Applied Soft Computing 112 (2021) 107838

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A fast variable neighborhood search approach for multi-objective community detection

Sergio Pérez-Peló, Jesús Sánchez-Oro, Antonio Gonzalez-Pardo, Abraham Duarte *

Dept. Computer Science, Universidad Rey Juan Carlos, C/Tulipán, S/N, Móstoles, Spain

ARTICLE INFO

Article history:
Received 16 February 2021
Received in revised form 14 August 2021
Accepted 15 August 2021
Available online 27 August 2021

Keywords:
Community detection
Variable Neighborhood Search
Greedy Randomized Adaptive Search Procedure
Metaheuristic
Multi-objective

ABSTRACT

Community detection in social networks is becoming one of the key tasks in social network analysis, since it helps analyzing groups of users with similar interests. This task is also useful in different areas, such as biology (interactions of genes and proteins), psychology (diagnostic criteria), or criminology (fraud detection). This paper presents a metaheuristic approach based on Variable Neighborhood Search (VNS) which leverages the combination of quality and diversity of a constructive procedure inspired in Greedy Randomized Adaptative Search Procedure (GRASP) for detecting communities in social networks. In this work, the community detection problem is modeled as a bi-objective optimization problem, where the two objective functions to be optimized are the Negative Ratio Association (NRA) and Ratio Cut (RC), two objectives that have already been proven to be in conflict. To evaluate the quality of the obtained solutions, we use the Normalized Mutual Information (NMI) metric for the instances under evaluation whose optimal solution is known, and modularity for those in which the optimal solution is unknown. Furthermore, we use metrics widely used in multi-objective optimization community to evaluate solutions, such as coverage, ϵ-indicator, hypervolume, and inverted generational distance. The obtained results outperform the state-of-the-art method for community detection over a set of real-life instances in both, quality and computing time.

Code metadata

Permanent link to reproducible Capsule: https://doi.org/10.24433/CO.9934279.v1.

1. Introduction

In recent years, the growth and development of social networks has caused scientists from different areas of knowledge to be interested in the study of their structure and its implications. Users of social networks are increasing every day, which has made them a very common source of data. Analyzing how the users are related between them, or how the information that they are sharing is intertwined, we can potentially obtain additional information that can be useful for other interests. For example, we can estimate the potential impact of a marketing campaign, what the general opinion about a certain topic is, what the users think about a company, a person or a service, etc.

In addition, the growth of big data techniques and concepts such as smart cities have led to the need for real-time analysis of large amounts of information quickly and efficiently. However, most of the traditional techniques are not adapted to deal with vast amounts of data, becoming unsuitable for most of the current challenges in social network analysis [1]. An efficient and effective analysis of social networks can report a high amount of benefits, so it is interesting to have a set of powerful algorithms that allow us to perform that analysis.

Most of the networks represent complex models with a large amount of data and interactions. The analysis of social networks are able to evaluate properties such as small world [2] or scale free [3], easing the understanding of those real-world networks. There is a special property that attracts the gaze of the scientific community, which is the community structure [4]. This property is able to shed light over several problems of different research areas: from social science to biological science, among others. See [5] for a thorough survey on community detection.

This work proposes a Variable Neighborhood Search (VNS) algorithm [6] for solving the Community Detection Problem (CDP) from a multi-objective perspective, resulting in the Multi-Objective Community Detection Problem (MOCDP). Every traditional objective function considered for the single-objective CDP presents one or more drawbacks for modeling the community structure of a given solution. In order to deal with this problem, the model of CDP as a multi-objective optimization problem is

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail addresses: sergio.perez.pelo@urjc.es (S. Pérez-Peló), jesus.sanchezoro@urjc.es (J. Sánchez-Oro), antonio.gpardo@urjc.es (A. Gonzalez-Pardo), abraham.duarte@urjc.es (A. Duarte).

https://doi.org/10.1016/j.asoc.2021.107838

becoming more relevant for the scientific community. Notice that considering more than one objective at the same time, which may complement among them, allows us to model the community structure of a network more precisely.

Although VNS has been traditionally considered for single-objective optimization, recent works have adapted the original VNS framework for tackling multi-objective optimization problems, resulting in robust algorithms which are competitive with the best methods in the literature [7,8].

The main contributions of this work are the following:

- An adaptation of the well-known VNS metaheuristic is proposed for dealing with multi-objective optimization problems. It considers that a solution in the VNS framework is the complete set of non-dominated solutions.
- The initial set of non-dominated solutions for the VNS is generated using the constructive phase of Greedy Randomized Adaptive Search Procedure (GRASP). This feature allows VNS to start the search from a set of diverse and high-quality solutions.
- A new greedy function that can be efficiently computed is proposed for the constructive procedure and for the local search method, with the aim of guiding the search for promising regions of the search space without requiring large computing times.
- Two local search methods are presented: the first one, Combined Search Procedure, follows the traditional multi-objective local optimization methods. However, the Independent Search Procedure is a new proposal which tries to further improve the set of non-dominated solutions by independently improving each considered objective.
- The quality of the proposal is analyzed through multi-objective optimization perspective. As far as we know, previous works only consider the metrics related with social network analysis, but it is interesting to evaluate the set of non-dominated solutions under multi-objective optimization metrics which are specifically designed to that end.

The paper is structured as follows: Section 2 formally defines the considered problem, as well as the metrics proposed for the evaluation of solutions; Section 3 briefly reviews the most relevant papers related with our research; Section 4 presents the new Variable Neighborhood Search-based procedure proposed for detecting communities and exposes how the initial solution set is generated with a constructive procedure, as well as the neighborhood structures considered within Variable Neighborhood Search; Section 5 introduces the computational experiments performed to test the quality of the proposal; and finally Section 6 draws some conclusions on the research.

2. Problem definition

A social network, conformed with a set of users and a set of relations among them, can be modeled as a graph $G = (V, E)$. Users are represented by the set of nodes V, with $|V| = n$, while relations among users are represented by the set of edges E, with $|E| = m$. Notice that an edge $(u, v) \in E$ indicates that users $u, v \in V$ are related in the social network. The kind of relation between the users strictly depends on the purpose of the social network (friendship, work, etc.). This paper considers bidirectional relationships. Thus, if there is a relation (u, v), then the relation (v, u) is also contemplated (i.e., G is an undirected graph).

The aim of this work is to deal with the Community Detection Problem (CDP) following a multi-objective approach. A community $C_i \subseteq V$ inside a network G is defined as a set of users and the relations that connect those users. In other words, the community C_i is represented by the induced subgraph $G_i = (C_i, E_i)$, where $E_i = \{(u, v) \in E : u, v \in C_i\}$. The CDP then consists in separating the complete social network into communities or groups. Although there is not a formal definition for community in the literature, the most widely accepted definition considers that a community is a group of users that are closely related to each other (i.e., share some properties/interests in the social network).

In terms of graphs, a well-detected community is the one whose nodes are densely connected among them and sparsely connected to nodes which do not belong to the community. Given a community C_i, the edges that connect nodes in the same community are usually known as intra-community edges, $\mathcal{E}_{\blacktriangleleft}(C_i)$, while those connecting nodes in different communities are named as inter-community edges, $\mathcal{E}_{\blacktriangleright}(C_i)$. In mathematical terms,

$$\mathcal{E}_{\blacktriangleleft}(C_i) = \{(u, v) \in E : u, v \in C_i\}$$
$$\mathcal{E}_{\blacktriangleright}(C_i) = \{(u, v) \in E : u \in C_i \wedge v \notin C_i\}$$

Following this definition, a community C_i in a social network is well defined if it presents a large number of intra-community edges $\mathcal{E}_{\blacktriangleleft}(C_i)$ and, at the same time, a small number of inter-community ones $\mathcal{E}_{\blacktriangleright}(C_i)$.

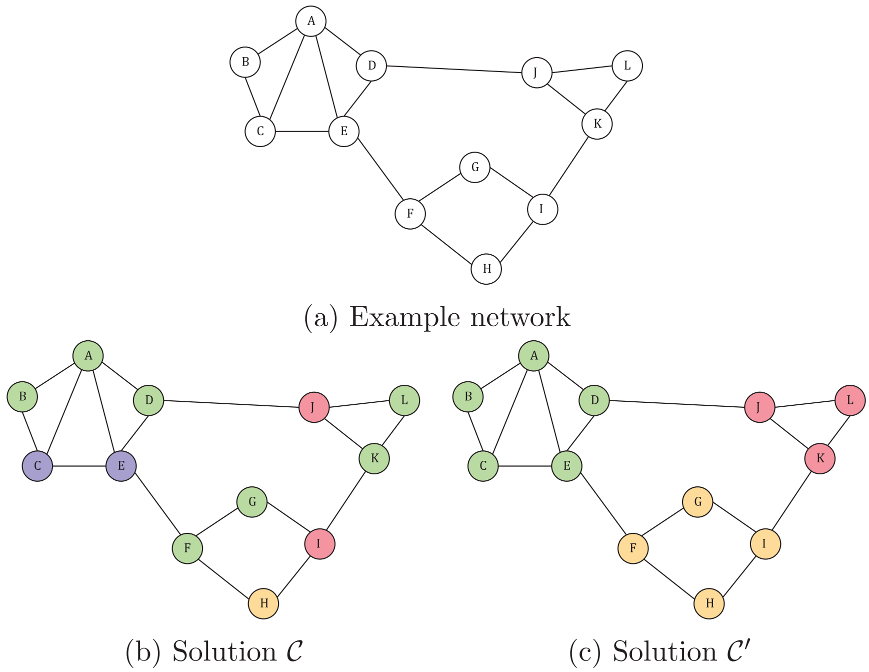
Given a social network, the CDP consists in assigning each user to a single community. Each community is labeled with an integer number i, with $1 \leq i \leq c \leq n$, being c the number of communities detected. Depending on the problem under consideration, the number of communities may be fixed or not [9]. In the CDP variant tackled in this paper the number of communities is not fixed *a priori*.

A solution $\mathcal{C}$ for the CDP is modeled as the set of communities $\mathcal{C} = \{C_1, C_2, \ldots, C_c\}$ of the network. Then, a solution for the CDP is feasible when all the nodes have been assigned to a single community, i.e., $\sum_{i=1}^{c} |C_i| = n$ and $C_i \cap C_j = \emptyset$ for $1 \leq i, j \leq j$ with $i \neq j$.

