Overlapping community detection using Pareto-based MOEAs and Spark GraphX Framework

Darian H. Grass-Boada[1], Airel Pérez-Suárez[1], Javier Fernández-Machin[1], Rafael Bello[2], and Alejandro Rosete[3]

[1] Advanced Technologies Application Center (CENATAV), Havana, Cuba
{dgrass,asuarez,jmachin}@cenatav.co.cu
[2] Dept. of Computer Science, Universidad Central "Marta Abreu" de Las Villas, Cuba.
{rbellop}@uclv.edu.cu
[3] Facultad de Ingeniería Informática, Universidad Tecnológica de la Habana "José Antonio Echeverría"(Cujae), Cuba.
{rosete}@ceis.cujae.edu.cu

Abstract. Overlapping community detection on social networks has received a lot of attention nowadays and it has been recently addressed as Multi-objective Optimization Evolutionary Algorithms (MOEAs). In this work we propose a multi-objective and evolutionary algorithm for overlapping community detection in social networks, named MOGR-PESA2, which builds an initial set of communities seeds.
Our algorithm employs the PESA-II framework and it proposes a new probabilistic evolutionary operator, which uses the information contained in the Pareto set in order to improve the heuristic search over the solution space. Moreover, we also propose the use of the Spark GraphX API in order to speeding up the building of the communities seeds.
The experimental evaluation over synthetic networks showed that our proposal is promising and effective for overlapping community detection in social networks. In addition, the inclusion of the Spark GraphX API allows our proposal to significantly accelerate the identification of community seeds.

Key words: Social Network Analysis, Overlapping Community Detection, Multi-objective Optimization Overlapping

1 Introduction

Social Network Analysis (SNA) [1] is a research field that deals with knowledge extraction tasks from domains characterized by the social relationships between individuals (i.e., social networks). Several algorithms from this field have been widely applied in different contexts, such as blogs networks, email exchange networks and public opinion analysis tasks, among others.

Complex systems like social networks show a modular organization in which elements of the same module or group have a certain similarity or affinity. The task of determining such modules is known as Community Detection in Social Networks and it has received a lot of attention nowadays due to its proved practical application [2]. Moreover, taken into account that most real-world networks have overlapping community structure [3], the study of the community detection problem has focused lately on the detection of overlapping communities.

Taking into account the NP-hard nature [4] of the community detection problem and knowing the drawbacks of using only a function for capturing the notion of community, many community detection algorithms model the problem as a Multi-objective Optimization Problem. In fact, most of the algorithms addressing the overlapping community detection problem in the literature use MOEAs for solving the Multi-objective Optimization Problem.

In [5] the authors propose a Pareto-dominance based MOEAs algorithm, named MOGR-OV, for overlapping community detection in social networks. MOGR-OV showed to outperforming the state-of-the-art overlapping algorithms, in terms of its accuracy, over real and synthetic social networks. However, this algorithm has two main drawbacks. The first one is that MOGR-OV does not make use of the information contained in the Pareto set in order to improve or to speeding up the evolutionary search process. The second one is that MOGR-OV is a single-solution based algorithm; that is, it produces each time a sole solution and thus, it takes longer to go through the search space.

In the Big Data era, where tons of data are created and stored continuously, the efficient detection of the communities presented in the social networks created from the above mentioned massive amount of data is an open issue. According to Nebro *et al.* in [7], the algorithms based on methaheuristics and evolutionary approaches are a good option for addressing the above mentioned issue. Nevertheless, addressing the community detection problem in the context of Big Data, from a multi-objective optimization approach is still a challenge [7].

Taking into account the aforementioned issues, in this work we propose a multi-objective and evolutionary algorithm for overlapping community detection in social networks, named MOGR-PESA2. Like MOGR-OV [5], our algorithm builds an initial set of communities seeds but it defines a property over the seeds which allow MOGR-PESA2 to select a most promising set of seeds. Our algorithm employs the PESA-II [8] framework and it proposes a new probabilistic evolutionary operator, which uses the information contained in the Pareto set in order to improve the heuristic search over the solution space. Additionally, with the aim of speeding up the construction and posterior evaluation of the solutions contained in the population, we use the JMetalSP framework propossed in [7] for large information volume processing. Moreover, we also propose the use of the Spark GraphX API in order to speeding up the building of the communities seeds, which is the more time-consuming task of our algorithm.

We conducted an experimental evaluation over several synthetic networks, generated from the LFR benchmark [10], in which we compared the accuracy our method attains against that of the related state-of-the-art algorithms; for mea-

suring the accuracy we use the NMI index [9]. These experimental results showed that MOGR-PESA2 outperforms the related overlapping community detection methods. Additionally, we evaluate the time our proposal spent for processing the whole network with and without the Spark GraphX implementation of the process where the communities seeds are built. From this last experiment we can conclude that the inclusion of the Spark GraphX API allows our proposal to speeding this process up to almost 6x in the best case.

