

Spotting Key Members in Networks: Clustering-Embedded Eigenvector Centrality

Emilio Carrizosa, Alfredo Marín, and Mercedes Pelegrín 

Abstract—Identifying key members in a social network is critical to understand the underlying system behavior. Whereas there are several measures designed to discern the most central member, they fail to identify a central set of members and at the same time reveal the spheres of influence of the individuals in such central set. Here, we combine eigenvector centrality with clustering to design a mathematical programming formulation capable of detecting key members while preventing their spheres of influence from overlapping. Our computational experience reproduces these two features as different aspects of the same phenomenon. The optimal set of key members and their spheres of influence are identified in real-life networks and synthetic ones. For the former, community structures are consistent with existing knowledge about the instances. For the latter, network underlying organization is known *a priori* and it is perfectly uncovered. Experiments further reveal previously neglected nodes to be optimal key members. The size of the instances tested reach several hundreds of nodes and thousands of links.

Index Terms—Clustering, eigenvector centrality, mathematical optimization, social networks.

I. INTRODUCTION

NETWORKS are the basic architectures to represent systems components and their interrelations in diverse domains, including biological, social, or communication systems. Given their versatility, it is not surprising that research communities from wide ranging disciplines such as computer sciences, biology, psychology, or sociology, have focused their efforts on improving their insight on networks dynamics [1]–[5]. One crucial step to understand and modify the processes that networks encode is to uncover their relevant nodes. Identification of such nodes has profound consequences for cyber-physical systems concerning, e.g., terrorist organizations [4] or viral marketing [5]. But, what do we understand as *relevant*? If the goal is

to prevent a terrorist attack, one would target those individuals whose eradication disrupts the network [6]. However, should a marketer aim to persuade people about their deepest desires, she would need to identify those personalities with ultimate influence and communication power [7].

The concept of relevance is always linked to the idea that there are a few key nodes that play a distinguished role within the network. These nodes are usually referred as *influencers* [8], *leaders* [9], *central* [10], *vital* [11], *spreaders* [12], or *prominent* [13]. The interest towards key nodes in networks emerged in the past century as a subject of mathematical sociology [9], [10], [14] and graph theory [15]–[17], and has grown to the point of becoming one of the most noteworthy challenges towards understanding social systems [5], [18], [19].

When social networks analysis was still in its infancy, different strategies were explored in order to determine the relevance of a single node, which gave rise to the so-called centrality measures. Some of the classics are based on local criteria such as the number of connections with other nodes of the network (*degree centrality*), the number of shortest paths that contain the node (*betweenness centrality* [10], [20]) or the distance between the node in question and the rest of the nodes in the network (*closeness centrality* [17], [21]). A different approach assumes that one node's importance not only depends on its connections to the rest of the network, but also on the importance of its neighbors. Translating such recursive definition into a mathematical formula yields the search of the eigenvectors of a matrix, the *matrix of relationships* in sociology and the *adjacency matrix* in graph theory. The result is a decentralized measure that has been known as *eigenvector centrality* [14], [22] and inspired popular Google's method for rating web pages, PageRank [23]. Among later approaches to measure the relevance of individual nodes, we can find *coreness* [19] or *h-index* [24].

State-of-the-art challenges in network analysis include discerning the relevance of a group of nodes [12], [25], [26]. In order to accomplish the new task, a first thought could be to leverage previous knowledge of the problem on a single node. Indeed, computing some of the abovementioned measures for every node gives a ranking from where selecting those with highest individual relevance, but says nothing about their potential influence power as a group, though. Hence, previous measures to quantify relevance of single nodes have to be rethought. One of the aspects that become crucial when determining joint relevance of multiple members is their relative distance [12], [26]. Think about the marketer who wants to sell a product. Targeting the two top influencers among its potential clients seems a sensible

Manuscript received May 13, 2019; revised September 25, 2019, January 28, 2020, and March 8, 2020; accepted March 9, 2020. Date of publication April 16, 2020; date of current version September 2, 2020. This work was supported in part by the Spanish Ministerio de Economía y Competitividad under Grant MTM2015-65915-R and in part by the Ministerio de Educación, Cultura y Deporte under Grant FPU15/05883. (Corresponding author: Mercedes Pelegrín.)

Emilio Carrizosa is with the Department of Statistics and Operational Research, Instituto de Matemáticas, Universidad de Sevilla, 41003 Sevilla, Spain (e-mail: ecarrizosa@us.es).

Alfredo Marín is with the Department of Statistics and Operational Research, Universidad de Murcia, 30001 Murcia, Spain (e-mail: amarín@um.es).

Mercedes Pelegrín was with the Department of Statistics and Operational Research, Universidad de Murcia, 30001 Murcia, Spain. He is now with the Laboratoire d'Informatique de l'X (LIX), École Polytechnique, 91128 Palaiseau, France (e-mail: pelegringarcia@lix.polytechnique.fr).

Digital Object Identifier 10.1109/JSYST.2020.2982266

strategy. However, if these celebrities happen to have lots of followers in common, it seems more reasonable to replace them by a pair whose influence covers a bigger portion of clients, even if they are less popular.

First attempts to approach joint relevance rely upon adaptations of classical degree, closeness, and betweenness centralities [1], [27] and also on the search of the minimal set of nodes whose simultaneous removal will fragment the network [1], [6]. More sophisticated approaches appeared with time, based on techniques borrowed from physics and statistics, including optimal percolation [25], diffusion models [12], [26], and proportional hazard models [8]. What distinguishes these latter approaches is the fact that they substantiate on influence propagation over time rather than taking an individualistic view of influence. One of these studies suggests that influential people with influential friends may be instrumental in the spread of information [8]. However, possible adaptations of eigenvector centrality, which relies on this idea, remain unexamined.

We propose to embed eigenvector computation into a k -centroids clustering framework. Real networks usually display a modular structure that emanates from the combination of compartments or functional subunits, identified by social scientists as *communities* [28]. Some existing approaches already address network modularity, identifying key nodes in a first step and using them as centers for clustering afterwards [29], [30]. Our approach is conceptually different because community discovery and key nodes identification go hand in hand. A representative or *centroid* is chosen from each compartment or *cluster* to be in the group of key members. The introduction of a mixing parameter in the model guarantees cluster cohesion and allows us to reinterpret clusters as network communities. Eigenvector centrality within each cluster determines the relevance of the corresponding representatives, i.e., we assume that one node's importance depends on that of their neighbors in the same cluster.

A mathematical programming formulation is proposed to maximize group relevance. In mathematical programming, problem solution is coded by decision variables. Modeling eigenvector computation over clusters with decision variables involves nonlinear equations; normalizing them adds difficulty to an already challenging optimization problem. Another key aspect in our model is symmetry breaking. Given that clusters are interchangeable, node-cluster assignment readily implies symmetry issues. Here, we use symmetry breaking constraints to impose some rules to regulate node-cluster assignments.