Fig. 1(a) shows an example of a network with 12 nodes and 17 edges. Figs. 1(b) and 1(c) shows two feasible solutions $\mathcal{C}$ and $\mathcal{C}'$, respectively, for the CDP, where each node is colored with a different color that corresponds to its community (1-green, 2-red, 3-yellow, 4-blue). The first solution is represented as $\mathcal{C} = \{C_1, C_2, C_3, C_4\}$, where $C_1 = \{\mathtt{A}, \mathtt{B}, \mathtt{D}, \mathtt{F}, \mathtt{G}, \mathtt{K}, \mathtt{L}\}$, $C_2 = \{\mathtt{J}, \mathtt{I}\}$, $C_3 = \{\mathtt{H}\}$, $C_4 = \{\mathtt{C}, \mathtt{E}\}$. Similarly, the second solution is defined as $\mathcal{C}' = \{C'_1, C'_2, C'_3\}$, where $C'_1 = \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}\}$, $C'_2 = \{\mathtt{J}, \mathtt{K}, \mathtt{L}\}$, $C'_3 = \{\mathtt{F}, \mathtt{G}, \mathtt{H}, \mathtt{I}\}$.

Although solution depicted in Fig. 1(c) is clearly more visually appearing than the one presented in Fig. 1(b), there is not a common criterion to decide whether a solution presents a good community detection or not. There are several widely accepted metrics to evaluate the community structure of a solution. In particular, modularity [10] is one of the most extended metrics, and it has been used by several bioinspired algorithms [5,11] to find high quality solutions. However, it has some disadvantages. On the one hand, maximizing modularity is an $\mathcal{NP}$-hard problem [12]. On the other hand, a large value of modularity does not necessarily indicates that the communities detected are realistic since, in some cases, random networks without community structure can present large modularity values [13]. Last but not least, modularity has the well-known problem of resolution limitation [14]. This problem refers to the fact that the maximization of modularity is not able to reveal communities which are smaller than a certain scale, depending on the network size and on the degree of connections among real communities.

Most of the previous works are focused on the single-objective variant of the CDP (see for instance [15,16]). However, it may be interesting to consider more than one objective at the same time since it could lead us to find new and more reliable communities



(a) Example network

(b) Solution $\mathcal{C}$

(c) Solution $\mathcal{C}'$

Fig. 1. Example of two different solutions for the CDP over an example network.

in the networks. In a multi-objective optimization problem, two or more objectives are in conflict with each other, which means that improving one of the objectives usually leads to deteriorate the other objectives. Therefore, there is not a single solution with the optimal value in all the considered objectives. The main goal in multi-objective programming is to find a set of non-dominated solutions.

This work tackles the CDP from a multi-objective perspective, resulting in the Multi-Objective Community Detection Problem (MOCDP). In particular, two conflicting objectives are considered: Negative Ratio Association (NRA) and Ratio Cut (RC). The former measures the percentage of intra-community edges that exists with respect to the size of the community, while the latter evaluates the percentage of inter-community edges in a community with respect to its size. In mathematical terms, NRA and RC of a cluster i are defined as:

$$NRA(C_i) = -\frac{\mathcal{E}_{\blacktriangleleft}(C_i)}{|C_i|} \qquad RC(C_i) = \frac{\mathcal{E}_{\blacktriangleright}(C_i)}{|C_i|}$$

Similarly, the NRA and RC of a complete solution $\mathcal{C}$ are defined as:

$$NRA(C) = \sum_{C_i \in C} NRA(C_i) \qquad RC(C) = \sum_{C_i \in C} RC(C_i)$$

being c the number of communities in the corresponding solution.

Following these definitions, a solution with small NRA and RC values presents a good community structure. These two metrics were proven to be in conflict in [17]. Analyzing the objective functions independently, optimizing only NRA usually results in solutions with small communities which are densely connected, while focusing only in RC leads us to obtain solutions with large communities. Notice that, dealing with both metrics simultaneously, allows us to overcome the drawbacks of each metric when considered independently. The MOCDP is focused on minimizing both objectives, NRA and RC.

Let us illustrate how we can evaluate these two metrics by considering the example introduced in Fig. 1. Specifically, for solution $\mathcal{C}$, we have:

$$NRA(\mathcal{C}) = -\frac{4}{7} - \frac{0}{2} - \frac{0}{1} - \frac{1}{2} = -1.07$$

$$RC(\mathcal{C}) = \frac{11}{7} + \frac{6}{2} + \frac{2}{1} + \frac{5}{2} = 9.07$$

Similarly, for solution $\mathcal{C}'$:

$$NRA(\mathcal{C}') = -\frac{7}{5} - \frac{3}{3} - \frac{4}{4} = -3.4$$

$$RC(\mathcal{C}') = \frac{2}{5} + \frac{2}{3} + \frac{2}{4} = 1.57$$

As it can be derived from the equations, the best solution with respect to both NRA and RC is $\mathcal{C}'$, since it presents the minimum values in both objective functions. Therefore, we can also conclude that $\mathcal{C}'$ dominates $\mathcal{C}$.

3. Literature review

Community detection problems (CDP) have attracted the interest of the scientific community in the last years, mainly due to the relevance of the results derived from this research. It is possible to find relevant research works in the literature that describe how community detection can be applied in real-world environments, thus finding an interesting utility for this area of research. For example, some works use Community Detection techniques in the area of cybersecurity, where the goal is to reduce the threats over a certain cluster of actives using these techniques. As an example, in [18] authors propose a modularity-based adaptive algorithm applied to social-aware message forwarding strategy in MANETs (Mobile Ad Hoc Networks) and worm propagation containment in Online Social Networks. A different field in which these algorithms could be useful is Business Intelligence and Business Science, where certain topics can be modeled as a network. For instance, in [19], we can find an application in business science in terms of topological features and nodal attributes.

Other research field that focus the attention of the research community is politics. In the last years there has been a huge increase in the use of social networks for political purposes. In this domain, there are two main problems that can be solved: topic opinion and political polarization. The former refers to those works whose goal is to understand what users think about a specific topic. The latter contains the works that try to align SN users with the different political parties. An example of topic opinion is [20] where authors tried to classify citizen's voting intention based on the tweets published during the Scottish Independence Referendum in 2014.

An example about political polarization, is the work published by Borge-Holthoefer et al. [21]. In this work, authors analyzed the social structure and the content of the tweets published by the users to understand the opinion evolution in Egypt during the summer of 2013. In this summer, there was a military takeover that resulted in an increase of polarized tweets but authors did not observe an ideological shift in the users.

In addition to the application domains, it is necessary to know which algorithms, or techniques, are the most popular for detecting the communities. In this sense, it is important to highlight that community detection it is a complex problem that it is difficult to solve by using classical algorithms. For this reason, it is really common to find researchers that use heuristics algorithms, and more precisely, bioinspired techniques [22].

In the literature, exact methods devoted to solve the community detection problem can be found, although they are not very efficient in solving the problem when the networks to be analyzed are too large. However, recent works have been focused on proposing new exact algorithms for dealing with large networks. Srinivas and Rajendran [23] propose a mathematical model for finding community structure on influential nodes, testing it in large scale networks. Although the performance is close to the state of the art, it is not able to reach a solution for networks with 115 nodes in a time limit of six hours, highlighting the relevance of using heuristic approaches. In the same way, Alinezhad et al. [24] propose a mathematical formulation for solving the community detection problem in attributed networks, considering both topological and node attributes. They limit the computing time to 7200 s for the exact procedure, since it is not able to converge in a reasonable computing time. The computational requirements of the exact procedures confirms the necessity of considering heuristic approaches for obtaining high quality solutions in small computing times.

In the area of bioinspired computation, evolutionary approaches are the most popular. It is important to highlight the review performed by Pizzuti in [25] about Evolutionary Computation (EC) techniques to detect communities in networks. An interesting work about EC is the work published by Said et al. [26], where authors designed a clustering coefficient-based genetic algorithm able to detect cohesive groups from dense graphs and also, communities in sparse networks. Other relevant work is [27] that presents a genetic algorithm that uses a multi-individual ensemble learning-based crossover function. The algorithm is improved with a local search strategy to speed up the convergence.

Other well-known bioinspired algorithms are the ones belonging to swarm intelligence. In this new group, the most popular algorithms are Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO). These two algorithms are inspired by the social behavior of birds within a flock, and the behavior of ants seeking a path from the nest to the source of food, respectively. PSO has been successfully used for CDP in [28], where a discrete PSO algorithm is used to extract the communities in large-scale social networks by optimizing the modularity. Regarding ACO algorithm, this algorithm has been used to extract high-quality communities in Ego Networks [16].

All these works face the CDP from a single-objective perspective, usually considering modularity as optimization criterion (see [29] for a recent and complete survey on CDP). As far as we know, the most recent approach for solving the single objective CDP considering a metaheuristic framework is presented in [15]. In particular, the authors propose a Greedy Randomized Adaptive Search Procedure devoted to maximize the modularity of the communities detected.

Nevertheless, there are other works that try to solve the CDP by optimizing a multi-objective function (MOCDP). Multi-objective optimization has evolved in the last years with novel approaches for generating robust approximations of the Pareto front. For instance, [30] proposes a novel interactive preference-based multi-objective evolutionary algorithm for designing a bolt supporting network, while [31] presents a dynamic robust multi-objective optimization method for solving problems where the time is a key factor. The rationale behind following a multi-objective approach in CDP is that the optimization metrics, traditionally considered isolatedly, always have one or more handicaps, resulting in the loss of information related to the community structure. The MOCDP emerges as a possible solution for this problem, considering two or more metrics simultaneously for improving the community detection in a social network. The goal in this case is to find the different communities of a network by considering different conflicting objectives to be optimized [32]. Most of the works are focused on adapting well-known evolutionary algorithms such as NSGA-II to solve different multi-objective community detection problems. An evolutionary algorithm based on decomposition [33] is designed for maximizing the density of internal degrees while minimizing the density of external degrees. Another evolutionary algorithm for solving MOCDP is presented in [34], considering as objective functions the maximization of the intra-link strength of the communities and the minimization of the inter-link strength, which are very similar to those considered in [33]. Finally, another bioinspired algorithm, based on enhanced firefly methodology is presented in [35], which maximizes the in-degree of the nodes in each community while minimizing their out-degree. Notice that, although each previous work considers different objective functions, they are very similar among them, focusing on locating in the same community the most connected nodes. As far as we know, [17] presents the most recent multi-objective approach for solving the MOCDP, considering the NRA and RC metrics previously defined.

4. Algorithmic approach

Heuristic algorithms are designed for reaching a local optimum in short computing times. However, they usually stagnate in those local optima, reducing the portion of the search space explored. Metaheuristic algorithms emerge as a solution to overcome this situation by guiding the search of the heuristic method, thus reaching further regions of the search space [36].