The remainder of this paper is as follows: Section 2 briefly describes the algorithms most related with our research. On the other hand, in Section 3, we introduce our proposal, named MOGR-PESA2, whilst Section 4 presents an experimental evaluation, over synthetic networks, in which the performance of MOGR-PESA2 is tested and compared against other related algorithms. Finally, in Section 5 the conclusions as well as future work directions are given.

2 Related work

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 $G = \langle V, E \rangle$ is a given network, such that V is the set of vertices and E the set of edges among the vertices, 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 objectives functions. As it is known, the goal is to find a set of *Pareto* optimal solutions [4].

To the best of our knowledge, the multi-objective algorithms reporting in the literature for addressing the problem of overlapping community detection are: MEA_CDPs [11], IMOQPSO [14], OMO [12], iMEA_CDPs [13], MOEA-OCD [15] and MOGR-OV [5].

MEA_CDPs algorithm [11] uses an undirected representation of the solution and the classical NSGA-II framework with the reverse operator, in order to search for the solutions optimising three objective functions. iMEA_CDPs [13] uses the same representation as MEA_CDPs but it uses other objective functions as well as the MOEA/D as the optimization framework, together with the PMX and simple mutation operators. IMOQPSO [14] uses a center-based representation of the solution together with a combination of QPSO and HSA optimization frameworks, in order to find a set of nodes that optimises two previously defined objective functions. OMO [12] and MOEA-OCD [15] use the classical NSGA-II optimization framework and a representation based on adjacencies between edges of the network. OMO uses two objective functions whilst MOEA-OCD employs three. MOGR-OV is a single-solution MOEAs that starts by building an initial set of communities seeds which are then iteratively processed in order to detect overlapping zones in the network, to improve the overlapping quality

of these zones, and to merge communities having high overlapping, using three steps MOGR-OV introduces.

From all these methods, the MOGR-OV showed in [5] evidence that it achieves better accuracy results than the previous algorithms; however, MOGR-OV has two main drawbacks that could affect its behavior. The first one is that MOGR-OV does not make use of the information contained in the Pareto set in order to improve or to speeding up the evolutionary search process. The second one is that MOGR-OV is a single-solution based algorithm; that is, it produces each time a sole solution and thus, it could take longer to go through the search space. Moreover, MOGR-OV does assign the same importance to each community seed independently of the number of vertices it shares with its neighbors. This could have an impact both in the accuracy of the resulting set of communities as well as in the overall time MOGR-OV spends to process the whole network.

3 Our proposal

In this section, we introduce MOGR-PESA2 algorithm. MOGR-PESA2 is a population-based MOEAs which retains the strengthen of MOGR-OV by using some of its steps for building an initial population of solutions. This initial population is after used in an evolutionary process which aims to iteratively use the information contained in the Pareto Set in order to speeding up the whole process as well as to improve the accuracy of the solutions.

Following we describes the overall structure of our proposal.

3.1 MOGR-PESA2 for overlapping community detection in social networks

MOGR-PESA2 starts by computing the *similarity class* of each vertex in the network as well as its *score*. This similarity classes are viewed as communities seeds.

We will say that a vertex $v_j \in V$ is related with a vertex $v_i \in V$, denoted as $v_i R v_j$, iff $|N(v_i) \cap N(v_j)| > \frac{1}{2} \cdot |N(v_j)|$; where $N(v)$ is the set of adjacent vertices of v. The set built from all the vertices related to a vertex v_i forms the so called similarity class of v_i.

The score of the similarity class defined by v_i, denoted as $Score(v_i)$, expresses the goodness of the similarity class of v_i for being selected as a community seed and it is computed as follows:

$$Score(v_i) = \frac{\sum_{v_j \in N(v_i)} \frac{|N(v_i) \cap N(v_j)|}{|N(v_j)|}}{|N(v_i)|} \tag{2}$$

Once the similarity class and the score of each vertex were computed, we use them in order to build a solution of the problem.

Let $Gr = \{g_1, g_2, \ldots, g_n\}$ be the set of the subgraphs induced by the similarity class of each vertex $v_i \in V$. For building a solution, we iteratively apply the

roulette wheel selection method over Gr, where the probability of being selected of a subgraph $g_j \in Gr$ is computed by using the number of unclustered vertices belonging to g_j plus the $Score\,(v_i)$. The selected subgraph g_j is then processed using the *expansion*, *improving* and *merging* steps introduced by MOGR-OV. Once the solution is built, it is added to the Pareto set iff it is a nondominated solution; for this purpose we employ the same two objective functions used by MOGR-OV. This whole process is repeated until we have an initial population of size m.

