Physica A 602 (2022) 127628

Contents lists available at ScienceDirect

Physica A

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

A mathematical programming approach to overlapping community detection

Stefano Benati [a,1], Justo Puerto [b,1], Antonio M. Rodríguez-Chía [c,1], Francisco Temprano [b,*,1]

[a] Dipartimento di Sociologia e Ricerca Sociale, Università di Trento, Via Verdi 26, 38122 Trento, Italy
[b] IMUS, Universidad de Sevilla, Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
[c] Faculty of Sciences, Universidad de Cádiz, Avda. República Saharaui, 11510 Puerto Real (Cádiz), Spain

ARTICLE INFO

Article history:
Received 25 November 2021
Received in revised form 24 February 2022
Available online 2 June 2022

Keywords:
Complex networks
Overlapping community detection
Modularity
Fuzzy membership
Mathematical programming

ABSTRACT

We propose a new optimization model to detect overlapping communities in networks. The model elaborates suggestions contained in Zhang et al. (2007), in which overlapping communities were identified through the use of a fuzzy membership function, calculated as the outcome of a mathematical programming problem. In our approach, we retain the idea of using both mathematical programming and fuzzy membership to detect overlapping communities, but we replace the fuzzy objective function proposed there with another one, based on the Newman and Girvan's definition of modularity. Next, we formulate a new mixed-integer linear programming model to calculate optimal overlapping communities. After some computational tests, we provide some evidence that our new proposal can fix some biases of the previous model, that is, its tendency of calculating communities composed of almost all nodes. Conversely, our new model can reveal other structural properties, such as nodes or communities acting as bridges between communities. Finally, as mathematical programming can be used only for moderate size networks due to its computation time, we proposed two heuristic algorithms to solve the largest instances, that compare favourably to other methodologies.

1. Introduction

The community detection problem is one of the most studied and interesting problems in networks science. It consists in classifying the units of a population into groups using only information about their links, so that units of the same group can be interpreted as communities. The model is formulated on a network $G = (V, E)$ in which the vertices V stand for the units, such as individuals, companies, and so on, and the edges E stand for the relations between units, such as kinship, commercial alliances, and so on. Community detection applications can be found in several disciplines, such as biology [1], ecology [2,3], economics [4], sociology [5,6], and many more.

A crucial assumption of a *standard* community detection model is that communities form a partition of V, that is, every unit belongs to one and only one community. This assumption is problematic, as there are applications in which units can realistically belong to two or more communities. For example, the sociological literature documented the role of

* Corresponding author.
E-mail addresses: stefano.benati@unitn.it (S. Benati), puerto@us.es (J. Puerto), antonio.rodriguezchia@uca.es (A.M. Rodríguez-Chía), ftgarcia@us.es (F. Temprano).

[1] All the authors contributed evenly to all the aspects of this research.

https://doi.org/10.1016/j.physa.2022.127628

individuals who close gaps between communities as members of *two* communities, see [7], and playing an important role on the functioning of the network as they are *bridges*, for example, they foster information spreading. Another example is the case of the protein network described in [1], in which proteins are nodes and communities are proteins that carry on one task, but one protein can interact with other communities to accomplish other functions and so it belongs to two or more communities. A standard community detection model may fail to recognize these units, and so emerged the quest for determining what is the best way for finding *overlapping* communities, as documented in the seminal paper [8] and the survey [9].

The principles that were applied by the algorithms detecting overlapping communities are the same principles used to detect non-overlapping communities, for example constructive methods, Fellows et al. [10], hierarchical clustering, Lancichinetti et al. [11], optimization methods such as mathematical programming in [8], and so on: Other approaches can be found in [12–16] and the survey by Xie et al. [9]. Here, we consider the methods that are based on the comparison of an objective function: That is, given two possible partitions, the best is the one with the highest objective value. When an objective function is used, then there is a strong consensus that the modularity, as defined by Newman and Girvan [17], is the most appropriate measure to detect disjoint communities. Nevertheless, the question of measuring the goodness (or quality) of overlapping communities is more controversial, as different measures were proposed by the literature. In [8], Newman's modularity function is extended using the fuzzy-c mean, so that a fuzzy membership function accounts for the possibility of multiple communities membership. As that objective function is non-linear, optimal solutions were not calculated and applications are tested using an heuristic algorithm. The contribution in [18] follows the same stream, as a new modularity function is introduced to account for multiple memberships. In this case, group memberships are represented as probabilities instead of fuzzy measures and a genetic heuristic algorithm is proposed to calculate the communities. In [1], the standard Newman's modularity function is optimized twice: Firstly, to find a node partition in which some nodes could be identified as bridges. Secondly, modularity is applied as an optimization model in which some node can be assigned to more than one community, through an appropriate modification of the mathematical programming model. In [19], a cooperative game is defined on the network and its characteristic function is used to measure players' Shapley values. Communities are identified as stable coalitions, e.g. the ones in which no player can unilaterally improve its Shapley values by moving into another coalition. Again, no attempt is made to formalize the optimization problem and communities are calculated through heuristics. Unfortunately, the use of heuristic methods to solve a well-posed mathematical programming problem can bring about biases in computing the correct overlapping communities. Indeed, heuristics are devised combining mathematical programming concepts with constructive rule-of-thumb considerations, to the point that empiric solutions can be very far from the optimal. As a matter of fact, when we tested some of these models with more accuracy, we found that sometimes they calculate inconsistent communities, for example, this is the case of the model proposed in [8]. As we documented in our contribution, when we calculated the optimal solution of that mathematical programming problem, we discovered that some overlapping communities can be the same community counted twice.

Here, we propose an amendment of the model of Zhang et al. [8], that we could prove it calculates meaningful overlapping communities, e.g., they reveal structural properties of nodes, or of group of nodes, that were not detected by the previous contribution. The new model improves basic ideas contained in [8] with new contributions. They are:

- Using a fuzzy membership function u_{ik} to determine whether a node i belongs to a community C_k;
- Calculating u through the optimization of an objective function;
- Using as objective function a variation of the Newman and Girvan's modularity index.

To begin with, our tests revealed that the fuzzy modularity function optimized in [8] biases optimal solutions to grand communities, a.g. communities that are formed by all or almost all nodes. Therefore, we introduce a different fuzzy modular objective function, that avoids this bias. Next, we formulate a Mixed-Integer Linear Programming (MILP) model that maximizes the fuzzy modular function under some linear and integer constraints. The advantage of this approach is that, at least for instances of moderate size, the optimal solution can be calculated *exactly* with off-the-shelf solvers. Calculating *optimal* instead of *empiric* overlapping communities has several advantages, the most important is that the quality and reliability of the communities can be established without the biases that are due to the use of an heuristic. When we applied our new model to some standard benchmark networks, we found meaningful overlapping communities. Unfortunately, MILP problems can be solved for moderate size instances only. However, both the new objective function and the MILP structure lead naturally to new heuristic procedures, that takes advantage of the mathematical formulation of the problem. When applied to large datasets, the heuristics are fast and reliable.

After this introduction, the paper is organized as follows. In Section 3 we provide an exact formulation of Zhang et al.'s model as Mixed Integer Concave Problem (MICP) so we can calculate its optimal solution for some test problems *exactly*. We discover that optimal solutions are quite different from the heuristic ones reported in [8] and, unfortunately, they are to a large extent meaningless, as they are the same community counted twice, or they are communities formed by all nodes but one. The misbehaviour is due to the fuzzy modularity index that was initially proposed, so we suggest a way to correct the index to avoid those inconsistent results. Using the new index, a new MILP model is formulated and successfully tested in Section 4. Next, in Section 5, we propose new heuristic algorithms to calculate overlapping communities for large size networks. It can be seen that they find optimal communities in short computing times. The paper concludes with some remarks and suggestions for future research in Section 6.

2. The modularity function for overlapping communities

Let $G = (V, E)$ be a non-oriented and non-weighted network (or graph), with vertex set $V = \{1, \ldots, n\}$ and edge set E, represented by the adjacency matrix A_{ij}, e.g., $a_{ij} = 1$ if $(i, j) \in E$, $a_{ij} = 0$ otherwise. Let $m = |E|$ and k_i the adjacency degree of node $i \in V$. In [17], modularity optimization is proposed to detect the non-overlapping communities of a graph. The modularity function compares the edge density between nodes of the same community with the expected edge density between the same nodes, but under the assumption that they do not form a community. The expected edge density is obtained from what is called the configuration model. The configuration model is the random graph obtained when edges are placed between nodes randomly, but keeping constant the adjacency degree of each node. In the configuration model, for two nodes i and j the expected number of edges between them is $\frac{k_i k_j}{2m}$. Let $P = \{C_1, \ldots, C_q\}$ be a partition of V. Then, the modularity function of the partition $P = \{C_1, \ldots, C_q\}$ is:

$$\frac{1}{2m} \sum_{i,j \in V} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(i, j), \tag{1}$$

where δ is the Kronecker function, attaining the value 1 if i and j belong to the same community, 0 otherwise. The formula of the modularity can be extended to weighted graphs as well. The entries A_{ij} are replaced by weights W_{ij}, k_i by the sum of the weights associated with arcs adjacent to i and m is replaced by the total sum of weights $W = \sum_{(i,j) \in E} W_{ij}$, as described in [20].

Let n_c be the optimal number of communities and V_k be the set of nodes that belong to community k. The previous modularity function (1) can be expressed as:

$$\sum_{k=1}^{n_c} \left(\frac{\sum_{i,j \in V_k} A_{ij}}{2m} - \left(\frac{\sum_{i \in V_k} k_i}{2m} \right)^2 \right). \tag{2}$$

Maximizing the modularity reveals the network community structure, defined as a *partition* of V. Nevertheless, there are some applications in which a *hard* partition (hard in the sense that a node can belong to only one community) cannot reveal interactive effects between nodes of different communities. When communities overlap, functions (1) or (2) are not sufficient to determine the network structure, so that, in [8], it is proposed to combine modularity with *soft* partitions.

Instead of assuming that the membership of unit i to community k is represented by a 0-1 number, there is a fuzzy value $0 \leq u_{ik} \leq 1$ that represents the membership of node i to community k. This value u_{ik} is called membership function: If $u_{ik} = 1$, then i belongs to community k for certain, if $u_{ik} = 0$, then i does not belong to community k for sure, while values u_{ik} in the range represent the uncertainty of the membership. It follows that the membership sum is one: $\sum_{k=1}^{n_c} u_{ik} = 1 \quad \forall i \in V$, where n_c is the number of communities.

Membership functions identify the communities structure. For a given threshold value λ, a community V_k is defined as the set of nodes whose membership exceeds the threshold λ, $V_k = \{i \in V : u_{ik} > \lambda\}$. From the definition of u and λ, communities V_k, $k = 1, \ldots, n_c$ may not form a partition, but are admitted to overlap: A node i may belong to more than one community. Clearly, overlapping communities depend on λ: If $\lambda > 0.5$, than nodes belong to one community at most and no community can overlap. Next, if $0.334 < \lambda \leq 0.5$, each node can belong to a maximum of 2 communities. In general, assuming p is a positive integer, if $\frac{1}{p+1} < \lambda \leq \frac{1}{p}$, each node can belong to a maximum number p of communities.

