International Journal on Artificial Intelligence Tools
Vol. 25, No. 4 (2016) 1650021 (44 pages)

DOI: 10.1142/S0218213016500214

An Evolutionary and Local Refinement Approach for Community Detection in Signed Networks

Alessia Amelio
DIMES, University of Calabria, Via P. Bucci 44, 87036 Rende (CS), Italy
a.amelio@dimes.unical.it

Clara Pizzuti*
National Research Council of Italy (CNR)
Institute for High Performance Computing and Networking (ICAR)
Via P. Bucci 7/11C, 87036 Rende (CS), Italy
clara.pizzuti@icar.cnr.it

Received 30 June 2015
Accepted 21 May 2016
Published 17 August 2016

An approach to detect communities in signed networks that combines Genetic Algorithms and local search is proposed. The method optimizes the concepts of modularity and frustration in order to find network divisions far from random partitions, and having positive and dense intra-connections, while sparse and negative inter-connections. A local search strategy to improve the network division is performed by moving nodes having positive connections with nodes of other communities, to neighboring communities, provided that there is an increase in signed modularity. An extensive experimental evaluation on randomly generated networks for which the ground-truth division is known proves that the method is competitive with a state-of-art approach, and it is capable to find accurate solutions. Moreover, a comparison on a real life signed network shows that our approach obtains communities that minimize the positive inter-connections and maximize the negative intra-connections better than the contestant methods.

Keywords: Evolutionary computation; community detection; multiobjective clustering; signed networks; local search.

1. Introduction

In the last few years the rapid diffusion of Internet and social networking has allowed people to connect and exchange opinions and information. The representation of such connections through the concept of network, where a node denotes an individual, and an edge denotes the link between two individuals, has primarily been interpreted as positive. Thus relationships have typically expressed collaboration,

*Corresponding author

common interests, membership to the same group, friendship. However, since the primary studies on structural balance theory of Heider,[1] relative to attitude and perception of social organization of individuals, later generalized by Cartwritght and Harary,[2] it has been pointed out that relationships between individuals can be either positive or negative, such as like-dislike, friends-enemies, love-hate, trust-distrust. *Signed networks* are an extension of networks to include the additional information of positive and negative links. Thus positive links denote friendly relations, while negative links represent antagonistic relations. Detecting community structure on these kind of networks is an important research topic since it allows to determine instability inside relationships, and, consequently, to predict changes in group organization.

Approaches to find dense groups of nodes for unsigned networks are mainly based on the optimization of the concept of *modularity*.[3] The extension of such concept to signed networks has been introduced by Gómez *et al.*[4]

As regards signed networks, Doreian and Mrvar[5] were the first that proposed a partitioning method by introducing the concept of *frustration*, which expresses the number of positive ties among different groups and the number of negative links inside the same group.

In this paper a method that combines Genetic Algorithms[6] and a local refinement strategy to detect communities in signed networks is proposed. The method, named *SN-MOGA* (*Signed Networks with MultiObjective Genetic Algorithms*) optimizes the concepts of *modularity*[3] and *frustration*[5] by applying a Multiobjective Genetic Algorithm.[7] The maximization of modularity allows to detect network divisions far from random divisions, while the minimization of frustration guarantees to have as few negative intra-connections and positive inter-connections as possible. The *SN-MOGA* algorithm evolves a population of candidate solutions by trying to obtain the best trade-off between high modularity and low frustration. At the end of the evolutionary process a solution from the Pareto front is chosen, and a local search strategy is performed to improve signed modularity, by moving those nodes having positive connections with nodes of other communities, to neighboring communities, provided that there is an increase in signed modularity.

The idea of modeling community detection in both signed and unsigned networks as a problem of optimizing multiple objectives is not new. Moreover, studies on which kinds of objectives for unsigned networks should be selected to improve the performance of a method, along with the advantages of using multiobjective optimization when objective functions are negatively correlated, have been discussed by Shi *et al.*[8] In this context, the main contributions of our work consist in coupling multiobjective optimization with a local search strategy to improve the solution obtained from the Pareto front. A correlation analysis of the two objective functions employed by the method shows that the two objectives are negatively correlated, thus suitable for multiobjective optimizations, according to the observations reported by Shi *et al.*[8] that negatively correlated objectives lead to better performances compared with single-objective or positively correlated objectives.

An extensive experimentation on synthetic networks and real life networks shows that our multiobjective approach optimizing signed modularity and frustration is capable to divide signed networks in groups of nodes having high accuracy and low edge misclassification. Comparison with other state-of-the-art methods indicates that *SN-MOGA* obtains network partitioning more meaningful and closer to the ground truth division.

The paper is organized as follows. In the next section an overview of existing approaches to community detection in signed networks is given. In Section 3 preliminary definitions are introduced, and the problem is clearly stated. Section 4 presents the algorithm. Section 5 describes the evaluation measures adopted for assessing the method results. Section 6 evaluates the performance of our method on synthetic generated networks for which the ground truth division is known, and a real life network. Moreover, a comparison with existing state-of-the art methods is reported. Section 7 compares *SN-MOGA* with the Particle Swarm Optimization method of Gong *et al.*[9] Section 8 analyzes the running time of the method. Section 9, finally, concludes the paper.

2. Related Work

In this section we give an overview of the main proposals to find communities in signed networks.

Signed networks originate from the studies of Heider[1] on *structural balance theory*. The idea underlying balancing is that if two people i and j belonging to the same group like each other, then their evaluation regarding other people should be consistent, that is if i and j like each other, then they both either dislike or like the same people, and if i and j dislike each other, they disagree in evaluating others. A triad is defined balanced if the product of its edge signs is positive. If all the triads in a network are balanced, the network is balanced. It has been proved that in a balanced network the set of vertices can be divided into two clusters such that positive links are only within clusters, while negative links are between clusters. However, rarely a network has a 2-way partitioning, thus Davis[10] extended the concept of balance to *k-balance*. A network is *k-balanced* if it can be divided into k groups such that, edges within groups are positive and edges between groups are negative. In such a case the network is also said *partitionable* or *clusterable*, while the term *balanced* is generally used for 2-way balance. k-balancing is an important research topic since balancing assures stability, while imbalance generates tension inside a group.

One of the first partitioning approaches to structural balance has been proposed by Doreian and Mrvar.[5] The method randomly divides the nodes in a fixed number k of clusters, and then tries to optimize a criterion function by moving nodes among neighboring partitions. The criterion function they proposed is *frustration* (see next section for a formal definition). The neighbors of each partition are computed, then, either a node is moved to a neighboring cluster, or two nodes are exchanged between two neighboring groups. These neighbors are examined at random, and, if the new

partition has a lower value of frustration, the new solution is accepted. The main drawbacks of the method are that the number of groups must be given as input parameter, and that it disregards the density of links, which is one of the main characteristics exploited in unsigned community detection methods.

More recently, because of the increasing interest in signed networks, several approaches have been proposed. Many of these methods extends concepts used to detect communities in unsigned networks, to take into account the sign of links.

Yang *et al.*[11] proposed an algorithm that uses the concept of random walk and adopts an agent-based heuristic to extract communities. The method starts with an arbitrary node. From this node an agent performs a random walk for a number of steps by visiting one of the neighboring nodes on the base of the transition probability computed from the network connectivity degree. The method, named FEC, is composed of two main phases. The FC (*Find a Community*) phase transforms the adjacency matrix of the graph by applying iterative operations, in order to compute aggregate transition probabilities, and sorts them for each row. The EC (*Extract the sink Community*) phase divides the transformed matrix in two blocks by applying a cutoff criterion. One of the two blocks is identified as a community, called the *sink community*, while the remaining block is recursively processed in the same way. A main problem is the definition of the cutoff value. To this end the authors proposed a variation of the *cut* concept used in spectral clustering[12] that takes into account the sign of edges. FEC needs as input parameter the number l of steps the agent performs before arriving to a destination node. A sensitivity analysis of this parameter shows that when l is greater than a range of values between 10 and 20, the FC phase is insensitive to this parameter.

Approaches that extend the concept of modularity of Newman and Girvan[3] have been proposed by Traag and Bruggeman[13] and Gómez *et al.*[4] The former extended modularity with negative signs and formalized the concept as a Potts model.[14] They defined a Hamiltonian for the positive part and another for the negative one, by extending the approach of Reichardt and Bornholdt.[14] Then minimizing the Hamiltonian is shown to be equivalent to maximizing modularity. To this end the authors modified the simulated annealing approach of Reichardt and Bornholdt,[14] and showed an application of the method to a network of conflicts and alliances between countries. Gómez *et al.*,[4] instead, generalized the concept of modularity to signed networks and proposed to maximize signed modularity to detect communities. They applied the approach to a real network related to retail stores in the city of Lyon, and found that the results they obtained were better when compared with the classification provided by a public institution.

Spectral graph theory is another important concept extensively used to find communities. In this context, Kunegis *et al.*,[15] in order to deal with signed networks, defined the signed Laplacian matrix of a graph, investigated its properties, gave a definition of signed ratio cut, and proposed a spectral approach to find a clustering by minimizing ratio cut. Moreover, they exploited these concepts also for graph visualization and link prediction. Chiang *et al.*[16] proposed a multilevel clustering

algorithm that introduces new k-way objectives and kernels. These objectives are shown to be equivalent to a general weighted kernel k-means objective, thus the optimization of these objectives can be performed by using a kernel k-means like algorithm. Furthermore the authors show that the approach of Kunegis *et al.*,[15] presents some weakness when directly generalizes the signed Laplacian to k-way clustering. Anchuri and Magdon-Ismail[17] proposed a two step spectral approach to detect communities in signed social networks. An input parameter to fix the value of the leading eigen vector, used to assign nodes to communities, must be given. They consider the concepts of frustration[5] and modularity[3] and detect communities by optimizing only one of these two objectives at a time. After that, they try to improve the chosen objective by moving nodes among communities. The authors formalize the problem of minimizing frustration and maximizing modularity to that of maximizing the mathematical form $f(M, s) = s^T Ms$, where $s \in \{-1, +1\}^n$ is an n-dimensional vector. When s is equal to the eigenvector corresponding to the maximum eigenvalue of M, it maximizes $f(M, s)$. The top eigenvector is computed by using the *Power Iteration* method. Since the method finds a partitioning in two communities, it can be extended to a higher number of communities by iteratively dividing communities until the objective cannot be improved any more. Experiments on two real-life signed networks show that when the objective function is modularity, the two step approach obtains the minimum frustration value with respect to modularity maximization without improvement, and k-means approaches. Same results are obtained when minimizing frustration with and without improvement.

