Overlapping Community Detection Using Multi-objective Approach and Rough Clustering

Darian Horacio Grass-Boada[1(✉)], Airel Pérez-Suárez[1], Leticia Arco[2], Rafael Bello[3], and Alejandro Rosete[4]

[1] Advanced Technologies Application Center (CENATAV), Havana, Cuba
{dgrass,asuarez}@cenatav.co.cu
[2] AI Lab, Computer Science Department, Vrije Universiteit Brussel, Brussels, Belgium
larcogar@vub.be
[3] Department of Computer Science, Universidad Central "Marta Abreu" de Las Villas, Santa Clara, Cuba
rbellop@uclv.edu.cu
[4] Facultad de Ingeniería Informática, Universidad Tecnológica de la Habana "José Antonio Echeverría" (Cujae), Havana, Cuba
rosete@ceis.cujae.edu.cu

Abstract. The detection of overlapping communities in Social Networks has been successfully applied in several contexts. Taking into account the high computational complexity of this problem as well as the drawbacks of single-objective approaches, community detection has been recently addressed as Multi-objective Optimization Evolutionary Algorithms (MOEAs). One of the challenges is to attain a final solution from the set of non-dominated solutions obtained by the MOEAs. In this paper, an algorithm to build a covering of the network based on the principles of the Rough Clustering is proposed. The experiments in a synthetic networks showed that our proposal is promising and effective for overlapping community detection in social networks.

Keywords: Social network analysis · Community detection · Multi-objective Optimization · Rough clustering

1 Introduction

The Analysis of Social Networks has received a lot of attention due to its wide range of applications in several contexts [1]. Specifically, in Social Network Analysis, the Community Detection Problem (CDP) plays an important role [5]. Community detection in social networks aims to organize the nodes of the network in groups or communities such that nodes belonging to the same community are densely interconnected but sparsely connected with the remaining nodes in the network [2]. Even though most of the community detection algorithms assume

R. Bello et al. (Eds.): IJCRS 2020, LNAI 12179, pp. 416–431, 2020.
https://doi.org/10.1007/978-3-030-52705-1_31

that communities are disjoint, according to Palla *et al.* in [6], most real-world networks have overlapping community structure, that is, a node can belong to more than one community.

On the other hand, since the community detection problem has an NP-hard nature, most reported approaches use heuristics to search for a set of nodes that optimises an objective function which captures the intuition of community, these single-objective optimization approaches face two main difficulties: a) the optimization of only one function confines the solution to a particular community structure, and b) returning one single partition may not be suitable when the network has many potential structures. To overcome the aforementioned problems, many community detection algorithms model the problem as a Multi-objective Optimization Problem, and specifically, they use Multi-objective Optimization Evolutionary Algorithms (MOEAs) to solve them.

Once the set of non-dominated solutions is obtained by the MOEAs, one of the main challenges is to accomplish a final solution. Most of the proposed algorithms [5,7–9] use the internal criteria (e.g., Modularity Index [10]) or the external criteria (e.g., Normalized Mutual Information (NMI) [3]) to select the final solution. The drawbacks of these approaches are that the internal criteria does not often correspond to the objective function used by MOEAs and the external criteria uses the ground truth of the network, which it is not always known. Also, the selected final solutions obtained by both approaches do not use the knowledge of the overlapping communities (Pareto set) obtained by MOEAs.

Rough Set Theory (RST) may be used to evaluate significance of attributes, to deal with inconsistent data, and to describe dependencies among attributes, to mention just some uses in machine learning and data mining [22].

The main advantage of Rough Set Theory in data analysis is that it does not need any preliminary or additional information about data [17]. RST allows to approximate a rough concept by a pair of exact concepts, called the lower and upper approximations. The lower approximation is the set of objects definitely belonging to a vague concept, whereas the upper approximation is the set of objects possibly belonging to the mentioned vague concept [17]. The upper and lower approximations can be used in a broader context such as clustering, denoted as Rough Clustering [13].

In our proposal, we focus on describing the relationship between the elements of the network (vertices) only taking into consideration their belonging to the communities of the Pareto Set. Then, we use Rough Clustering to obtain a final covering of the network, that describes the communities with their lower and upper approximations. The lower approximation is the set of vertices belonging to the community without uncertainty, whereas the upper approximation is the set of vertices possibly belonging to this community, therefore located at the boundary of it. Hence, the selected final solution uses the knowledge of the overlapping communities (Pareto set) obtained by MOEAs.

In this paper, we propose an Overlapping Community Detection Algorithm using Multi-objective approach and Rough Clustering, denoted as MOOCD-RC. Our algorithm allows selecting the final solution based on the subjective

information as the number of vertices located in the cores or boundaries of the communities. As a consequence, it helps decision-makers (DM) incorporate their domain knowledge into the community detection process. Our main contributions are as follows:

1. We define an indiscernibility relationship between vertices of the network by taking the number of communities in the Pareto Set where they match.
2. We use the Rough Clustering foundation to build and describe the final covering of the network through the lower and upper approximations of the communities.