The fuzzy modularity function for overlapping communities, introduced in [8], is:

$$\sum_{k=1}^{n_c} \left(\frac{\sum_{i,j \in V_k} A_{ij} \frac{u_{ik} + u_{jk}}{2}}{2m} - \left(\frac{\sum_{i,j \in V_k} A_{ij} \frac{u_{ik} + u_{jk}}{2} + \sum_{i \in V_k, j \notin V_k} A_{ij} \frac{u_{ik} + (1 - u_{jk})}{2}}{2m} \right)^2 \right). \tag{3}$$

This function is a modification of the original modularity function (2). It is obtained by weighting each edge (i, j) by the average of u_{ik} and u_{jk} if $i, j \in V_k$, or by the average of u_{ik} and $1 - u_{jk}$ if $i \in V_k, j \notin V_k$. Note that the expression (3) can be applied to weighted graphs, too. It is sufficient to change the entry A_{ij} with an edge weight, W_{ij}, and m replaced by the total sum of weights $W = \sum_{(i,j) \in E} W_{ij}$.

3. An exact mathematical programming formulation of the maximum fuzzy modularity problem

Finding the n_c overlapping communities that maximizes the function (3) is a hard problem to solve. Indeed, it is a mixed binary polynomial optimization problem which can be easily proven to be NP-hard [21] and, in [8], only a heuristic procedure is proposed indeed. Unfortunately, heuristic procedures may result with sub-optimal solutions that can be very different from the optimal. Therefore, to test the effectiveness of that model, we formulated a mathematical programming model that exactly maximizes function (3).

Problem variables are the membership functions u_{ik}, such that $0 \leq u_{ik} \leq 1$ for all i and k. Next, hard membership functions x_{ik}, for all i and k, that depend on variables u_{ik} and threshold parameter λ, are defined in the following way:

$$x_{ik} = \begin{cases} 1, & \text{if node } i \text{ is assigned to community } k, \text{ that is, if } u_{ik} > \lambda, \\ 0, & \text{otherwise.} \end{cases}$$

The constraint between variables x_{ik} and u_{ik} is:

$$x_{ik} \geq u_{ik} - \lambda, \quad \forall i = 1, \ldots, n, \ k = 1, \ldots, n_c, \tag{4}$$

$$x_{ik} \leq 1 + u_{ik} - \lambda, \quad \forall i = 1, \ldots, n, \ k = 1, \ldots, n_c. \tag{5}$$

The objective function (3) can be formulated using variables x. The term $\sum_{i,j \in V_k} A_{ij} \frac{u_{ik}+u_{jk}}{2}$ is rewritten:

$$\sum_{i,j=1}^{n} A_{ij} \frac{u_{ik} + u_{jk}}{2} x_{ik} x_{jk} = \sum_{i,j=1}^{n} A_{ij} \frac{u_{ik}}{2} x_{ik} x_{jk} + \sum_{i,j=1}^{n} A_{ij} \frac{u_{jk}}{2} x_{ik} x_{jk}.$$

Assuming a non-oriented network, e.g. $A_{ij} = A_{ji}$, then the previous expression is:

$$\sum_{i,j=1}^{n} A_{ij} u_{ik} x_{ik} x_{jk}.$$

Next, the term $\sum_{i \in V_k, j \notin V_k} A_{ij} \frac{u_{ik}+(1-u_{jk})}{2}$ is rewritten as:

$$\sum_{i,j=1}^{n} A_{ij} \frac{u_{ik} + (1 - u_{jk})}{2} x_{ik}(1 - x_{jk}) =$$

$$\sum_{i,j=1}^{n} \frac{A_{ij}}{2} (u_{ik} x_{ik} + x_{ik} - u_{jk} x_{ik} - u_{ik} x_{ik} x_{jk} - x_{ik} x_{jk} + u_{jk} x_{ik} x_{jk}).$$

Matrix A is symmetric, therefore: $\sum_{i,j=1}^{n} A_{ij}(u_{jk} x_{ik} x_{jk} - u_{ik} x_{ik} x_{jk}) = 0$, and, after simplifying, we obtain:

$$\sum_{i,j=1}^{n} \frac{A_{ij}}{2} (u_{ik} x_{ik} + x_{ik} - u_{jk} x_{ik} - x_{ik} x_{jk}), \tag{6}$$

that is a quadratic polynomial. For computational purposes, it is convenient to replace quadratic with linear terms, e.g., $w_{ijk} = u_{ik} x_{ik} x_{jk}$, $s_{ijk} = u_{jk} x_{ik}$ and $z_{ijk} = x_{ik} x_{jk}$, and then to introduce additional linear constraints to represent these identities. Finally, the mathematical programming formulation of maximizing function (3) is:

$$\text{(F-MOD)} \quad \max \frac{1}{2m} \sum_{k=1}^{n_c} \left(\left(\sum_{i,j=1}^{n} A_{ij} w_{ijk} \right) - \frac{\left(\sum_{i,j=1}^{n} A_{ij} w_{ijk} + \sum_{i,j=1}^{n} A_{ij} \frac{s_{iik} - s_{ijk} - z_{ijk} + x_{ik}}{2} \right)^2}{2m} \right) \tag{7}$$

$$s.t. : (4), (5),$$

$$\sum_{k=1}^{n_c} x_{ik} \geq 1, \quad \forall i = 1, \ldots, n, \tag{8}$$

$$\sum_{k=1}^{n_c} u_{ik} = 1, \quad \forall i = 1, \ldots, n, \tag{9}$$

$$w_{ijk} \leq x_{ik}, \quad \forall i, j = 1, \ldots, n, \ k = 1, \ldots, n_c, \tag{10}$$

$$w_{ijk} \leq x_{jk}, \quad \forall i, j = 1, \ldots, n, \ k = 1, \ldots, n_c, \tag{11}$$

$$w_{ijk} \leq u_{ik}, \quad \forall i, j = 1, \ldots, n, \ k = 1, \ldots, n_c, \tag{12}$$

$$w_{ijk} \geq u_{ik} + x_{ik} + x_{jk} - 2, \quad \forall i, j = 1, \ldots, n, \ k = 1, \ldots, n_c \tag{13}$$

$$s_{ijk} \leq x_{ik}, \quad \forall i, j = 1, \ldots, n, \ k = 1, \ldots, n_c, \tag{14}$$

$$s_{ijk} \leq u_{jk}, \quad \forall i, j = 1, \ldots, n, \ k = 1, \ldots, n_c, \tag{15}$$

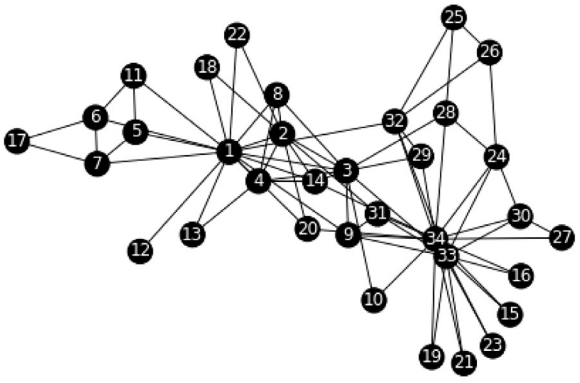
$$s_{ijk} \geq u_{jk} + x_{ik} - 1, \quad \forall i, j = 1, \ldots, n, \ k = 1, \ldots, n_c, \tag{16}$$

$$z_{ijk} \leq x_{ik}, \quad \forall i, j = 1, \ldots, n, \ k = 1, \ldots, n_c, \tag{17}$$

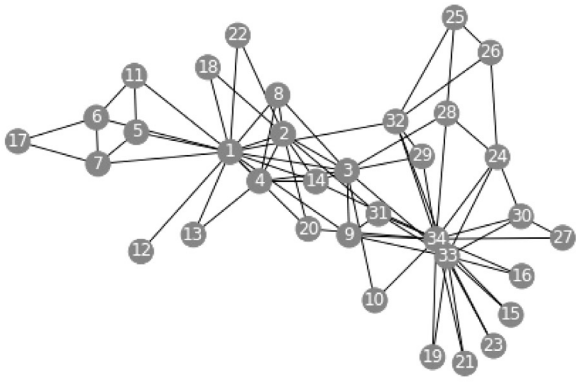
$$z_{ijk} \leq x_{jk}, \quad \forall i, j = 1, \ldots, n, \ k = 1, \ldots, n_c, \tag{18}$$

$$z_{ijk} \geq x_{ik} + x_{jk} - 1, \quad \forall i, j = 1, \ldots, n, \ k = 1, \ldots, n_c, \tag{19}$$

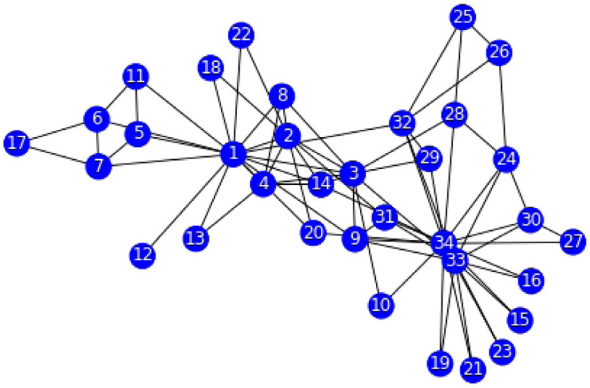
$$w_{ijk} \in [0, 1] \quad \forall i, j = 1, \ldots, n, \ k = 1, \ldots, n_c, \tag{20}$$



Community 1.



Community 2.



Community 3.

Fig. 1. Optimal communities of the Zachary's karate club structure for the formulation (F-MOD) with $n_c = 3$ and $\lambda = 0.25$.

$$z_{ijk}, s_{ijk}, \in [0, 1] \quad \forall i, j = 1, \ldots, n,\ k = 1, \ldots, n_c, \tag{21}$$

$$u_{ik} \in [0, 1], \quad \forall i = 1, \ldots, n,\ k = 1, \ldots, n_c, \tag{22}$$

$$x_{ik} \in \{0, 1\}, \quad \forall i = 1, \ldots, n,\ k = 1, \ldots, n_c. \tag{23}$$

The objective function (7) is the fuzzy modularity function with soft membership variables u. Constraints (4) and (5) represent the node memberships to communities, that depend on threshold λ. Constraints (8) impose that every node belongs to at least one community. Membership sum is equal to one for (9). Finally, the families of constraints (10)–(19) represent the identities: $w_{ijk} = u_{ik}x_{ik}x_{jk}$, $s_{ijk} = u_{jk}x_{ik}$ and $z_{ijk} = x_{ik}x_{jk}$, as linear constraints.

The objective function of problem (F-MOD) is concave, constraints are linear, variables are continuous or binary, so that problem (F-MOD) is MICP that can be solved by off-the-shelf mathematical programming solver as Gurobi, Cplex, and others. In our tests we embedded the Gurobi solver in a Python program. The advantage of problem (F-MOD) as MICP is that solutions are optimal, so that substantial interpretation of communities is not biased by the way in which a heuristic procedure is implemented and calculate the sub-optimal solution to the problem. To compare the two approaches, optimal vs suboptimal, we consider two networks: Zachary's karate club and American college football teams, for which suboptimal solutions are contained in [8]. We will show that the way in which the heuristic is implemented strongly biases the calculation of the overlapping communities.

We begin applying formulation (F-MOD) to the Zachary's karate club network, Zachary [22], with the same parameters as in [8]: $n_c = 3$, $\lambda = 0.25$. In Fig. 1, the optimal communities from MICP are reported. The model identifies as communities the whole set of nodes V, counted three times, as $n_c = 3$. The result is meaningless for a substantial analysis, casting a doubt about the validity of the index (3). In Fig. 2, communities calculated by the heuristic algorithm in [8] are reported for the same parameters. It can be seen that they are very different from the optimal, casting a further doubt about the validity of the heuristic.