As an illustrative example, Fig. 1 shows the group of seven most relevant characters in the novel *Les Misérables* (diamond nodes), together with their communities. Here, links connect any pair of characters that coappear in the novel—weights to quantify the number of such coappearances were also considered, as compiled by Knuth [31]. Valjean, the main character, has been identified as a member of the top group. However, Javert, the antagonist, has been not. A community only admits one leader; given two competing nodes, the model automatically discerns whether the sphere of influence of one of them should absorb the other candidate or, conversely, there is room for two leaderships. In this case, Javert is eclipsed by Valjean and embedded in his community. Many other major characters

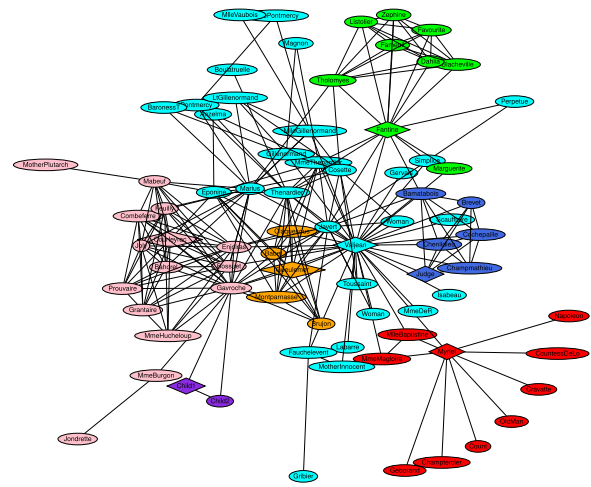


Fig. 1. Optimal solution for *Les Misérables* network: $p = 7$ and $\mu = 0.46$.

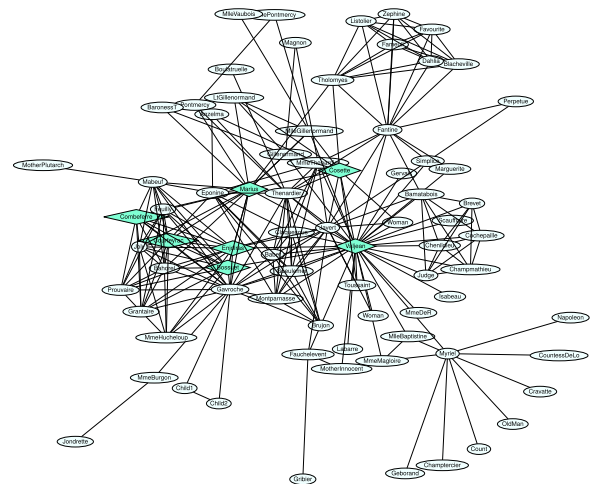


Fig. 2. Top-7 nodes for *Les Misérables* network, according to eigenvector centrality.

are covered by Valjean's influence sphere. In their place, our method prefers secondary characters such as bishop Myriel or Courfeyrac to build a more balanced and covering group of leaders. Case of Child1 is especially remarkable. The character, of little significance for the novel, is, according to our solution, among the seven most relevant ones. Indeed, Child1 and Child2 are to some extent isolated from the rest, and it is precisely this distinct role what the model detects. Gueulemer is identified as the leader of the group consisting of the gang of criminals made of Montparnasse, Claquesous, Babet and himself, and Brujon, another criminal in the novel. To illustrate comparison with eigenvector centrality, Fig. 2 depicts the seven nodes with highest scores according to this measure when applied to the whole network of characters. Evidence that it fails to comply with network coverage is noticeable.

Solution depicted on Fig. 1 has been obtained under mixing parameter $\mu = 0.46$. This means that node relations within a cluster represents, for each node, at least a 54% of its relations. The mixing parameter is an input of our model that can be tuned to control the cohesiveness of the spheres of influence (i.e., communities) in the solution.

As shown by the previous example, eigenvector centrality needs for an adaptation if one wants to address group relevance. In this article, we propose such an adaptation, to the best of our knowledge the first one in the literature. The main contribution of this work is an innovative model that combines optimization of group relevance and community discovery in the same process. Unlike other approaches, which consider the two problems separately in different steps, the proposed mathematical programming model is able to reproduce the interrelation between them. The rest of this article is organized as follows. Section II presents the preliminaries about eigenvector centrality and introduces some notation. The proposed mathematical programming formulation is presented in Section III. Our computational experience is reported and analyzed in Section IV. Finally, Section V closes the article with some conclusions.

II. PRELIMINARIES

Let (V, E, W) be an undirected network with n nodes, $V = \{1, \dots, n\}$, some links, collected in a set E , and weights on these links, arranged in a matrix W . Set E consists of unordered pairs of distinct nodes, i.e., $(i, j) = (j, i)$ and $i \neq j$. Matrix W is symmetric and positive: $W = (w_{ij})$ with $w_{ij} = w_{ji}$ and $w_{ij} \geq 0$ for every pair $i, j \in V$. Weight w_{ij} represents the strength of link ij , if it exists, and it is 0 otherwise.

Eigenvector centrality of one node $j \in V$, c_j , is defined as a weighted sum of the eigenvector centralities of its neighbors [23], [32]

$$\lambda c_j = \sum_{i=1}^n w_{ij} c_i. \quad (1)$$

Equation (1) indicates that one node's importance depends on how notably it is linked to other nodes and how important are such. This equation is equivalent to determining some of the left eigenvectors of W and its associated eigenvalue. Indeed, (1) yields the eigenvector equation $\lambda c = cW$. Solutions for such equation are only guaranteed to exist in \mathbb{R}^n under some conditions. Particularly, when all the entries of W are strictly positive and the sum of any of its rows equals one, W is row stochastic, W can be reinterpreted as the *transition matrix* of an irreducible, aperiodic, and positive recurrent Markov chain [33]. This fact ensures that $\lambda = 1$ is the largest eigenvalue of W with associated eigenvector c unique and such that $c_i > 0$ and $\sum_{i=1}^n c_i = 1$. Eigenvector c is, then, called a *stationary distribution* of the Markov chain.

To guarantee solution existence, a fixed score e_j is usually added to the eigenvector centrality of j

$$\lambda c_j = \sum_{i=1}^n w_{ij} c_i + e_j \quad (2)$$

that is, $\lambda c = cW + e$, where $e = (e_i)$, $e_i > 0$. Note that, if $\sum_{i=1}^n c_i = 1$, we can rewrite previous equation as $\lambda c = c(W + e \times \mathbb{1})$ where $\mathbb{1} = (1, \dots, 1)_{1 \times n}$, i.e., c is a left eigenvector of $W + e \times \mathbb{1}$, whose entries are strictly positive. If $W + e \times \mathbb{1}$ was row stochastic, $\lambda c = c(W + e \times \mathbb{1})$ would have a solution with $\lambda = 1$ and associated eigenvector c in $(\mathbb{R}^+)^n$ such that $\sum_{i=1}^n c_i = 1$, which would be also a solution to (2). Vector e might have different interpretations, including the “exogenous

contribution” to a person's status [32] and the probability that an Internet surfer clicks on links at random [23].

Similarly to k -means clustering, one solution features a partition of the nodes of the network in p clusters, where p is an input parameter. Eigenvector centrality is calculated independently for every cluster in the solution. In other words, we compute an eigenvector for every submatrix of the matrix of weights corresponding to one cluster.

For introducing eigenvector computation inside the network subunits, let $\{C_k\}_{k=1}^p$ denote a partition of V in p clusters, i.e., $\bigcup_{k=1}^p C_k = V$. We consider $W^\epsilon = (w_{ij}^\epsilon)$, where $w_{ij}^\epsilon = w_{ij} + \epsilon$ and $\epsilon > 0$. For each node i belonging to C_k , we denote with π_{ik} the eigenvector centrality of i within C_k . That is, node eigenvector centrality within cluster C_k is computed as follows:

$$\pi_{jk} = \sum_{i \in C_k} \pi_{ik} \frac{w_{ij}^\epsilon}{\sum_{\ell \in C_k} w_{i\ell}^\epsilon} \quad \forall j \in C_k \quad (3)$$

$$\text{subject to: } 0 < \pi_{ik} < 1 \quad \forall i \in C_k \quad (4)$$

$$\sum_{i \in C_k} \pi_{ik} = 1. \quad (5)$$

Equation (3) is equivalent to $\pi_{.k} = \pi_{.k} W^{\epsilon, k}$, where $W^{\epsilon, k} := (\frac{w_{ij}^\epsilon}{\sum_{\ell \in C_k} w_{i\ell}^\epsilon})_{i, j \in C_k}$ is the submatrix of W^ϵ induced by C_k and normalized by rows. Normalization conceptually means that the strength of one link ij is relative to the total influence power exerted from i , computed as the sum of the corresponding row. Since $W^{\epsilon, k}$ has positive entries and is row stochastic ($\sum_{j \in C_k} \frac{w_{ij}^\epsilon}{\sum_{\ell \in C_k} w_{i\ell}^\epsilon} = 1$ for all $i \in C_k$), the system (3)–(5) has a unique solution for each C_k . Note that the introduction of parameter ϵ is the equivalent of taking uniform vector $e = (\epsilon)$. Despite its possible interpretations [23], [32], adding ϵ responds to a technical need: solution existence cannot be guaranteed otherwise. Therefore, we advise to set ϵ sufficiently small to prevent network nature from disruption.