This paper is arranged as follows. Section 2 briefly introduces the necessary notions of multi-objective community detection problem and Rough Clustering. In Sect. 3, we introduce our proposal. Section 4 presents the experimental evaluation of our proposal and compared against other related state-of-the-art algorithms over synthetic networks. Finally, Sect. 5 gives the conclusions and some ideas about future work.

2 Background

This section introduces the necessary background knowledge for understanding the proposed method. First, the definition of multi-objective community detection problem and multi-objective algorithms of the related work are presented. Next, we will give the basics about Rough Set Theory and Rough Clustering.

2.1 Multi-objective Community Detection Problem

Let $G = (V, E)$ be a given network, where V is the set of vertices and E is the set of edges among the vertices. A multi-objective community detection problem aims to search for a partition P^* of G such that:

$$F(P^*) = min_{P \in \Omega} \left(f_1(P), f_2(P), \ldots, f_r(P)\right), \tag{1}$$

where P is a partition of G, Ω is the set of feasible partitions, r is the number of objective functions, f_i is the ith objective function and $min(\cdot)$ is the minimum value obtained by a partition P taking into account all the objective functions. With the introduction of the multiple objective functions, there is usually no absolute optimal solution, thus, the goal is to find a set of *Pareto* optimal solutions [2]. A commonly used way to solve a multi-objective community detection problem is by using MOEAs [9].

The first algorithm using MOEAs for detecting overlapping communities is named Multiobjective Evolutionary Algorithm to solve CDP (MEA_CDP) [5]. MEA_CDP uses an undirected representation of the solution and the classical Nondominated Sorting Genetic Algorithm II (NSGA-II) with the reverse operator to search for the solutions optimising the average community fitness, the average community separation and the overlapping degree among communities.

On the other hand, the Improved Multiobjective Evolutionary Algorithm to solve CDP (iMEA_CDP) [7] uses the same representation and optimization framework of MEA_CDP but it proposes to employ the PMX crossover operator and the simple mutation operator as evolutionary operators. iMEA_CDPs employs the Modularity function [10] and a combination of the average community separation and overlapping degree as its objective functions.

The Overlapping Community Detection Algorithm based on MOEA (MOEA-OCD) [9] uses the classical NSGA-II optimization framework and a representation based on adjacents among edges of the network. On the other hand, MOEA-OCD uses the negative fitness sum and the unfitness sum as objective functions. Unlike previously mentioned algorithms, in MOEA-OCD algorithm, a local expansion strategy is introduced into the initialization process to improve the quality of initial solutions.

Another algorithm is the Maximal Clique based on MOEA (MCMOEA) [8] which first detects the set of maximal cliques of the network and then it builds the maximal-clique graph. Starting from this transformation, MCMOEA uses a representation based on labels and the Multiobjective Evolutionary Algorithm based on Decomposition (MOEA/D) in order to detect the communities optimising the Radio Cut (RC) and Kernel K-Means (KKM) objective functions [11].

In [16] the authors combine Granular Computing and a multi-objective optimization approach for discovering overlapping communities in social networks. This algorithm, denoted as MOGR-OV, starts by building a set of seeds that is afterwards processed for building overlapping communities, using three introduced steps, named *expansion*, *improving* and *merging*.

Most of the exiting works focus on developing MOEAs to detect overlapping communities but not addresses the problem of selecting a final solution from the set of the obtained non-dominated solutions.

2.2 Foundations of Rough Clustering

The main components in the Rough Set Theory are an information system and an indiscernibility relation [17]. The classical RST was originally proposed using on a particular type of indiscernibility relations called equivalence relations (i.e., those that are symmetric, reflexive and transitive). Yao et al. [19] described various generalizations of rough sets by relaxing the assumptions of an underlying equivalence relation.

RST takes a pair of precise concepts to study the vagueness of a concept, named the lower and upper approximations. The lower approximation composes of all objects which surely belong to the concept, whereas the upper approximation contains all objects which perhaps belong to the concept. The boundary region of the vague concept is the difference between the upper and the lower approximations [18].

Lingras et al. [15] define another generalization of the approximate sets, seeing them as interval sets. The authors propose the rough k-means algorithm, where the concept of k-means is extended by viewing each cluster as an interval

or rough set. The core idea is to separate discernible from indiscernible objects and to assign objects to lower $\underline{A}(X)$ and upper $\overline{A}(X)$ approximations of a set X. This proposal allows overlaps between clusters [20]. The upper and lower approximation concepts require to follow some of the basic rough set properties such as [14]:

1. An object v can be part of at most one lower approximation. This implies that any two lower approximations do not overlap.
2. An object v that is member of a lower approximation of a set is also part of its upper approximation. This implies that a lower approximation of a set is a subset of its corresponding upper approximation.
3. If an object v is not part of any lower approximation it belongs to two or more upper approximations. This implies that an object cannot only belong to a single boundary region.