This paper presents a metaheuristic algorithm based on the Variable Neighborhood Search (VNS) [6] framework. VNS methodology was originally designed to escape from local optima by performing systematic changes of neighborhood. It is worth mentioning that, as a metaheuristic approach, it cannot guarantee the optimality of the obtained solutions.

The effectiveness of VNS methodology has lead the scientific community to develop several variants, which can be classified according to the balance between diversification and intensification. On the one hand, Reduced VNS (RVNS) [37,38] is focused in diversification, considering stochastic changes of neighborhoods. On the other hand, Variable Neighborhood Descent (VND) [39,40] is devoted to intensification by performing deterministic changes of neighborhoods. Finally, Basic VNS (BVNS) [41] arises as a compromise between intensification and diversification by combining stochastic and deterministic changes of neighborhoods. As a result of the success of the methodology, several new variants have been proposed: General VNS (GVNS) [42], Variable Neighborhood Decomposition Search (VNDS) [43], Skewed VNS (SVNS) [44], or Variable Formulation Search (VFS) [45], among others.

VNS methodology was originally designed for tackling single objective optimization problems. Recently, the VNS framework has been also adapted for dealing with multiobjective problems [7]. Multi-objective VNS has lead to several recent successful research, emerging as one of the most robust methodologies in the area [8,46]. In this work, we adapt the multi-objective VNS presented in [7] for solving the multi-objective community detection problem, focusing in the BVNS variant (MOBVNS). Algorithm 1 presents the general framework of MOBVNS.

Algorithm 1 *MOBVNS* $(S, k_{\max})$

1: $k \leftarrow 1$
2: **while** $k \leq k_{\max}$ **do**
3: $\quad S' \leftarrow Shake(S, k)$
4: $\quad S'' \leftarrow Improve(S')$
5: $\quad k \leftarrow NeighborhoodChange(S, S'', k)$
6: **end while**
7: **return** S

The algorithm starts from an initial set of non-dominated solutions denoted with S. In the context of VNS, the initial front can be generated either at random, or using a more elaborated constructive procedure. In our case, this initial set is generated with the Greedy Randomized Adaptive Search Procedure (GRASP) described in Section 4.1. The second input parameter of the MOBVNS algorithm is the maximum neighborhood to be explored during the search, $k_{\max}$. As stated in previous works [47], the maximum neighborhood to be considered in the VNS algorithm is usually small, to avoid exploring completely different solutions in each iteration, which will eventually lead to a multi-start approach.

The algorithm starts by considering the neighborhood $k = 1$ (step 1). Then, MOBVNS iterates until reaching the maximum neighborhood $k_{\max}$ (steps 2–6). In each iteration, a perturbed set of solutions S' is generated with the shake method presented in Section 4.2. Then, S'' is created as the set of non-dominated solutions derived from applying the local search method introduced in Section 4.3 to each solution contained in S' to reach a local optimum of each perturbed solution. Finally, the neighborhood change procedure (Section 4.4) is responsible for selecting the next neighborhood to be explored.

The traditional neighborhood change method inside single-objective VNS restarts the search from the first neighborhood ($k = 1$) every time an improvement is found. Otherwise, the search continues in the next neighborhood ($k = k + 1$). In the context of multi-objective optimization, the definition of improvement is slightly modified. Specifically, neighborhood change method considers that an improvement is found if a solution has been able to enter in the set of non-dominated ones.

The algorithm ends when no improvement for the set of non-dominated solutions is found in any of the neighborhoods (i.e., the algorithm has not been able to insert a new solution in it), returning the resulting set of non-dominated solutions.

4.1. Generation of the initial set of non-dominated solutions

The main objective of a good constructive method in a multi-objective problem is to generate a front with high quality solutions (i.e, those that are non-dominated) while maintaining the diversity among them. With this aim, we propose a constructive procedure based on the Greedy Randomized Adaptive Search Procedure (GRASP). This metaheuristic is originally presented in [48] and formally defined in [49] which consists of two different phases: construction and local search. We refer the reader to [50] for a recent survey on this methodology and some extensions recently studied.

In this paper, we only consider the first stage (i.e., the construction phase) of the GRASP to populate an initial set of non-dominated solutions. Algorithm 2 shows the associated pseudocode. This procedure starts by creating one community for each node in the network (step 1) and initializing the set of non-dominated solutions S with it (step 2). The next step corresponds to compute all the possible new communities that can be created by merging two of the existing ones, creating a Candidate List (CL) with them (step 3).

Algorithm 2 *Construction* $(G = (V, E), \alpha)$.

1: $\mathcal{C} \leftarrow \{C_1, C_2, \ldots, C_n : C_i = \{v_i\}, v_i \in V \wedge 1 \leq i \leq n\}$
2: $S \leftarrow \{\mathcal{C}\}$
3: $CL \leftarrow \{\langle C_i, C_j \rangle, \forall C_i, C_j \in \mathcal{C}, 1 \leq i < j \leq n\}$
4: **while** $|CL| > 1$ **do**
5: $\quad g_{\min} \leftarrow \min_{\langle C_i, C_j \rangle \in CL} g(C_i, C_j)$
6: $\quad g_{\max} \leftarrow \max_{\langle C_i, C_j \rangle \in CL} g(C_i, C_j)$
7: $\quad \mu \leftarrow g_{\max} - \alpha \cdot (g_{\max} - g_{\min})$
8: $\quad RCL \leftarrow \{\langle C_i, C_j \rangle \in CL : g(\langle C_i, C_j \rangle) \geq \mu\}$
9: $\quad \langle C_i, C_j \rangle \leftarrow Random(RCL)$
10: $\quad \mathcal{C} \leftarrow \left(\mathcal{C} \setminus \{C_i, C_j\}\right) \cup (C_i \cup C_j)$
11: $\quad CL \leftarrow \{\langle C_i, C_j \rangle, \forall C_i, C_j \in S, 1 \leq i < j \leq n\}$
12: $\quad updateNDS(S, \mathcal{C})$
13: **end while**
14: **return** S

Next steps are repeated until the CL contains a single candidate (i.e., there are only two communities in the solution). In each iteration, all the candidates are evaluated under a certain greedy criterion g (steps 5 and 6). Then, these two values are used to define the threshold μ (Line 7) that depends on the parameter $\alpha \in [0, 1]$, which controls the randomness/greediness of the method. On the one hand, when $\alpha = 0$, $\mu = g_{\max}$ and only those communities with maximum value of the greedy function are included in the Restricted Candidate List (RCL). On the other hand, if $\alpha = 1$, $\mu = g_{\min}$, all the communities are included in the RCL and then the algorithm becomes totally random. Therefore, the threshold defines the size of the RCL because only the most promising candidates of CL will belong to RCL (step 8). Once the RCL is constructed, an entry is randomly selected (step 9), specifying the two communities that will be merged (step 10).

Then, the algorithm updates the CL (step 11), by removing all the pairs in which the two communities that have been merged, C_i and C_j, were involved, and including a new candidate to merge the new community created $C_i \cup C_j$ with every other community in the solution (step 11). Finally, the resulting community is evaluated to be considered in the set of non-dominated solutions (step 12).

In order to define whether two given communities (i.e. C_i and C_j) should be merged, the algorithm uses a greedy function denoted as $g(C_i, C_j)$. This function takes into account the number of edges that starts and ends in nodes belonging to C_i and C_j, and the size of the resulting community (Eq. (1)).

$$g(C_i, C_j) = \frac{|\{(u, v) \in E \ \forall u, v \in C_i \cup C_j\}|}{|C_i \cup C_j|} \tag{1}$$

Although the construction phase is based on the greedy algorithm, the multi-objective optimization is used when the built solution has to be included in the reference front. Not all constructed solutions are finally included in the approximation of the pareto front but only those that are non-dominated solutions according to Negative Ratio Association (NRA) and Ratio Cut (RC). It is worth mentioning that a solution is non-dominated if it is better than other solution already in the front in any of the two objectives. When a non-dominated solution is included in the front, all those solutions that are dominated by the new one are removed.

4.2. Perturbing solutions for the CDP

The *Shake* procedure is responsible for escaping from local optima within the VNS framework. In order to do so, the method resorts to a random solution in the neighborhood of the solution under exploration. In order to properly adapt this method to the context of multi-objective optimization, it is needed to previously define a basic movement of MOCDP, which consists in removing a vertex v from its current community, say for instance C_j, and inserting it in a different community, for example C_i, with $i \neq j$. More formally, given a solution $\mathcal{C} = \{C_1, C_2, \ldots, C_c\}$, a vertex $v \in C_j \subset V$, and a community C_i, with $i \neq j$:

$$Move(\mathcal{C}, v, C_i) = \begin{cases} C_i \leftarrow C_i \cup \{v\} \\ C_j \leftarrow C_j \setminus \{v\} \end{cases}$$

The neighborhood of a solution is defined as the set of solutions that can be reached by performing the aforementioned move. Specifically,

$$N(\mathcal{C}) = \{\mathcal{C} \leftarrow Move(\mathcal{C}, v, C_i) : \forall v \in V \setminus C_i \wedge 1 \leq i \leq c\}$$

We rely on this definition to introduce the neighborhood $N_k(\mathcal{C})$. In particular, $N_k(\mathcal{C})$ is conformed with the set of solutions that can be obtained when performing exactly k consecutive basic moves to $\mathcal{C}$.

Notice that solution $\mathcal{C}'$ obtained in the neighborhood N_k of solution $\mathcal{C}$ is usually worse than $\mathcal{C}$ (in terms of objective functions value). However, the main objective of the *Shake* method is to escape from local optima, continuing the search from a completely different region of the search space. Additionally, all the solutions explored in the *Shake* methods are guaranteed to be feasible, being unnecessary to check the constraint of the problem, which is one of the most time consuming parts of the algorithm.

The output solution obtained in a *Shake* procedure is not necessarily a local optimum with respect to the defined neighborhood and, therefore, the local search method is applied to locally optimize the newly generated solution.

4.3. Local optimization

Population-based metaheuristics (i.e., Genetic Algorithms, Particle Swarm Optimization, Differential Evolution, etc.), are extended in the multi-objective context since they consider a set of solutions, which can be easily identified with the efficient set. Symmetrically, trajectory-based metaheuristics (VNS, Tabu Search, Simulated Annealing, etc.), use only one solution. It is possible to overcome this situation by considering the whole set of non-dominated solutions as the incumbent solution to a multi-objective problem [7].

We propose in this paper an improvement procedure (see step 4 in Algorithm 1) that receives the perturbed set of solutions, obtained with the *Shake* method, and returns a locally optimal set. Specifically, this method randomly scans each solution and then improves it with a local search algorithm. We consider two different strategies that follow the first improvement approach, which means that the search will restart when an improvement in the current solution under evaluation has been found.