III. MATHEMATICAL FORMULATION

In order to model both, node-cluster assignment and eigenvector calculation, we define the following mathematical programming variables for all $i \in V$ and $k = 1, \dots, p$:

$$\begin{aligned} x_{ik} &\in \{0, 1\}, & x_{ik} &= 1 \text{ iff } i \in C_k; \\ \Pi_{ik} &\in [0, 1], & \Pi_{ik} &= \text{the eigenvector centrality of } i \text{ within } C_k \text{ if } i \in C_k, 0 \text{ otherwise;} \\ y_{ik} &\in \{0, 1\}, & y_{ik} &= 1 \text{ iff } i \text{ is the node with maximum eigenvector centrality within } C_k; \\ z_k &\in [0, 1], & z_k &= \text{the maximum eigenvector centrality in } C_k, \text{ i.e.,} \\ & & z_k &= \sum_{i=1}^n \Pi_{ik} y_{ik}. \end{aligned}$$

Variables Π encode p eigenvectors, one per cluster. For each k , positive components of the vector of variables $\Pi_{.k} = (\Pi_{1k}, \dots, \Pi_{nk})^T$ correspond with node eigenvector centralities within C_k . Observe that we model eigenvector centralities as normalized vectors, i.e., $\Pi_{ik} \in [0, 1]$. Variables y represent clusters centroids, i.e., $y_{ik} = 1$ iff $\Pi_{ik} = \max\{\Pi_{jk} : j \in C_k\}$.

Given a network partition $\{C_k\}_{k=1}^p$, our vectors of variables $\Pi_{.k} = (\Pi_{1k}, \dots, \Pi_{nk})^T$, $k = 1, \dots, p$, must satisfy the following constraints (stationary distribution):

- i) $\Pi_{jk} = x_{jk} \sum_{i=1}^n x_{ik} \Pi_{ik} \frac{w_{ij}^\epsilon}{\sum_{\ell=1}^n w_{i\ell}^\epsilon x_{\ell k}}$, for all $j = 1, \dots, n$;
- ii) $0 \leq \Pi_{ik} \leq 1$ for all $i = 1, \dots, n$;
- iii) $\sum_{i=1}^n \Pi_{ik} = 1$.

These conditions are just a translation of system (3)–(5) into our modeling language with decision variables. Note that condition (i) not only indicates eigenvector computation inside C_k , but also ensures that Π_{ik} components are zero for nodes outside C_k .

Any solution must additionally fulfill some constraints regarding clustering cohesion. Such constraints restrict the cluster-external relations of one node to represent at most a fraction μ of its relations within the entire network. The incorporation of this parameter to the model avoids dummy solutions in which clusters have disconnected components or are singletons. But, more interestingly, μ allows us to reinterpret clusters as network communities. A community is usually defined as a group of nodes such that the density of links between nodes of the group is higher than the average link density in the network [28]. In fact, $1 - \mu$ determines community cohesion and μ has been known as the *mixing parameter* [28].

The group of top relevance nodes—the cluster centroids—is made of those with the highest eigenvector component in each cluster. Our formulation finds the partition in clusters that maximizes the sum of centroids' eigenvector components; eigenvectors normalization prevents the optimization to produce biased solutions. When $p = 1$, the approach becomes trivially a single eigenvector computation on the complete network, serving as a generalization of classical eigenvector centrality.

We have now the ingredients to present the following mixed-integer nonlinear formulation for the problem of group centrality, where $i, j \in V$ and $k = 1, \dots, p$:

$$\max \sum_{k=1}^p z_k \quad (6)$$

$$\text{s.t.} \quad \sum_{k=1}^p x_{ik} = 1 \quad \forall i \quad (7)$$

$$\sum_{i=1}^n y_{ik} = 1 \quad \forall k \quad (8)$$

$$\sum_{i=1}^n \Pi_{ik} = 1 \quad \forall k \quad (9)$$

$$\Pi_{jk} = x_{jk} \sum_{i=1}^n x_{ik} \Pi_{ik} \frac{w_{ij}^\epsilon}{\sum_{\ell=1}^n w_{i\ell}^\epsilon x_{\ell k}} \quad \forall i, k \quad (10)$$

$$\Pi_{ik} \leq x_{ik} \quad \forall i, k \quad (11)$$

$$y_{ik} \leq x_{ik} \quad \forall i, k \quad (12)$$

$$z_k \leq \Pi_{ik} + 1 - y_{ik} \quad \forall i, k \quad (13)$$

$$y_{ik} \leq 1 + \Pi_{ik} - \Pi_{jk} \quad \forall i, j, k \quad (14)$$

$$\sum_{j=1}^n (w_{ij} + w_{ji}) x_{jk} \geq$$

$$(1 - \mu) x_{ik} \sum_{j=1}^n (w_{ij} + w_{ji}) \quad \forall i, k \quad (15)$$

$$\Pi_{ik}, z_k \geq 0 \quad \forall i, k$$

$$x_{ik}, y_{ik} \in \{0, 1\} \quad \forall i, k.$$

The objective function to maximize (6) is the sum of the greatest eigenvector components in each cluster. Constraints (7) impose that each node is assigned to only one cluster, and (8) guarantee that every cluster has only one centroid. Regarding eigenvectors, constraints (9) and (10) ensure that stationary distribution equations (iii) and (i) are satisfied. On the other hand, (11) and (12) prevent Π_{ik} and y_{ik} to be positive when i is not in cluster C_k . Constraints (13) have no effect when i is not a centroid and upperly bound z_k by Π_{ik} when it is; since (6) is to be maximized, z_k will attain this bound in the optimal solution. Constraints (14) ensure that binary variable y_{ik} is zero if there is a node with eigenvector component greater than Π_{ik} . Finally, (15) are to guarantee clusters cohesion. Given that $i \in C_k$ [i.e., $x_{ik} = 1$ on the right-hand side of (15)], the overall relations of i with nodes inside C_k (on the left-hand side) must be greater than a $(1 - \mu) \cdot 100\%$ of its overall relations within the network, $\sum_{j=1}^n (w_{ij} + w_{ji})$. On the other hand, observe that cluster cohesion is imposed using W — and not W^ϵ — as a reference. These inequalities will be referred as *mixing constraints*. Last lines in the formulation determine the type of the variables— positive (Π and z) and binary (x and y).