The way to incorporate rough sets into k-means clustering requires adapting the calculation of the centroids and deciding whether an object is assigned to a lower or upper approximation of a cluster. In the first moment, the centroids of clusters are calculated including the effects of lower as well as upper approximations. Next, an object is assigned to the lower approximation of a cluster when the distance (similarity) between the object and the particular cluster center is smaller than the distances to the remaining other cluster centers [14].

3 Proposal

The proposed algorithm obtains a final covering through two steps. It starts building sets of indiscernible (similar) objects that form basic granules of knowledge on the network $G = (V, E)$, where V represents the set of nodes and E represents the set of edges which connect nodes. Thus, a partition of the set V is obtained allowing us to define an equivalence relation in V. From our point of view, two vertices should be related if they share many communities at the Pareto Set. Next, through the Rough Clustering foundations, specifically the rough k-means algorithm ideas [15], we build the final covering of the network by viewing each community as a rough set, which allows us to obtain overlapping communities.

3.1 First Step: Build the Granules of Indiscernible Objects

In this step, we build a set of granules which represents a partition of V. First of all, we describe a series of useful concepts that we are applying in our proposal.

Definition 1 (Thresholded similarity graph). *Let $V = \{v_1, v_2, \ldots, v_n\}$ be the set of vertices of the network $G = (V, E)$, β a user-defined parameter and $S(v_i, v_j)$ a symmetric similarity function between vertices v_i and v_j, a thresholded similarity graph is an undirected graph $G_\beta = (V, E_\beta)$ where $(v_i, v_j) \in E_\beta$ if and only if $S(v_i, v_j) \geq \beta$.*

Definition 2 (Subgraph). *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs. $G_1 = (V_1, E_1)$ is a subgraph of $G_2 = (V_2, E_2)$, denoted as $G_1 \subseteq G_2$, if and only if $V_1 \subseteq V_2$ and $E_1 \subseteq E_2$.*

Definition 3 (Induced subgraph). *Let $V^{'} \subseteq V$ be a set of vertices, the subgraph of G induced by $V^{'}$ is $G^{'} = (V^{'}, E^{'})$, such that $E^{'} = \{(v_i, v_j) \in E \mid v_i, v_j \in V^{'}\}$.*

Definition 4 (β-Connected component). *Let $G_\beta = (V, E_\beta)$ be a thresholded similarity graph and $G^{'} = (V^{'}, E^{'})$ a subgraph of G_β. The subgraph $G^{'}$ is a β-connected component in G_β if and only if satisfies the following conditions:*

1. *$\forall u, v \in V^{'}, u \neq v$, exists $v_1, v_2, \ldots, v_q \in V^{'}$, such that $\forall i = 1 \ldots q$, $(v_i, v_{i+1}) \in E^{'}$ and also $v_1 = u$ and $v_q = v$ or $v_1 = v$ and $v_q = u$.*
2. *do not exist another subgraph of $G_\beta, G_1 = (V_1, E_1)$ with $G_1 \neq G^{'}$, that pleases the condition 1 and also $G^{'} \subseteq G_1$.*

Let $S_{_{ps}}(v_i, v_j)$ be the similarity function between v_i and v_j. $S_{_{ps}}(v_i, v_j)$ employs the solutions in the Pareto Set, denoted as PS. Let CV_i be a solution of PS and G_{v_i} the set of communities where v_i belongs. Let $mc(v_i, v_j)$ be the number of matching clusters between v_i and v_j in CV_i. The function $S_{_{ps}}(v_i, v_j)$ is defined as follows:

$$S_{_{ps}}(v_i, v_j) = \frac{\sum_{CV_i \in PS} match(v_i, v_j)}{ps} \tag{2}$$

where ps is the number of solutions in PS and $match(v_i, v_j) = \frac{mc(v_i, v_j)}{|G_{v_i}| \cdot |G_{v_j}|}$.

We build the thresholded similarity graph $G_\beta = (V, E_\beta)$ based on Eq. 2 and the user-defined parameter β ($\beta \in [0, 1]$). Let $G^{'}_r = \{G^{'}_{r_1}, G^{'}_{r_2}, \ldots, G^{'}_{r_q}\}$ be the β-connected component set. By definition, the connected component set in a graph constitutes a partition of the set of vertices.

We will say that a vertex $v_i \in V$ is related with a vertex $v_j \in V$, denoted as $v_i R_{_{ps}} v_j$, if and only if $\exists G^{'}_{r_i} \in G^{'}_r$ such that $v_i, v_j \in G^{'}_{r_i}$, being $R_{_{ps}}$ a equivalence relation. The set built from all the vertices related to a vertex v_i forms the so called *equivalence class* of v_i, denoted as $[v_i]_{R_{ps}}$. Therefore, $[v_i]_{R_{ps}}$ is the set of $v_j \in V$ such that share the same connected component $G^{'}_{r_i}$. This means that the vertices belonging to the same connected component have a strong relationship in terms of sharing the equal communities of PS. This strong relationship is measured by $S_{_{ps}}(v_i, v_j)$.