To demonstrate further the previous findings, we analysed the case of the American college football teams network, Girvan and Newman [23], with parameters $n_c = 10$ and $\lambda = 0.1$. Due to the network size, we stopped the computation after

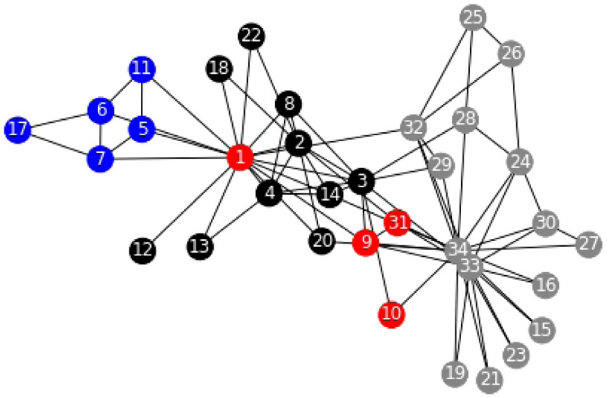


Fig. 2. Zachary's karate club community structure obtained by procedure in [8] assuming $n_c = 3$ and $\lambda = 0.25$.

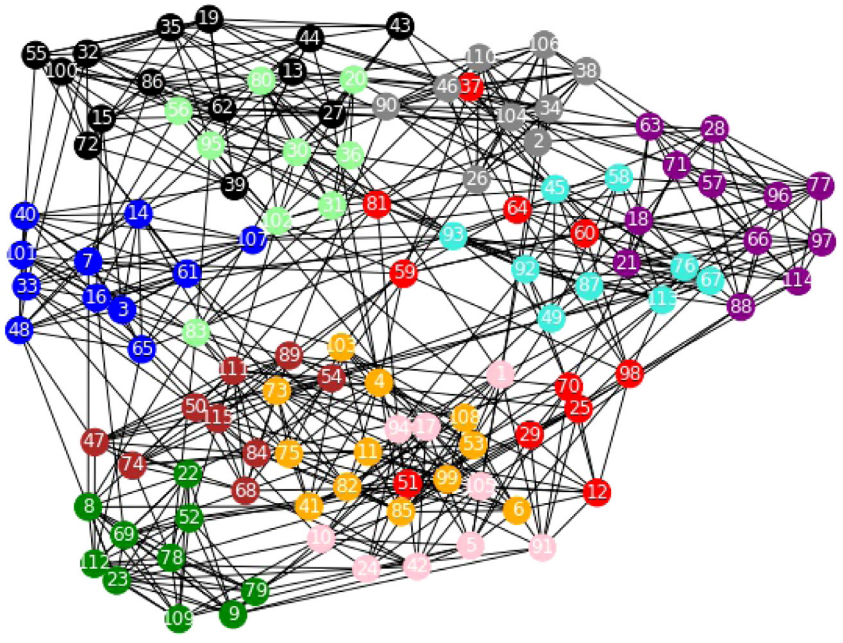


Fig. 3. American college football team community structure obtained by the procedure in [8] with $n_c = 10$ and $\lambda = 0.1$, where red nodes represent intersection nodes.

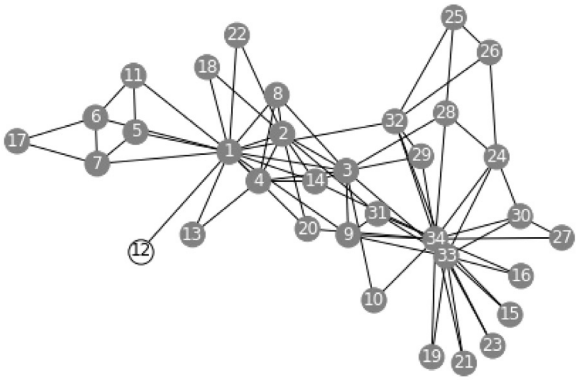
24 h and we relied to a suboptimal solution of problem (F-MOD). The algorithm terminates with a gap of 7.75% between the best feasible solution and the best upper bound. The best feasible solution is ten communities all consisting of all nodes. In Fig. 3, the community structure calculated by the algorithm in [8] for the same parameters is reported. Again, it proves that the heuristic developed by Zhang et al. [8] does not optimize function (3).

One may wonder whether the previously documented biases were due to the MICP formulation instead, as it allows coincident communities. So, we repeated the previous experiments with a MICP formulation in which not only coincident, but also included communities are forbidden. That is, solutions for which $C_k \subseteq C_r$ for some k and r are not allowed.

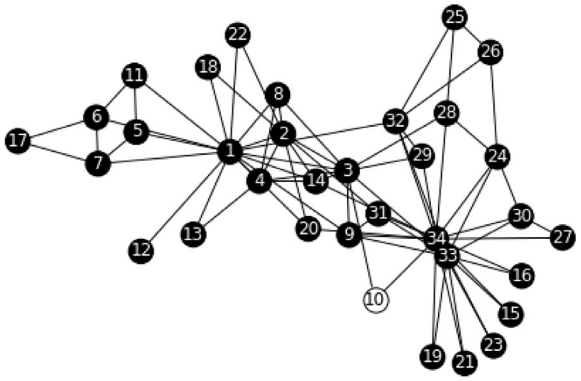
To prevent the inclusion between communities, the next variables and constraints, for $i = 1, \ldots, n$ and $1 \leq k < r \leq n_c$, must be introduced:

$$h_{ikr} = \begin{cases} 1, & \text{if } i \text{ belongs to community } r \text{ and not to community } k, \\ 0, & \text{otherwise.} \end{cases}$$

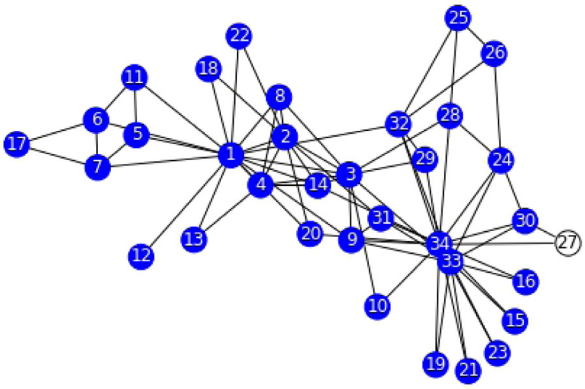
h-Variables depend on x-variables, as $h_{ikr} = x_{ir}(1 - x_{ik})$. Moreover, we can assume that optimal communities are ordered from the one with the largest size to the one with the smallest size, so that a community with an index k cannot



Community 1.



Community 2.



Community 3.

Fig. 4. Optimal communities provided by formulation (F-MOD-NI) for Zachary's karate club structure with $n_c = 3$ and $\lambda = 0.25$.

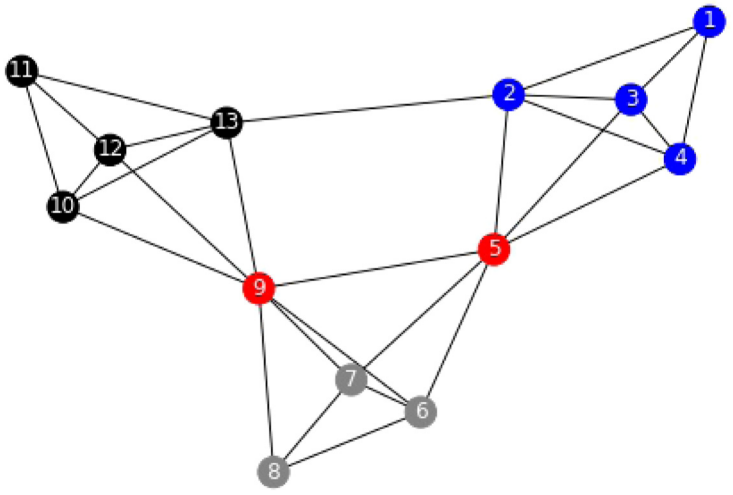
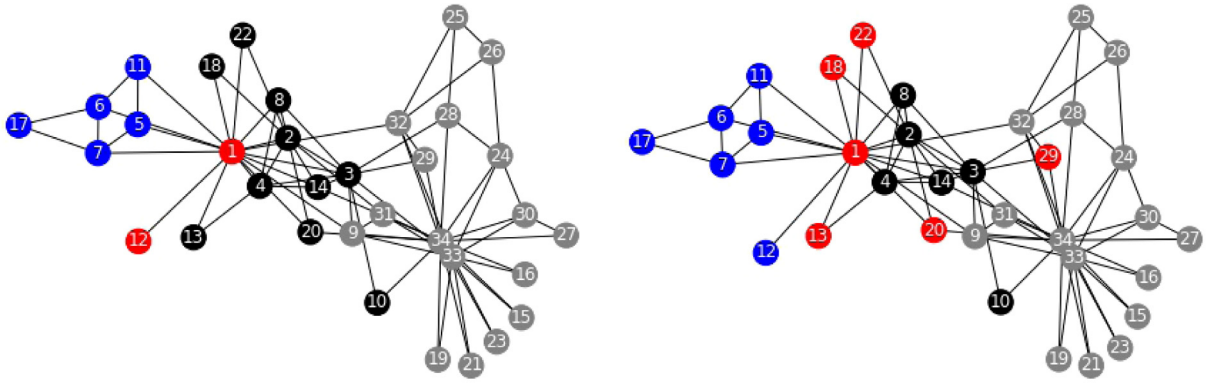


Fig. 5. Community structure obtained by formulation (NEW-MOD) on the network of first example in [8], where the red nodes represent the overlap between communities assuming $n_c = 3$ and $\lambda = 0.15$. Our result matches the one reported in [8].



(a) Zachary's club communities with $n_c = 3$ and $\lambda = 0.25$. (b) Zachary's club communities with with $n_c = 3$ and $\lambda = 0.1$.

Fig. 6. Zachary's club community structures.

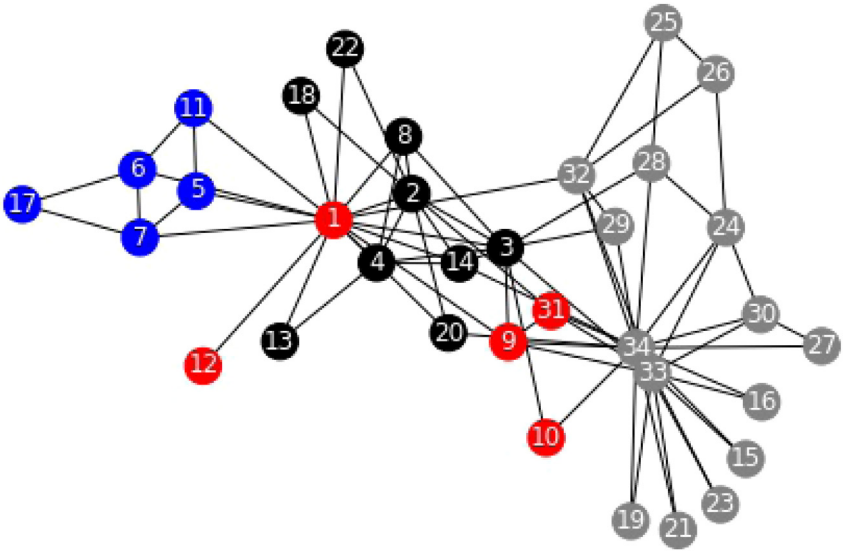


Fig. 7. Zachary's club communities with $n_c = 3$, $\lambda = 0.25$ and bridge constraints on nodes 1, 9, 10, 31.

be smaller than the one indexed by $k+1$ for all $k = 1, \ldots, n_c - 1$. This can be enforced with the constraints:

$$\sum_{i=1}^{n} x_{ik} \geq \sum_{i=1}^{n} x_{i,k+1}, \quad \forall k = 1, \ldots, n_c - 1. \tag{24}$$