Even though the previous is a valid formulation, it has a main drawback, namely that (10) is highly nonlinear. In order to linearize it, we introduce two families of continuous variables T and F

$$T_{ik} := \frac{x_{ik} \Pi_{ik}}{\sum_{\ell=1}^n w_{i\ell}^\epsilon x_{\ell k}} \quad \forall i \in V, \forall k = 1, \dots, p \quad (16)$$

$$F_{ijk} := T_{ik} x_{jk} \quad \forall i, j \in V, \forall k = 1, \dots, p. \quad (17)$$

Using T variables, eigenvector (10) can be rewritten as

$$\Pi_{jk} = x_{jk} \sum_{i=1}^n w_{ij}^\epsilon T_{ik} \quad \forall i \in V, \forall k = 1, \dots, p \quad (18)$$

while, using F variables, (16) reads

$$\sum_{\ell=1}^n w_{i\ell}^\epsilon F_{i\ell k} = x_{ik} \Pi_{ik} \quad \forall i \in V, \forall k = 1, \dots, p. \quad (19)$$

The introduction of these variables allows us to rewrite previous nonlinear model into the following mixed-integer linear formulation:

(p -leaders)

$$\max \sum_{k=1}^p z_k$$

$$\text{s.t.} \quad (7) - (9), (11) - (15)$$

$$F_{ijk} \leq T_{ik} \quad \forall i, j, k \quad (20)$$

$$F_{ijk} \leq M x_{jk} \quad \forall i, j, k \quad (21)$$

$$F_{ijk} \geq T_{ik} - M(1 - x_{jk}) \quad \forall i, j, k \quad (22)$$

$$\sum_{\ell=1}^n w_{i\ell}^\epsilon F_{i\ell k} = \Pi_{ik} \quad \forall i, k \quad (23)$$

$$T_{ik} \leq M x_{ik} \quad \forall i, k \quad (24)$$

$$\sum_{i=1}^n w_{ij}^\epsilon T_{ik} \leq \Pi_{jk} + M'(1 - x_{jk}) \quad \forall i, k \quad (25)$$

$$\sum_{i=1}^n w_{ij}^\epsilon T_{ik} \geq \Pi_{jk} - 1 + x_{jk} \quad \forall i, k \quad (26)$$

$$\Pi_{ik}, z_k, T_{ik}, F_{ijk} \geq 0 \quad \forall i, j, k$$

$$x_{ik}, y_{ik} \in \{0, 1\} \quad \forall i, k.$$

Here, (20)–(22) serve to linearize (17). In effect, when $x_{jk} = 1$, (20) and (22) guarantee that $F_{ijk} = T_{ik}$, and $F_{ijk} = 0$ otherwise due to (21). Constant M should be large enough and can be computed in terms of W . Equations (19) stand thanks to (23) and (24). On the one hand, (23) are precisely (19) when $x_{ik} = 1$. On the other hand, if $x_{ik} = 0$, the right-hand side of (23) is zero because of (11), and so is its left-hand side due to (24) together with (20). Finally, (25) and (26) account for eigenvector (18) when $x_{jk} = 1$. Otherwise, (25) and (26), where M' is again a large enough constant, have no effect and (11) ensure that (18) stand.

Formulation (p -leaders) can be now implemented in a computer and solved with any commercial optimizer for mixed-integer linear programming such as Cplex, Gurobi, or Xpress. All these solvers feature exact procedures, i.e., the solution obtained is guaranteed to be optimal. Nevertheless, a careful design of formulations is crucial for solvers success. In the following, we explore some modifications to improve (p -leaders).

A. Formulation Improvements

The size of (p -leaders) can be reduced since $w_{ij} = w_{ji}$ for all $(i, j) \in E$. Indeed, not all variables T_{ik} are needed. Suppose that $i, j \in C_k$. We claim that $T_{ik} = T_{jk}$, i.e.,

$$\frac{\Pi_{ik}}{\sum_{\ell=1}^n w_{i\ell}^\epsilon x_{\ell k}} = \frac{\Pi_{jk}}{\sum_{\ell=1}^n w_{j\ell}^\epsilon x_{\ell k}}.$$

Equivalently

$$\Pi_{ik} \sum_{\ell=1}^n w_{j\ell}^\epsilon x_{\ell k} = \Pi_{jk} \sum_{\ell=1}^n w_{i\ell}^\epsilon x_{\ell k}.$$

Now, we can use (10) to replace Π_{ik} , Π_{jk} , and then apply the distributive property

$$\begin{aligned} & \left(\sum_{s=1}^n x_{sk} \Pi_{sk} \frac{w_{si}^\epsilon}{\sum_{r=1}^n w_{sr}^\epsilon x_{rk}} \right) \sum_{\ell=1}^n w_{j\ell}^\epsilon x_{\ell k} \\ &= \left(\sum_{s=1}^n x_{sk} \Pi_{sk} \frac{w_{sj}^\epsilon}{\sum_{r=1}^n w_{sr}^\epsilon x_{rk}} \right) \sum_{\ell=1}^n w_{i\ell}^\epsilon x_{\ell k}, \\ & \sum_{s=1}^n \left(\frac{x_{sk} \Pi_{sk}}{\sum_{r=1}^n w_{sr}^\epsilon x_{rk}} \sum_{\ell=1}^n w_{si}^\epsilon w_{j\ell}^\epsilon x_{\ell k} \right) \\ &= \sum_{s=1}^n \left(\frac{x_{sk} \Pi_{sk}}{\sum_{r=1}^n w_{sr}^\epsilon x_{rk}} \sum_{\ell=1}^n w_{sj}^\epsilon w_{i\ell}^\epsilon x_{\ell k} \right) \end{aligned}$$

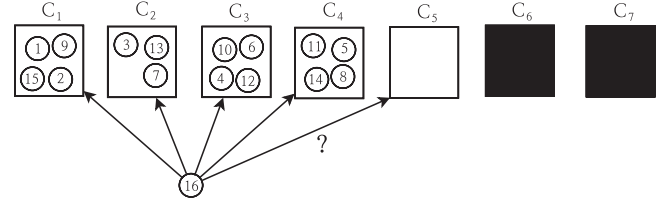


Fig. 3. Realization of (27) when $k = 5$ and $i = 16$.

$$\begin{aligned} & \sum_{s=1}^n \sum_{\ell=1}^n \left(\frac{x_{sk} \Pi_{sk}}{\sum_{r=1}^n w_{sr}^\epsilon x_{rk}} w_{si}^\epsilon w_{j\ell}^\epsilon x_{\ell k} \right) \\ &= \sum_{s=1}^n \sum_{\ell=1}^n \left(\frac{x_{sk} \Pi_{sk}}{\sum_{r=1}^n w_{sr}^\epsilon x_{rk}} w_{sj}^\epsilon w_{i\ell}^\epsilon x_{\ell k} \right). \end{aligned}$$

The last equality stands because W is symmetric. Consequently, we have that $T_{ik} = T_{jk} \forall i, j \in C_k$ and the size of the formulation can be reduced.

Based on clustering, our approach involves symmetry issues regarding node-group assignments, and so does (p -leaders). Given a feasible solution to (p -leaders), several (symmetric) solutions can be produced by permuting clusters members, all them having same objective value (6). This represents a serious hindrance for the solving procedure, which is based on branching on the binary variables: the branching tree grows exponentially towards the different equivalent feasible solutions. A natural approach to avoid this is to eliminate such symmetric solutions obtained by permutations, i.e., to make them unfeasible for the model. In our case, imposing some constraints that rule node-cluster assignment will do the work. For each node i and cluster C_k , we introduce the following mathematical programming constraints:

$$\sum_{s=1}^k x_{is} + \sum_{j=k}^{i-1} x_{jk} \geq 1 \quad \forall i > k \quad (27)$$

$$x_{ik} = 0 \quad \forall i < k, \quad \forall j \in V. \quad (28)$$

Constraints (28) state that node 1 is assigned to C_1 , node 2 is assigned either to C_1 or C_2 and, in general, i is not assigned to C_k if $k > i$. These constraints also imply $y_{ik} = \Pi_{ik} = T_{ik} = F_{ijk} = 0 \forall i < k, \forall j \in V$. On the other hand, (27) are a particular case of shifted column inequalities for the partitioning orbitope [34]. For an interpretation of (27), assume without loss of generality that nodes are assigned to their clusters in order by index, i.e., node 1 is assigned first, node 2 is assigned in second place, and so on so forth. If i is the next node to assign and C_k is empty, i.e., $\sum_{j=k}^{i-1} x_{jk} = 0$, then i will not be assigned to a cluster with index greater than k , i.e., $\sum_{s=1}^k x_{is} = 1$. Fig. 3 illustrates (27) when $k = 5$, $i = 16$, and $p = 7$. In this example, nodes from 1 to 15 have been placed inside clusters C_1, \dots, C_4 . Constraints (27) forbid that the next node, 16, is placed into clusters C_6 or C_7 , because C_5 is empty. Constraints (27) together with (28) indeed constitute the following rule for node-cluster assignment: “every node is assigned either to some of the already used clusters or to the first empty one”.