Let $EC = \{[v_1]_{R_{ps}}, [v_2]_{R_{ps}}, \ldots, [v_q]_{R_{ps}}\}$ be a set of equivalence classes under the indiscernibility relation $R_{_{ps}}$. The elements of EC are disjoint sets. Let $G_r = \{G_{r_1}, G_{r_2}, \ldots, G_{r_q}\}$ be the set of subgraphs induced by EC on $G = (V, E)$. Hence, G_{r_i} is a subgraph on $G = (V, E)$ induced from $[v_i]_{R_{ps}}$. Therefore, G_r is viewed as granules of indistinguishable elements which do not share vertices. These granules constitutes our initial granularity criterion [21], and also we will use them to build the final covering of the network.

3.2 Second Step: Build the Final Covering of $G = (V, E)$

We take the k biggest granules, $G_{r_i} \in G_r$, according to the number of vertices, as prototypes of clusters and the remaining of them are assigned to those selected ones. Therefore, the foundation is to initially covering the network with those granules of indistinguishable vertices that give greater coverage of the network. The variable $k, 1 \leq k \leq q$ receives the median value of the number of clusters that form the solutions at the Pareto Set. For this purpose, we define a similarity function between any two granules $G_{r_i}, G_{r_j} \in G_r$. This function is defined as follows:

$$S_{G_r}(G_{r_i}, G_{r_j}) = \frac{\sum_{v_i \in G_{r_i}} \sum_{v_j \in G_{r_j}} S_{ps}(v_i, v_j)}{|G_{r_i}| \cdot |G_{r_j}|} \tag{3}$$

As described in Sect. 2, the use of k-means clustering in Rough Clustering requires adapting the calculation of the centroids (cluster prototype) and decides whether an object is assigned to a lower or upper approximation of a cluster. In our case, we selected as prototypes of communities the k biggest granules, according to their number of vertices. Next, the remaining granules are assigned to those selected ones. A granule G_{r_i} is assigned to the lower approximation of a community when the similarity between G_{r_i} and the particular prototype of the community $G_{r_j}, 1 \leq j \leq k$, is much greater than the similarity to the remaining other prototypes. In this case, the similarity function defined in the Eq. 3 is used for deciding whether the remained granules are assigned to a lower or upper approximation of the selected k granules.

Worth noting that in this step, the assignation process uses the granules obtained in the previous step, $G_r = \{G_{r_1}, G_{r_2}, \ldots, G_{r_q}\}$. The selected k biggest granules represent the initial communities of network and also the lower approximations of them. The remaining granules $G_{r_i}, k < i \leq q$ will be part of the lower or upper approximations of the communities according to the similarity S_{G_r} and the γ user-defined parameter ($\gamma \in [0, 1]$).

The pseudocode of MOOCD-RC is shown in Algorithm 1. It is important to notice that the used Pareto Set is the result of using the MOGR-OV algorithm [16]. In MOOCD-RC, initially the cover CV is formed by the k greatest granules in G_r, which ones represent the lower approximations of the communities. These k selected granules represent the prototypes of communities to be built. Afterly, the remaining granules are included in the lower or upper approximations of the communities in CV according to S_{G_r}. Worth noting that the lower approximation of those communities are formed by the vertices that definitely belong to them, whereas the upper approximations are formed by the vertices that are located at the boundary of the communities. These vertices represent the overlapping in themselves.

In the first step, the building of the equivalence classes is tightly bound to the thresholded similarity graph $G_\beta = (V, E_\beta)$, which in turn depends on the β user-defined parameter. The higher the value of β the smaller granules will be obtained and vice versa. On the other hand, in the second step the dimensions of the lower and upper approximations of the communities depend on γ user-defined

Algorithm 1: MOOCD-RC algorithm

Input: $G = (V, E)$, Pareto Set with overlapping communities ($PSetOC$)
Output: Covering of the network $CV = \{CV_1, CV_2, \ldots, CV_k\}$
First Step: build the granules of indiscernible objects
for $v_i, v_j \in V$ **do**
  Take $PSetOC$ and compute $S_{ps}(v_i, v_j)$;
Build a thresholded similarity graph $G_\beta = (V, E_\beta)$;
Identify a β-connected component in G_β;
Compute $[v_i]_{R_{ps}}$ for each $v_i \in V$;
Build the set $Gr = \{G_{r_1}, G_{r_2}, \ldots, G_{r_q}\}$, subgraphs induced by each $[v_i]_{R_{ps}}, v_i \in V$;
Second Step: build the final covering of $G = (V, E)$
Sort descending Gr by number of vertices;
Select the first k granules $G_{r_i} \in Gr$ as prototypes of communities $CV_i \in CV$;
for $i = 1$ *to* k **do**
  $CV_i \leftarrow G_{r_i}$;
for $j = k + 1$ *to* q **do**
  Determine the most similarity between G_{r_j} and the k granules $G_{r_i} \in Gr$:
  $G_{r_{max}} \leftarrow \max_{1 \le i \le k} S_{G_r}(G_{r_j}, G_{r_i})$;
  $T \leftarrow \{\}$;
  for $i = 1$ *to* k **do**
    if $S_{G_r}(G_{r_j}, G_{r_i}) / S_{G_r}(G_{r_j}, G_{r_{max}}) \le \gamma$ **then**
      Add G_{r_i} to T;
  if $|T| > 1$ **then**
    $\forall G_{r_i} \in T$ take the community CV_i associated;
    Add G_{r_j} to $\overline{CV_i}$;
  else
    Take take the community CV_i associate to $G_{r_{max}}$;
    Add G_{r_j} to $\underline{CV_i}$ and $\overline{CV_i}$;
return CV