Finally, we obtain the formulation of problem (F-MOD) that prevents the inclusion between communities:

$$\text{(F-MOD-NI)} \quad \max \ \frac{1}{2m} \sum_{k=1}^{n_c} \left(\left(\sum_{i,j=1}^{n} A_{ij} w_{ijk} \right) - \frac{\left(\sum_{i,j=1}^{n} A_{ij} w_{ijk} + \sum_{i,j=1}^{n} A_{ij} \frac{s_{iik} - s_{ijk} - z_{ijk} + x_{ik}}{2} \right)^2}{2m} \right)$$

$$s.t.: (4), (5), (8) - (24),$$

$$h_{ikr} \leq 1 - x_{ik}, \quad \forall i = 1, \ldots, n, \ k, r = 1, \ldots, n_c, \ k < r, \tag{25}$$

$$h_{ikr} \leq x_{ir}, \quad \forall i = 1, \ldots, n, \ k, r = 1, \ldots, n_c, \ k < r, \tag{26}$$

$$x_{ir} - x_{ik} - h_{ikr} \leq 0, \quad \forall i = 1, \ldots, n, \ k, r = 1, \ldots, n_c, \ k < r, \tag{27}$$

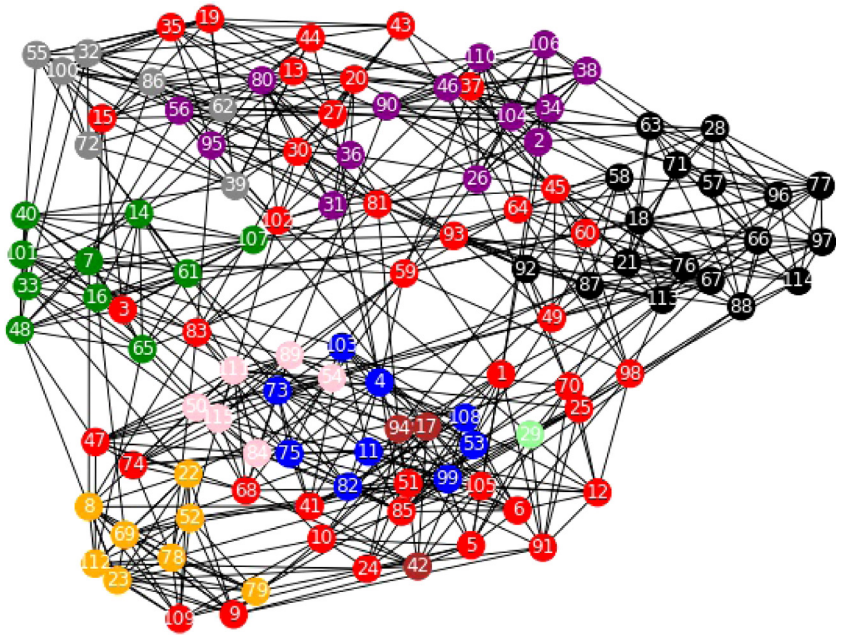


Fig. 8. American college football team communities with $n_c = 10$ and $\lambda = 0.1$.

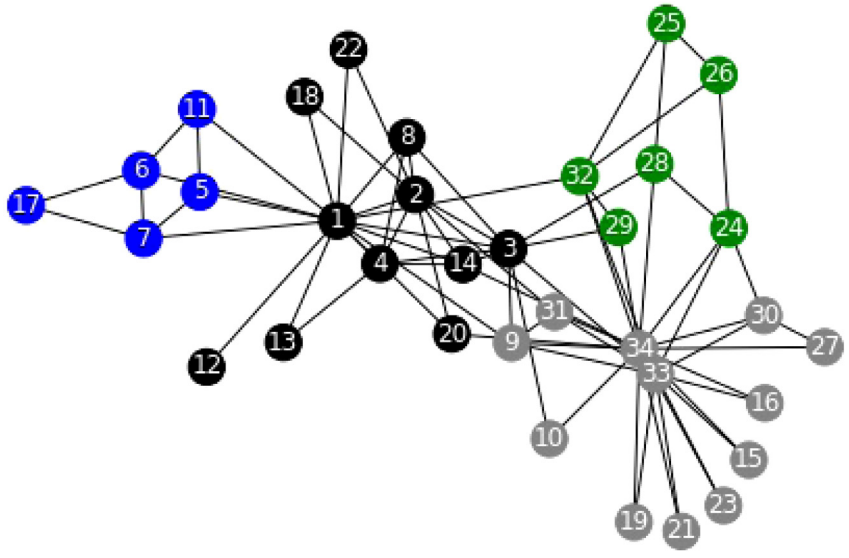


Fig. 9. Zachary's Karate club communities maximizing Newman and Girvan's modularity.

$$\sum_{j=1}^{n} h_{jkr} \geq x_{ir}, \quad \forall i = 1, \ldots, n,\ k, r = 1, \ldots, n_c,\ k < r, \tag{28}$$

$$h_{ikr} \in [0, 1] \quad \forall i = 1, \ldots, n,\ k, r = 1, \ldots, n_c,\ k < r. \tag{29}$$

The family of constraints (25)–(27) represents the identities: $h_{ikr} = x_{ir}(1 - x_{ik})$. Then, constraints (28), jointly with (24), impose that in community r, that contains less units than community k as $k < r$, there is at least one element that is not contained in k.

Even though problem (F-MOD-NI) has been designed to avoid the inconsistency that could be due to implicit assumptions that communities cannot be included, still the outcome appears biased as before. When problem (F-MOD-NI) has been applied to Zachary's karate club, we obtained the results of Fig. 4. That is, all communities are composed by all the nodes except one. The same happens for the American Football networks (data not reported here).

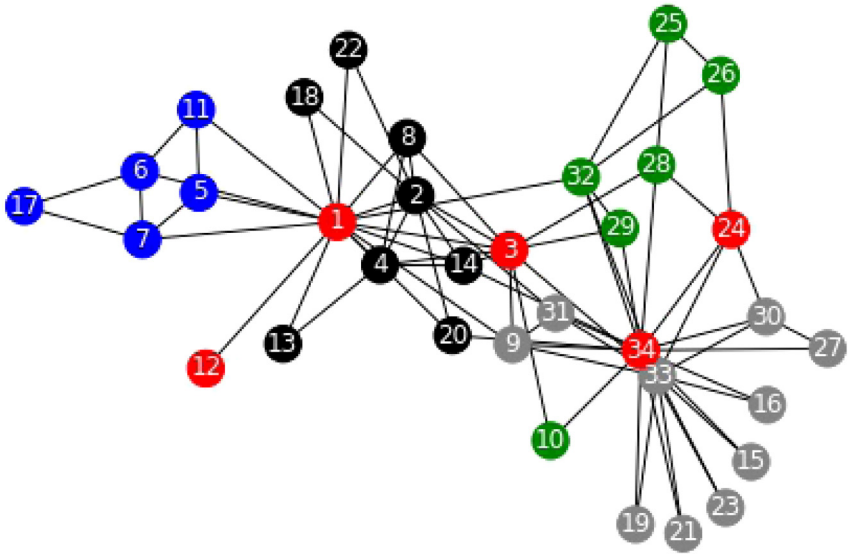


Fig. 10. Zachary's Karate club communities maximizing Problem (NEW-MOD) with $n_c = 4$ and $\lambda = 0.25$.

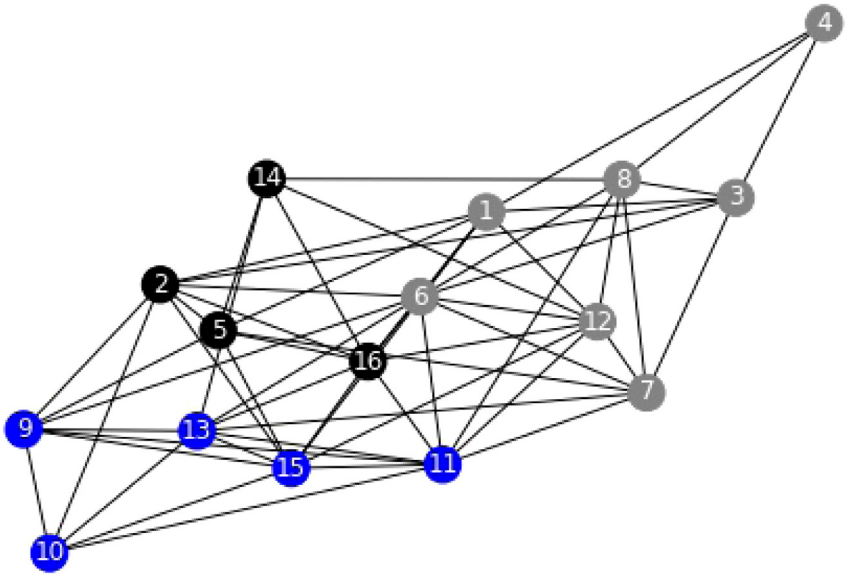


Fig. 11. Highland tribes communities maximizing the Newman and Girvan's modularity.

To summarize the previous tests, there is some flaw in the definition of optimal overlapping communities, as proposed in [8], that regards the formulation of the objective function (3). The previous contribution could not recognize the inconsistency, because optimization was implemented through the use of an heuristic procedure. The heuristic procedure stacked in favour of sub-optimal solutions that were actually so far from optimal ones that they confuse *reasonable* communities with *optimal*. To motivate this argument, note that a set of overlapping communities correspond to fixing hard assignment variables x to 0 or 1. Then, with x being fixed, problems (F-MOD) and (F-MOD-NI) can be solved to calculate the corresponding soft assignment variables u. Finally, the objective function (3) can be calculated and compared using various x. Table 1 reports, the objective function for: (i) overlapping communities calculated by problem (F-MOD), (ii) overlapping communities calculated by problem (F-MOD-NI), (iii) overlapping communities calculated by the heuristic procedure in [8] and reported in Figs. 2 and 3. As can be seen, the objective function of the suboptimal solutions are very far from the optimal ones.

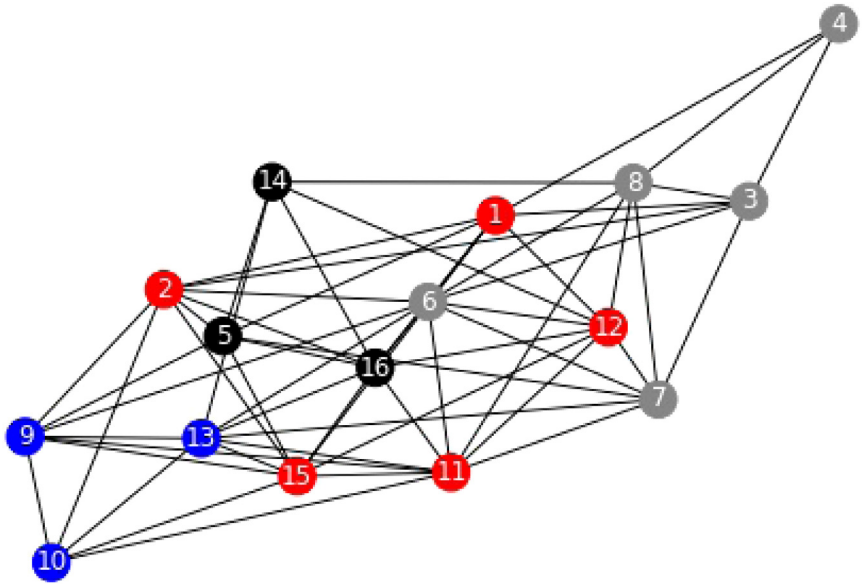
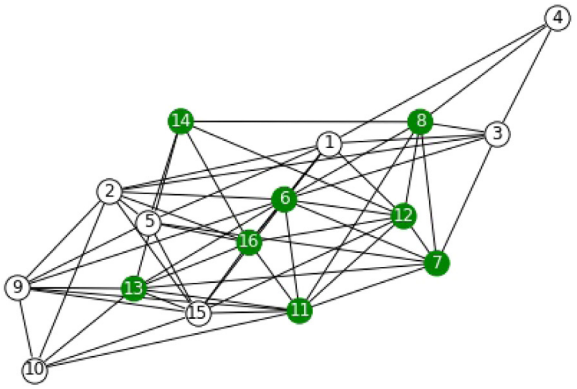
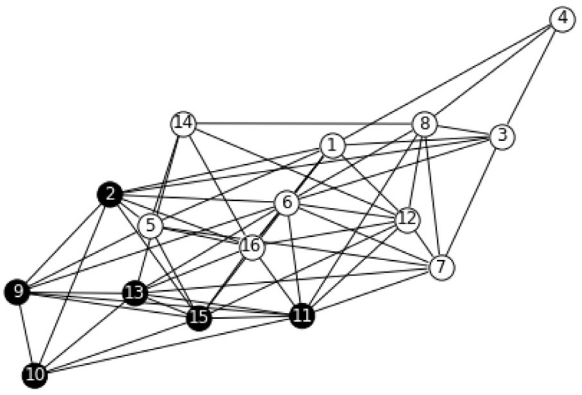