IV. EXPERIMENTS AND ANALYSIS

This section discusses the computational performance of the proposed approach. Model (p -leaders) can be solved by any standard optimization software. This kind of software, usually called solvers, are based on a branch and bound scheme and on some linear programming algorithms, such as the Simplex. Using the latter, the linear programming relaxation of the model is first solved at the root. Then, two subproblems (the child nodes of the root) are created by fixing one of the binary variables to 0 and 1, respectively. The process is repeated for the new nodes in the tree, until all the search space is explored. The search tree can be pruned by using the bounds provided by the linear relaxations together with the current best objective value. The time spent at each node, the bounds and the size of the tree strongly depend on the mathematical formulation used, which will have a significant impact on the algorithm overall efficiency. In this sense, improvements proposed in Section III-A are oriented to boost the algorithm's performance. An introduction to mathematical programming methods and algorithms can be found in [35].

We conducted experiments on two real-life networks and twenty synthetically generated networks. Our experiments were run on an i7-6700 k 4.0 GHz \times 8 machine with 16 GB memory and the solver used was CPLEX v12.6.3.

A. Real-Life Networks

As first real-word example, we use the Zachary's Karate Club weighted network [36], which is a standard for testing community detection algorithms. This network collects relative strengths of the relations in a university-based karate club. Fig. 4 shows several illustrations that emphasize different social structures within the club. Nodes are labeled with numbers from 0 to 33 that represent anonymous club members, except for the instructor Hi (node 0) and the administrator John A. (node 33). After a conflict, the club separated into those members that formed a club around John A. and those who remained with Hi. Fig. 4(a) shows the community structure observed by Zachary, which only misclassified node 8 with respect to the actual split. When nodes are ranked by eigenvector centrality, John A. and Hi occupy the first and second place, respectively. As opposed to most approaches, our model reveals previously unnoticed node 5 for a choice of two members with minimum cluster cohesion $1 - \mu = 0.62$ [see Fig. 4(b)]. This exemplifies the model's preference for small communities, given that they satisfy constraints regarding cohesion. When a set of three key members is selected, node 5 together with "ground truth" leaders John A. and Hi are nominated. The increment entails a loss of cluster cohesion, which drops from 0.62 to 0.58. Modular structure of the solution coincides with existing knowledge and features the same communities frontiers originally observed by Zachary [see Fig. 4(c)]. However, Hi's faction is divided into two subunits: one led by member 5, which coincides with that of Fig. 4(b), and other led by himself. Finally, Fig. 4(d) illustrates the optimal group of four key members and their relative communities, which inevitably present significant interrelations ($\mu = 0.53$). This time, unlike other approaches that split the green group on Fig. 4(c) into two subcommunities, our model

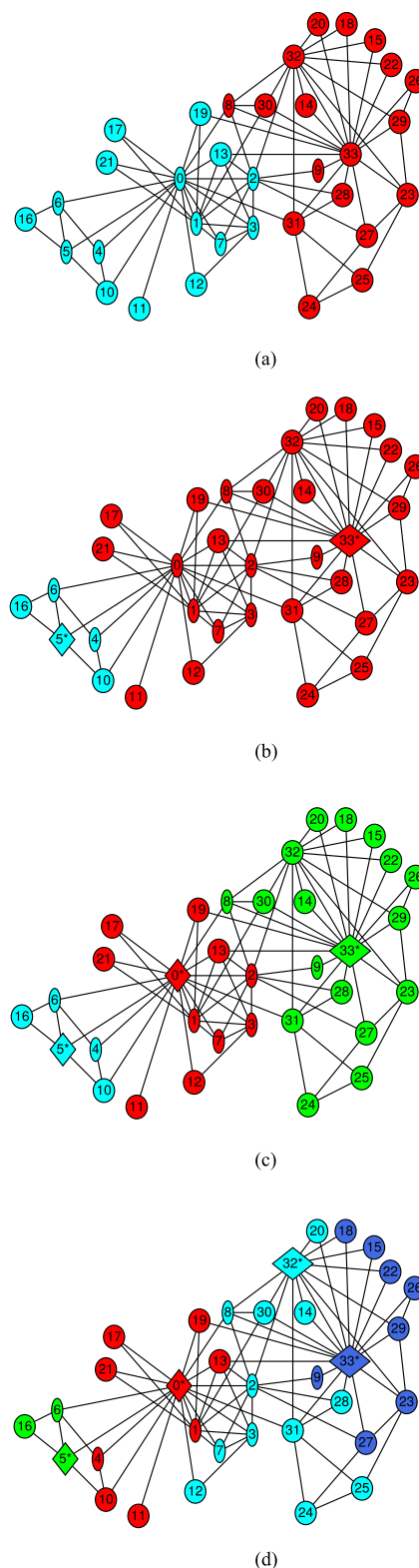


Fig. 4. Social structures in the Zachary's Karate Club weighted network [36] revealed by different methods. (a) Ford-Fulkerson binary community detection algorithm applied by Zachary [36]. (b) Our model with $p = 2$, $\mu = 0.38$. While 0 and 33 are frequently selected [29], [30], our solution reveals node 5 as key member. (c) Our model with $p = 3$, $\mu = 0.42$. The depicted community structure was identified in previous studies [37], [38], [39]. (d) Our model with $p = 4$, $\mu = 0.53$ In [40], group $\{0, 5, 31, 33\}$ was obtained with clustering and information transfer probabilities; and $\{0, 1, 32, 33\}$ with the greedy of [41]. Communities also differ from previous studies [39], [40].

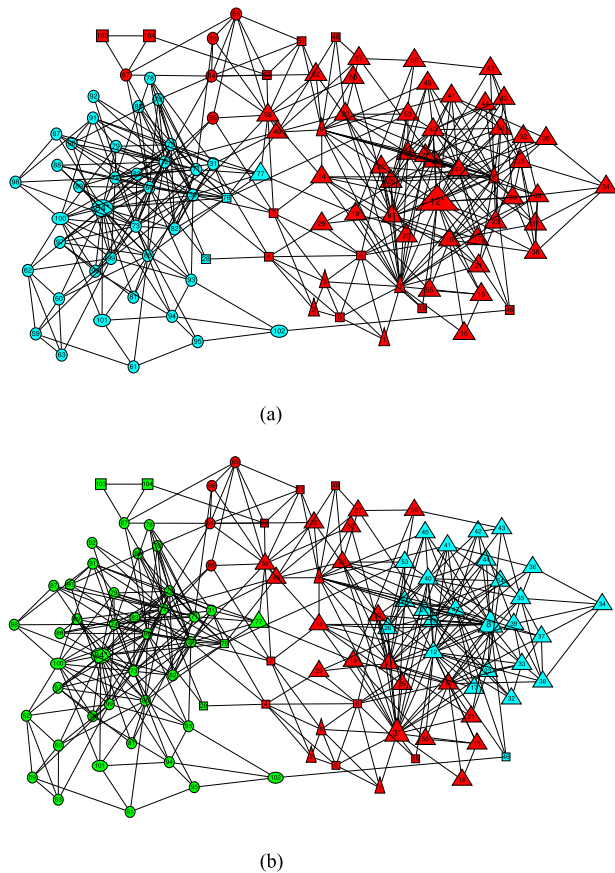


Fig. 5. Key members and communities in the American Political Books (unweighted) network [38]. Circle, triangle and square nodes represent ground-truth partition in liberal, conservative and neutral books, resp., which features modularity 0.4149, see [29]. (a) Our model with $p = 2$, $\mu = 0.4$. Nodes in the optimal group, 12 and 84, have centralities 0.0496 and 0.0644. Partition modularity is 0.4424. In [29], 8 and 72 were selected (which have centralities 0.0496 and 0.0561 in our solution) and modularity was 0.4454. (b) Our model with $p = 3$, $\mu = 0.48$. Nodes 84, 8 and 3 form an optimal group of key members and partition modularity is 0.4924.