Afterwards, MOGR-PESA2 computes the *aptitude* for each similarity class of G. The aptitude of a similarity class is computed as the sum of its score and the ratio of solutions in the population which contain that similarity class as part of one of their communities. From this point onwards, MOGR-PESA2 performs an evolutionary process comprised of four steps, for a predefined number of iterations. In the first step, the selection operator of the PESA-II [8] framework is used to obtain a subset P' of the current Pareto Set. After, in the second step the aptitude of each similarity class is updated using P'. In the third step, MOGR-PESA2 builds an offspring of size m, using the same procedure described in the previous paragraph but using the aptitude of a similarity class instead of its score for computing the probability of being selected of that similarity class. Using this offspring the non dominated solutions are added to the Pareto Set in the fourth step, and all the previous steps are repeated until the stop condition is fulfilled.

3.2 Using Spark GraphX for speeding up the similarity class computation

Taking into account that the computation of the similarity class of each vertex has a computational complexity of $O(n^3)$, as well as the time MOGR-PESA2 spends in this step in preliminary experiments, we decided to implement this step using the Spark GraphX API. For this purpose, each edge (v, u) are processed using *aggregateMessage* function provided by this API, in such a way that v sends a message to u iff $v_i R v_j$, and vice versa.

4 Experimental evaluation

In this section, we conduct several experiments for evaluating the effectiveness of our proposal.

The experiments were focused on: 1) to evaluate the accuracy attained by our proposal and to compare it against the one attained by MOGR-OV [5] algorithm, which has reported the best results; and 2) to measure the time consuming by our algorithm with and without the Spark GraphX implementation of the process where the similarity class are built.

The experiments were conducted over 11 synthetic networks generated using the Lancichinetti-Fortunato-Radicchi (LFR) benchmark [10], which is suitable for both separated and overlapping situations.

In LFR benchmark networks, both node degrees and community sizes follow the power-law distribution and they are regulated using 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 in the network has with other communities. The smaller the value of μ, the more significant community structure the LFR benchmark network has. For the two first experiments we set network size to 1000, $\tau_1 = 2$, $\tau_2 = 1$, the node degree is in $[0, 50]$ with an average value of 20, whilst the community sizes varies from 10 to 50 elements. Using previous parameters values we vary μ from 0.1 to 0.6 with an increment of 0.05 and consequently, we built 11 different synthetic networks.

As we mentioned before, in the first experiment we compare the accuracy attained by our proposal against that attained by MOGR-OV algorithm, over the 11 synthetic networks. For evaluating the accuracy of each algorithm we used the NMI external evaluation measure, proposed by Lancichinetti et al. in [9]. 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.

For each algorithm, we executed it over each network and we selected the highest NMI value attained by a solution of each resulting Pareto set. This experiment is repeated twenty times for each network and, we computed the average of the highest NMI values attained all these times. Table 1 shows the average NMI attained by each algorithm over each network.

Table 1. Comparison MOGR-OV with MOGR-PESA2, regarding the NMI value. Best values appears bold-faced

Algorithms	Net 1	Net 2	Net 3	Net 4	Net 5	Net 6	Net 7	Net 8	Net 9	Net 10	Net 11
MOGR-OV	0.98	0.95	0.96	0.96	0.95	0.96	0.94	0.94	0.91	0.72	0.7304
MOGR-PESA2	0.98	**0.97**	0.96	**0.97**	0.95	0.96	**0.95**	**0.95**	0.91	**0.74**	**0.736**

As it can be seen in Table 1, when the structure of the networks is well defined, MOGR-OV, and MOGR-PESA2 have a performance almost stable. However, when the structure of the communities is uncertain, MOGR-PESA2 has better results.

Finally, in the last experiment we generate other five synthetic networks. For this purpose, we use as network sizes 10K, 20K, 30K, 50K and 100K and, for each of these sizes, we set $\tau_1 = 2$, $\tau_2 = 1$, $\mu = 0.1$; the average degree and the community sizes were settle as the 2% and 5% of the network size, respectively.

Using these networks, we evaluate the time consuming by our algorithm with and without the Spark GraphX implementation of the process where the communities seeds are built (see Section 3.2). Figure 1 shows the results of this comparison. In this figure, the black line refers to our proposal with the Spark GraphX implementation, whilst the gray line refers to our original proposal.

The Spark GraphX implementation run in the cluster of 4 virtual machines, each one with 8 cores Intel(R) Xeon(R) CPU E5-2670 v2 @ (2.50 GHz), 48 GB RAM. These virtual machines are used as Slave nodes with the role of TaskTracker (Spark). The Master node, which coordinates the distribution of tasks, is hosted in a different machine with 8 Intel(R) Xeon(R) CPU E5-2670 v2 @ (2.50 GHz), 12 GB RAM. The original proposal run in PC Intel Core i5-4440 (3.10 GHz) CPU, 4 GB RAM.