A different approach based on simulated annealing has been presented by Bogdanov *et al.*[18] The authors proposed a framework for building signed networks from content generation flow. Validation is performed on two case studies of articles extracted from Wikimedia download site. Since the number k of clusters to find must be provided in input, the authors vary the value of k from 2 to 10, and compute the criterion they optimize to obtain a partitioning. They choose the value of k for which a higher value does not increase the optimization criterion.

One of the main limitations of these approaches is that the number k of clusters must be given as input parameter. Thus some strategy must be introduced to determine k, such as executing the method for a range of k values, and then choosing the k giving the best value of the criterion that the method optimizes.

Recently, however, methods based on evolutionary computation, that automatically determine the number of partitions, have been proposed. Li *et al.*[19] presented and compared two evolutionary algorithms, named *EA-SN* and *CSA-SN*, and two memetic algorithms, named EA_{HC}*-SN* and CSA_{HC}*-SN*. The latter two differ with respect to the formers since they include a hill-climbing strategy. All the algorithms adopt the character string encoding of individuals, i.e. each node is associated with the label of the cluster it belongs to, and use as objective function to optimize the *improved modularity* and the *improved modularity density*. The first one is the modularity extended by Gómez *et al.*[4] to signed networks, while the latter is a

generalization to networks with signs of the modularity density concept, proposed by the same authors for unsigned networks. Experiments on different networks show that the memetic approaches outperform the evolutionary approaches.

Liu *et al.*[20] proposed a multiobjective evolutionary method to find communities in signed networks, named *MEA_s-SN*. The two objectives to optimize are based on the concepts of positive and negative cluster similarity. The authors extend the definition of similarity of Huang *et al.*[21] between two neighboring nodes to signed links, and define the first objective as the positive internal and external similarity of a community structure, while the second objective as the negative internal and external similarity of a community structure. Moreover, they propose a representation of individuals consisting of two components. The first component is a node permutation, the second component denotes the cluster label the node belongs to. In order to determine this label, *MEA_s-SN* performs a community detection method that starts by an empty cluster and adds a node, provided that a criterion, named signed tightness, increases. This approach allows the method to assign a node to multiple communities. The method has been compared with the algorithm FEC of Yang *et al.*,[11] with CSA_{HC}-*SN* of Li *et al.*,[19] and an extension of the Blondel *et al.*[22] method. The authors showed that their approach outperforms the competitors.

A different bio-inspired approach has been proposed by Gong *et al.*[9] They introduced a multiobjective discrete particle swarm optimization algorithm, called *MODPSO*, to solve the network clustering problem by optimizing two objective functions, the kernel k-means and the ratio-cut. Though the method is proposed for unsigned networks, the authors extended the two fitness functions for signed networks, and presented results also on 4 small sized networks, used by Yang *et al.*[11] to evaluate the FEC algorithm.

The method we propose, analogously to Liu *et al.*,[20] is based on multiobjective optimization. However the two approaches are different in many aspects. First of all *MEA_s-SN* uses an individual representation that combines both cluster label and node permutation, *SN-MOGA*, instead, as will be clear in Section 4, adopts the locus-based representation. The objective functions the two algorithms optimize are also different. *MEA_s-SN* adapts the community fitness introduced by Lancichinetti *et al.*[23] to signed networks, while *SN-MOGA* uses signed modularity and frustration. In the experimental result section we compare *SN-MOGA* with *MEA_s-SN* on synthetic networks, and with *MEA_s-SN* and Chiang *et al.*[16] method on the real-life network Wikipedia. Moreover, a comparison with *MODPSO* is also reported on four popular signed networks. We show that *SN-MOGA* is very competitive with respect to these methods.

3. Notation and Definitions

A signed social network can be modeled as a graph $G = (V, E, W)$, where V is the set of n nodes (vertices) and E is the set of m edges. $W : V \times V \rightarrow \{-1, 0, 1\}$

is a function which assigns +1 to edges connecting positively a pair of nodes, −1 to edges that connect negatively a pair of nodes, and 0 if an edge does not exist between the nodes.

Let A denote the weighted adjacency matrix associated with G, i.e. $A_{i,j} = W(i,j)$. The matrix A can be split into two adjacency matrices corresponding to positive and negative edges by setting $A^+_{i,j} = A_{i,j}$ if $A_{i,j} > 0$, zero otherwise, and $A^-_{i,j} = -A_{i,j}$ if $A_{i,j} < 0$, zero otherwise. Thus

$$A = A^+ - A^- \tag{1}$$

Given a node $i \in V$, a^+_i and a^-_i are defined respectively as the positive degree and the negative degree of i.

Now consider a division $C = \{C_1, \ldots, C_k\}$ of the graph G into k communities.

Frustration $F(C)$ of a network partition $C = \{C_1, \ldots, C_k\}$ is defined as the sum of the number of positive edges between nodes belonging to different communities and the number of negative edges between nodes inside the same community.[5]

$$F(C) = \sum_{i,j \in V} \alpha A^-_{i,j} \delta(c_i, c_j) + (1-\alpha) A^+_{i,j} (1 - \delta(c_i, c_j)) \tag{2}$$

where c_i (c_j) is the community of node i (j) and $\delta(c_i, c_j)$ is the Kronecker delta function which takes the value 1 if nodes i and j belong to the same community, 0 otherwise, and $0 \leq \alpha \leq 1$ is a parameter that allows to give a different weight to positive and negative links. In the following we do not differentiate the importance of links, thus we consider frustration without this parameter.

Frustration can be rewritten as:

$$F(C) = \sum_{r=1}^{k} \left[\sum_{i,j \in C_r} A^-_{i,j} + \sum_{i \in C_r, j \notin C_r} A^+_{i,j} \right] \tag{3}$$

now let

$$l^-_r = \sum_{i,j \in C_r} A^-_{i,j} \tag{4}$$

and

$$\gamma^+_r = \sum_{i \in C_r, j \notin C_r} A^+_{i,j} \tag{5}$$

Then frustration can be expressed as:

$$F(C) = \sum_{r=1}^{k} (l^-_r + \gamma^+_r) \tag{6}$$

The concept of *modularity* has been introduced by Newman and Girvan in Ref. 3. Intuitively, it is the difference between the fraction of edges inside a community, and the expected value of the fraction of edges that would be in the network if edges fell at random without regard to community structure. For signed networks

the definition of modularity is modified to take into account the contribution of both positive and negative edges.

Signed modularity can be defined as:[4]

$$Q_S = \frac{1}{2m^+ + 2m^-} \sum_{i,j \in V} \left(A_{i,j} + \frac{a_i^- a_j^-}{2m^-} - \frac{a_i^+ a_j^+}{2m^+} \right) \delta(c_i, c_j) \tag{7}$$

where m^+ and m^- are the number of positive and negative entries in A, respectively. Let $m = m^+ + m^-$. Since the Kronecker function $\delta(c_i, c_j)$ is equal to 1 if the nodes i and j are in the same community, similarly to frustration, the signed modularity can be reformulated as:

$$Q_S = \sum_{r=1}^{k} \left[\sum_{i,j \in C_r} \frac{A_{i,j}}{2m} + \sum_{i \in C_r} \frac{(a_i^-)^2}{2m^-(2m)} - \sum_{i \in C_r} \frac{(a_i^+)^2}{2m^+(2m)} \right] \tag{8}$$

Now let

$$l_r = \sum_{i,j \in C_r} A_{i,j} \tag{9}$$

$$d_r^+ = \sum_{i \in C_r} (a_i^+) \tag{10}$$

$$d_r^- = \sum_{i \in C_r} (a_i^-) \tag{11}$$

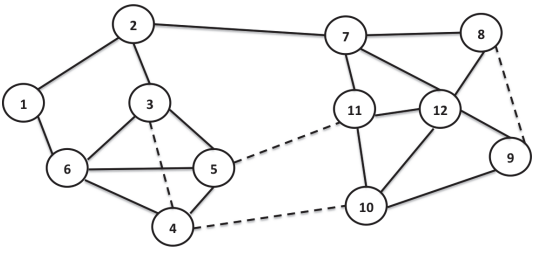
Signed modularity can be rewritten as :

$$Q_S = \sum_{r=1}^{k} \left[\frac{l_r}{2m} + \frac{(d_r^-)^2}{2m^-(2m)} - \frac{(d_r^+)^2}{2m^+(2m)} \right] \tag{12}$$

Note that, if either m^+ or m^- are zero, then Q_S cannot be computed, thus we assume that its value is zero.

Given a graph $G = (V, E, W)$ modeling a signed network, our objective is to find a partitioning of G in k clusters such that: (1) intra-connections are dense and most edges within clusters are positive; (2) inter-connections between clusters are sparse and most of these edges are negative.

Shi *et al.*[8] performed an experimental study aiming at comparing different objective functions in multiobjective community detection methods, in order to choose the objectives leading to better performances. To this end, for a given network, they generated random partitions, and then computed the values of different objective functions. After that, they estimated the Pearson correlation coefficient among the objectives and observed that only negatively correlated objectives are suitable for multiobjective optimization. In fact they (1) provoke opposite effects on the number of communities, (2) avoid an algorithm to converge to trivial solutions, (3) enhance diversity and avoid premature convergence. Positively correlated objectives, instead, are equivalent to single objective methods.



(a)

2	3	5	6	3	1	8	7	12	11	7	7

(b)

Fig. 1. A signed network (a) and the genetic representation (b) corresponding to the ground truth division into the two groups $\{1, 2, 3, 4, 5, 6\}, \{7, 8, 9, 10, 11, 12\}$.

In order to verify whether the two objectives of signed modularity and frustration satisfy the property of being negatively correlated, we performed an analogous study on the synthetic networks described in Section 6. The Pearson correlation coefficient we obtained was -0.3244, showing, thus, a negative correlation. This result strengthens the suitability of these two objectives in discovering meaningful solutions.

4. *SN-MOGA* Description

In this section we give a description of the multiobjective algorithm *SN-MOGA* for signed networks, the representation adopted for partitioning the network, and the variation operators used.