redistribute community memberships of that solution. Spotted key members are $\{0, 5, 32, 33\}$, as opposed to the top group $\{0, 2, 32, 33\}$ according to a ranking by eigenvector centrality. The four key members identified by [37], $\{0, 5, 31, 33\}$, lie in the four communities in Fig. 4(d) separately. However, node 31 has eigenvector centrality 0.137 in our solution while 32, the centroid, has 0.164.

For a second test with real-word data, we use the American Political Books network compiled by Krebs [38] [see Fig. 5]. Here, nodes are 105 books about US politics sold on Amazon, and edges represent frequent copurchasing of two books by the same buyers. When the books are ranked by eigenvector centrality, books 8 and 12 draw as top ones, followed by 3 and 84, which also end in a tie. Conversely, our formulation identifies 84 and 12 as optimal group, which lie in different books categories, as opposed to 8 and 12, which are both conservative books [see Fig. 5(a)]. Alternatively, 84 and 8 are also an optimal solution to our model. For the three most popular books, our method finds 84, 3, and 8 [see Fig. 5(b)]. This time ties between 8 and 12 are broken, and 12 is absorbed by the community of 8, which is selected as one of the key books. Furthermore, nodes 3 and

84 are identified as leading books of distinct communities. Blue and green communities fairly respect the categories identified by Newman, while the red one contains a mixture of the three types of books.

B. Synthetic Networks

The use of a public generator of synthetic networks that feature modular structures [28] allows us to test the scalability of the model while validating community discovery. The resulting testbed gathers networks of 50, 100, 150, 200, and 500 nodes, with links densities ranging from 1.5 to 16 (%). The benchmarks feature between 3 and 13 communities whose interrelation accounts for at most a 58% of the links of one node in the less cohesive network and a 8% in the most cohesive one. We compiled in total 20 benchmarks, four of each size.

For evaluating consistency of the method, for each of the 20 instances, we solve the model for different values of p without exceeding the number of communities in the network according to the benchmark generator [28]. We observe whether the group of key members is stable when p increases. Parameter μ , the maximum mixing allowed between clusters, have to be tuned in each experiment. For instance, with $p = 2$, we begin with a small μ of 0.1 or 0.2 and increment it until the model is feasible. Feasibility is rapidly discarded, which allows us to tune by inspection. The number of clusters is increased at the cost of losing cohesiveness, and μ has to be consequently updated with the increase of p . As an illustrative example, Fig. 6 shows the optimal solutions on a network of 50 nodes with $p = 2, 3, 4$, and 5. Different clusters are depicted in different colors and key members are marked with an asterisk and displayed as diamond nodes. Fig. 6(d) illustrates the optimal solution for $p = 5$, which perfectly matches the community structure produced by the benchmark generator. Taking then this figure as reference, one can observe that communities frontiers are maintained for $p < 5$. Indeed, when p is decreased, the original five communities are placed one inside another, like matryoshkas. Inspecting now how the optimal group of members varies when p decreases, we find a singularity when $p = 4$. Member 37 is in the optimal group of four leaders depicted on Fig. 6(c), while it is not among the optimal five key members in Fig. 6(d). Despite the apparent contradiction, both solutions seem natural, since 37 is linked to members 42 and 44, the leaders of the original communities on Fig. 6(d) that are merged to form a bigger community on Fig. 6(c) whose leader is now 37. These fluctuations in leaderships can account for quite of the experiments made on one instance and may not occur for another. On average, we have found that such changes occur in a 4.7% of our experiments. This tendency may suggest the use of some ad hoc strategy to approximate a solution by successive steps.

Table I summarizes the experiments compiled. The benchmarks, whose nodes range from 50 to 500, are displayed on different rows. For each of them, we have executed our method with p ranging from the number of connected components of the network to the number of communities according to the benchmark generator [28]. The table shows, for each instance and p , the minimum mixing parameter under which an optimal solution is obtained. In other words, there is not a partition of

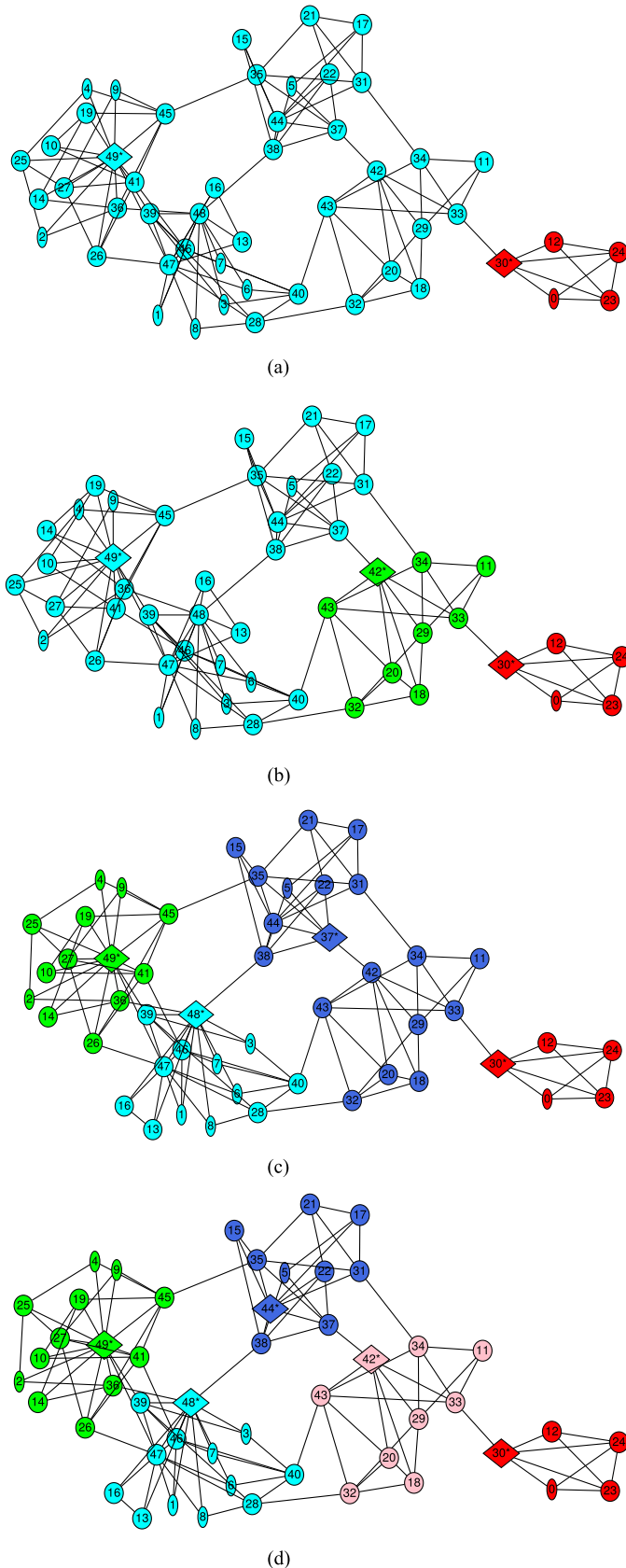


Fig. 6. Optimal solutions to instance 50_1 for different p . (a) $p = 2$, $\mu = 0.21$, $\text{OPT} = 0.34$. (b) $p = 3$, $\mu = 0.21$, $\text{OPT} = 0.51$. (c) $p = 4$, $\mu = 0.23$, $\text{OPT} = 0.77$. (d) $p = 5$, $\mu = 0.23$, $\text{OPT} = 0.98$.