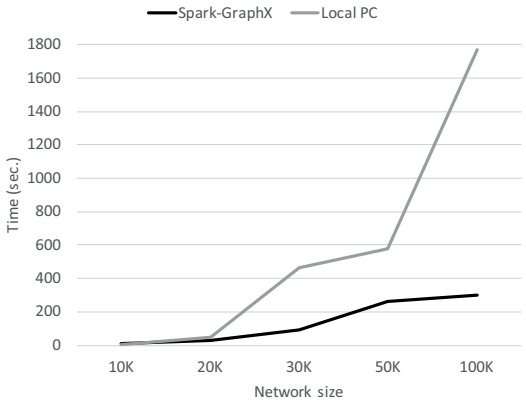

Fig. 1. Time consuming by our algorithm with and without the Spark GraphX implementation of the first step.

As it can be seen in Figure 1, the implementation of the aforementioned step in Spark GraphX API allows our proposal to consume less time. In fact, we get an speeding up of almost 6x in the best case.

5 Conclusions

In this work, we proposed a multi-objective and evolutionary algorithm for overlapping community detection in social network. Our algorithm, named MOGR-PESA2, uses the information contained in the Pareto set in order to improve the heuristic search over the solution space.

Additionally, with the aim of speeding up the construction and posterior evaluation of the solutions contained in the population, we use the JMetalSP framework and we propose a variant of MOGR-PESA2 which uses the Spark GraphX API to speeding up its most time-consuming step.

Our proposal was evaluated over 11 synthetic networks in terms of its accuracy, measured using the NMI index, and compared against the related algorithm showing the best performance in the literature. These experiments showed our proposal is promising and effective for overlapping community detection in social networks. Finally, these experiments showed also that the use of the Spark

GraphX API is a good alternative for scale our proposal for large and very large networks.

As future works, we would like to obtain a full Spark GraphX implementation of our proposal in order to being able to process bigger networks.

References

1. Wasserman, S., Faust, K.: Social network analysis: Methods and applications. Volume 8. Cambridge university press (1994)
2. Zhou, Y., Wang, J., Luo, N., Zhang, Z.: Multiobjective local search for community detection in networks. Soft Computing (2015)
3. Palla G, Derznyi I, Farkas I, Vicsek T: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435:814-818 (2005)
4. Shi, C., Yan, Z., Cai, Y., Wu, B.: Multi-objective community detection in complex networks. Applied Soft Computing 12(2) (2012)
5. Grass-Boada, D.H., Prez-Surez, A., Bello, R., Rosete, A.: Multiobjective overlapping community detection algorithms using granular computing. Book chapter Uncertainty Management with Fuzzy and Rough Sets, Studies in Fuzziness and Soft Computing, pp 233-256, 2019
6. Yao, Y.Y. et al.: Granular computing: basic issues and possible solutions. In: Proceedings of the 5th joint conference on information sciences, pp 186-189, 2000
7. Barba-Gonzaléz, C., García-Nieto, J., Nebro, Antonio J., Aldana-Montes, José F.: Multi-objective big data optimization with jmetal and spark. In: International Conference on Evolutionary Multi-Criterion Optimization, 16-30, Springer (2017)
8. Corne, David W., Jerram, Nick R., Knowles, Joshua D., Oates, Martin J.: PESA-II: Region-based selection in evolutionary multiobjective optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001), (2001)
9. A. Lancichinetti, S. Fortunato, J. Kertesz: Detecting the overlapping and hierarchical community structure of complex networks. New Journal of Physics 11(3):033015 (2009)
10. A. Lancichinetti, S. Fortunato, F. Radicchi: Benchmark graphs for testing community detection algorithms. Phys. Rev. E 78(4):046110 (2008)
11. 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)
12. Liu, B., Wang, C., Wang, C., Yuan, Y.: A New Algorithm for Overlapping Community. In: Proceeding of the 2015 IEEE International Conference on Information and Automation Detection, 813-816 (2015)
13. Liu, C., Liu, J., Jiang, Z.: An improved multi-objective evolutionary algorithm for simultaneously detecting separated and overlapping communities. An international journal of Natural Computing, 15(4):635-651 (2016)
14. Li, Y., Wang, Y., Chen, J., Jiao, L., Shang, R.: Overlapping community detection through an improved multi-objective quantum-behaved particle swarm optimization. Journal of Heuristics, 21(4):549-575 (2015)
15. Yuxin, Z., Shenghong, L., Feng, J.: Overlapping community detection in complex networks using multi-objective evolutionary algorithm. Computational and Applied Mathematics, 36(1):749-768 (2017)