parameter. In the way of this parameter changes we will obtain boundaries of communities more or less tight.

The parameters β and γ allow decision-makers to obtain a final covering of the network by adjusting the cores or boundaries of the communities. In our experiments, we set $\beta = 0.75$ and $\gamma = 0.1$. We chose these values according to the related works [13,14,20].

4 Experimental Results

In this section, we conduct several experiments for evaluating the effectiveness of our proposal. Since the built-in communities in benchmark networks are already

known, we use the Normalized Mutual Information external evaluation measure to test the performances of different community detection algorithms.

Hence, the experiments were focused on evaluating the accuracy attained by our proposal in terms of the NMI value. Our algorithm was applied to synthetic networks generated from the Lancichinetti–Fortunato–Radicchi (LFR) benchmark dataset [4]. Its performances were compared against the one attained by MEA_CDP [5], iMEA_CDP [7], MCMOEA [8] and MOEA-OCD [9] algorithms, described in Sect. 2.

The algorithms of the related works do not build a final covering from the communities of the Pareto Set. Thus, we choose the best solution in the Pareto Set, according to the NMI, and compare this solution with respect to the ones obtained by our algorithm.

The NMI takes values in [0, 1] and it evaluates a set of communities based on how much these communities resemble a set of communities manually labeled by experts, where 1 means identical results and 0 completely different results.

In LFR benchmark networks, both node degrees and community sizes follow the power-law distribution and they are regulated using the parameters τ_1 and τ_2. Besides, the significance of the community structure is controlled by a mixing parameter μ, which denotes the average fraction of edges each vertex has with others from other communities in the network. The smaller the value of μ, the more significant community structure the LFR benchmark network has. The parameter O_n is specially defined for controlling the overlapping rate of communities in the network. O_n is the number of overlapping nodes, evaluating overlapping density among communities. Similar to μ, the higher the value of O_n, the more ambiguous the community structure is.

In the first part of the experiment, we set the network size to $N = 1000$, $\tau_1 = 2$, $\tau_2 = 1$, the node degree is in $[0, 50]$ with an average value of 20, whilst the community sizes vary from 10 to 50 elements. Using previous parameter values we vary μ from 0.1 to 0.6 with an increment of 0.05. After, we set $\mu = 0.1$ and $\mu = 0.5$, and we vary the percent of overlapping nodes existing in the network (parameter O_n of LFR Benchmark) from $0.1N$ to $0.5N$ with an increment of 0.1; the other parameters remain the same as the first experiment.

The average NMI value attained for each algorithm over the LFR benchmark when μ varies from 0.1 to 0.6 with an increment of 0.05, as show in Fig. 1. As the value of μ increases the performance of each algorithm deteriorates, being both MOEA-OCD and MOOCD-RC those that performing the best. As the mixing parameter μ exceeds 0.5, the MOEA-OCD algorithm begins to decline in its performance and it is outperformed by MOOCD-RC. Figure 1 shows the good performance of our method.

For summarizing the above results, we evaluated the statistical significance of the NMI values using the Friedman test as Non-Parametric Statistic Procedure included in the KEEL Software Tool. Also, we used the Holms and Finner as post hoc methods. Table 1 shows the average ranks obtained by each method in the Friedman test. Our method ranks second, however, Table 2 shows the overall performance of MOEA-OCD with respect to the remaining algorithms, where

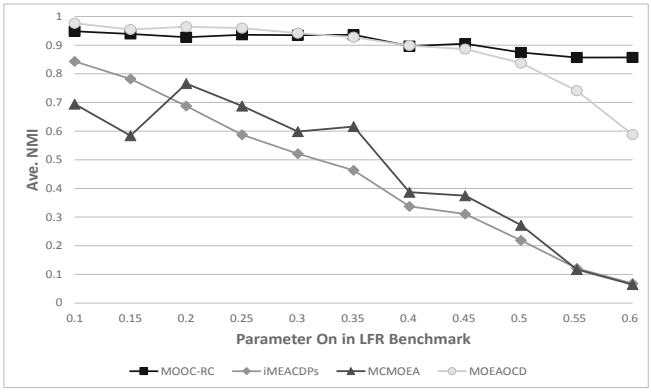

Fig. 1. Average NMI value attained by each algorithm on LFR benchmark networks when μ varies from 0.1 to 0.6 with an increment of 0.05.