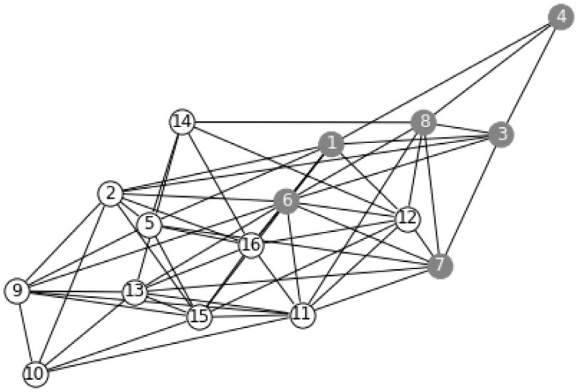
Fig. 12. Highland tribes communities maximizing Problem (NEW-MOD) with $n_c = 3$ and $\lambda = 0.25$.



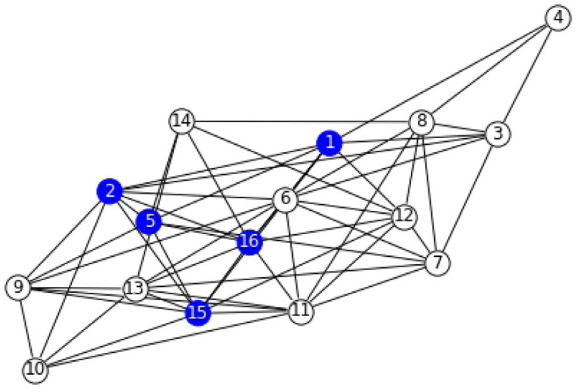
Community 1.



Community 2.

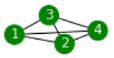


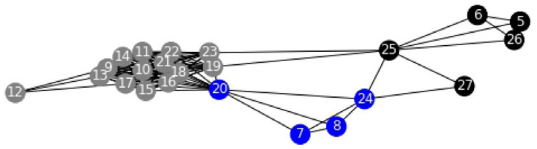
Community 3.



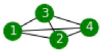
Community 4.

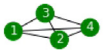
Fig. 13. Optimal Highland tribes communities maximizing (NEW-MOD) with $n_c = 4$ and $\lambda = 0.4$.

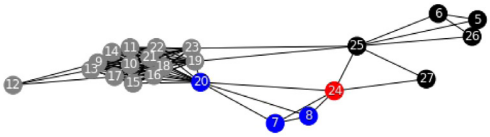




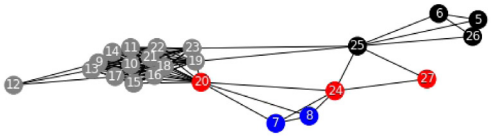
Zebra communities maximizing Newman and Girvan's modularity.







Zebra communities maximizing Problem (NEW-MOD) with $n_c = 4$ and $\lambda = 0.4$.



Zebra communities maximizing Problem (NEW-MOD) with $n_c = 4$ and $\lambda = 0.25$.

Fig. 14. Zebra communication community structures.

4. A new fuzzy modularity function for overlapping community detection

Our previous experiments showed that formula (3) is not appropriate to detect overlapping communities, as it leads to optimal solutions composed of almost all the vertices. The reason to this is that, if node $i \in V_k$ and $j \notin V_k$, weighting the edges between i and j by the average of u_{ik} and $1 - u_{jk}$ introduces a bias in the sum. Indeed, if $j \notin V_k$, then $(1 - u_{jk})$ is large, but this term appears as a subtraction in (3), and therefore this is an incentive to create large communities, just to avoid these negative terms. Therefore, a reasonable transformation of formula (3) could be to exclude those negative terms and to retain only the positive ones, [24,25].

To formulate the new measure, we introduce some hypothesis, as those that inspired the modularity function in [17]. There, similarity between units are compared to the ones that were obtained randomly, e.g., the ones of the so-called configuration graph. In our proposal, instead of generating one random graph, we generate n_c random graphs and we assume edge weights calculated as the average of their membership for each of the n_c communities. Therefore, the new objective function is:

$$\sum_{k=1}^{n_c} \sum_{i,j \in V_k} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \frac{u_{ik} + u_{jk}}{2}, \tag{30}$$

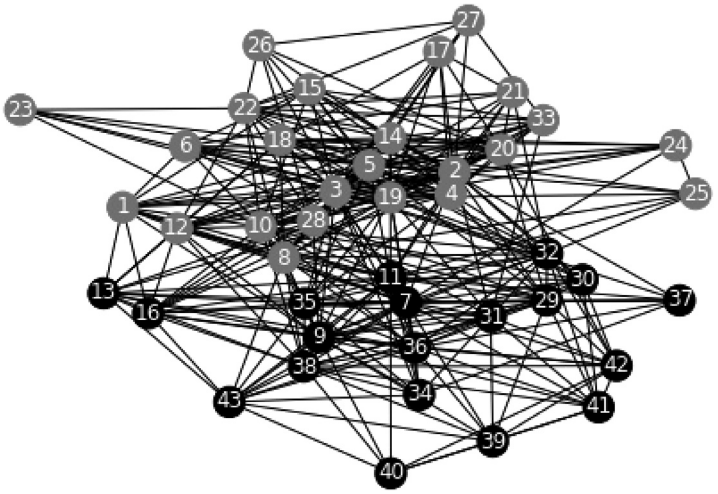


Fig. 15. Windsurfers communities maximizing Newman and Girvan's modularity are the same maximizing Problem (NEW-MOD) with $n_c = 2$ and $\lambda = 0.4, 0.25$.

where the condition $i \in V_k$ corresponds to the hard assignment resulting from $u_{ik} \geq \lambda$. Note that the objective function (30) is linear, while the objective function (3) is quadratic, leading to a mixed-integer optimization problem that should be solved faster than its quadratic counterpart. Moreover, Eq. (30) can be extended to consider weighted graphs too. Entry i, j of the adjacency matrix, A_{ij} is replaced by W_{ij}, k_i, and the weights sum is $W = \sum_{(i,j)\in E} W_{ij}$.

The new model is:

$$\text{(NEW-MOD)} \quad \max \quad \frac{1}{2m} \sum_{k=1}^{n_c} \sum_{i,j=1}^{n} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \frac{w_{ijk} + w_{jik}}{2} \tag{31}$$

$$\text{s.t.:} \quad (4), (5), \text{ from } (8) \text{ to } (13), (20), (22), (23).$$

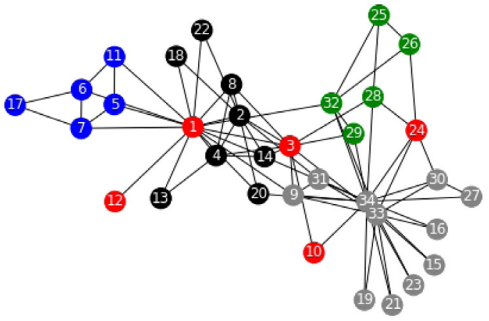
In Problem (NEW-MOD), two parameters appear as input: They are the membership threshold λ and the total number of communities n_c. There is not an optimal choice for these parameters, rather, there is a trade-off between them and the value of the objective function, as it happens similarly for the k-means model. The correct choice depends on the application at hand. Here, we propose to solve maximum modularity with disjoint community to calculate n_c, and then using it as the input of Problem (NEW-MOD) to let these communities overlap, e.g. controlling whether the possibility of multiple memberships increases the objective function. Nevertheless, this choice is not exclusive and other rule-of-thumb can be used to the purpose. Next, parameter λ controls for the maximum number of communities to which a node can belong, for example and as discussed previously, for $\lambda \in [0.34, 0.50)$ a node can belong to two communities at most. Therefore, admissible values are $\lambda \in \{\frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{n_c}\}$. Nevertheless, choosing the right one depends on the application.

After having implemented formulation (NEW-MOD) in Python and solved with the Gurobi's solver, we applied it to the first example in [8] and on the Zachary's karate club network. Our results are reported in Figs. 5 and 6(a), respectively. As can be seen, for the first example, Fig. 5 replicates the communities that have been found in [8]. For what concerns the karate club, we can compare Fig. 2 with 6(a) and observe that the results of the new model are communities similar to the ones that were calculated with a constructive algorithm in [8]. To see the effect of varying λ, the results with $\lambda = 0.1$ are reported in Fig. 6(b). Similar overlapping communities can be seen, but more intersection nodes appear due to the smaller value of λ.

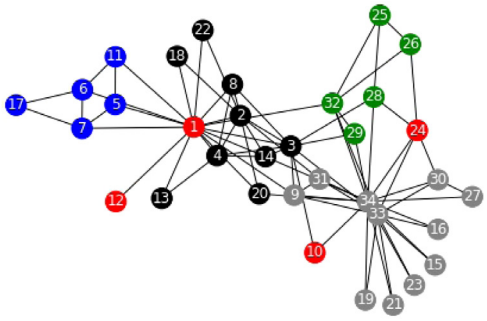
One important advantage of using mathematical programming to formulate and solve the overlapping community detection is that additional features or constraints that one expects from communities can be explicitly modelled as linear inequalities of the problem constraints. For example, as suggested in [1], a researcher may know from other qualitative sources that some nodes, e.g. actors of the networks, are acting as bridges between groups. In this case, these nodes should be included in at least two communities from the beginning, that is, from the problem formulation. This property can be imposed over a node i by the inequality:

$$\sum_{k=1}^{n_c} x_{ik} \geq 2$$

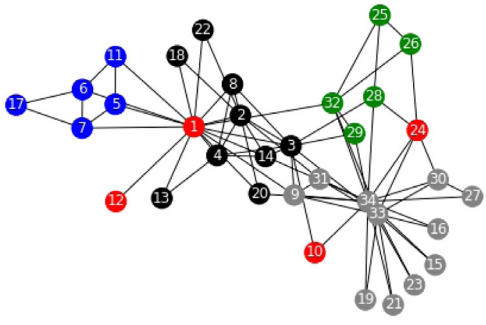
in the constraints of Problem (NEW-MOD). Another possibility is to require that full inclusion among overlapping communities is explicitly forbidden. This can be enforced including in the formulation inequalities (25)–(29).



Communities obtained by the Algorithm 1.i.