Both strategies try to optimize NRA and RC of each solution, but they differ in the way that these two objectives are considered. The first strategy, named Independent Search Procedure (ISP), independently improves each objective starting from the very same solution (in the reference front). The second strategy is called Combined Search Procedure (CSP) since both metrics (NRA and RC) are alternatively considered in the procedure.

Algorithm 3 shows a general scheme of the improvement method that needs to be particularized for each strategy. Specifically, this method considers a generic objective function denoted with f. On the one hand, the first local search strategy takes into account $f(\mathcal{C}) = NRA(\mathcal{C})$ and then $f(\mathcal{C}) = RC(\mathcal{C})$. On the other hand, the second strategy alternatively takes in each iteration of the while-loop either $f(\mathcal{C}) = NRA(\mathcal{C})$ or $f(\mathcal{C}) = RC(\mathcal{C})$. After finishing any of the two local search strategies, non-dominated solutions found are tested to be admitted (or not) in the reference set.

Algorithm 3 *Improve* $(\mathcal{C} = \{C_1, C_2, \ldots, C_c\}, S)$

```
1:  Improve ← True
2:  while Improve do
3:      Improve ← False
4:      for C_i ∈ 𝒞 do
5:          for v ∈ V \ C_i do
6:              S' ← Move(𝒞, v, C_i)
7:              updateNDS(S, 𝒞')
8:              if f(𝒞') < f(𝒞) then
9:                  𝒞 ← 𝒞'
10:                 Improve ← True
11:                 Restart the search at step 3
12:             end if
13:         end for
14:     end for
15: end while
```

Algorithm 3 receives as input parameter a solution. For the sake of brevity, we omit the inclusion of either NRA and RC as input parameter since they are generalized by the function f. The method iterates over all the communities and vertices (steps 4–14). In each iteration, we evaluate the impact of moving v from its current community to a different one in the solution under evaluation. In order to do so, we use the defined $Move(\mathcal{C}, v, C_i)$, see step 6. The procedure tries to include the neighbor solution in the reference front by using the procedure *updateNDS* (step 7). It basically tests whether $\mathcal{C}'$ is dominated by other solution belonging to S. If so, the efficient front remains unaltered; otherwise, $\mathcal{C}'$ is included in S, removing those solutions in S that become dominated.

After that, if an improvement is found with respect to the criteria under evaluation (either NRA and RC), the incumbent solution is updated and the search restarts again. The method ends when no improvement is found. It is worth mentioning that no return is needed since step 7 already includes all the non-dominated solutions found during the search.

The local optimization process in a multi-objective optimization problem is usually designed to simultaneously optimize all the considered objectives. In the context of MOCDP, we propose two different local optimization strategies: Combined Search Procedure (CSP) and Independent Search Procedure (ISP).

The first one, CSP, follows the traditional approach, where the two considered objectives are optimized at the same time. Specifically, each iteration of the local search method, presented in Algorithm 3, is focused on optimizing either NRA or RC. In particular, the even iterations finds a local optimum with respect to NRA, while the odd iterations focuses on finding a local optimum with respect to RC.

The second method, ISP, follows a different criterion with the aim of finding better solutions for each objective. In particular, each solution is optimized with each considered objective, i.e., NRA and RC, as in a single-objective optimization problem. In other words, given a certain solution, the ISP finds a local optimum with respect to NRA and then with respect to RC, starting from the same initial solution. All the solutions found during the optimization process are evaluated to be included in the set of non-dominated solutions.

The termination criterion for both, ISP and CSP, is the same: the search stops when no new non-dominated solutions have been found after a complete execution of the local optimization method.

4.4. Neighborhood change

The main objective of the Neighborhood Change method within VNS is the selection of the next neighborhood to be explored. In the context of single objective optimization, the Neighborhood Change method usually receives three input parameters: the best solution found so far, the candidate solution to be evaluated, and the current neighborhood being explored (k). Then, it verifies whether the incumbent solution outperforms the best one. If so, the search is restarted from the first neighborhood, updating the best solution found so far. Otherwise, the search continues in the next neighborhood.

The Neighborhood Change method has been adapted to the multi-objective nature of the problem considered in this work. The most significant modification affects to the concept of improvement. In particular, we introduce the method *isNotDominated* in charge of comparing two non-dominated set of solutions. Algorithm 4 illustrates the pseudo-code of this procedure. It tests whether points in S' are dominated, or not, by any point in S. If there is at least one non-dominated point in S, then the method returns `True`; otherwise (i.e., all points in S are dominated), it returns `False`.

Algorithm 4 *isNotDominated* (S', S)

```
for all C ∈ S' do
    if (C ∉ S ∧ ¬Dominated(C, S')) then
        return True
    end if
end for
return False
```

The pseudo-code of the Neighborhood Change method is shown in Algorithm 5. The input parameters are now the current best non-dominated set S', the reference front under evaluation S, and the neighborhood under exploration (k). As it was aforementioned, the most significant modification affects to step 1. The second relevant modification consists in updating the non-dominated set of solutions by merging S' and S with the procedure *updateNDS* (step 2). See Section 4.3 for further details. As it is customary in this method, if we do not find an improvement, the search continues in the next neighborhood (step 5).

Algorithm 5 *NeighborhoodChange* $(E^\star, E, k)$

```
1: if isNotDominated(E*, E) then
2:     updateNDS(E*, E)
3:     k ← 1
4: else
5:     k ← k + 1
6: end if
7: return k
```

4.5. Computational complexity

In this section, the computational complexity of each component is analyzed and, then, the complete complexity of the proposed algorithm is computed. First of all, it is necessary to evaluate the cost of generating the initial non-dominated set of solutions with the constructive procedure. Analyzing the pseudocode presented in Algorithm 2, the complexity of constructing the candidate list CL is $O(n^2)$, since it requires to traverse the complete set of communities (initially one community per node) and, then, to create a candidate community to be merged with every other community. Notice that when the construction evolves, the number of available communities is reduced since it merges two communities in each iteration. Also, as this construction is included in a while loop until the CL contains a single community, the computational complexity is bounded $O(n^3)$.

We now compute the complexity of the local optimization method presented in Algorithm 3. In each iteration of the local search procedure, the method needs to traverse all the nodes, resulting in a complexity of $O(n)$. Each node is evaluated to enter in every other community, resulting in a complexity of $O(n)$ (in the worst case, there is a community for each node). A naive implementation of the move operator would result in a complexity of $O(m)$, since updating the inter and intra-community edges requires to evaluate every edge after performing the move. However, the proposed algorithm leverages the data structure called UnionFind, which basically assigns a representative node for each community in such a way that the evaluation of the modifications in inter and intra-community edges can be performed in $O(1)$. Therefore, the computational complexity of each iteration of local search procedure is bounded by $O(n^2)$ instead of $O(n^2 \cdot m)$ which would be obtained by the naive implementation. Since the local search is executed while an improvement is found, it is not possible to determine the complexity of the complete method since it highly depends on how close to a local optimum is the input solution.

The perturbation method presented in Section 4.2 again leverages the UnionFind structure to reduce complexity, which is bounded by $O(k)$, since k moves are performed, being the complexity of each move $O(1)$.

Finally, the complexity of the complete VNS algorithm is computed. As it was aforementioned, the complexity of generating the initial front is bounded by the maximum between the constructive, $O(n^3)$, and the local search procedure, $O(n^2)$ in each iteration. This results in a complexity $O(n^3)$ for generating the initial front. Then, k iterations are performed, where it is executed a shake procedure, with a complexity of $O(k)$, a local improvement, with a complexity of $O(n^2)$ in each iteration, and the neighborhood change method which presents a complexity of $O(1)$ since it only requires to select the next neighborhood to be explored. Therefore, the complexity of the complete algorithm is $O(k) \cdot \max\{O(n^3), O(n^2), O(1)\} = O(k \cdot n^3)$.

5. Experiments and results

In this section we will expose the experiments performed to test the effectiveness and efficiency of the proposed algorithm and to compare it with the best method found in the related literature [17]. All algorithms are executed over two different datasets: synthetic and real-world networks. For the former, we have used the network generator developed by Lancichinetti et al. [51] to construct synthetic instances,[1] where the node degree distribution and the community size follow a power-law highly configurable. The main advantage of these instances is

[1] Lancichinetti, Fortunato, and Radicchi (LFR) networks.

that the optimal ground truth for the community structure is known by construction. We have considered different configurations for the network generator and we have generated different network instances for each configuration (totalizing 52 different networks). In particular, we have considered networks with a range from 500 to 7500 nodes, and the edge probability p is defined as $p \in [0.1, 0.8]$ with an interval of 0.1 for instances with 500, 1000 and 5000 nodes, and as $p \in [0.1, 0.3]$ with an interval of 0.1 for instances with 5500 to 7500 nodes.

To complement these instances, we additionally consider 12 real-world networks: Zacharys karate club [52] (32 nodes and 78 edges), dolphin social network [53] (64 nodes and 159 edges), American college football [54] (115 nodes and 613 edges), jazz [55] (with 198 nodes and 2742 edges), and netscience [56,57] (1589 nodes and 2742 edges), facebook large page-page network [58] (22 570 nodes and 171 002 edges), and the set of Twitch Social Networks [58] (with number of nodes in range from 1912 to 9498 and edges in range from 31 299 to 153 138). Notice that the ground truth for karate, dolphin, and football networks are known beforehand.

The computational experiment is divided into two different phases: on the one hand, we carry out a set of preliminary experiments to tune the parameters of our algorithms. In these experiments a subset of all instances will be used (18 out of 62), with the aim to avoid the overfitting of the algorithm. On the other hand, we run the final experiments over the whole benchmark to compare our best identified method with those presented in the state of the art. We additionally compare the proposed MOBVNS with the most extended algorithms for solving the CDP following a single-objective approach, with the aim of evaluating the relevance of modeling the CDP as a multi-objective optimization problem.

For sake of fairness, we have executed both algorithms with a time limit of 1800 s. Both algorithms have been executed in a computer with an AMD Ryzen 5 3600 AM4 core (3.6 GHz) with 16GB RAM. All algorithms were implemented using Java 9. With the aim of facilitating further comparisons, the dataset and the source code of the proposed algorithm are publicly available at http://grafo.etsii.urjc.es/mocdp.