TABLE I
MIXING PARAMETER μ (IN%) FOR DIFFERENT p THROUGHOUT THE BENCHMARKS

	p												
	1	2	3	4	5	6	7	8	9	10	11	12	13
50_1	0	21	21	23	23								
50_2	0	41	41	44	48	48	48						
50_3			0	17									
50_4	0	36	36										
100_2		0	29	29	30	34	34	37					
100_3							0	25	26	-			
100_5	0	22	22	23	23	27							
100_6	0	27	27	28	30	35	35	-					
150_1	0	27	27	28	33	-	-	-					
150_4	0	22	23	24	24	25	25	27	30	30			
150_5	0	26	26	28	29	31							
150_7	0	26	26	27	-	-							
200_1	0	21	23	24	26	29	32						
200_2		0	17	17	18	18	19	19	20	26			
200_3	0	-	-	-	-	-	-	21	-	23			
200_4		0	14	14	15	15	15	16	16	17	18		
500_1						0	5	6	6	8			
500_2					0	7	7	7	11				
500_3			0	19	19	20							
500_4					0	19	24	30	30	-	-	44	

Entries are in Bold When p Equals the Number of Communities Indicated by the Benchmark Generator.

TABLE II
RUNNING TIME (IN SECONDS), TL INDICATES THAT TIME LIMIT (5000 s.) WAS REACHED

	p												
	1	2	3	4	5	6	7	8	9	10	11	12	13
50_1	0	1	1	5	1								
50_2	0	25	5	58	2787	787	410						
50_3		0	0										
50_4	0	31	7										
100_2		0	1	1	1	8	2	15					
100_3							0	0	1	TL			
100_5	0	1	1	3	1	2							
100_6	0	2	2	6	7	46	11	TL					
150_1	2	10	16	45	23	TL	TL	TL					
150_4	2	4	30	81	83	293	119	368	3146	73			
150_5	2	9	9	18	42	173							
150_7	2	35	9	171	TL	TL							
200_1	10	10	7	248	495	324	2028						
200_2			5	755	83	998	155	3710	390	844	125		
200_3	10	TL	TL	TL	TL	TL	TL	76	TL	689			
200_4		5	15	129	1050	239	145	2627	533	233	329		
500_1							96	78	321	317	351		
500_2					309	319	748	156	1546				
500_3			196	130	186	1224							
500_4						169	608	2644	1313	1921	TL	TL	4902

the given network in p clusters with mixing parameter smaller than that indicated in Table I. Zero entries appear when p equals the number of connected components of the network. In this case, our method simply computes eigenvector centrality in each connected component. Dash entries indicate unsolved configurations. Finally, bold entries correspond to experiments in which p matches the number of communities identified by the benchmark generator. In those experiments, the community structure produced by the generator was perfectly uncovered, i.e., it coincides with the optimal cluster partition identified by our method. On the other hand, Table II shows the running time spent to find the optimal solution for each instance and p .

When $n = 50$, the method is able to find the optimal key members for all the experiments run, even when the mixing parameter is close to 0.5 (see Table I). The underlying communities

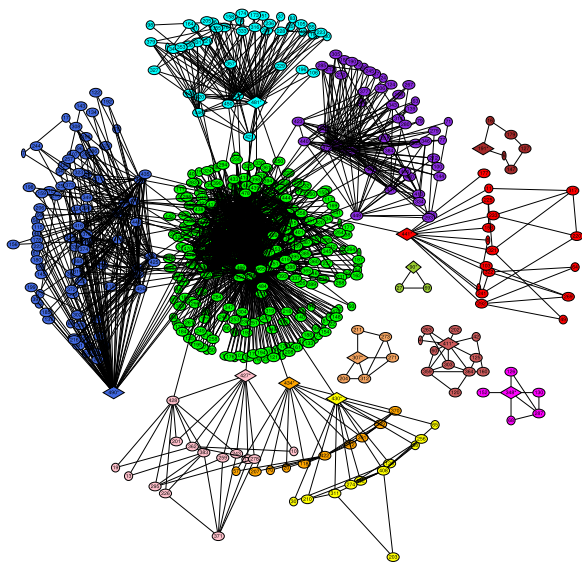


Fig. 7. Optimal solution for synthetic benchmark 500_4 with $n = 500$, $p = 13$, $\mu = 0.44$.

according to the benchmark generator are discovered in all the cases. When $n = 100$ or $n = 150$, the method also finds optimal solutions for wide ranging sizes of the key members group. However, sometimes p does not reach the number of underlying communities when they are not well-cohered (mixing parameter close to 0.5 or exceeding 0.5 in the underlying structures). For the last group of benchmarks, those with $n = 200$ and $n = 500$, community structures produced by the generator are always uncovered (we find bold entries for these instances on Table I). Note that running times are not only related to the network size but also to the mixing parameter (see Table II). For instance, it took 410 s to solve an instance with $n = 50$ and $\mu = 0.48$ and 329 s to solve one with $n = 200$ and $\mu = 0.18$. Despite comparison with underlying structures for testing purposes, one should keep in mind that parameters p and μ are not known *a priori* for a given network. Moreover, these parameters present some degree of uncertainty in real applications, especially for large networks. In this vein, difficulties in finding an optimal solution to, for instance, network 500_4 and $p = 12$, are palliated by the discovery of 13 key members and their communities, which, as a matter of fact, fit better with the actual network layout (see Fig. 7). Indeed, if the model struggles to find a solution, it is probably the case that the structures that are being searched for (which are determined by p and μ) are far from the intrinsic network organization.

Unlike most existing solving strategies, which are of heuristic nature, results reported here provide the best group of key members for the proposed approach. Finding such optimum obviously has a computational cost, which can turn expensive depending on the network at hand. Our computational experience on the proposed model demonstrates extremely good performance for the smallest instances and high flexibility for the largest ones, even if sometimes optimal solutions are not achieved within a reasonable time limit through the full range of possible values of p .

V. CONCLUSION

This article proposes a new approach to identify the group of most relevant nodes, where relevance is based on eigenvector centrality. Originally thought to give support to eigenvector computation, clusters became a key aspect of our proposal, whose result is twofold. Clustering partition and group relevance optimization interact in a two-directional feedback mechanism to reveal both network modular structure and key members of the network. Network partition directly affects group relevance estimation, while optimizing such results in a more suitable partition in clusters.

Experiments on real-life networks of small size show interesting results that reveal previously unnoticed key members. Additionally, clusters are consistent with previous knowledge on the community structure of the networks. Our computational experience on larger synthetic networks demonstrates an adequate scalability of the method, which is able to find optimal solutions for networks of hundreds of members and thousands of links. As future work, the use of some heuristic, such as large neighborhood search (LNS), together with the proposed formulation could be explored with the aim of solving bigger instances.