We used the *Nondominated Sorting Genetic Algorithm* (*NSGA-II*) proposed by Srinivas and Deb in Ref. 24 and implemented in the *Global Optimization Toolbox* of MATLAB. *SN-MOGA* has been adapted with a customized population type that suitably represents a partitioning of a network and endowed with the complementary objectives.

Genetic Representation. The algorithm uses the locus-based adjacency representation proposed in Ref. 25. An individual of the population consists of n genes $g_1, \ldots, g_n$ and each gene assumes a value j in the range $\{1, \ldots, n\}$. Each gene corresponds to a node of the graph G modeling the network. If the value of the ith gene is j, it means that there is an edge between nodes i and j, and that both i and j belong to the same cluster. In this representation the number of clusters is determined by the number of connected components contained in an individual. Figure 1(a) shows a signed network of 12 nodes clusterable in the two groups $\{1, 2, 3, 4, 5, 6\}$ and $\{7, 8, 9, 10, 11, 12\}$. Dashed lines correspond to negative links, while solid lines to positive edges. The genotype corresponding to this division is

shown in Fig. 1(b) and it is interpreted as: node 1 is connected with node 2, node 2 with node 3, node 3 with node 5, and so on.

Initialization. A random individual is generated such that if in the ith position there is a value j, then j must be one of the neighbors of i, i.e. the edge (i, j) must exist.

Uniform Crossover. *SN-MOGA* uses a standard uniform crossover operator. First a crossover mask of length n, i.e. the number of nodes, is randomly generated. Each value on the mask is either 0 or 1. An offspring is created by selecting from the first parent the genes where the mask is a 0, and from the second parent the genes where the mask is a 1. Uniform crossover guarantees the maintenance of the effective connections of the nodes in the network in the child individual.

Mutation. Analogously to initialization, fixed a position i, mutation randomly selects one of the neighbors of i and assigns this value to the ith gene.

Fitness Functions. The two objectives to optimize are signed modularity (formula (12)) and frustration (formula (6)).

Solution Selection. Multiobjective optimization techniques do not return a unique solution to a problem, but a set of solutions are found through the use of *Pareto optimality theory*.[26] In this context, since a vector of competing objectives must be simultaneously optimized, the goal is to obtain *Pareto-optimal* solutions, i.e. *nondominated* solutions for which an improvement in one objective requires a degradation of another (*Pareto front*). Thus the *Pareto front* represents the compromise solutions satisfying all the objectives as best as possible. However, a solution, out of the Pareto front, should be selected. In our case we show the results when choosing the solution having the minimum frustration and that having the maximum signed modularity.

The pseudo-code of the algorithm is reported in Fig. 2. *SN-MOGA* starts with a randomly generated population of individuals (step 1) and performs multiobjective optimization for a number of generations (steps 2–4). Then it chooses a solution from the Pareto front (step 5) and tries to improve signed modularity by moving nodes, having positive connections with nodes belonging to other clusters, to neighboring communities (steps 6–8). In the experimental result section we will show that *SN-MOGA* is able to obtain highly accurate partitioning of the signed networks we consider.

Computational Complexity. *SN-MOGA*, as already described, uses the NSGA-II[24] method. In Ref. 27 it has been proved that the run-time complexity of the NSGA-II algorithm is $O(gp\ log^{h-1}\ p)$, where g is the number of generations, p is the population size, and h is the number of objective functions. Since the number h of objectives of *SN-MOGA* is two, its complexity is $O(gp\ log\ p)$. As regards genetic

SN-MOGA Method:

Input: A signed network $\mathcal{SN}$ and the graph $\mathcal{G} = (V, E, W)$ modeling it

Output: A node cluster labeling that partitions $\mathcal{SN}$ in the optimal community structure

```
1   create a population of random individuals whose
    length equals the number N =| V | of nodes of G
2   while not maxGen
3     Perform a multiobjective GA with objectives
        3.1 F(C) (formula (6))
        3.2 Q_S (formula (12))
4   end while
5     choose the solution C = {C_1, ... C_k} of the Pareto front having
      the either maximum signed modularity or minimum frustration value;
6     for each node v_j of a cluster C_i having at least
      a positive link with a node belonging to a cluster C_l
7       Move v_j to C_l provided that signed modularity Q_S augments
8     end for
```

Fig. 2. The pseudo-code of the *SN-MOGA* algorithm.

operators, at each generation, crossover needs $O(n)$ time, mutation $O(1)$ time, while fitness computation is composed of three terms: decoding of an individual in connected components, modularity and frustration computation. Decoding requires $O(n\ log\ n)$ time.[28] To compute modularity and frustration, for each node i its a_i^+ and a_i^- neighbors must be considered, then the time complexity is $O(m)$, where m is the total number of edges. Fitness computation can thus be computed in $O(n\ log\ n) + O(m) + O(m)$ time. The overall complexity of *SN-MOGA* is thus $O((gp\ log\ p) \times (n\ log\ n + m))$.

Before presenting the results, in the next section the measures used to evaluate the method are described.

5. Evaluation Measures

To validate our approach and compare it with other methods, we consider two evaluation measures: the *error*, as defined by Yang *et al.*,[11] useful when no information regarding the community structure is available, and a modified version of the well known information theory concept of *normalized mutual information* (NMI),[29] applicable when the ground-truth division of the network is given.

Error. Yang *et al.*[11] employed the frustration concept to define the error rate of a signed network partitioning C as

$$error(C) = \frac{F(C)}{\sum_i \sum_j \mid A_{i,j} \mid} \times 100\% \tag{13}$$

As pointed out by the authors, this error function considers only the sign of the links, and completely disregards the edge density.

Normalized Mutual Information (NMI). When the ground-truth division of a network is known, a very popular measure to compare community structures, based on information theory,[30] is the *Normalized Mutual Information* (NMI).

The normalized mutual information $\mathrm{NMI}(A, B)$ of two divisions A and B of a network is defined as follows. Let C be the confusion matrix whose element C_{ij} is the number of nodes of community i of the partition A that are also in the community j of the partition B.

$$\mathrm{NMI}(A, B) = \frac{-2\sum_{i=1}^{c_A}\sum_{j=1}^{c_B} C_{ij} log(C_{ij} n / C_{i.} C_{.j})}{\sum_{i=1}^{c_A} C_{i.} log(C_{i.}/n) + \sum_{j=1}^{c_B} C_{.j} log(C_{.j}/n)} \tag{14}$$

where c_A (c_B) is the number of groups in the partition A (B), $C_{i.}$ ($C_{.j}$) is the sum of the elements of C in row i (column j), and n is the number of nodes. The denominator is a normalization factor that limits the range of values in the interval $[0, 1]$. Different types of normalizations have been proposed.[31,32] We adopt the same used by Danon *et al.*[29] for complex networks. If $A = B$, $\mathrm{NMI}(A, B) = 1$, if A and B are completely different, $\mathrm{NMI}(A, B) = 0$.

Weighted Normalized Mutual Information (WNMI). Recently, it has been proved that NMI suffers of the so called *selection bias*,[33] i.e. the leaning to choose clustering solutions having many clusters or with fewer data points when compared with the ground-truth clustering. This provokes an unfair favorable behavior towards those methods that find a high number of clusters, independently from the true effective number. Consider, for instance, the toy example of Fig. 1(a). The division of the network reported in Fig. 3 into the three clusters $\{1, 2, 3, 6, 7\}$, $\{5, 11\}$, $\{4, 8, 9, 10, 12\}$, when compared with the ground truth division, has an NMI value of 0.1866. However, if consider the partitioning constituted by 12 singleton communities the NMI value is 0.4362, which is rather unintuitive. Thus, the importance

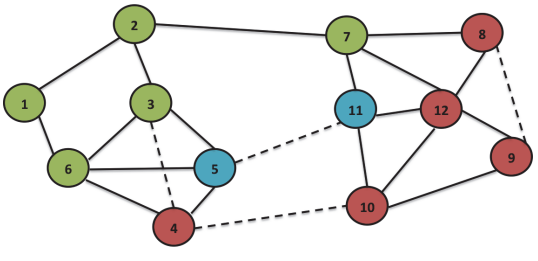


Fig. 3. Division of the network reported in Fig. 1 into the three clusters $\{1, 2, 3, 6, 7\}$, $\{5, 11\}$, $\{4, 8, 9, 10, 12\}$. The NMI value in this case is 0.1866. The NMI value for 12 singleton communities is 0.4362.

of correcting NMI when the number of clusters is high with respect to the data size[34,35,33,36] has been discussed, and modifications proposed.

In this paper we adopted the *adjusted* NMI measure proposed by Amelio and Pizzuti[36] because, as experimentally demonstrated, it is fast to compute, differently from the high computing time required by the measure of Romano *et al.*,[33] and avoids to consider very similar a predicted and the ground truth clustering when the former consists of a too few or too high number of communities with respect to the latter.

Let A and B be the ground-truth division of a network in c_A communities, and the partitioning in c_B communities obtained by a method, respectively. The *weighted* NMI is defined as follows:

$$\text{WNMI} = e^{-\frac{|c_A - c_B|}{c_A}} \times \text{NMI} \tag{15}$$

The exponent of the exponential function is 0 when the predicted number C_B and the true number C_A are the same. In this case, thus, the weighted NMI and NMI values coincide. However, as the difference between c_A and c_B increases, both if either a lower or a higher number c_B of communities is obtained, the value of WNMI proportionally decreases.

In the next section we test our method and compare it with other state-of-the-art approaches by showing both the NMI and WNMI values the methods obtain.

6. Experimental Results

In this section we evaluate the capability of our approach in obtaining meaningful partitions of signed networks. As regards the parameters needed by *SN-MOGA*, in order to set crossover and mutation rate, we executed the algorithm on the synthetic networks described in detail in the next section, by considering values between 0.1 and 0.4 for mutation, and 0.1 until 1 for crossover. Figure 4 shows the NMI and modularity values for the combinations of these values. The figure points out that there are a number of combinations that give high NMI. However, since it is known that high mutation rate could destroy good solutions, while low values do not help in escaping from local optima, we fixed it to 0.2 along with crossover fraction 0.8, which gives both high modularity and NMI values. Moreover, we set elite reproduction 10% of the population size, roulette selection function, population size 100, number of generations 100. These values have already been experimented for community detection in unsigned networks and showed to give good results. The algorithm has been executed 10 times and the average values of error rate, NMI and weighted NMI have been computed, together with standard deviation. For all the experiments, the statistical significance of the results produced has been checked by performing a t-test at the 5% significance level. The p-values returned are very small, between 2.3534e-65 and 1.1120e-30, thus the significance level is very high since the probability that a community could be obtained by chance is very low.