5.1. Multi-objective metrics

In this paper, we deal with the variant of the Community Detection Problem, where the Negative Ratio Association (NRA) and Ratio Cut (RC) are optimized simultaneously. Notice that these two objectives have been already proven that are in conflict. Then, in order to compare the performance of the proposed algorithms we use metrics that evaluate the quality of an approximation of the Pareto front. Specifically, we have considered four of the most extended multi-objective metrics [59]: coverage, hypervolume, ϵ-indicator, and inverted generational distance. Given two non-dominated set of solutions (S and S'), the **coverage** metric, $CV(S, S')$, evaluates the number of solutions within the approximation front S that are dominated by solutions in S'. In our experiments, we evaluate the quality of S derived from an specific algorithm with respect to a reference set constructed with all non-dominated solutions found with all algorithms tested in the corresponding experiment. Given this definition, the smaller the value, the better. For the sake of brevity, we denote $CV(S, S')$ as CV, being S the set of non-dominated solutions under evaluation and S' the reference set (as indicated above).

The **hypervolume** metric, HV, measures the size of the space covered by the set of non-dominated solutions. In other words, it computes the hypervolume of the portion of the objective space that is weakly dominated by an approximation front. Then, large values of HV implies that the set of non-dominated solutions obtained with the algorithm is better.

The ϵ-**indicator**, $EPS(S, S')$, evaluates the smallest distance needed to transform every point of the approximation front under evaluation (S) in the closest point of the reference set S' (equivalent to the coverage metric). Therefore, if we obtain low values of ϵ-indicator, it indicates that the reference front generated by the algorithm under evaluation is better than others. As indicated above, we denote $EPS(S, S')$ as EPS.

Finally, the **inverted generational distance**, $IGD{+}(S, S')$, is an inversion of the well-known generational distance metric with the aim of measuring the distance from the incumbent set of non-dominated solutions (S) to the reference set obtained during the experiment (S'). Therefore, small values of $IGD{+}$ indicate a high proximity to the reference front, which is better. Finally, the computing time of all the algorithms is also presented, with the aim of evaluating the efficiency of the procedures. As it was aforementioned, we simplify the notation of $IGD{+}(S, S')$ as $IGD{+}$.

5.2. Context-based metrics

In social network analysis, there exists two popular performance metrics usually referred to as normalized mutual information (NMI) [60] and the modularity (Q) [10]. NMI requires for a ground truth since it evaluates the difference between the community structure detected for the incumbent algorithm and the true one. It is worth mentioning that the ground truth is known by construction for all LFR instances. Additionally, it is also available for karate, dolphin, and football instances.

The modularity can be evaluated in any network since it does not depend on the ground truth. This metric compares the structure of the communities against a random graph. More precisely, this metric measures how likely the communities are created at random. For this reason, modularity metric is particularly useful for real-world instances such as jazz or netscience, where the ground truth is unknown.

Notice that we are dealing with a multi-objective optimization problem. Therefore, instead of having a single solution, we have a set of non-dominated solutions. In order to provide a value of either NMI or Q, we follow the methodology proposed in [17]; i.e., to traverse the complete front finding the solution that presents the largest value in each metric. It implies that the solution that reaches the best Q value is not necessarily the one that provides the best result in terms of Normalized Mutual Information (NMI).

5.3. Preliminary experimentation

The first experiment is oriented to determine the best value of the α parameter (see Section 4.1). In particular, we test $\alpha = \{0.25, 0.50, 0.75, RND\}$, where RND indicates that the value is selected randomly in the range [0, 1] for each construction. These values cover from an almost greedy constructive method to a semi-random one. For each instance used in this experiment, we execute the constructive algorithm for 100 independent iterations, returning the best solution found. Table 1 shows the associated results, where we report average values across the subset of preliminary instances for the CV, HV, EPS, and $IGD{+}$. We additionally include the average computing time required by the algorithms (column $T(s)$).

In view of these results, the best configuration of the constructive method is the one that uses $\alpha = 0.25$. In particular, based on the results obtained for the coverage metric, we can affirm that most of the points in the reference front also belong to the constructive method executed with a $\alpha = 0.25$. Also, the hypervolume metric (HV) is larger than the one attained with the other approaches, though it is closely followed by the constructive method configured with RND. The same occurs with

Table 1
Comparison of the reference front built by the constructive procedure with different values for the α parameter. Best results are highlighted with bold font.

α	*CV*	*HV*	*EPS*	*IGD+*	*T* (s)
0.25	**0.38**	**0.73**	**0.05**	**0.01**	119.04
0.50	0.81	0.68	0.09	0.04	123.04
0.75	0.81	0.59	0.18	0.13	170.30
RND	0.65	0.71	0.05	0.02	**117.06**

Table 2
Comparison of the reference front obtained with the local search procedures designed in this work. Best results are highlighted with bold font.

Algorithm	*CV*	*HV*	*EPS*	*IGD+*	*T* (s)
$C(0.25)$+ISP	**0.14**	**0.70**	**0.03**	**0.00**	117.99
$C(0.25)$+CSP	0.81	0.45	0.43	0.23	**116.86**

Table 3
Comparison of the reference front obtained with different values of k for the MOBVNS method. Best results are highlighted with bold font.

$k_{\max}$	*CV*	*HV*	*EPS*	*IGD+*	*T* (s)
0.1	0.71	0.71	0.07	0.03	118.76
0.2	0.67	**0.72**	0.06	0.02	**118.43**
0.3	**0.66**	**0.72**	**0.05**	0.02	119.14
0.4	0.67	**0.72**	0.07	0.02	122.75
0.5	0.68	**0.72**	0.06	**0.01**	123.80

the ϵ-indicator, with $\alpha = 0.25$ the solutions provides the smallest value in the comparison, and the *RND* value is the second best approach. Regarding the inverted generational distance, $\alpha = 0.25$ again obtains the best results, closely followed by *RND*. Analyzing the computing time, we can clearly see that there are no differences among all considered variants, as expected.

Once we know what the best configuration for our constructive method is, we conduct an additional experiment to determine the performance of the proposed local search algorithms. From now one, we will refer to constructive algorithm with $\alpha = 0.25$ simply as $C(0.25)$. The results of the metrics obtained with both local search approaches (described in Section 4.3) are shown in Table 2.

As we can clearly see, the ISP performs better in this problem. Specifically, with respect to *IGD+*, coupling the ISP with the best version of our constructive method lead the algorithm to find an efficient set of solutions which is much closer to the reference front than CSP. Attending to hypervolume, we can see that ISP obtains a considerably larger value than the second variant. Analogously, the ϵ-indicator is also the smallest one in the comparison, being heavily smaller than its competitor. Although the computational time required by CSP is smaller than ISP, there is not significant differences between these two procedures, and the great results obtained by ISP with the other metrics clearly justifies the choice of ISP.

In the next experiment, we test the best k value for the *MOBVNS* algorithm. Table 3 shows the obtained results for each configuration of the algorithm.

Analyzing these results, we can see that all of them are quite similar, becoming difficult to choose the best value for $k_{\max}$ parameter. In particular, the hypervolume metric is not deterministic since almost all the variants report the same value. However, the coverage and ϵ-indicator metrics suggest that the best value is $k_{\max} = 0.3$. Additionally, we can clearly see that the larger the value of $k_{\max}$, the more computationally demanding. Therefore, we select $k_{\max} = 0.3$ as the best value for the *MOBVNS* algorithm.

For the sake of brevity, we refer with MOBVNS as the multi-objective VNS variant that uses $C(0.25)$, ISP, and $k_{\max} = 0.3$.

Table 4
Comparison of the reference fronts obtained when applying the constructive method and when coupling it with the local search procedure ISP. Best results are highlighted with bold font.

Algorithm	*CV*	*HV*	*EPS*	*IGD+*
$C(0.25)$+ISP	**0.03**	**0.67**	**0.00**	**0.00**
$C(0.25)$	0.89	0.31	0.61	0.39

Table 5
Comparison of the reference front obtained with full MOBVNS framework and MOBVNS framework without Local Search Procedure. Best results are highlighted with bold font.

Algorithm	*CV*	*HV*	*EPS*	*IGD+*
MOBVNS	**0.28**	**0.63**	**0.17**	**0.10**
MOBVNS (without ISP)	0.29	0.48	0.42	0.11

5.4. Analysis of the effect of each component of the proposed algorithm

This section is devoted to clarify the contribution of each component of the algorithm in the final configuration. The algorithm is conformed with three main components: constructive procedure, local improvement method, and the combination of both of them in the VNS framework.

The first experiment is designed to evaluate the effect of the local search procedure over the results obtained by the constructive procedure isolatedly. Table 4 shows the results obtained in this experiment.

As it can be seen, the use of a local search procedure after constructing an initial non-dominated front with constructive method significantly improves the quality of the final non-dominated front obtained. In particular, the coverage of 0.03 obtained when coupling the constructive procedure with the local search method indicates that almost all the initial solutions are improved, while the value of 0.89 obtained by the constructive procedure indicates that almost all the initial solutions are dominated by the ones obtained with the local search procedure. The hypervolume, ϵ-indicator, and inverted generational distance values supports these results.

The second experiment is intended to study the influence of the local search procedure within the VNS framework. Table 5 shows the obtained results in this comparison.

In this case, the coverage metric is not determinant since both algorithms present similar values, although considering the local search obtains better results, as well as when considering the inverted generational distance. However, analyzing the hypervolume and the ϵ-indicator metrics, the relevance of the local search inside the VNS framework is confirmed, being the variant with local search twice better than the one without local improvement phase. Therefore, adding an improvement method result in more robust non-dominated sets of solutions.

Finally, having shown the relevance of the constructive and local search procedure, it is interesting to evaluate the effect of the initial front generated by GRASP algorithm in the final MOBVNS. In order to do so, the algorithm with GRASP for generating the initial front is compared with the same VNS algorithm but considering an initial random population of the reference front. Table 6 show the results obtained in this comparison.

The results clearly show that the contribution of starting from a good initial front to the final algorithm is justified. In particular, the coverage is reduced to 0.00 when considering a GRASP generation of the initial front, and the hypervolume of the random initial front is close to 0.00. Additionally, the ϵ-indicator also confirms the superiority of the GRASP initialization. If we analyze the inverted generational distance, we can see that considering a

Table 6
Comparison of the reference front obtained by the final proposed algorithm when considering an initial population generated with GRASP and Random constructions. Best results are highlighted with bold font.

Algorithm	*CV*	*HV*	*EPS*	*IGD+*
GRASP initial front	**0.00**	**0.32**	**0.58**	0.39
Random initial front	0.11	0.01	0.99	**0.32**

Table 7
Comparison of the reference front obtained with the best configuration for MOBVNS and the LMOEA proposed by [17]. Best results are highlighted with bold font.

Algorithm	*CV*	*HV*	*EPS*	*IGD+*	*T* (s)
MOBVNS	**0.07**	**0.14**	**0.86**	0.22	**214.64**
LMOEA	0.36	0.02	0.27	**0.21**	1800.00

random initial front provides more diversity, resulting in slightly better results when considering this metric.