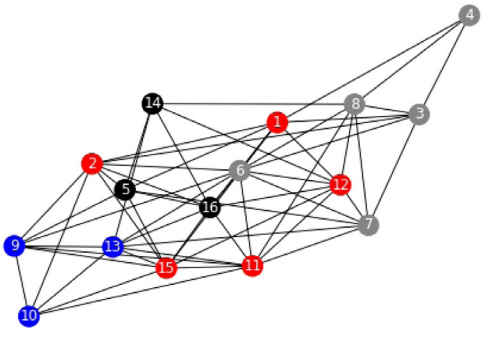


Communities obtained by the Algorithm 1.ii.

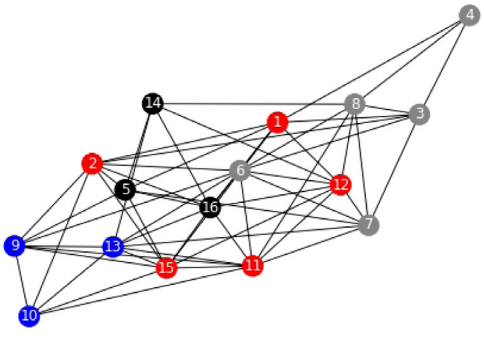


Communities obtained by the Algorithm 1.iii

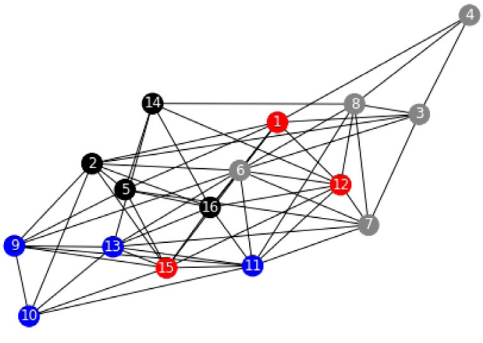
Fig. 16. Zachary's karate club community structure obtained by the Algorithm 1 with $\lambda = 0.25$.



Communities obtained by the Algorithm 1.i.

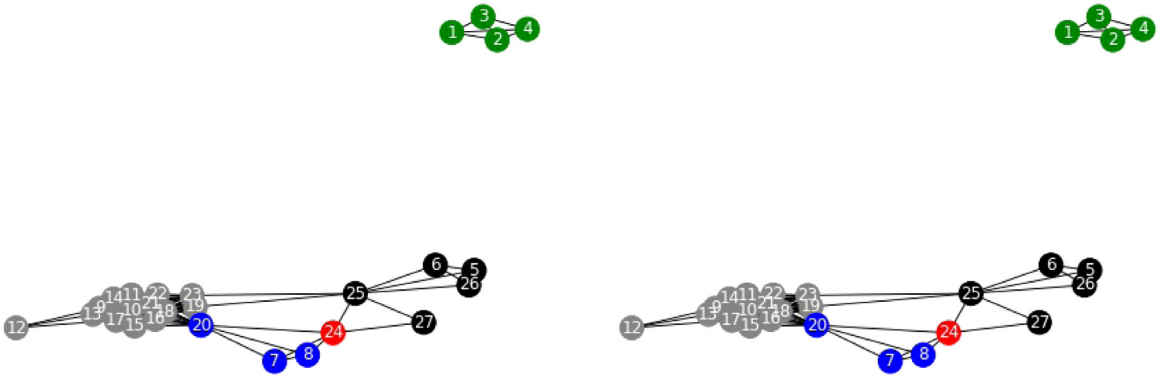


Communities obtained by the Algorithm 1.ii.



Communities obtained by the Algorithm 1.iii

Fig. 17. Highland tribes community structure obtained by the Algorithm 1 with $\lambda = 0.25$.



Communities obtained by the Algorithm 1.i. with $\lambda = 0.4$.

Communities obtained by the Algorithm 1.ii. with $\lambda = 0.4$.

Communities obtained by the Algorithm 1.iii. with $\lambda = 0.4$.

Communities obtained by the Algorithm 1.i. with $\lambda = 0.25$.

Communities obtained by the Algorithm 1.ii. with $\lambda = 0.25$.

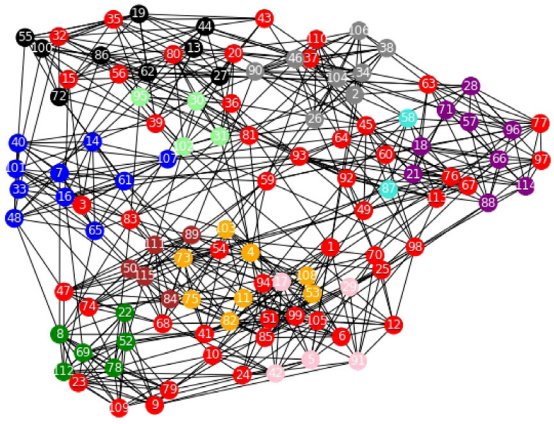
Communities obtained by the Algorithm 1.iii. with $\lambda = 0.25$.

Fig. 18. Zebra communication community structure obtained by the Algorithm 1.

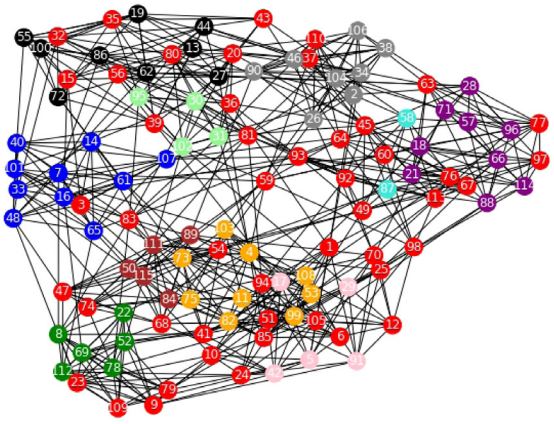
Table 1
Fuzzy (3) comparisons of the most outstanding structures.

Dataset	n_c	λ	F-MOD	F-MOD-NI	[8]
Zachary's karate club	3	0.25	0.667	0.65	0.445
American college football team	10	0.1	0.9	0.887	0.619

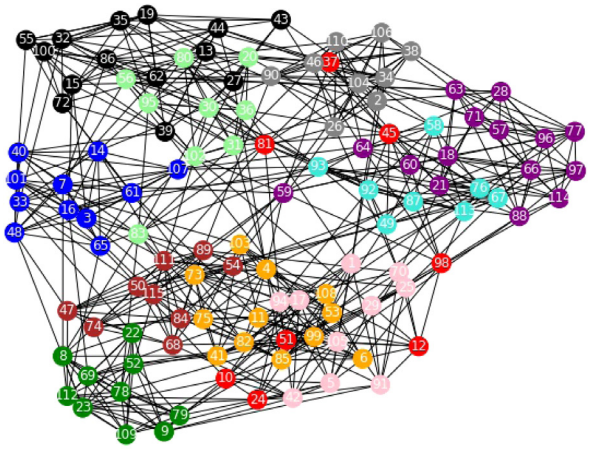
As an example of the previous remark, we calculated the structure of the Zachary's karate club network with $n_c = 3$ and $\lambda = 0.25$ but imposing some nodes to belong to two communities, acting as bridges. Following the solution reported in [8], we let nodes 1, 9, 10 and 31 belong to two communities. The solution obtained is depicted in Fig. 7. The same



Communities obtained by the Algorithm 1.i.



Communities obtained by the Algorithm 1.ii.



Communities obtained by the Algorithm 1.iii.

Fig. 19. American college football team community structure obtained by the Algorithm 1 with $\lambda = 0.1$.

overlapping structure as in Fig. 6(a) are obtained, but now nodes 9,10 and 31 are at the intersection of the communities black and grey.

Applied to the American football instance, after 24 h of computing time, the linear solver could not certify the optimality of its incumbent solution to problem (NEW-MOD). This incumbent solution is depicted in Fig. 8. Even though the solution has not been proved to be the best, still its objective function, (3) e.g $f = 0.632$, is higher than the objective function of the solution calculated in [8]. Moreover, it can be observed in Fig. 8 that there are many intersection nodes, as λ has a low value, communities tend to overlap more.

In the following, we apply formulation (NEW-MOD) using the following procedure: First, we calculate the optimal disjoint communities that maximizes the Newman and Girvan's modularity to determine the parameter n_c, the number of these communities. Then we apply Problem (NEW-MOD) with varying n_c and λ as input parameters.

The first example is again Zachary's karate club. The optimal non-overlapping communities are reported in Fig. 9, the output is $n_c = 4$. When admitting overlapping communities, the results are reported in Fig. 10. As can be seen, overlapping communities identify nodes that are bridges between communities, as their edges are adjacent to different groups, as is the case of nodes 1, 3, 12, 24, 34. This result shows that overlapping communities provide important information about the structural properties of nodes, information that is not available when communities are disjoint. The second example is the alliances network between Highland tribes of New Guinea, Read [26]. The optimal non-overlapping communities are reported in Fig. 11, the output is $n_c = 3$. When admitting overlapping communities, and with $\lambda = 0.25$, the results are reported in Fig. 12, while, for $n_c = 4$ and $\lambda = 0.4$, the results are reported in Fig. 13. The detected communities contain a high internal edge density and the intersection nodes share connections with various communities. These results can be interpreted easily and robust to different parameters combination The third example is the zebra communication network, [27]. The optimal non-overlapping communities are reported in subfigure (a) of Fig. 14, the output is $n_c = 4$.

Table 2
Computational results of the solution methods.

Dataset	n_c	λ	Solving method	Time (s)	Objective value
Zachary's karate club	4	0.25	Exact model	42315	0.442831
			Algorithm 1.i.	4.26	0.441979
			Algorithm 1.ii.	0.66	0.440787
			Algorithm 1.iii.	0.4	0.440787
Zachary's karate club	3	0.25	Exact model	11573	0.41415
			Algorithm 1.i.	131	0.41415
			Algorithm 1.ii.	14.68	0.41415
			Algorithm 1.iii.	5.47	0.41415
Highland tribes	3	0.25	Exact model	23715	0.191439
			Algorithm 1.i.	0.43	0.191439
			Algorithm 1.ii.	0.17	0.191439
			Algorithm 1.iii.	0.08	0.184379
Zebra communication	4	0.4	Exact model	1463	0.282266
			Algorithm 1.i.	0.72	0.282266
			Algorithm 1.ii.	0.22	0.282266
			Algorithm 1.iii.	0.19	0.282266
Zebra communication	4	0.25	Exact model	3682	0.284342
			Algorithm 1.i.	1.33	0.284342
			Algorithm 1.ii.	0.35	0.284342
			Algorithm 1.iii.	0.15	0.282911
American college football team	10	0.1	Exact model	86400	0.619
			Algorithm 1.i.	1902	0.6345487
			Algorithm 1.ii.	136	0.6345
			Algorithm 1.iii.	56.78	0.616872

When admitting overlapping communities, and testing for $\lambda = 0.25$, and 0.4, it can be seen that as λ is smaller, there are more nodes in the intersections between communities. The fourth example is the windsurfers network, [28]. The optimal non- overlapping communities result in $n_c = 2$. When admitting overlapping communities, still disjoint communities are detected, see Fig. 15. This is because there are two distinguished communities and the model cannot find nodes behaving as bridges between groups.

From the four applications we can conclude that:

- Overlapping communities provide additional information about the structure of the connection between groups, for example, identifying nodes interpreted as bridges.
- Varying parameter λ is an effective tool to let communities of different shape emerge, identifying what nodes are also the most influential, as they belong to more than two communities.
- The overlapping model is flexible enough to guarantee a non-overlapping community as the outcome, when data suggest so.