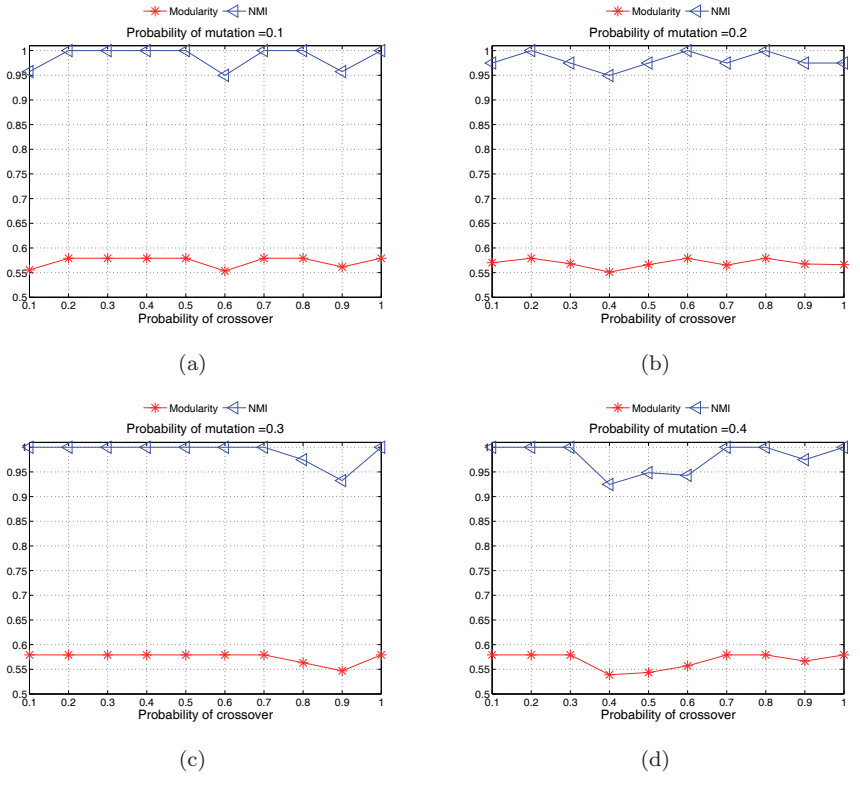
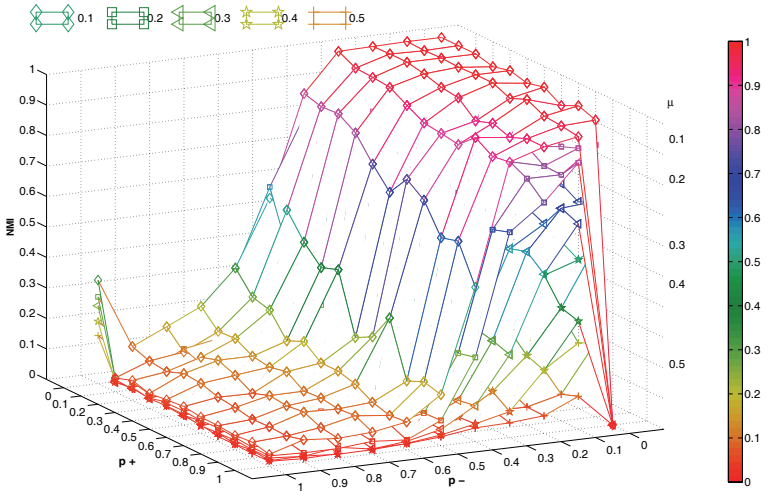


Fig. 4. (Color online) NMI and modularity values for mutation rate varying in the interval $[0.1, 0.2, 0.3, 0.4]$ and crossover fraction from 0.1 to 1.

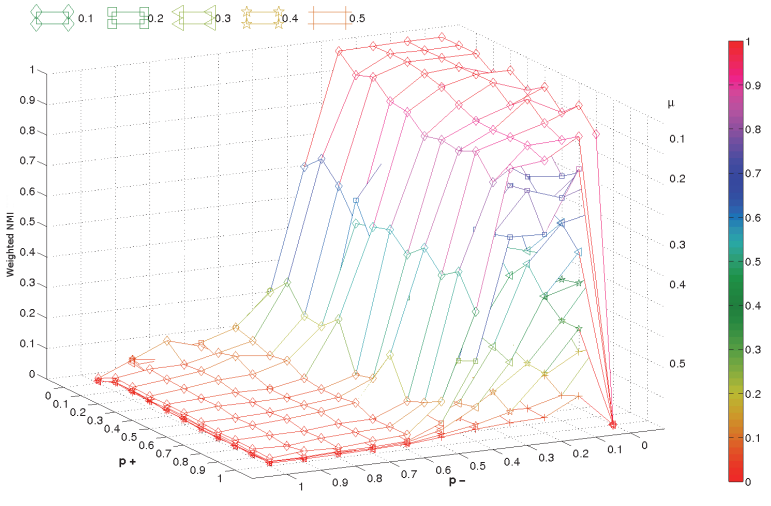
6.1. *Evaluation on synthetic networks*

In this section a more deep study on synthetic networks generated analogously to Yang *et al.*[11] with control parameters that determine the structure of communities, is performed. In particular, we modified the benchmark proposed by Lancichinetti *et al.*,[37] which is an extension of the classical benchmark of Girvan and Newman,[38] by assigning a controlled sign to edges.

The networks consist of 128 nodes divided into four communities of 32 nodes each. Every node has an average degree of 16 and shares a fraction μ of edges with the other nodes of the network, and $1-\mu$ of links with the nodes of its community. μ is called the *mixing parameter*. When $\mu < 0.5$ the neighbors of a node inside its group are more than the neighbors belonging to the other three groups, thus a good algorithm should discover them. We generated 10 different networks for values of μ ranging from 0.1 to 0.5. In order to make the networks signed, analogously to Yang *et al.*,[11] we used two parameters p_-, denoting the probability of negative links appearing within communities, and p_+, denoting the probability of positive links appearing between communities. Thus, for all the combinations of p_- and p_+ values



(a)

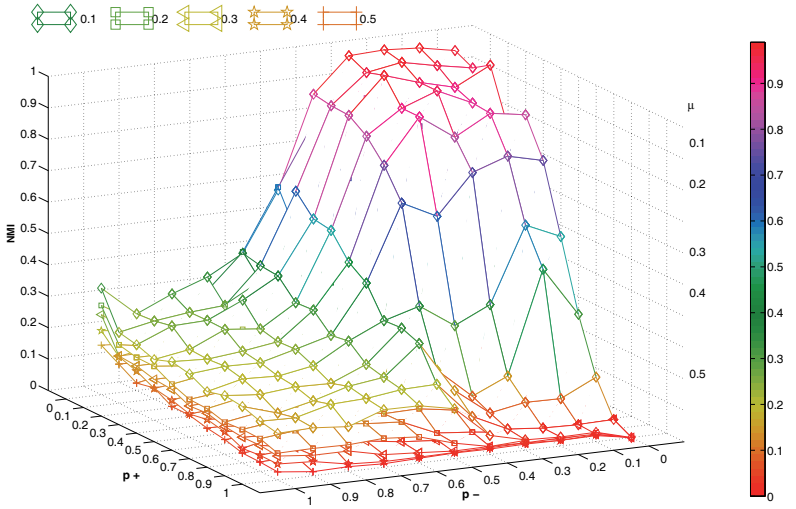


(b)

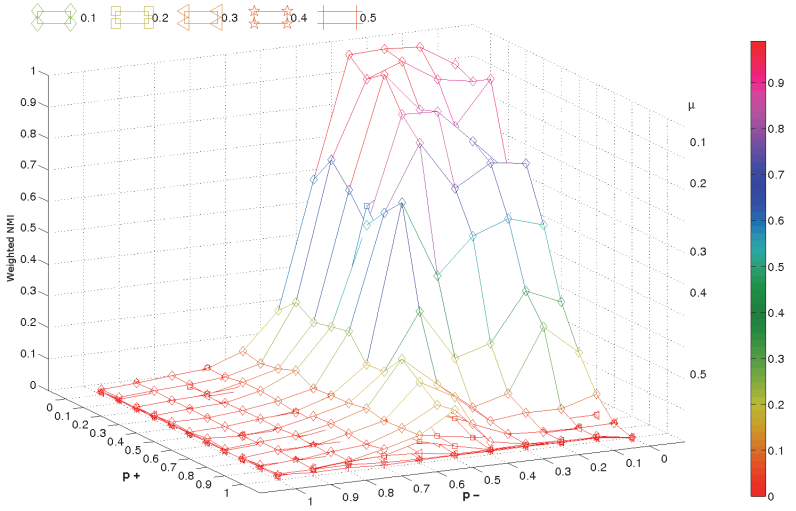
Fig. 5. (Color online) (a) NMI and (b) Weighted NMI corresponding to the maximum modularity values obtained from *SN-MOGA* for all the possible $p+$ and $p-$ values at different values of the μ parameter.

ranging in the interval $[0, 0.1, \ldots, 1]$, we randomly assigned a negative sign to edges inside a community with probability p_-, and a positive sign to edges between two different communities with probability p_+.