Therefore, the contribution of each component of the main algorithm (constructive procedure, local search method, and complete VNS framework) is confirmed.

5.5. Final experiments

Having made the necessary adjustment in the proposed algorithm and having chosen the best parameters for each configuration of the procedure, we proceed to make the comparison with the best previous method found in the literature, denoted as LMOEA [17]. To do this, we will execute both algorithms over the full set of 62 instances. We first compare the obtained results with those metrics employed in preliminary experiments.

The parameter setting of LMOEA considers a population of 100 individuals, 100 generations, a crossover probability of 0.9, a mutation probability of 0.1 and a neighborhood size of 40 (see [17] for further details). Considering that we are comparing heuristics algorithms, we include an additional termination criterion based on the maximum allowed computing time. Specifically, we fix 1800 s as the maximum time spent in a single instance. If this additional termination condition is met, we halt the corresponding algorithm, returning the best solution found during that time horizon.

Table 7 shows the results obtained for the considered multi-objective metrics *CV*, *HV*, *EPS*, and *IGD+* as well as the average computational time required for each algorithm under evaluation. Notice that the results of each metric is the average value obtained across the complete set of 62 instances. Due to the different sizes of the considered instances, all the metrics are normalized in the range [0, 1] in the comparison.

As we can see, the proposed MOBVNS provides the best results in all four measures but IGD+, where the results are rather similar. Analyzing the computational time, there is a significant difference between the performance of both algorithms. Whereas LMOEA spends the budget time of 1800 s for every instance under evaluation, MOBVNS needs, in average, only 214.64 s, obtaining higher quality solutions in considerably smaller computing times.

Once we have compared both methods using the classical multi-objective metrics, we further analyze the quality of their solution by considering the experimental framework described in [17]. In particular, we first graphically depict the NMI for each LFR instance.

In order to have more robust conclusions, both algorithms were executed for 20 independent executions, reporting the average results. Fig. 2 shows these results for $n = 500$ and $n = 1000$, where p varies from 0.1 to 0.8 in steps of 0.1. As we can observe in this figure, MOBVNS presents high quality solutions for $p < 0.5$ with values of NMI close to 1.0. As expected, for larger values of p the behavior gets worse since these networks are harder to be solved. LMOEA seems to be more stable in these instances, ranging the NMI values from 0.55 to 0.7. Indeed, LMOEA is able to outperform MOBVNS in $p = 0.8$ and $n = 1000$.

The value of p in the LFR generator indicates the average ratio between inter-community edges and the total edges in the optimal community detection provided by construction. Therefore, a large value of p results in communities with several edges to nodes in other communities when comparing it with the total number of edges of the considered community. This value leads to networks which do not accurately represent real-world networks since, in them, the number of edges to nodes in other communities is usually small.

For that reason, in the instances in which $p \geq 0.5$, the number of edges to other communities is considerably larger than the number of edges in the same community, resulting in networks where the community structure is not necessarily preserved. Since the MOBVNS is designed for detecting communities in networks that present community structure, the main ideas of its design are not useful for these special instances, presenting similar or even worse performance than LMOEA.

Notice that there exists a considerable performance difference between the results of LMOEA that we present in these figures and those reported in [17]. This discrepancy might come from the fact that we have considered an additional termination condition (i.e., 1800 s of CPU time), not allowing the algorithm to reach the maximum number of generations. Indeed, in our computer, LMOEA is able to evolve the population for less than 50 generations (on average) in 1800 s, which is half of number used in [17].

In the next experiment, we compare both algorithms over the set of real-world instances. As it was aforementioned, the results obtained in this benchmark must be separated into two groups. On the one hand, those where the ground truth is known and, on the other hand, those where the ground truth is unknown. Therefore, for karate, dolphin, and football, we report NMI and Q. Whereas for jazz and netscience, we only show the Modularity.

We report in Table 8 the average NMI of 20 independent executions (and the associated standard deviation) for both, MOBVNS and LMOEA, over each instance. We additionally include the CPU time (notice that both algorithms were executed in the same computer). To facilitate the comparison, we additionally include the NMI values published in [17], where the LMOEA is executed for 100 generations (without time limit). As we can see in this experiment, MOBVNS obtains competitive results in all networks by spending few seconds of computing time. It consistently finds better results than LMOEA (executed for 1800 s) and it is relatively close to LMOEA executed without time limit. The standard deviation smaller than 0.01 on average reached by MOBVNS confirms the robustness of the method, reaching the best values or close to best in most of the executions.

We now show in Table 9 the modularity obtained with MOBVNS and LMOEA when considering the whole set of real-world networks. As was aforementioned, the maximum CPU time is limited to 1800 s. As before, we also include the results of LMOEA reported in [17] where the authors did not consider any time limit. We report for each network the average Q value of 20 executions. Notice that, in the case of LMOEA, no solutions are generated after 1800 s for two instances, which is indicated with an asterisk in the corresponding cell of the table. The dashes in the LMOEA [17] indicates that these are new instances which were not tested in the original work and, therefore, there are not results for them. This experiment again confirms the good performance of the proposed algorithm. As can be observed, our method consistently produces better outcomes. The proposed

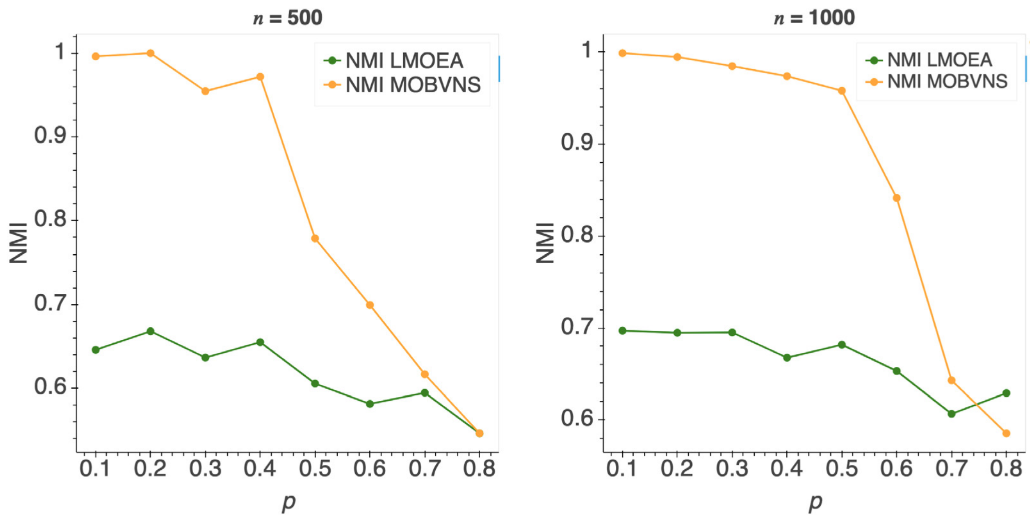


Fig. 2. NMI for $n = 500$ and $n = 1000$ instances.

Table 8
Summary of the results of NMI metric obtained by the proposed MOBVNS and LMOEA when solving the real world instances.

Instance	LMOEA				MOBVNS			
	Avg. NMI	Avg. Time (s)	Best NMI	Best Time (s)	Avg. NMI	Avg. Time (s)	Best NMI	Best Time (s)
dolphin	0.050	1800.00	0.069	1800.00	0.751	0.41	0.770	0.11
football	0.020	1800.00	0.033	1800.00	0.864	1.80	0.877	0.27
karate	0.100	1800.00	0.100	1800.00	0.439	0.07	0.439	0.03

Table 9
Summary of the results of modularity metric obtained by the proposed MOBVNS and LMOEA when solving the real world instances.

Instance	LMOEA				MOBVNS			
	Avg. Q	Avg. Time (s)	Best Q	Best Time (s)	Avg. Q	Avg. Time (s)	Best Q	Best Time (s)
dolphin	0.059	1800.00	0.067	1800.00	0.728	0.41	0.736	0.11
football	0.029	1800.00	0.067	1800.00	0.789	1.80	0.827	0.27
karate	0.061	1800.00	0.069	1800.00	0.500	0.07	0.508	0.03
jazz	0.027	1800.00	0.027	1800.00	0.899	11.99	0.899	10.87
netscience	0.058	1800.00	0.058	1800.00	0.972	1800.00	0.972	1800.00
musae_DE	0.001	1800.00	0.012	1800.00	0.032	1800.00	0.043	1800.00
musae_ENGB	–	1800.00	–	1800.00	0.095	1800.00	0.108	1800.00
musae_ES	0.001	1800.00	0.003	1800.00	0.067	1800.00	0.069	1800.00
musae_FR	0.001	1800.00	0.001	1800.00	0.030	1800.00	0.030	1800.00
musae_RU	–	1800.00	–	1800.00	0.201	1800.00	0.201	1800.00

Table 10
Summary of the results obtained by the proposed MOBVNS and LMOEA when solving the 62 instances considered in this work.

Algorithm	#Best Q	#Best NMI	#Avg. Q	Avg. NMI
MOBVNS	**62**	**41**	**0.27**	**0.77**
LMOEA	0	14	0.01	0.66

MOBVNS method reaches the best values of Modularity in the 5 considered networks. Indeed, only in netscience network, our method spends the whole budget of CPU time. Again, the standard deviation is smaller than 0.01, confirming the robustness of the proposal, which is able to reach the best value in most executions and stay close to it in those cases in which the best value is not found.

We summarize the results over the whole set of instances in Table 10. Specifically, we report in this table, the number of instances in which each algorithm shows the best results using the metrics just mentioned, as well as the average value of these two metrics.

As it can be seen in this table, taking into account Modularity the proposed algorithm (MOBVNS) provides the best solution in all the instances compared with LMOEA executed with a time limit of 1800 s. Regarding the NMI, MOBVNS obtains the best results in 41 out of 62 networks (66% of the instances). Analyzing the average modularity and average NMI, MOBVNS provides again better results as these mean values are higher.

In order to validate these results, we have conducted the well-known non-parametric Wilcoxon statistical test for pairwise comparisons, which answers the question: do the solutions generated by both algorithms represent two different populations? We consider a typical level of significance of 1%. The resulting value is smaller than 0.0005 when comparing MOVNS with LMOEA, confirming the superiority of the proposed algorithm. Therefore, MOVNS emerges as one of the most competitive algorithms for the MOCDP, being able to reach high quality solutions in reduced computing times.

Finally, we have performed an additional experiment to validate the results obtained by MOBVNS. In particular, we have included in the comparison the most extended algorithms for community detection in social networks, which are focused in

Table 11
Comparative of MOBVNS Algorithm against other classical heuristic methods found in literature.

