynamic Networks, *PloS One*, Vol.9, No.2, 2014, p. e86891.

[24] R.I. Lung and D. Dumitrescu, Computing Nash Equilibria by Means of Evolutionary Computation, *International Journal of Computers, Communications and Control*, Vol.3, 2008, pp. 364–368.

[25] A.F. McDaid, D. Greene and N. Hurley, Normalized Mutual Information to Evaluate Overlapping Community Finding Algorithms, *ArXiv Preprint ArXiv:1110.2515*, 2011.

[26] P. J. McSweeney, K. Mehrotra and J.C. Oh, A Game Theoretic Framework for Community Detection, *ASONAM*, 2012, pp. 227–234.

[27] D. Monderer and L.S. Shapley, Potential Games, *Games and Economic Behavior*, Vol.14, No.1, 1996, pp. 124–143.

[28] J. Nash, Non-cooperative Games, *Annals of Mathematics*, 1951, pp. 286–295.

[29] M.E. Newman, Fast Algorithm for Detecting Community Structure in Networks, *Physical Review E*, Vol.69, No.6, 2004, p. 066133.

[30] M.E. Newman, Spectral Methods for Community Detection and Graph Partitioning, *Physical Review E*, Vol.88, No.4, 2013, p. 042822.

[31] M.E. Newman and M. Girvan, Finding and Evaluating Community Structure in Networks, *Physical Review E*, Vol.69, No.2, 2004, p. 026113.

[32] N.P. Nguyen, T.N. Dinh, S. Tokala and M.T. Thai, Overlapping Communities in Dynamic Networks: Their Detection and Mobile Applications, *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, 2011, pp. 85–96.

[33] N.P. Nguyen, T.N. Dinh, Y. Xuan and M.T. Thai, Adaptive Algorithms for Detecting Community Structure in Dynamic Social Networks, *INFOCOM*, 2011, pp. 2282–2290.

[34] I. Psorakis, S. Roberts, M. Ebden and B. Sheldon, Overlapping Community Detection Using Bayesian Non-negative Matrix Factorization, *Physical Review E*, Vol.83, No.6, 2011, p. 066114.

[35] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto and D. Parisi, Defining and Identifying Communities in Networks, *Proceedings of the National Academy of Sciences of the United States of America*, Vol.101, No.9, 2004, pp. 2658–2663.

[36] J. Reichardt and S. Bornholdt, Statistical Mechanics of Community Detection, *Physical Review E*, Vol.74, No.1, 2006, p. 016110.

[37] H. Shen, X. Cheng, K. Cai and M.B. Hu, Detect Overlapping and Hierarchical Community Structure in Networks, *Physica A: Statistical Mechanics and its Applications*, Vol.388, No.8, 2009, pp. 1706–1712.

[38] Y. Sun, J. Tang, J. Han, M. Gupta, and B. Zhao, Community Evolution Detection in Dynamic Heterogeneous Information Networks, *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, 2010, pp. 137–146.

[39] S. White and P. Smyth, A Spectral Clustering Approach To Finding Communities in Graph, *SDM*, Vol.5, 2005, pp. 76–84.

[40] J. Yang and J. Leskovec, Overlapping Community Detection at Scale: a Nonnegative Matrix Factorization Approach, *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, 2013, pp. 587–596.

[41] W.W. Zachary, An Information Flow Model for Conflict and Fission in Small Groups, *Journal of Anthropological Research*, 1977, pp. 452–473.

[42] 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, 2007, pp. 483–490.