Figures 5(a) and (b) depict the NMI and WNMI values obtained by running *SN-MOGA* for all the combinations of parameters $\mu = [0.1, \ldots, 0.5]$, $p_- = [0, 0.1, \ldots, 1]$, and $p_+ = [0, 0.1, \ldots, 1]$ when selecting from the Pareto front the community



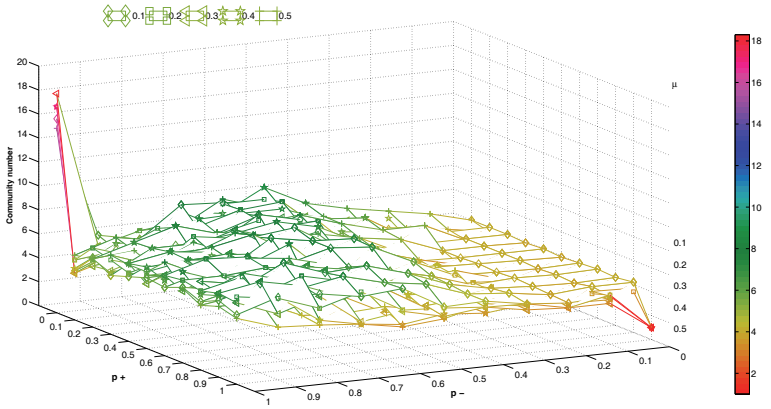
(a)



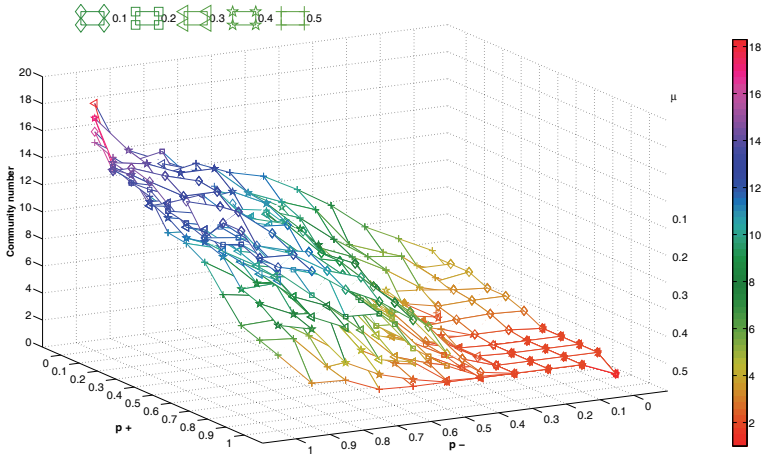
(b)

Fig. 6. (Color online) (a) NMI and (b) Weighted NMI corresponding to the minimum frustration values obtained from *SN-MOGA* for all the possible p_+ and p_- values at different values of the μ parameter.

structure having the highest modularity value. The figures point out that the NMI and WNMI values do not sensibly differ, meaning that the number of communities found by *SN-MOGA* is close to the true number, which is 4. From the figures it can be observed that, fixed a μ value, the method is not sensitive to the increase of the number of positive edges between communities. As regards p_-, until



(a)



(b)

Fig. 7. (Color online) Average number of clusters obtained by *SN-MOGA* for p_+ and p_- varying in the interval $[0, 1]$, at different values of the μ parameter, when (a) maximum modularity and (b) minimum frustration are selected from the Pareto front.

$p_- \leq 0.4$, *SN-MOGA* maintains high values, however it is negatively influenced by the augmentation above 0.4 of negative links within a community.

A similar behavior can be observed in Figs. 6(a) and (b), where the solutions having the minimum frustration are now selected from the Pareto front. In this case the NMI and WNMI values obtained are lower with respect to the previous case, moreover *SN-MOGA* is again insensitive to the variation of negative edges for $p_- \leq 0.4$.

Figures 7(a) and (b) show the average number of clusters obtained by *SN-MOGA* for p_+ and p_- varying in the interval $[0, 1]$, when maximum modularity and

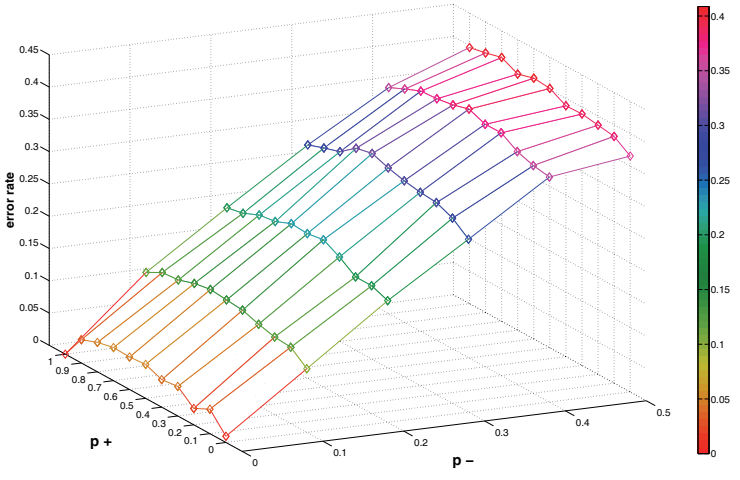


Fig. 8. (Color online) Error rate values obtained by *SN-MOGA* for p_+ and p_- varying in the interval $[0, 1]$ and $\mu = 0.1$.

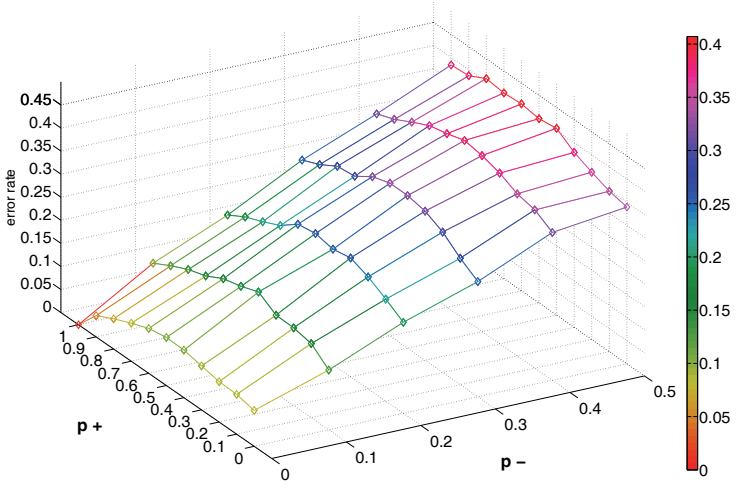


Fig. 9. (Color online) Error rate values obtained by *SN-MOGA* for p_+ and p_- varying in the interval $[0, 1]$ and $\mu = 0.2$.

minimum frustration, respectively, are selected from the Pareto front. It can be observed that, in the former case, the number of communities is 4 for a good range of p_- and p_+ combinations, and it almost always is not greater than 8. Solutions with minimum frustration divide the networks in more communities, particularly when p_- increases. This implies a decrease in NMI and WNMI values.

Figures 8–12 show the error rate obtained by *SN-MOGA*, when minimum frustration solutions are chosen from the Pareto front, for increasing values of the

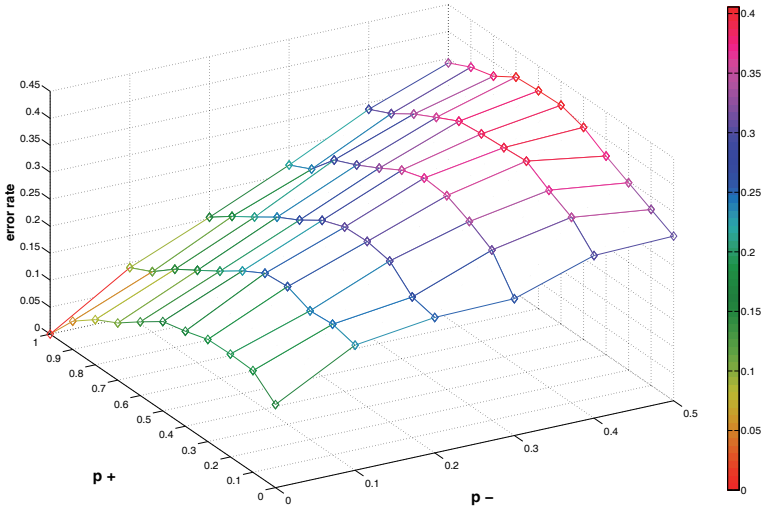


Fig. 10. (Color online) Error rate values obtained by *SN-MOGA* for p_+ and p_- varying in the interval $[0, 1]$ and $\mu = 0.3$.

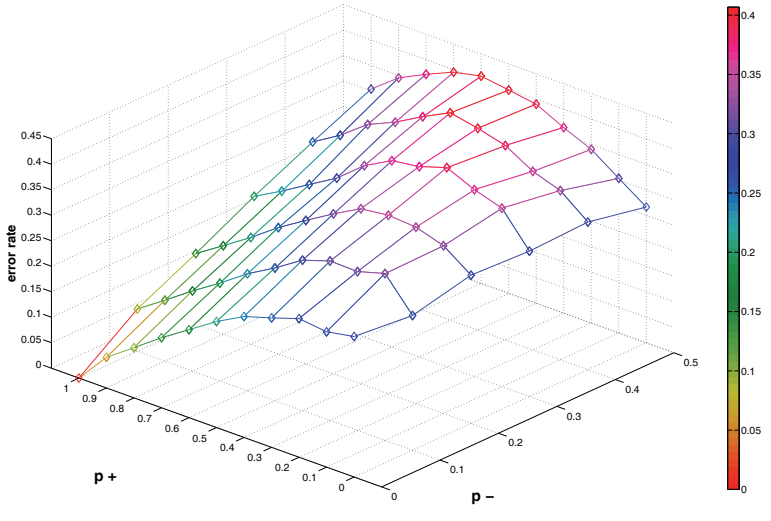


Fig. 11. (Color online) Error rate values obtained by *SN-MOGA* for p_+ and p_- varying in the interval $[0, 1]$ and $\mu = 0.4$.

mixing parameter μ, and combinations of p_- and p_+ values. The figures point out that the error rate is insensitive to increasing values of p_+, i.e. the augmentation of positive links between different communities does not provoke abrupt changes in the frustration value. However, the error rate increases as the percentage p_- of negative links inside the same community augments.