REFERENCES

- [1] S. Borgatti, "Identifying sets of key players in a social network," *Comput. Math. Org. Theory*, vol. 12, pp. 21–34, 2006.
- [2] F. Hao, G. Min, Z. Pei, D. Park, and L. Yang, " k -clique community detection in social networks based on formal concept analysis," *IEEE Syst. J.*, vol. 11, no. 1, pp. 250–259, Mar. 2017.
- [3] D. Lusseau, K. Schneider, O. Boisseau, P. Haase, E. Slooten, and S. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behavioral Ecology Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003.
- [4] V. Krebs, "Mapping networks of terrorist cells," *Connections*, vol. 24, no. 3, pp. 43–52, 2002.
- [5] F. Probst, L. Grosswiele, and R. Pflieger, "Who will lead and who will follow: Identifying influential users in online social networks," *Bus. Inf. Syst. Eng.*, vol. 5, no. 3, pp. 179–193, 2013.
- [6] A. Arulselvan, C. Commander, L. Elefteriadou, and P. Pardalos, "Detecting critical nodes in sparse graphs," *Comput. Operations Res.*, vol. 36, pp. 2193–2200, 2009.
- [7] J. Arndt, "Role of product-related conversations in the diffusion of a new product," *J. Marketing Res.*, vol. 4, no. 3, pp. 291–295, 1967.
- [8] S. Aral and D. Walker, "Identifying influential and susceptible members of social networks," *Science*, vol. 337, pp. 337–341, 2012.
- [9] E. Hollander and W. Webb, "Leadership, followership, and friendship: An analysis of peer nominations," *J. Abnormal Social Psychol.*, vol. 50, no. 2, pp. 163–167, 1955.
- [10] L. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.
- [11] L. Lü, D. Chen, X. Ren, Q. Zhang, Y. Zhang, and T. Zhou, "Vital nodes identification in complex networks," *Physical Rep.*, vol. 650, pp. 1–63, 2016.
- [12] M. Kitsak *et al.*, "Identification of influential spreaders in complex networks," *Nature Phys.*, vol. 6, no. 11, pp. 888–893, 2010.
- [13] D. Knoke and R. Burt, "Prominence," in *Applied Network Analysis*, R. Burt and M. Minor, Eds. Beverly Hills, CA, USA: Sage, 1983, pp. 195–222.
- [14] P. Bonacich, "Factoring and weighting approaches to status scores and clique identification," *J. Math. Sociology*, vol. 2, no. 1, pp. 113–120, 1972.
- [15] G. Flament, *Applications of Graph Theory to Group Structure*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1963.
- [16] C. Jordan, "Sur les assemblages de lignes," *J. für die Reine und Angewandte Mathematik*, vol. 70, no. 185, pp. 185–190, 1869.
- [17] G. Sabidussi, "The centrality index of a graph," *Psychometrika*, vol. 31, no. 4, pp. 581–603, 1966.

- [18] P. Bonacich, "Power and centrality: A family of measures," *Amer. J. Sociology*, vol. 92, no. 5, pp. 1170–1182, 1987.
- [19] S. Dorogovtsev, A. Goltsev, and J. Mendes, " k -core organization of complex networks," *Physical Rev. Lett.*, vol. 96, no. 4, 2006, Art. no. 040601.
- [20] M. Newman, "A measure of betweenness centrality based on random walks," *Social Netw.*, vol. 27, no. 1, pp. 39–54, 2005.
- [21] L. Freeman, "Centrality in social networks conceptual clarification," *Social Netw.*, vol. 1, no. 3, pp. 215–239, 1978.
- [22] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.
- [23] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Tech. Rep., Stanford InfoLab., 1999.
- [24] J. Hirsch, "An index to quantify an individual's scientific research output," *Proc. Nat. Acad. Sci.*, vol. 102, no. 46, pp. 16 569–16 572, 2005.
- [25] F. Morone and H. Makse, "Influence maximization in complex networks through optimal percolation," *Nature*, vol. 524, pp. 65–68, 2015.
- [26] J. Zhang, D. Chen, Q. Dong, and Z. Zhao, "Identifying a set of influential spreaders in complex networks," *Scientific Rep.-U.K.*, vol. 6, no. 27823, 2016, Art. no. 27823.
- [27] M. Everett and S. Borgatti, "The centrality of groups and classes," *J. Math. Sociology*, vol. 23, no. 3, pp. 181–201, 1999.
- [28] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Rev. E*, vol. 78, no. 4, 2008, Art. no. 046110.
- [29] Y. Li, C. Jia, and J. Yu, "A parameter-free community detection method based on centrality and dispersion of nodes in complex networks," *Physica A, Statistical Mechanics Appl.*, vol. 438, pp. 321–334, 2015.
- [30] A. Stanoev, D. Smilkov, and L. Kocarev, "Identifying communities by influence dynamics in social networks," *Physical Review E*, vol. 84, no. 4, 2011, Art. no. 046102.
- [31] D. Knuth, "The stanford graphbase: A platform for combinatorial computing," New York, NY, USA: ACM Press; Addison-Wesley Publishing Company, 1993.
- [32] C. Hubbell, "An input-output approach to clique identification," *Sociometry*, vol. 28, no. 4, pp. 377–399, 1965.
- [33] R. Serfozo, *Basics of Applied Stochastic Processes*. Berlin, Germany: Springer, 2009.
- [34] V. Kaibel and M. Pfetsch, "Packing and partitioning orbitopes," *Math. Program.*, vol. 114, no. 1, pp. 1–36, 2008.
- [35] L. Wolsey, *Integer Programming*. Hoboken, NJ, USA: Wiley, 1998.
- [36] W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropology Res.*, vol. 33, no. 4, pp. 452–473, 1977.
- [37] X. Zhang, J. Zhu, Q. Wang, and H. Zhao, "Identifying influential nodes in complex networks with community structure," *Knowl.-Based Syst.*, vol. 42, pp. 74–8, 2013.
- [38] V. Krebs, "Books about us politics," 2004. [Online]. Available: <http://www.orgnet.com/>



Emilio Carrizosa was born in Córdoba, Spain, in 1966. He received the B.S. and Ph.D. degrees in mathematics from the Universidad de Sevilla, Seville, Spain, in 1989 and 1992, respectively.

He is currently a Full Professor with the Department of Statistics and Operational Research, Faculty of Mathematics, Universidad de Sevilla, and the Director of the Instituto de Matemáticas de la Universidad de Sevilla. His main research interests include topics in Optimization (mixed integer nonlinear programming, global optimization) and Big Data (visualization, support vector machines, clustering).

Prof. Carrizosa a President of the Spanish Society of Statistics and Operations Research from 2017 to 2019.



Alfredo Marín was born in Murcia, Spain, in 1962. He received the B.S. degree from the Universidad de Granada, Seville, Spain, in 1987, and the Ph.D. degree from the Universidad de Murcia, Murcia, Spain, in 1996, both in mathematics.

He is currently a Full Professor with the Department of Statistics and Operational Research, Faculty of Mathematics, Universidad de Murcia. His main research interests include integer programming, combinatorial optimization and discrete location.

Prof. Marín was a member of the board of the EURO Working Group on Locational Analysis from 2007 to 2011.



Mercedes Pelegrín was born in Murcia, Spain, in 1992. She received the B.S. degree in mathematics and computer engineering and the Ph.D. degree in mathematics from the Universidad de Murcia, Murcia, Spain, in 2015 and 2019, respectively.

She is currently a Postdoctoral Researcher with the École Polytechnique in Palaiseau, France. She was a Visiting Student of the Université Libre de Bruxelles, in 2017, and the Instituto de Matemáticas de la Universidad de Sevilla (IMUS), in 2018. Her research interests include mathematical programming,

combinatorial optimization, and facility location.