	EB	FG	LP	LE	ML	WT	IM	CL	MOBVNS
1000_0.1_prev	0.011	0.014	0.011	0.021	0.012	0.011	0.011	0.012	**0.450**
1000_0.1network	0.433	0.428	0.433	0.273	0.433	0.433	0.433	0.433	**0.454**
1000_0.2_prev	0.009	0.018	0.009	0.025	0.009	0.009	0.009	0.009	**0.435**
1000_0.2network	0.385	0.367	0.385	0.221	0.386	0.385	0.385	0.386	**0.428**
1000_0.3_prev	0.007	0.024	0.007	0.054	0.008	0.007	0.007	0.008	**0.396**
1000_0.3network	0.339	0.319	0.339	0.220	0.340	0.339	0.339	0.340	**0.414**
1000_0.4_prev	0.008	0.035	0.008	0.019	0.009	0.008	0.008	0.009	**0.369**
1000_0.4network	0.285	0.277	0.290	0.186	0.293	0.290	0.290	0.293	**0.407**
1000_0.5_prev	0.006	0.041	0.009	0.038	0.009	0.006	0.006	0.009	**0.391**
1000_0.5network	0.176	0.230	0.241	0.141	0.246	0.240	0.241	0.246	**0.411**
1000_0.6_prev	0.004	0.048	0.019	0.027	0.011	0.010	0.007	0.011	**0.357**
1000_0.6network	0.086	0.203	0.197	0.154	0.204	0.188	0.193	0.204	**0.386**
1000_0.7_prev	0.004	0.054	0.249	0.068	0.017	0.032	0.249	0.017	**0.311**
1000_0.7network	0.040	0.169	0.249	0.144	0.150	0.126	0.249	0.150	**0.317**
1000_0.8_prev	0.047	0.054	0.249	0.065	0.023	0.039	0.249	0.023	**0.275**
1000_0.8network	0.057	0.158	0.249	0.142	0.125	0.089	0.249	0.125	**0.272**
500_0.1_prev	0.023	0.023	0.023	0.020	0.023	0.023	0.023	0.023	**0.446**
500_0.1network	0.419	0.414	0.419	0.325	0.419	0.419	0.419	0.419	**0.424**
500_0.2_prev	0.020	0.026	0.020	0.039	0.020	0.020	0.020	0.020	**0.411**
500_0.2network	0.376	0.367	0.376	0.235	0.376	0.376	0.376	0.376	**0.432**
500_0.3_prev	0.017	0.028	0.020	0.061	0.017	0.017	0.017	0.017	**0.378**
500_0.3network	0.330	0.323	0.329	0.219	0.330	0.330	0.330	0.330	**0.382**
500_0.4_prev	0.015	0.032	0.015	0.053	0.016	0.015	0.015	0.016	**0.390**
500_0.4network	0.264	0.265	0.279	0.206	0.280	0.279	0.279	0.280	**0.362**
500_0.5_prev	0.013	0.041	0.025	0.027	0.017	0.015	0.015	0.017	**0.383**
500_0.5network	0.162	0.222	0.239	0.168	0.239	0.232	0.233	0.239	**0.374**
500_0.6_prev	0.008	0.054	0.249	0.036	0.016	0.015	0.015	0.016	**0.365**
500_0.6network	0.044	0.180	0.248	0.135	0.187	0.174	0.248	0.187	**0.342**
500_0.7_prev	0.056	0.051	0.249	0.068	0.022	0.019	0.249	0.022	**0.317**
500_0.7network	0.066	0.175	0.249	0.133	0.145	0.127	0.249	0.145	**0.329**
500_0.8_prev	0.004	0.047	0.249	0.043	0.026	0.049	0.249	0.026	**0.323**
500_0.8network	0.066	0.151	0.249	0.116	0.124	0.151	0.249	0.124	**0.310**
dolphins	0.513	0.484	0.471	0.487	0.507	0.477	0.512	0.507	**0.727**
football	0.587	0.540	0.591	0.481	0.592	0.591	0.588	0.592	**0.788**
karate	0.318	0.286	0.307	0.306	0.342	0.289	0.307	0.342	**0.500**

single-objective optimization. Specifically, we have tested: Edge-Betweenness (EB) [61], Fast-Greedy (FG) [62], Label Propagation (LP) [63], Leading Eigenvector (LE) [57], MultiLevel (ML) [64], Walktrap (WT) [65], InfoMap (IM) [66], Cluster Louvain (CL) [64]. These algorithms are included in every community detection framework due to their popularity. This comparison allows us to evaluate the relevance of dealing with the CDP following a multi-objective approach. Table 11 shows the individual results obtained for each considered instance in terms of modularity.

It is worth mentioning that most of the algorithms included in the comparison are directly focused on optimizing modularity, although some of them such as the Label Propagation uses a different criterion as optimization metric. As it can be seen in the table, MOBVNS consistently obtains the best results in terms of modularity. Although in some instances, such as `1000_0.1network`, the improvement obtained is negligible, in most of the instances the multi-objective modeling of the problem allows the algorithm to reach considerably better modularity values. These results support the interest of tackling the community detection as a multi-objective optimization problem.

6. Conclusions and future work

In this paper, we have proposed a new metaheuristic method for community detection in social network based on Variable Neighborhood Search (VNS), where the set of initial set of non-dominated solutions is generated with a constructive procedure based on Greedy Randomized Adaptive Search Procedure methodologies (GRASP). The use of GRASP for the initial set of non-dominated solutions allows VNS to start the search from a promising region of the search space. The problem is addressed by optimizing the Radio Cut (RC) and the Negative Ratio

Table A.12
Information about the number of nodes, edges, and density of the instances derived from the LFR dataset.

Instances	Nodes	Edges	Density
500_0.1	500	10 674	0.08
500_0.1_prev	500	10 386	0.08
500_0.2	500	9940	0.07
500_0.2_prev	500	9444	0.07
500_0.3	500	9684	0.07
500_0.3_prev	500	9870	0.07
500_0.4	500	10 338	0.08
500_0.4_prev	500	10 326	0.08
500_0.5	500	10 310	0.08
500_0.5_prev	500	9894	0.07
500_0.6	500	10 312	0.08
500_0.6_prev	500	10 258	0.08
500_0.7	500	10 232	0.08
500_0.7_prev	500	9442	0.07
500_0.8	500	10 194	0.08
500_0.8_prev	500	9798	0.07
1000_0.1	1000	20 868	0.04
1000_0.1_prev	1000	19 330	0.03
1000_0.2	1000	20 142	0.04
1000_0.2_prev	1000	20 032	0.04
1000_0.3	1000	19 432	0.03
1000_0.3_prev	1000	19 218	0.03
1000_0.4	1000	20 014	0.04
1000_0.4_prev	1000	19 668	0.03
1000_0.5	1000	20 770	0.04
1000_0.5_prev	1000	19 122	0.03
1000_0.6	1000	20 084	0.04
1000_0.6_prev	1000	19 940	0.03
1000_0.7	1000	20 150	0.04
1000_0.7_prev	1000	19 900	0.03
1000_0.8	1000	20 640	0.04
1000_0.8_prev	1000	19 072	0.03

Table A.13
Information about the number of nodes, edges, and density of the instances derived from the LFR dataset.

Instances	Nodes	Edges	Density
5000_0.1	5000	100 430	0.01
5000_0.2	5000	1 011 070	0.08
5000_0.3	5000	103 662	0.01
5000_0.4	5000	101 732	0.01
5000_0.5	5000	99 770	0.01
5500_0.1	5500	112 858	0.01
5500_0.2	5500	111 868	0.01
5500_0.3	5500	111 592	0.01
6000_0.1	6000	121 486	0.01
6000_0.2	6000	121 692	0.01
6000_0.3	6000	119 742	0.01
6500_0.1	6500	134 384	0.01
6500_0.2	6500	132 336	0.01
6500_0.3	6500	133 520	0.01
7000_0.1	7000	142 156	0.01
7000_0.2	7000	141 552	0.01
7000_0.3	7000	142 424	0.01
7500_0.1	7500	151 060	0.01
7500_0.2	7500	152 778	0.01
7500_0.3	7500	152 972	0.01

Table A.14
Information about the number of nodes, edges, and density of the instances derived from the real-world instances dataset.

Instances	Nodes	Edges	Density
dolphins	62	159	0.08
football	115	613	0.09
karate	34	78	0.13
netscience	1589	2742	0.01
jazz	198	2742	0.14
musae_DE_edgesnetwork	9498	153 138	0.01
musae_ENGB_edgesnetwork	7126	35 324	0.01
musae_ES_edgesnetwork	4648	59 382	0.01
musae_FR_edgesnetwork	6549	112 666	0.01
musae_RU_edgesnetwork	4385	37 304	0.01

Association (NRA) metrics simultaneously as a bi-objective optimization problem. The quality of the solutions are evaluated by considering classic multi-objective metrics (Coverage, Hypervolume, ϵ-indicator, and Inverted Generational Distance) and two well-known metrics in the context of social network analysis (Normalized Mutual Information and Modularity).

The performed experiments show that the combination of GRASP with VNS in a multiobjective optimization framework is able to produce high quality solutions for the Multi-Objective Community Detection Problem (MOCDP), outperforming the best method found in the literature, which is based on a multiobjective evolutionary algorithm (LMOEA, Local Multi-Objective Evolutionary Algorithm). Additionally, the efficient implementation of the proposed algorithm is almost ten times faster than the original LMOEA, becoming more suitable for large scale networks.

In future works, it would be interesting to analyze the performance of a multi-objective variant when compared with the best single-objective methods found in the state of the art, even including different conflicting objectives to be compared. Additionally, the VNS approach presented in this work will be tested in new variants of the Community Detection Problem, such as Dynamic Community Detection or Overlapping Community Detection, to validate the potential of this framework for dealing with Community Detection Problems.

CRediT authorship contribution statement

Sergio Pérez-Peló: Conceptualization, Methodology, Software, Implementation, Writing. **Jesús Sánchez-Oro:** Conceptualization, Methodology, Software, Implementation, Writing. **Antonio Gonzalez-Pardo:** Conceptualization, Methodology, Software, Implementation, Writing. **Abraham Duarte:** Conceptualization, Methodology, Software, Implementation, Writing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was funded by “Ministerio de Ciencia, Innovación y Universidades, Spain” under grant ref. PGC2018-095322-B-C22, “Comunidad de Madrid” and “Fondos Estructurales” of European Union with grant refs. S2018/TCS-4566, Y2018/EMT-5062.

Appendix. Instances description

See Tables A.12–A.14.