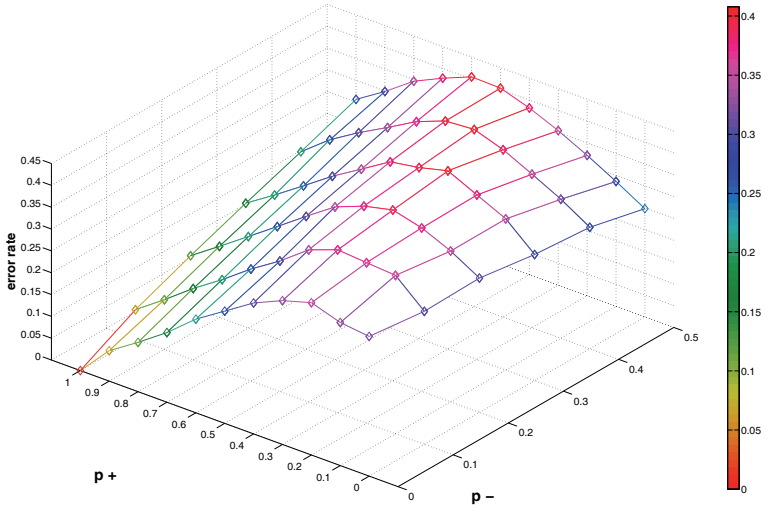
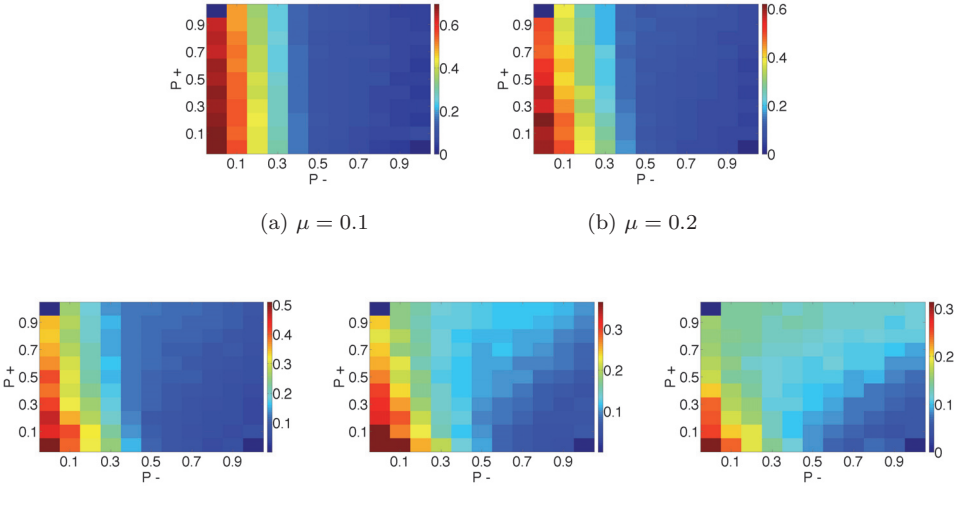


Fig. 12. (Color online) Error rate values obtained by *SN-MOGA* for p_+ and p_- varying in the interval $[0, 1]$ and $\mu = 0.5$.



(a) $\mu = 0.1$ (b) $\mu = 0.2$

(c) $\mu = 0.3$ (d) $\mu = 0.4$ (e) $\mu = 0.5$

Fig. 13. (Color online) Signed modularity values obtained by *SN-MOGA* for p_+ and p_- varying in the interval $[0, 1]$ and $\mu = 0.1, \ldots, 0.5$.

Finally, Fig. 13 depicts the signed modularity values obtained by *SN-MOGA* for p_+ and p_- varying in the interval [0, 1] and $\mu = 0.1, \ldots, 0.5$. It is worth observing that modularity values are high for $p_- \leq 0.4$, analogously to the NMI and WNMI values.

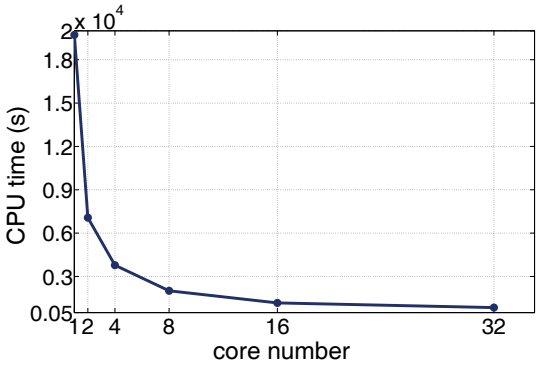


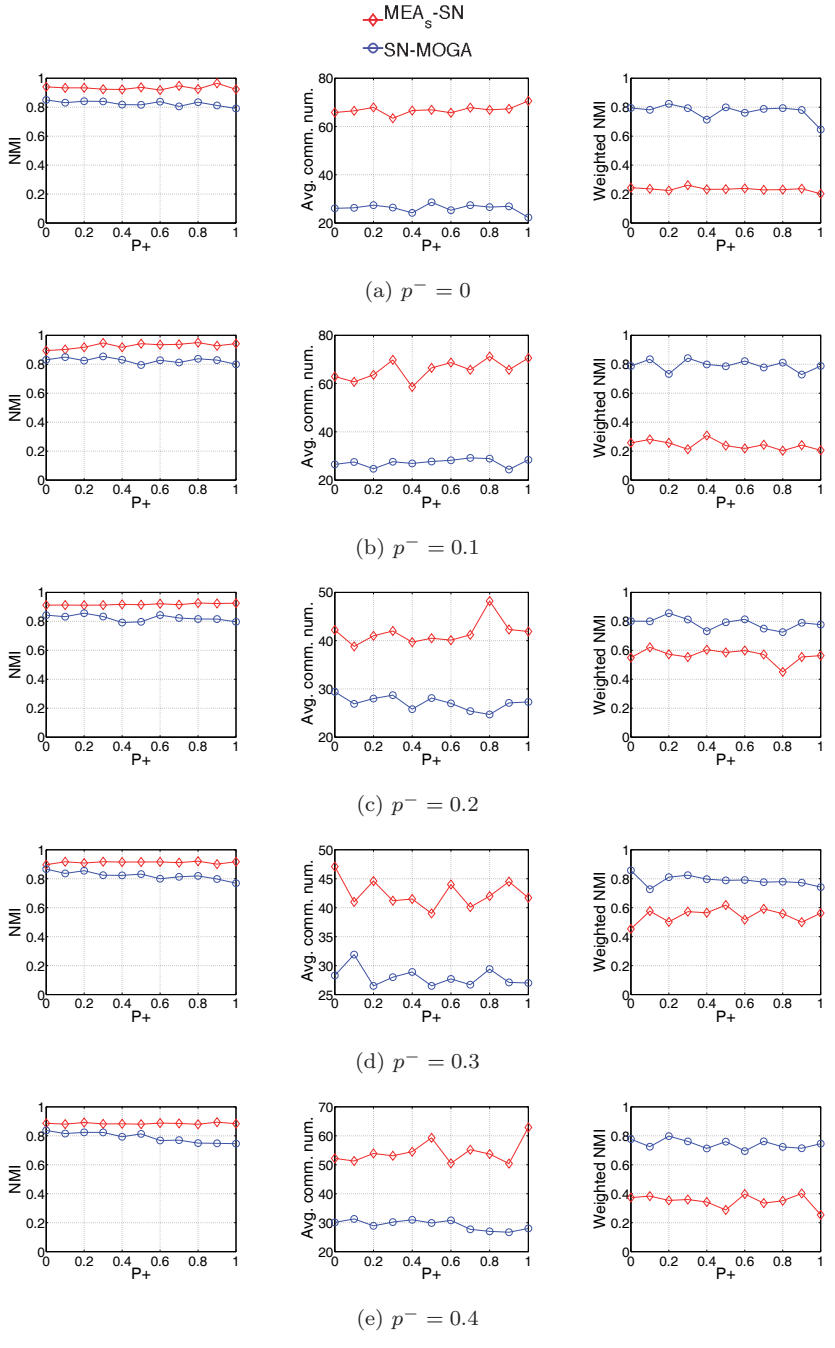
Fig. 14. CPU time required by *SN-MOGA* when the number of processors doubles from 1 until 32.

6.2. *Comparison with MEA_s-SN on synthetic networks*

In this section we compare *SN-MOGA* with the method proposed by Liu *et al.*[20] *MEA_s-SN* is one of the most recent evolutionary based proposals for signed networks, and it has been shown to outperform state-of-the-art methods. In order to compare the two methods, we generated an LFR benchmark, as proposed by Lancichinetti *et al.*[37] constituted by 1000 nodes, average node degree 20, maximum node degree 50, exponent of degree distribution -2, community size distribution -1, mixing parameter μ varying as $0 \leq \mu \leq 0.5$. Also for this benchmark, in order to obtain signed networks, for all the combinations of p_- and p_+ values ranging in the interval $[0, 0.1, \ldots, 1]$, we randomly assigned a negative sign to edges inside a community with probability p_-, and a positive sign to edges between two different communities with probability p_+. A benchmark with analogous characteristics has been used by Liu *et al.* to evaluate their method. We executed *MEA_s-SN* with the parameters suggested by the authors, i.e. number of generations 100, population size 100, crossover fraction 0.8, mutation rate 0.2, and then selected from the final population the solution having the maximum signed modularity. As regards *SN-MOGA* we fixed the same parameters of *MEA_s-SN*.

Figures 15–24 show the average values of NMI, number of communities obtained by the methods, and Weighted NMI for p_- and p_+ ranging in the interval $[0, 1]$. For each experiment, the true number of communities is also reported. In particular, when $\mu = 0.1, 0.2, 0.3, 0.4, 0.5$ the corresponding average ground truth number of communities is 28, 28, 33, 32, and 32, respectively.

From Figs. 15–24 we can observe that *MEA_s-SN* has the tendency to generate a number of communities much higher than the true numbers. For instance, when $\mu = 0.1$ and $p_- \leq 0.4$ (corresponding to the first 5 rows of Fig. 15) it partitions the networks in a number of communities between 40 and 70, depending on p_+. When $p_- \geq 0.5$, this number sensibly increases and can reach 300. The corresponding NMI is, in these cases, unreasonably high, always above 0.8, except for $p_- = 1$.