Table 1. LFR benchmark networks when μ varies from 0.1 to 0.6. Average Rankings of the algorithms (Friedman).

Algorithm	Ranking
MOOCD-RC	1.5455
iMEACDPs	3.6364
MCMOEA	3.3636
MOEAOCD	1.4545

Table 2. LFR benchmark networks when μ varies from 0.1 to 0.6. Post Hoc comparison where $\alpha = 0.05$ (Friedman).

i	Algorithm	$z = (R_0 - R_i)/SE$	p	Holm	Finner
3	iMEACDPs	3.96347	0.000074	0.000222	0.000222
2	MCMOEA	3.468036	0.000524	0.001049	0.000786
1	MOOCD-RC	0.165145	0.86883	0.86883	0.86883

there is not statistically significance between our proposal and MOEA-OCD. The Friedman statistic value distributed according to chi-square with three degrees of freedom is 26.6727. Besides, the p-value computed by the Friedman test is 0.000007.

The structures of the networks are well defined in the second part of the experiment, as shown in Fig. 2. Our proposal and MOEA-OCD have a performance almost stable, independently of the number of overlapping nodes in the network, being MOEA-OCD the one that performs the best. On the other

hand, when the structure of the communities is uncertain, the performance of the MOEA-OCD algorithm drops off when the overlapping in the network increases, being our proposal the one that performs better, as shown in Fig. 3.

Similar to the previous experiment, we evaluated the statistical significance of the NMI values. Table 3 shows the average ranks obtained by each algorithm in the Friedman test. The Friedman statistic value distributed according to chi-square with three degrees of freedom is 25.92. Besides, the p-value computed by the Friedman test is 0.00001. Our algorithm ranks second, however, like the previous experiment, Table 4 shows the overall performance of MOEA-OCD with respect to the remaining algorithms, where there is not statistically significance between our proposal and MOEA-OCD.

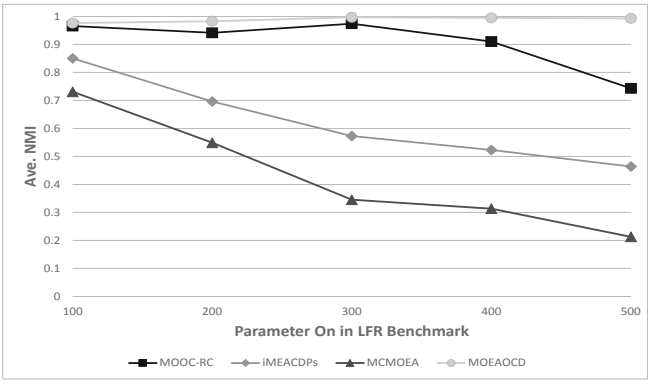

Fig. 2. Average NMI value attained by each algorithm on LFR benchmark networks when $\mu = 0.1$ and O_n varies from 100 to 500 with an increment of 100.

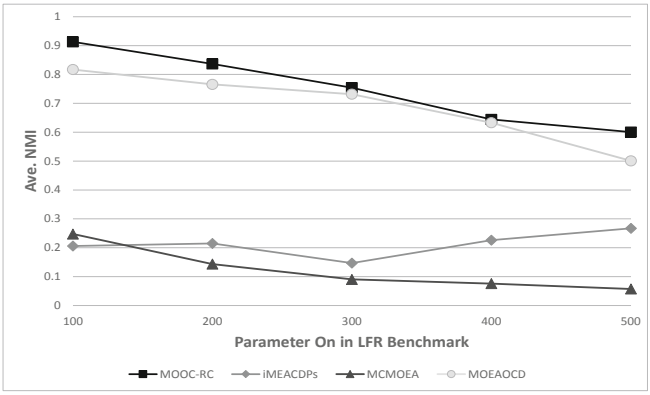

Fig. 3. Average NMI value attained by each algorithm on LFR benchmark networks when $\mu = 0.5$ and O_n varies from 100 to 500 with an increment of 100.

Table 3. LFR benchmark networks when $\mu = 0.1$, $\mu = 0.5$, and O_n varies from 100 to 500. Average Rankings of the algorithms (Friedman).