References

[1] D. Camacho, Á. Panizo-LLedot, G. Bello-Orgaz, A. Gonzalez-Pardo, E. Cambria, The four dimensions of social network analysis: An overview of research methods, applications, and software tools, Inf. Fusion 63 (2020) 88–120.
[2] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2004) 026113.
[3] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512.
[4] M. Girvan, M.E. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. 99 (12) (2002) 7821–7826.
[5] Q. Cai, M. Lijia, M. Gong, A survey on network community detection based on evolutionary computation, Int. J. Bio-Inspired Comput. 8 (2014).
[6] P. Hansen, N. Mladenović, Variable neighborhood search, in: Search Methodologies, Springer, 2014, pp. 313–337.
[7] A. Duarte, J.J. Pantrigo, E.G. Pardo, N. Mladenovic, Multi-objective variable neighborhood search: an application to combinatorial optimization problems, J. Global Optim. 63 (3) (2015) 515–536.
[8] E. de Siqueira, M. Souza, S. de Souza, A multi-objective variable neighborhood search algorithm for solving the hybrid flow shop problem, Electron. Notes Discrete Math. 66 (2018) 87–94.
[9] L. Peel, D.B. Larremore, A. Clauset, The ground truth about metadata and community detection in networks, Sci. Adv. 3 (5) (2017) e1602548.
[10] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2004) 026113.
[11] X. Li, C. Gao, A novel community detection algorithm based on clonal selection, J. Computational Information Systems 9 (2013) 1899–1906.
[12] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, D. Wagner, Maximizing modularity is hard, 2006, arXiv preprint Physics/0608255.
[13] J. Reichardt, S. Bornholdt, Statistical mechanics of community detection, Phys. Rev. E 74 (2006) 016110.
[14] S. Fortunato, M. Barthélemy, Resolution limit in community detection, Proc. Natl. Acad. Sci. 104 (1) (2007) 36–41.
[15] S. Pérez-Peló, J. Sánchez-Oro, R. Martín-Santamaría, A. Duarte, On the analysis of the influence of the evaluation metric in community detection over social networks, Electronics 8 (1) (2018).
[16] A. Gonzalez-Pardo, J.J. Jung, D. Camacho, ACO-based clustering for ego network analysis, Future Gener. Comput. Syst. 66 (2017) 160–170.
[17] F. Cheng, T. Cui, Y. Su, Y. Niu, X. Zhang, A local information based multi-objective evolutionary algorithm for community detection in complex networks, Appl. Soft Comput. 69 (2018) 357–367.
[18] N.P. Nguyen, T.N. Dinh, Y. Shen, M.T. Thai, Dynamic social community detection and its applications, PLoS One 9 (4) (2014) 1–18.

[19] V. Labatut, J.-M. Balasque, Detection and interpretation of communities in complex networks: Practical methods and application, in: Computational Social Networks: Tools, Perspectives and Applications, Springer London, London, 2012, pp. 81–113.
[20] A. Fang, I. Ounis, P. Habel, C. Macdonald, N. Limsopatham, Topic-centric classification of twitter user's political orientation, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2015, pp. 791–794.
[21] J. Borge-Holthoefer, W. Magdy, K. Darwish, I. Weber, Content and network dynamics behind egyptian political polarization on twitter, in: Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, ACM, 2015, pp. 700–711.
[22] E. Osaba, J. Del Ser, D. Camacho, M. Nekane Bilbao, X.-S. Yang, Community detection in networks using bio-inspired optimization: Latest developments, new results and perspectives with a selection of recent meta-heuristics, Appl. Soft Comput. 87 (2020) 106010.
[23] S. Srinivas, C. Rajendran, Community detection and influential node identification in complex networks using mathematical programming, Expert Syst. Appl. 135 (2019) 296–312.
[24] E. Alinezhad, B. Teimourpour, M.M. Sepehri, M. Kargari, Community detection in attributed networks considering both structural and attribute similarities: two mathematical programming approaches, Neural Comput. Appl. 32 (8) (2020) 3203–3220.
[25] C. Pizzuti, Evolutionary computation for community detection in networks: A review, IEEE Trans. Evol. Comput. 22 (3) (2018) 464–483.
[26] A. Said, R.A. Abbasi, O. Maqbool, A. Daud, N.R. Aljohani, CC-GA: A clustering coefficient based genetic algorithm for detecting communities in social networks, Appl. Soft Comput. 63 (2018) 59–70.
[27] M. Moradi, S. Parsa, An evolutionary method for community detection using a novel local search strategy, Physica A 523 (2019) 457–475.
[28] Q. Cai, M. Gong, L. Ma, S. Ruan, F. Yuan, L. Jiao, Greedy discrete particle swarm optimization for large-scale social network clustering, Inform. Sci. 316 (2015) 503–516.
[29] Z. Yang, R. Algesheimer, C.J. Tessone, A comparative analysis of community detection algorithms on artificial networks, Sci. Rep. 6 (2016) 30750.
[30] Y.-N. Guo, X. Zhang, D.-W. Gong, Z. Zhang, J.-J. Yang, Novel interactive preference-based multiobjective evolutionary optimization for bolt supporting networks, IEEE Trans. Evol. Comput. 24 (4) (2019) 750–764.
[31] Y. Guo, H. Yang, M. Chen, J. Cheng, D. Gong, Ensemble prediction-based dynamic robust multi-objective optimization methods, Swarm Evol. Comput. 48 (2019) 156–171.
[32] P. Bedi, C. Sharma, Community detection in social networks, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 6 (3) (2016) 115–135.
[33] M. Gong, L. Ma, Q. Zhang, L. Jiao, Community detection in networks by using multiobjective evolutionary algorithm with decomposition, Physica A 391 (15) (2012) 4050–4060.
[34] C. Shi, Z. Yan, Y. Cai, B. Wu, Multi-objective community detection in complex networks, Appl. Soft Comput. 12 (2) (2012) 850–859.
[35] B. Amiri, L. Hossain, J.W. Crawford, R.T. Wigand, Community detection in complex networks: Multi–objective enhanced firefly algorithm, Knowl.-Based Syst. 46 (2013) 1–11.
[36] M. Gendreau, J.-Y. Potvin, Handbook of Metaheuristics, second ed., Springer Publishing Company, Incorporated, 2010.
[37] A. Herrán, M.J. Colmenar, A. Duarte, A variable neighborhood search approach for the vertex bisection problem, Inform. Sci. 476 (2019) 1–18.
[38] J. Sánchez-Oro, A.D. López-Sánchez, M.J. Colmenar, A general variable neighborhood search for solving the multi-objective open vehicle routing problem, J. Heuristics (2017) 1–30.
[39] A. Duarte, J. Sánchez-Oro, N. Mladenović, R. Todosijević, Variable neighborhood descent, Handb. Heuristics (2018) 341–367.
[40] J. Sánchez-Oro, A. Martínez-Gavara, M. Laguna, A. Duarte, R. Martí, Variable neighborhood descent for the incremental graph drawing, Electron. Notes Discrete Math. 58 (2017) 183–190, 4th International Conference on Variable Neighborhood Search.
[41] A. Duarte, L.F. Escudero, R. Martí, N. Mladenovic, J.J. Pantrigo, J. Sánchez-Oro, Variable neighborhood search for the vertex separation problem, Comput. Oper. Res. 39 (12) (2012) 3247–3255.
[42] J. Sánchez-Oro, A.D. López-Sánchez, J.M. Colmenar, A general variable neighborhood search for solving the multi-objective open vehicle routing problem, J. Heuristics (2017).
[43] P. Hansen, N. Mladenović, D. Perez-Britos, Variable neighborhood decomposition search, J. Heuristics 7 (4) (2001) 335–350.
[44] P. Hansen, N. Mladenović, Variable neighborhood search, in: Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques, Springer US, Boston, MA, 2005, pp. 211–238.
[45] E.G. Pardo, N. Mladenović, J.J. Pantrigo, A. Duarte, Variable formulation search for the cutwidth minimization problem, Appl. Soft Comput. 13 (5) (2013) 2242–2252.
[46] X. Song, D. Jones, N. Asgari, T. Pigden, Multi-objective vehicle routing and loading with time window constraints: a real-life application, Ann. Oper. Res. (2019) 1–27.
[47] P. Hansen, N. Mladenović, J.A.M. Pérez, Variable neighbourhood search: methods and applications, Ann. Oper. Res. 175 (1) (2010) 367–407.
[48] T.A. Feo, M.G.C. Resende, A probabilistic heuristic for a computationally difficult set covering problem, Oper. Res. Lett. 8 (2) (1989) 67–71.
[49] T.A. Feo, M.G.C. Resende, S.H. Smith, A greedy randomized adaptive search procedure for maximum independent set., Oper. Res. 42 (5) (1994) 860–878.
[50] M.G.C. Resende, C.C. Ribeiro, GRASP: Greedy randomized adaptive search procedures, in: Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques, Springer US, Boston, MA, 2014, pp. 287–312.
[51] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E 78 (2008) 046110.
[52] W.W. Zachary, An information flow model for conflict and fission in small groups, J. Anthropol. Res. 33 (4) (1977) 452–473.
[53] S. Gregory, Finding overlapping communities in networks by label propagation, New J. Phys. 12 (10) (2010) 103018.
[54] M.E.J. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. 103 (23) (2006) 8577–8582.
[55] P.M. Gleiser, L. Danon, Community structure in jazz, Adv. Complex Syst. 6 (04) (2003) 565–573.
[56] R.A. Rossi, N.K. Ahmed, The network data repository with interactive graph analytics and visualization, in: AAAI, 2015.
[57] M.E. Newman, Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E 74 (3) (2006) 036104.
[58] B. Rozemberczki, C. Allen, R. Sarkar, Multi-scale attributed node embedding, 2019.
[59] M. Li, X. Yao, Quality evaluation of solution sets in multiobjective optimisation: A survey, ACM Comput. Surv. 52 (2) (2019) 26.
[60] L. Danon, A. Díaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, J. Stat. Mech. Theory Exp. 2005 (09) (2005) P09008.
[61] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, PNAS 99 (12) (2002) 7821–7826.
[62] A. Clauset, M.E.J. Newman, C. Moore, Finding community structure in very large networks, Phys. Rev. E 70 (2004) 066111.
[63] U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Phys. Rev. E 76 (2007) 036106.
[64] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. Theory Exp. 2008 (10) (2008) P10008.
[65] M. Rosvall, C.T. Bergstrom, Maps of random walks on complex networks reveal community structure, Proc. Natl. Acad. Sci. 105 (4) (2008) 1118–1123.
[66] M. Rosvall, D. Axelsson, C.T. Bergstrom, The map equation, Eur. Phys. J. Spec. Top. 178 (1) (2009) 13–23.