(a) $p^- = 0$

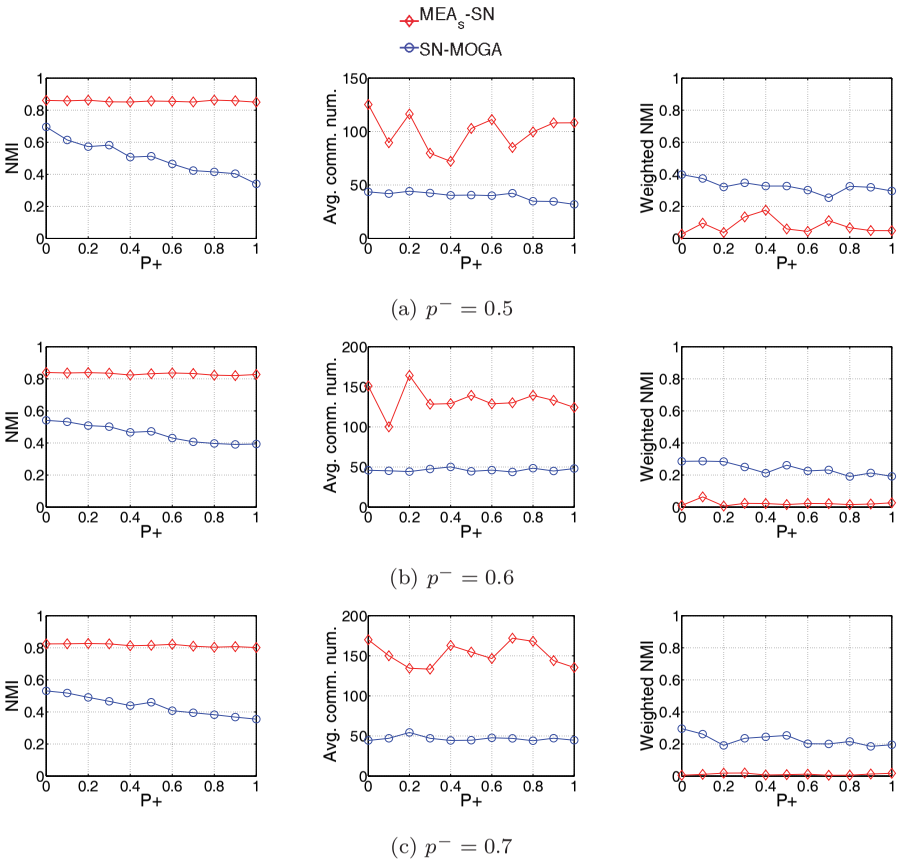
(b) $p^- = 0.1$

(c) $p^- = 0.2$

(d) $p^- = 0.3$

(e) $p^- = 0.4$

Fig. 15. (Color online) NMI, number of communities, and weighted NMI for all the combinations of $p^+ = \{0, \ldots, 1\}$ and $p^- = \{0, \ldots, 1\}$, when $\mu = 0.1$. The ground truth number of communities is 28. Each row corresponds to a p^- value, starting from $p^- = 0$ on the first row, and $p^- = 0.4$ on the last row.

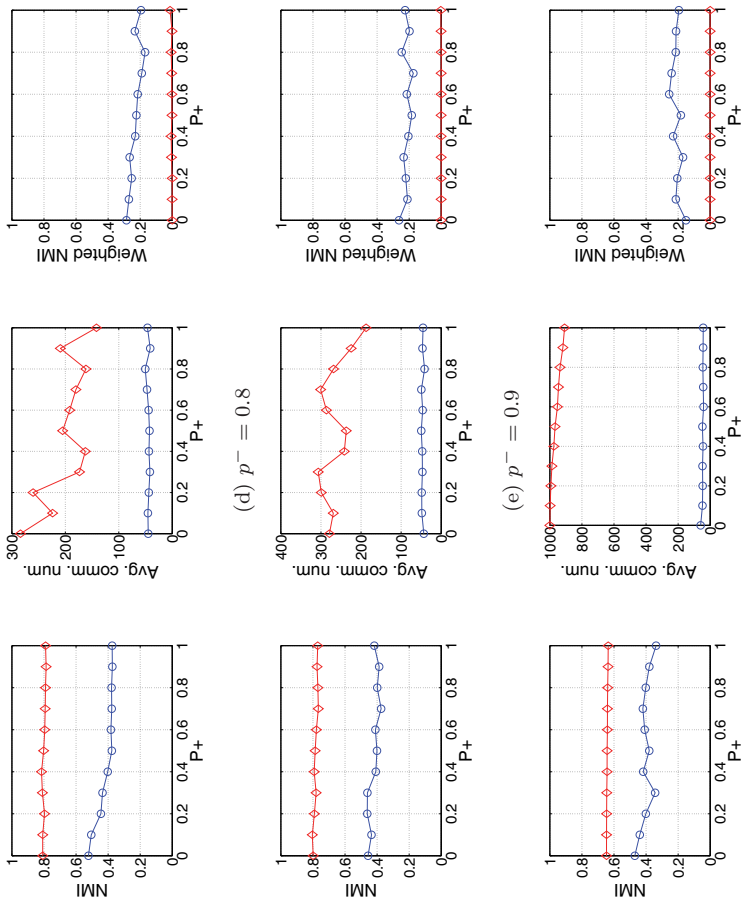


(a) $p^- = 0.5$

(b) $p^- = 0.6$

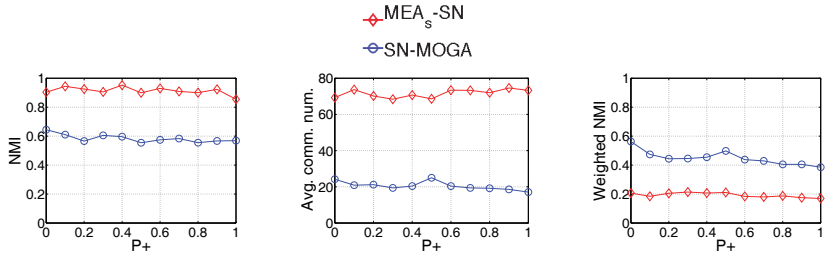
(c) $p^- = 0.7$

Fig. 16. (Color online) NMI, number of communities, and weighted NMI for all the combinations of $p^+ = \{0, \ldots, 1\}$ and $p^- = \{0, \ldots, 1\}$, when $\mu = 0.1$. The ground truth number of communities is 28. Each row corresponds to a p^- value, starting from $p^- = 0.5$ on the first row, and $p^- = 1$ on the last row.

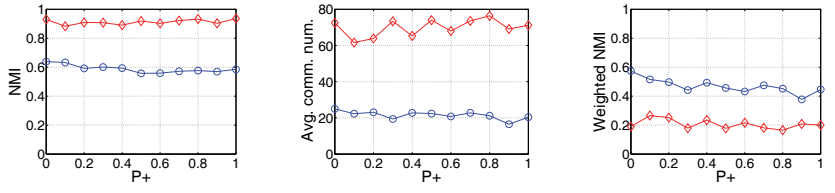


(d) $p^- = 0.8$ (e) $p^- = 0.9$ (f) $p^- = 1$

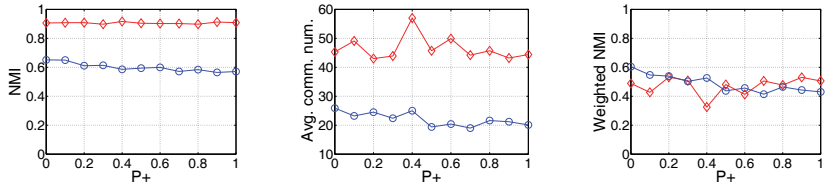
Fig. 16. (*Continued*)



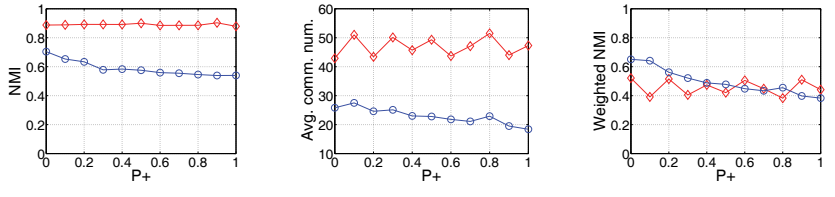
(a) $p^- = 0$



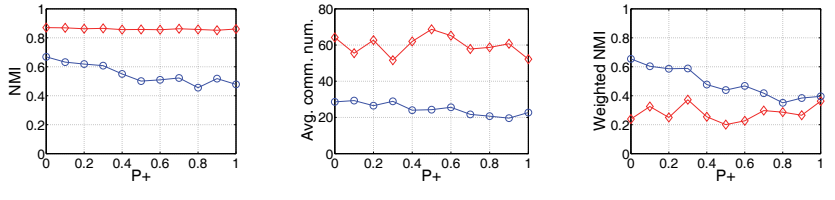
(b) $p^- = 0.1$



(c) $p^- = 0.2$

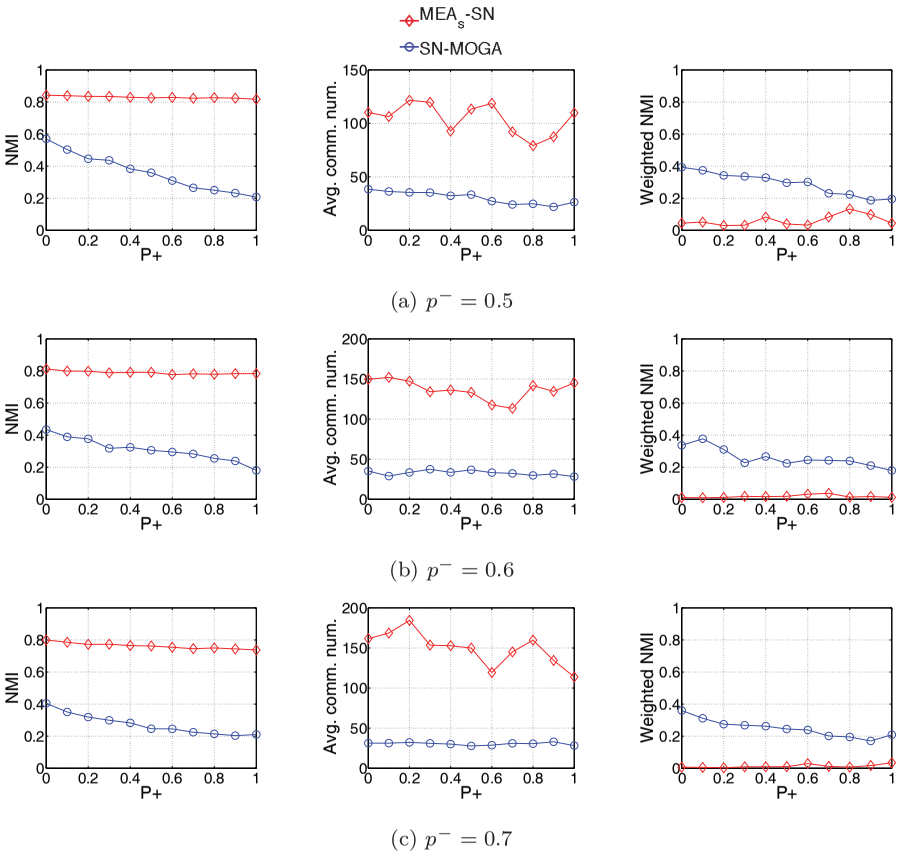


(d) $p^- = 0.3$



(e) $p^- = 0.4$

Fig. 17. (Color online) NMI, number of communities, and weighted NMI for all the combinations of $p^+ = \{0, \ldots, 1\}$ and $p^- = \{0, \ldots, 1\}$, when $\mu = 0.2$. The ground truth number of communities is 28. Each row corresponds to a p^- value, starting from $p^- = 0$ on the first row, and $p^- = 0.4$ on the last row.