5. Heuristic algorithms to approximate overlapping communities

If λ is fixed to 1, then Problem (NEW-MOD) is the maximum modularity problem, a problem that is known NP-hard, therefore NP-hard itself: They are problem for which a polynomial time algorithm does not exist, unless P = NP. In practice, this negative result implies that optimal solutions can be obtained only for instances of moderate size. As can be seen in the application to the American college football teams this size is of the order of a few tens. Nevertheless, the MILP formulation can be used to obtain good approximations of optimal solutions through the use of heuristic. The first algorithm that we propose is based on local search. It consists of an iterative method applied to feasible solutions that are improved by modifying some value, e.g. membership functions or hard assignments, until no improvement is possible. In that case, we say that a local optimum has been reached. More formally, let n_c be the maximum number of communities and λ be the threshold of the membership value, let $\Pi = \{V_1, \ldots, V_{n_c^*}\}$ be a set of n_c^* communities of V, $n_c^* \leq n_c$. We say that Π is a feasible solution if: (1) every node belongs to at least one community, that is, $\bigcup_{k=1}^{n_c^*} V_k = V$, (2) no community is a subset of a larger one, that is, $\nexists k, r = 1, \ldots, n_c, k \neq r$, such that $V_k \subseteq V_r$, and (3) there is a feasible membership solution u for the partition Π, i.e. $\forall i \in V$ the inequality $\frac{1}{|\{k=1,\ldots,n_c^*: i \in V_k\}|} \geq \lambda$ is fulfilled.

Assume that Π is a feasible solution. To improve the objective function we consider three types of changes of hard assignments: (1) to add a node to a community or (2) to remove a node from a community, and (3) to swap two nodes

Table 3
Computational results comparing Algorithm 2 with Clique percolation and Fuzzy c-means.

n	λ	Number of communities	Solving method	Time (s)	Objective value
333	0.5	13	Large scale algorithm	0.84	0.4381
		15	Fuzzy c-means	63.35	0.36
		172	Clique percolation	26.67	0.3135
333	0.2	13	Large scale algorithm	0.84	0.44
		13	Fuzzy c-means	64.32	0.3614
		109	Clique percolation	27.98	0.3871
747	0.5	6	Large scale algorithm	7	0.526
		9	Fuzzy c-means	262.59	0.506
		–	Clique percolation	–	–
747	0.2	6	Large scale algorithm	7.28	0.5286
		9	Fuzzy c-means	270.174	0.512
		–	Clique percolation	–	–
224	0.5	5	Large scale algorithm	0.51	0.2691
		12	Fuzzy c-means	37.33	0.254
		100	Clique percolation	1102.26	0.1679
224	0.2	5	Large scale algorithm	0.91	0.2933
		12	Fuzzy c-means	38	0.26169
		90	Clique percolation	1268.74	0.1893
534	0.5	10	Large scale algorithm	3.9	0.62656
		10	Fuzzy c-means	140.74	0.63
		246	Clique percolation	70.68	0.488
534	0.2	10	Large scale algorithm	4.5	0.6445
		10	Fuzzy c-means	149.83	0.635
		93	Clique percolation	77.8	0.6224
1034	0.5	6	Large scale algorithm	39.36	0.5357
		7	Fuzzy c-means	499.18	0.52428
		–	Clique percolation	–	–
1034	0.2	6	Large scale algorithm	46.9	0.5401
		7	Fuzzy c-means	519.15	0.52689
		–	Clique percolation	–	–

from two different communities. These moves can be applied if and only if the new obtained solution is feasible. For instance, the first and the second movement cannot be applied if it results in some inclusion between communities. More formally, for $i \in V$ and $k = 1, \ldots, n_c$, let the triplet $(i, k, 1)$ be the move of adding node i to community k and let the triplet $(i, k, 2)$ be the move of removing node i from community k, let the 5-tuple $(i, k, i', k', 3)$ be the move of swapping nodes i and i' between communities k and k' respectively.

Assume that Π is a feasible solution, represented by hard assignments x and membership functions u, then, to calculate the objective function (30), membership functions u must be determined too: u can be calculated in the following ways:

(i) Exact calculation of u: use formulation (NEW-MOD) to calculate the optimal objective function (30) and its corresponding u.
(ii) Approximate calculation of u: keep fixed the membership functions u corresponding to unchanged assignments x, and find an approximate value u_{ik} only for the assignments x_{ik} that were modified using formulation (NEW-MOD). This option is less accurate but also reduces complexity.
(iii) Approximate calculation of u: Approximate u as follows. If $x_{ik} = 0$ then $u_{ik} = 0$, while $u_{ik} = p$, with p a constant term, for the value for which $x_{ik} = 1$, so $p = \frac{1}{\sum_{j=1}^{n_c^s} x_{ij}}$. This is the least accurate approximation, but also the simplest and fastest, as it does not require any optimization.

Finally, the interchange heuristic is applied to the initial solution x calculated maximizing modularity (1) with non-overlapping solution. Further diversification can be obtained by choosing the initial solution x randomly, e.g. using the so-called random restart.

The pseudo-code of the algorithm is summarized in Algorithm 1:

Algorithm 1 Heuristic algorithm

procedure LOCAL SEARCH FOR OVERLAPPING COMMUNITIES
  $\Pi = \{V_1, \ldots, V_{n_c}\} \leftarrow$ *Initial_Subdivision* ▷ Π feasible subdivision obtained randomly or by another procedure
  $f \leftarrow$ *Extended_Modularity*(Π) ▷ The extended modularity (30) is computed by three different ways: i), ii) or iii).
  local_opt = *FALSE* ▷ Condition for a local optimum
  while *local_opt* = *FALSE* **do**
    $\Delta \leftarrow$ *Feasible_Moves*(Π) ▷ Δ: list of admissible movements for Π.
    for (i, k, d) in Δ **do**
      if d=1 **then**
        $V_k \leftarrow V_k \cup \{i\}$
        $\delta_{ikd} \leftarrow$ *Extended_Modularity*(Π)
        $V_k \leftarrow V_k \setminus \{i\}$
      end if
      if d=2 **then**
        $V_k \leftarrow V_k \setminus \{i\}$
        $\delta_{ikd} \leftarrow$ *Extended_Modularity*(Π)
        $V_k \leftarrow V_k \cup \{i\}$
      end if
    end for
    for $(i, k, i', k', 3) \in \Delta$ **do**
      if d=3 **then**
        $V_k \leftarrow V_k \cup \{i'\} \setminus \{i\}$
        $V_{k'} \leftarrow V_{k'} \cup \{i\} \setminus \{i'\}$
        $\delta_{iki'k'd} \leftarrow$ *Extended_Modularity*(Π)
        $V_k \leftarrow V_k \setminus \{i'\} \cup \{i\}$
        $V_{k'} \leftarrow V_{k'} \setminus \{i\} \cup \{i'\}$
      end if
    end for
    $(i^*, k^*, d^*) \in \arg\max\{\delta_{ikd} | (i, k, d) \in \Delta\}$ ▷ Select the move that increases the most
    $(i^*, k^*, i'^*, k'^*, d^*) \in \arg\max\{\delta_{iki'k'd} | (i, k, i', k', d) \in \Delta\}$ ▷ Select the move that increases the most
    if $\delta_{i^*k^*i'^*k'^*d^*} > \max\{f, \delta_{i^*k^*d^*}\}$ **then**
      $f \leftarrow \delta_{i^*k^*i'^*k'^*d^*}$ ▷ Update f
      $V_{k^*} \leftarrow V_{k^*} \cup \{i'^*\} \setminus \{i^*\}$
      $V_{k'^*} \leftarrow V_{k'^*} \cup \{i^*\} \setminus \{i'^*\}$ ▷ Update Π
    else
      if $\delta_{i^*k^*d^*} > f$ **then**
        $f \leftarrow \delta_{i^*k^*d^*}$ ▷ Update f
        if $d^* = 1$ **then**
          $V_{k^*} \leftarrow V_{k^*} \cup \{i^*\}$ ▷ Update Π
        else
          $V_{k^*} \leftarrow V_{k^*} \setminus \{i^*\}$ ▷ Update Π
        end if
      else
        local_opt = *TRUE*
      end if
    end if
  end while
  return Π ▷ Return the local optimum
end procedure

If the initial solution Π is obtained randomly, then the interchange can be repeated for a maximum of t_{max} initial solution. For each attempt t, we obtain local optimal objective function f_t and overlapping communities Π_t. Finally, approximate solution of problem (NEW-MOD) is the best local optimum.

In test problems, we run the algorithm with the three different methods to calculate membership function u. To assess their quality, we applied Algorithm 1 to the initial solution Π calculated by the maximum modularity (1), as in this way we prevent potential biases caused by random starting solutions. The communities found in this way are reported in Figs. 16, 17, 18 and 19. As can be seen, when the membership functions u are calculated exactly, e.g. by solving the optimization problem, then the heuristic communities are the optimal. While, when u are approximated, the heuristic communities only differ for very few nodes to the optimal ones.

Next, in Table 2, we compared Algorithm 1 with the three variants for computing u with the optimal solution of problem (NEW-MOD). It can be seen that the heuristic algorithms reduce the computational time at the cost of decreasing the objective function only to a small amount. When the optimal solution is not available, such as the case of the American football data, two of the heuristic algorithms could obtain a better objective value than the MILP truncation after 24 h of computing.

For the Zachary's karate club case with $n_c = 3$, since the optimal number of communities in the non-overlapping case is $n_c = 4$ and $\lambda = 0.25$, we do not use this as an initial solution. Instead, we perform a multistart strategy with ten iterations starting with 3 randomly chosen communities.

Algorithm 1 works well when input data are small or medium sized networks, e.g. networks with hundreds of nodes, but computational times might be too high for large sized networks, for example instances with more than 1000 nodes. For this reason, we developed a variation of the previous Algorithm, reported in Algorithm 2, in which some operations

are accelerated at the cost of further approximation of optimal decisions. Nevertheless, the method is appropriate for large size networks.

Algorithm 2 Large scale heuristic algorithm

procedure LOCAL SEARCH FOR OVERLAPPING COMMUNITIES
  $\Pi = \{V_1, \ldots, V_{n_c}\} \leftarrow$ *Initial_Subdivision* ▷ Π feasible subdivision obtained randomly or by another procedure (heuristic)
  $f \leftarrow$ *Extended_Modularity*(Π) ▷ The extended modularity (30) is computed by third way iii).
  local_opt = *FALSE* ▷ Condition for a local optimum
  while *local_opt* = *FALSE* **do**
    $\Delta \leftarrow$ *Feasible_Moves*(Π) ▷ Δ: list of admissible movements for Π.
    for (i, k, d) in Δ **do**
      if d=1 **then**
        $V_k \leftarrow V_k \cup \{i\}$
        $\delta_{ikd} \leftarrow$ *Extended_Modularity*(Π) $- f$
        $V_k \leftarrow V_k \setminus \{i\}$
      end if
      if d=2 **then**
        $V_k \leftarrow V_k \setminus \{i\}$
        $\delta_{ikd} \leftarrow$ *Extended_Modularity*(Π) $- f$
        $V_k \leftarrow V_k \cup \{i\}$
      end if
    end for
    for $(i, k, i', k', 3) \in \Delta$ **do**
      if d=3 **then**
        $V_k \leftarrow V_k \cup \{i'\} \setminus \{i\}$
        $V_{k'} \leftarrow V_{k'} \cup \{i\} \setminus \{i'\}$
        $\delta_{iki'k'd} \leftarrow$ *Extended_Modularity*(Π) $- f$
        $V_k \leftarrow V_k \setminus \{i'\} \cup \{i\}$
        $V_{k'} \leftarrow V_{k'} \setminus \{i\} \cup \{i'\}$
      end if
    end for
    $V \leftarrow \{1, \ldots, n\}$, $K \leftarrow \{1, \ldots, n_c\}$, *f_improve* $\leftarrow$ *TRUE*
    while $V \neq \emptyset, K \neq \emptyset,$ *f_improve* = *TRUE* **do**
      $(i^*, k^*, d^*) \in \arg\max\{\delta_{ikd} | (i, k, d) \in \Delta,\ i \in V,\ k \in K\}$ ▷ Select the move that increases the most
      $(i^*, k^*, i'^*, k'^*, d^*) \in \arg\max\{\delta_{iki'k'd} | (i, k, i', k', d) \in \Delta,\ i, i' \in V,\ k, k' \in K\}$ ▷ Select the move that increases the most
      if $\delta_{i^*k^*i'^*k'^*d^*} > \max\{0, \delta_{i^*k^*d^*}\}$ **then**
        $V \leftarrow V \setminus \{i^*, i'^*\}$
        $K \leftarrow K \setminus \{k^*, k'^*\}$
        $V_{k^*} \leftarrow V_{k^*} \cup \{i'^*\} \setminus \{i^*\}$
        $V_{k'^*} \leftarrow V_{k'^*} \cup \{i^*\} \setminus \{i'^*\}$ ▷ Update Π
        $f \leftarrow$ *Extended_Modularity*(Π) ▷ Update f
      else
        if $\delta_{i^*k^*d^*} > 0$ **then**
          $V \leftarrow V \setminus \{i^*\}$
          $K \leftarrow K \setminus \{k^*\}$
          if $d^* = 1$ **then**
            $V_{k^*} \leftarrow V_{k^*} \cup \{i^*\}$ ▷ Update Π
            $f \leftarrow$ *Extended_Modularity*(Π) ▷ Update f
          else
            $V_{k^*} \leftarrow V_{k^*} \setminus \{i^*\}$ ▷ Update Π
            $f \leftarrow$ *Extended_Modularity*(Π) ▷ Update f
          end if
        else
          f_improve = *FALSE*
          if $V = \{1, \ldots, n\}$ **then**
            local_opt = *TRUE*
          end if
        end if
      end if
    end while
  end while
  return Π ▷ Return the local optimum
end procedure