Algorithm	Ranking
MOOCD-RC	1.5
iMEA-CDPs	3.1
MCMOEA	3.9
MOEA-OCD	1.5

Table 4. LFR benchmark networks when $\mu = 0.1$, $\mu = 0.5$, and O_n varies from 100 to 500. Post Hoc comparison where $\alpha = 0.05$ (Friedman).

i	Algorithm	$z = (R_0 - R_i)/SE$	p	Holm	Finner
3	MCMOEA	4.156922	0.000032	0.000097	0.000097
2	iMEA-CDPs	2.771281	0.005584	0.011167	0.008364
1	MOOCD-RC	0	1	1	1

From the above experimental results, we can conclude that MOEA-OCD and our proposal have outstanding performances on LFR benchmark networks in most cases. However, our algorithm employs the information contained in the communities of Pareto Set to build a final covering of the network. Although the solutions of Pareto Set do not have overlapping communities, our proposal does not depend on this for building the final communities. Thus, our algorithm can be used by multi-objective evolutionary algorithms which build disjoint or overlapping community structures.

It should be noted that our proposal depends on the obtained non-dominated solutions. In these experiments we used the algorithm MOGR-OV [16] to generate the Pareto Set. On the other hand, the settings of β and γ have a narrow relationship over the obtained final covering. Following, we will give a brief description about this.

4.1 Community Structure Under Different Lower and Upper Approximation Scales

In the above experiments, the parameters β and γ are fixed to 0.75 and 0.1, respectively. We will have as results boundaries of communities more or less tight, depending on the way we change those parameters. Hence, both of them allow decision-makers to analyze the network according to the domain problem.

Using the synthetic network generated above with the parameters values $\mu = 0.1$ and $O_n = 0.1N$, we will show the overlapping communities with different

lower and upper approximation scales. For that, we change the γ parameter and keep the same β value used in the experiments. The parameter γ allows to tune the boundaries of communities. Thus, the higher the value of γ is, the wider the boundaries are and vice versa, which means that there is going to be more or less overlapping vertices, respectively.

Furthermore, we build two coverings of the obtained synthetic network by considering $\gamma = 0.1$ and $\gamma = 0.25$. For a better comprehension of the studied network we used the graph analysis tool Gephi. It employs both the network properties (e.g., vertex degree) and also the identified communities in the network in the visualization process. Figures 4 and 5 showed next were obtained using the Force Atlas 2 [23] method belonging to Gephi.[1]

As shown in Figs. 4 and 5, the covering obtained using $\gamma = 0.25$ shows boundaries of communities wider than the covering obtained with $\gamma = 0.1$. Thus, the communities showed in Fig. 5 have more overlapping vertices than communities showed in Fig. 4. The overlapped vertices are bigger visualized than others and they are placed in the boundaries of communities. As described before, the parameter γ allows the DM from its own knowledge to tight or wide the boundaries of communities. In this way, the decision maker has a mechanism to weigh the importance of lower and upper approximations in the obtained communities. However, the adjustment of β and γ has a direct control over the final covering. Worth noting that our algorithm builds the final covering only using the information about the communities of the Pareto Set.

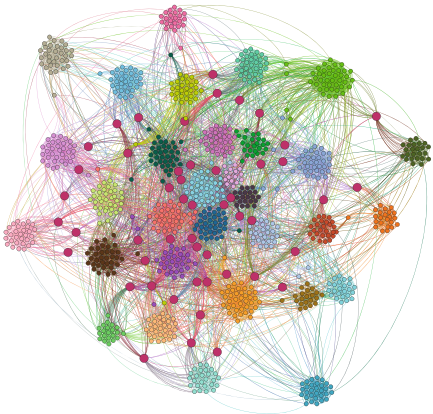

Fig. 4. Covering obtained over the obtained synthetic network based on the parameter values $\mu = 0.1$, $O_n = 0.1N$ and $\gamma = 0.1$.

[1] http://gephi.github.io/.

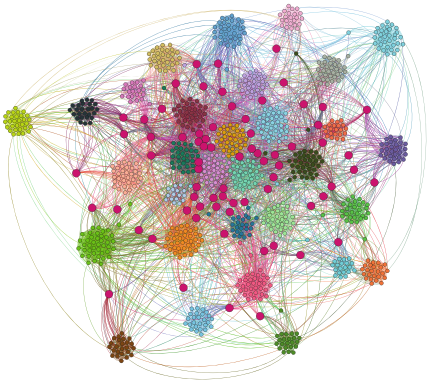

Fig. 5. Covering obtained over the obtained synthetic network based on the parameter values $\mu = 0.1$, $O_n = 0.1N$ and $\gamma = 0.25$.

5 Conclusions

In this paper, we proposed a new algorithm, named MOOCD-RC, for discovering overlapped communities through a combination of a multi-objective approach and Rough Clustering. It is composed of two steps: (a) build the granules of the indiscernible objects, and (b) build the final covering of network.

In the fist step, MOOCD-RC defined an equivalence relation between each pair of vertices of the network through the thresholded similarity graph. The obtained equivalence classes under the indiscernibility relation induce a granule set which constitutes our initial granularity criterion. We will also use them to build the final covering of the network. Afterward, in the second steps, the algorithm built the resulting communities through the Rough Clustering, taking the k greatest granules as prototypes of the communities; they also represent the lower approximations inside their own communities.