Algorithm 2 is based on the previous Algorithm 1, but modifying some of its steps. First of all, membership functions u are calculated by option (iii), e.g. the fastest approximation scheme among those proposed for Algorithm 1. Next, initial solution is calculated through an approximation of optimal modularity communities, that is, using agglomerative hierarchical clustering proposes in [29]. Finally, add, remove and interchange steps to improve the incumbent solution are applied only if they use nodes or communities that are not repeated, as it was already done in the heuristic presented in [30]. In this way, computational times can be reduced to a large extent.

We compare our method to two other algorithms: the clique percolation proposed in [31] and the fuzzy c-means proposed in [8]. The first method defines as communities the connected k-cliques, that are the cliques composed of k nodes. The algorithm depends on the parameter k, so we have calculated the fuzzy modularity function (30) for all the values from $k = 1$ to k the size of the greatest clique. The second method, the fuzzy c-means, is based on the eigenvectors of the normalized adjacency matrix and depends on two parameters: the maximum number of communities and the parameter m that appears on the fuzzy c-means expression, because membership functions are to the power of m in the

expression of the fuzzy-c means. Then, we run the algorithm with varying n_c from 2 to a fixed maximum number of communities.

All procedures are implemented in Python and applied to the Facebook friendship relation networks, [32]. In some cases, clique percolation could not solve the instance for the computational complexity of calculating k-cliques. Algorithm fuzzy c-means has been run with a maximum number of communities equal to $n_c = 15$ and different values of m in $\{1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2\}$ and we report the best results of these alternatives. Computational results are reported in Table 3. As can be seen, Algorithm 2 is the fastest method and in most cases its objective function is the highest, showing evidence that it is better than the two other methods.

6. Conclusion

In this paper, we have proposed a new model to detect overlapping communities in a network. Our model is based on the optimization of a fuzzy modularity function, following the suggestion contained in [8], However, we elaborated that model further, as we discovered that for that original contribution optimal communities are the whole set of nodes, or the whole set except a few nodes. We proposed a novel fuzzy modularity function and we proved by computational experiments that overlapping communities calculated in that way could reveal the network structure in a meaningful way, for example detecting the nodes with the structural property of being bridges.

As in [8], our method is mathematical programming with integer variables and the optimization problem is hard to solve: Computational times for large networks are too high for both methods being of practical use. Nevertheless, the new objective function is the core of a new heuristic method, based on the operations of add, drop, and interchange, that can be applied to large data sets. In this case, computational times are reasonable, only at a cost of a small approximation of the optimal solution. Nevertheless, further improvements can be obtained for the algorithm for large size instances, for example, testing other heuristic techniques such as genetic algorithms, tabu search or variable neighbourhood search. Heuristic could take advantage of the MILP formulation of the model, as the methods proposed in [30,33,34]. Finally, improvements can be obtained by designing hybrid techniques, and combining maximum fuzzy modularity with clique percolation and clustering.

CRediT authorship contribution statement

Stefano Benati: Supervision, Conceptualization, Methodology, Format analysis, Writing – original draft, Writing – review & editing, Visualization. **Justo Puerto:** Supervision, Conceptualization, Methodology, Format analysis, Writing – original draft, Writing – review & editing, Visualization. **Antonio M. Rodríguez-Chía:** Supervision, Conceptualization, Methodology, Format analysis, Writing – original draft, Writing – review & editing, Visualization. **Francisco Temprano:** Conceptualization, Methodology, Format analysis, Writing – original draft, Writing – review & editing, Visualization.

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.

Acknowledgements

This research has been partially supported by Spanish Ministry of Education and Science/FEDER grant number PID2020-114594GB02, and projects Junta de Andalucia, Spain P18-FR-1422, FEDER-US-1256951, CEI-3-FQM331 and *NetmeetData*: Ayudas Fundación BBVA a equipos de investigación científica 2019.

References

[1] L. Bennett, A. Kittas, S. Liu, L.G. Papageorgiou, S. Tsoka, Community structure detection for overlapping modules through mathematical programming in protein interaction networks, PLoS ONE 9 (11) (2014) e112821.
[2] J.A. Dunne, R.J. Williams, N.D. Martinez, Food-web structure and network theory: The role of connectance and size, Proc. Natl. Acad. Sci. 99 (20) (2002) 12917–12922.
[3] F. Maestre, D. Eldridge, S. Soliveres, S. Kéfi, M. Delgado-Baquerizo, M. Bowker, P. García-Palacios, J. Gaitán, A. Gallardo, R. Lázaro, M. Berdugo, Structure and functioning of dryland ecosystems in a changing world, Annu. Rev. Ecol. Evol. Syst. 47 (2016) 215–237.
[4] D. Garlaschelli, M.I. Loffredo, Structure and evolution of the world trade network, Physica A 355 (1) (2005) 138–144.
[5] M. Catino, S. Rocchi, G. Vittucci Marzetti, The network of interfamily marriages in 'ndrangheta, Social Networks 68 (2022) 318–329.
[6] P. Bearman, J. Moody, K. Stovel, Chains of affection: The structure of adolescent romantic and sexual networks, Am. J. Sociol. 110 (2004) 44–99.
[7] R.S. Burt, Structural holes and good ideas, Am. J. Sociol. 110 (2) (2004) 349–399.
[8] S. Zhang, R.-S. Wang, X. Zhang, Identification of overlapping community structure in complex networks using fuzzy c-means clustering, Physica A (374) (2007) 483–490.
[9] J. Xie, S. Kelley, B.K. Szymanski, Overlapping community detection in networks: The state-of-the-art and comparative study, ACM Comput. Survey 45 (43) (2013) 1–35.
[10] M.R. Fellows, J. Guo, C. Komusiewicz, R. Niedermeier, J. Uhlmann, Graph-based data clustering with overlaps, Discrete Optim. 9 (2011) 2–17.

[11] A. Lancichinetti, S. Fortunato, J. Kertész, Detecting the overlapping and hierarchical community structure in complex networks, New J. Phys. 11 (3) (2008).
[12] H.J. Li, Z. Bu, Z. Wang, J. Cao, Dynamical clustering in electronic commerce systems via optimization and leadership expansion, IEEE Trans. Ind. Inf. 16 (8) (2020a) 5327–5334.
[13] H.J. Li, Z. Wang, P. Jian, J. Cao, Y. Shi, Optimal estimation of low-rank factors via feature level data fusion of multiplex signal systems, IEEE Trans. Knowl. Data Eng. (2020c).
[14] C. Xia, Y. Luo, L. Wang, H.J. Li, A fast community detection algorithm based on reconstructing signed networks, ACM Trans. Knowl. Discov. Data 16 (2) (2021).
[15] H.J. Li, L. Wang, Y. Zhang, M. Perc, Optimization of identifiability for efficient community detection, New J. Phys. 22 (063035) (2020b).
[16] H.J. Li, W. Xu, S. Song, W.X. Wang, M. Perc, The dynamics of epidemic spreading on signed networks, Chaos Solitons Fractals 151 (111294) (2021).
[17] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026113.
[18] V. Nicosia, G. Mangioni, V. Carchiolo, M. Malgeri, Extending the definition of modularity to directed graphs with overlapping communities, J. Stat. Mech. Theory Exp. 69 (03) (2009) P03024.
[19] A. Jonnalagadda, L. Kuppusamy, A cooperative game framework for detecting overlapping communities in social networks, Physica A (491) (2018) 498–515.
[20] M.E.J. Newman, Analysis of weighted networks, Phys. Rev. E 70 (5) (2004) 056131.
[21] D. Pisinger, The quadratic knapsack problem—a survey, Discrete Appl. Math. 155 (5) (2007) 623–648, http://dx.doi.org/10.1016/j.dam.2006.08.007, URL https://www.sciencedirect.com/science/article/pii/S0166218X06003878.
[22] W.W. Zachary, An information flow model for conflict and fission in small groups, J. Anthropol. Res. 33 (4) (1977) 452–473.
[23] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. 99 (12) (2002).
[24] D. Chen, M. Shang, Y. Fu, Detecting overlapping communities of weighted networks via a local algorithm, Physica A (389) (2010) 4177–4187.
[25] J. Chitra Devi, E. Poovammal, An analysis of overlapping community detection algorithms in social networks, Procedia Comput. Sci. (89) (2016) 349–358.
[26] K.E. Read, Cultures of the central highlands, new guinea, Southwest. J. Anthropol. (1954) 1–43.
[27] S.R. Sundaresan, I.R. Fischhoff, J. Dushoff, D.I. Rubenstein, Network metrics reveal diverences in social organization between two wssion-fusion species, grevy's zebra and onager, Oecologia 151 (2007) 140–149.
[28] L.C. Freeman, S.C. Freeman, A.G. Michaelson, On human social intelligence, J. Soc. Biol. Struct. 11 (1988) 415–425.
[29] S. Fortunato, Community detection in graphs, Phys. Rep. (486) (2010) 75–174.
[30] S. Benati, J. Puerto, A.M. Rodríguez-Chía, Clustering data that are graph connected, European J. Oper. Res. 261 (1) (2017) 43–53.
[31] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (7043) (2005).
[32] S. Catanese, P. De Meo, E. Ferrara, G. Fiumara, A. Provetti, Extraction and analysis of facebook friendship relations, Comput. Soc. Netw. Min. Vis. 3 (2012) 291–324.
[33] E. Angelelli, R. Mansini, M.G. Speranza, Kernel search: a new heuristic framework for portfolio selection, Comput. Optim. Appl. 51 (2012) 345–361.
[34] Z. Li, X.S. Zhang, R.S. Wang, H. Liu, S. Zhang, Discovering link communities in complex networks by an integer programming model and a genetic algorithm, PLoS ONE 8 (12) (2013) e83739, http://dx.doi.org/10.1371/journal.pone.0083739.