The MOOCD-RC algorithm was evaluated over synthetic networks in terms of its accuracy and it was compared against four algorithms of the related work. From the above experimental results, we can draw the conclusion that MOEA-OCD and our algorithm have outstanding performances on LFR benchmark networks in most cases. Moreover, this evaluation showed that MOOCD-RC is promising and effective for overlapping community detection in complex networks. As future work, we would like to make a more automatic adjustment to the β and γ parameters.

References

1. Maivizhi, R., Sendhilkumar, S., Mahalakshmi, G.S.: A survey of tools for community detection and mining in social networks. In: Proceedings of the International Conference on Informatics and Analytics. ACM (2016)

2. Shi, C., Yan, Z., Cai, Y., Wu, B.: Multi-objective community detection in complex networks. Appl. Soft Comput. **12**(2), 850–859 (2012)
3. Lancichinetti, A., Fortunato, S., Kertesz, J.: Detecting the overlapping and hierarchical community structure of complex networks. New J. Phys. **11**(3), 033015 (2009)
4. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Phys. Rev. E **78**(4), 046110 (2008)
5. Liu, J., Zhong, W., Abbass, H., Green, D.G.: Separated and overlapping community detection in complex networks using multiobjective evolutionary algorithms. In: IEEE Congress on Evolutionary Computation (CEC), (2010)
6. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature **435**, 814–818 (2005)
7. Liu, C., Liu, J., Jiang, Z.: An improved multi-objective evolutionary algorithm for simultaneously detecting separated and overlapping communities. Int. Nat. Comput. **15**(4), 635–651 (2015). https://doi.org/10.1007/s11047-015-9529-y
8. Wen, X., et al.: A maximal clique based multiobjective evolutionary algorithm for overlapping community detection. IEEE Trans. Evol. Comput. **21**, 363–377 (2016)
9. Yuxin, Z., Shenghong, L., Feng, J.: Overlapping community detection in complex networks using multi-objective evolutionary algorithm. Comput. Appl. Math. **36**(1), 749–768 (2015). https://doi.org/10.1007/s40314-015-0260-1
10. Shen, H., Cheng, X., Cai, K., Hu, M.B.: Detect overlapping and hierarchical community structure in networks. Phys. A Stat. Mech. Appl. **388**(8), 1706–1712 (2009)
11. Gong, M., Cai, Q., Chen, X., Ma, L.: Complex network clustering by multiobjective discrete particle swarm optimisation based on decomposition. IEEE Trans. Evol. Comput. **18**(1), 82–97 (2014)
12. Zhou, A., Qu, B.Y., Li, H., Zhao, S.Z., Suganthan, P.N., Zhang, Q.: Rough-fuzzy collaborative clustering. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **36**(4), 795–805 (2006)
13. Lingras, P., Chen, M., Miao, D.: Qualitative and quantitative combinations of crisp and rough clustering schemes using dominance relations. Int. J. Approximate Reasoning **55**(1), 238–258 (2014)
14. Lingras, P., Chen, M., Miao, D.: Applying rough set concepts to clustering. In: Peters, G., Lingras, P., Ślezak, D., Yao, Y. (eds.) Rough Sets: Selected Methods and Applications in Management and Engineering. Advanced Information and Knowledge Processing, pp. 23–37. Springer, London (2012). https://doi.org/10.1007/978-1-4471-2760-4_2
15. Lingras, P., West, C.: Interval set clustering of Web users with rough k-means. Technical Report No. 2002–002, Department of Mathematics and Computer Science, St. Mary's University, Halifax, Canada (2002)
16. Grass-Boada, D.H., Pérez-Suárez, A., Bello, R., Rosete, A.: Multiobjective overlapping community detection algorithms using granular computing. In: Bello, R., Falcon, R., Verdegay, J.L. (eds.) Uncertainty Management with Fuzzy and Rough Sets. SFSC, vol. 377, pp. 233–256. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-10463-4_13
17. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic, Dordrecht (1991)
18. Pawlak, Z., Andrzej, S.: Rough sets: some extensions. Inf. Sci. **177**, 28–40 (2006)
19. Yao, Y.Y., Lin, T.Y.: Generalization of rough sets using modal logic. Intell. Autom. Soft Comput. **2**(2), 103–120 (1996)
20. Mitra, S.: An evolutionary rough partitive clustering. Pattern Recogn. Lett. **25**(12), 1439–1449 (2004)

21. Mitra, S.: Granular computing: basic issues and possible solutions. In: Proceedings of the 5th Joint Conference on Information Sciences, no. 1, pp. 186–189 (2000)
22. Slowinski, R., Vanderpooten, D.: A generalized definition of rough approximations based on similarity. IEEE Trans. Knowl. Data Eng. **12**(2), 331–336 (2000)
23. Mathieu, J., Tommaso, V., Sebastien, H., Mathieu, B.: ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. PLoS ONE **9**(6), e98679 (2014)