(a) $p^- = 0.5$

(b) $p^- = 0.6$

(c) $p^- = 0.7$

Fig. 18. (Color online) NMI, number of communities, and weighted NMI for all the combinations of $p^+ = \{0, \ldots, 1\}$ and $p^- = \{0, \ldots, 1\}$, when $\mu = 0.2$. The ground truth number of communities is 28. Each row corresponds to a p^- value, starting from $p^- = 0.5$ on the first row, and $p^- = 1$ on the last row.

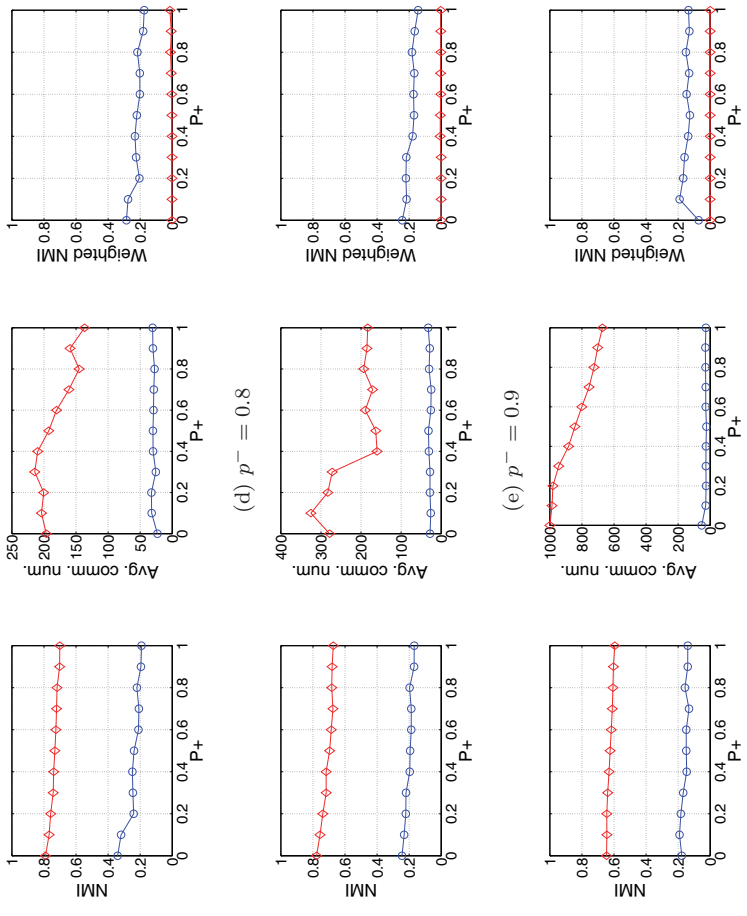
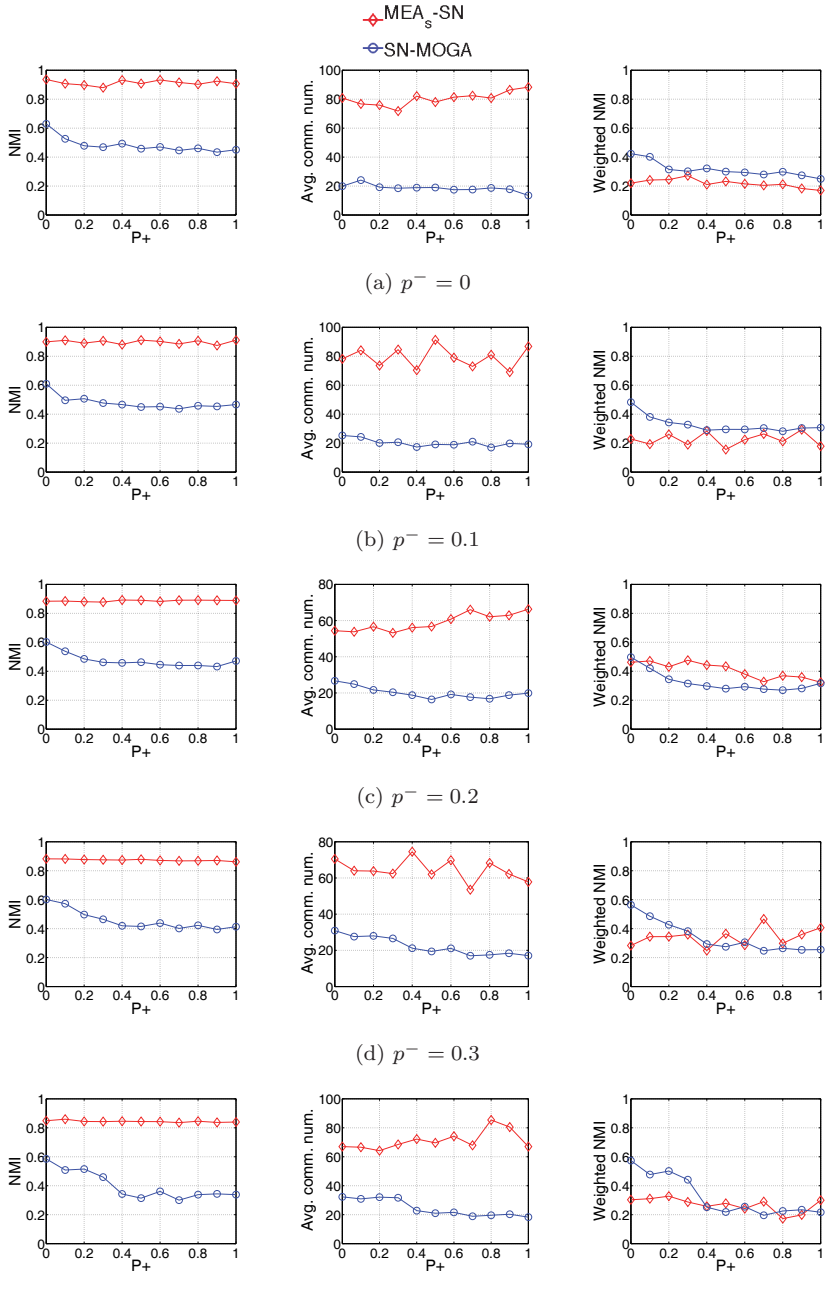


Fig. 18. (*Continued*)



(a) $p^- = 0$

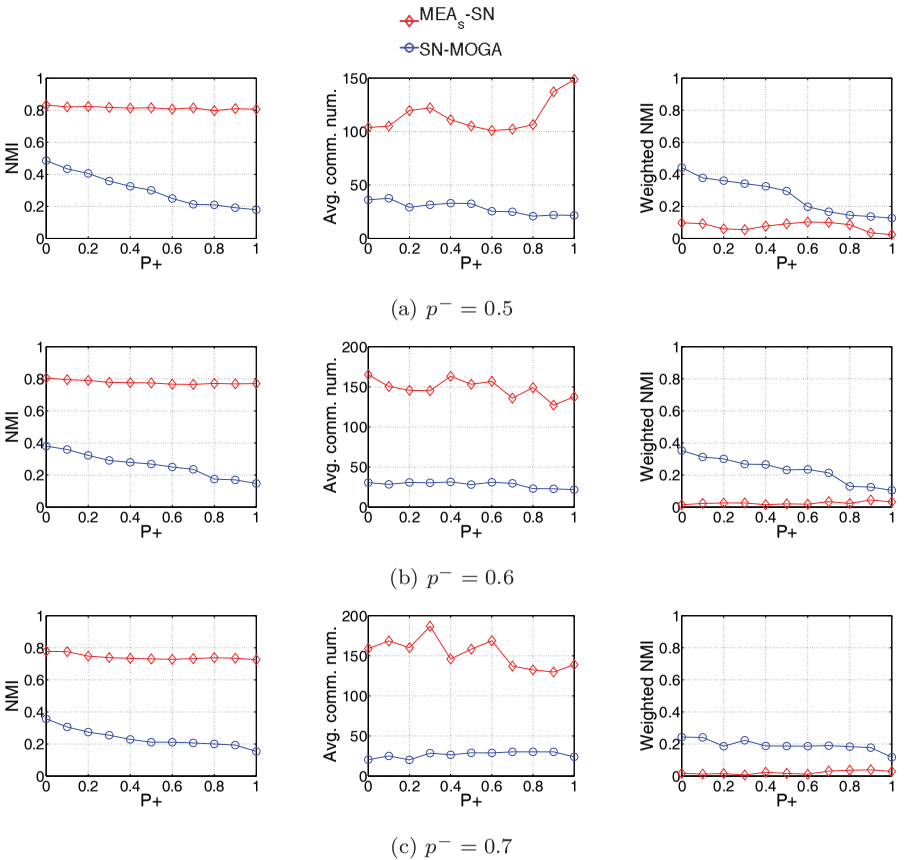
(b) $p^- = 0.1$

(c) $p^- = 0.2$

(d) $p^- = 0.3$

(e) $p^- = 0.4$

Fig. 19. (Color online) NMI, number of communities, and weighted NMI for all the combinations of $p^+ = \{0, \ldots, 1\}$ and $p^- = \{0, \ldots, 1\}$, when $\mu = 0.3$. The ground truth number of communities is 33. Each row corresponds to a p^- value, starting from $p^- = 0$ on the first row, and $p^- = 0.4$ on the last row.

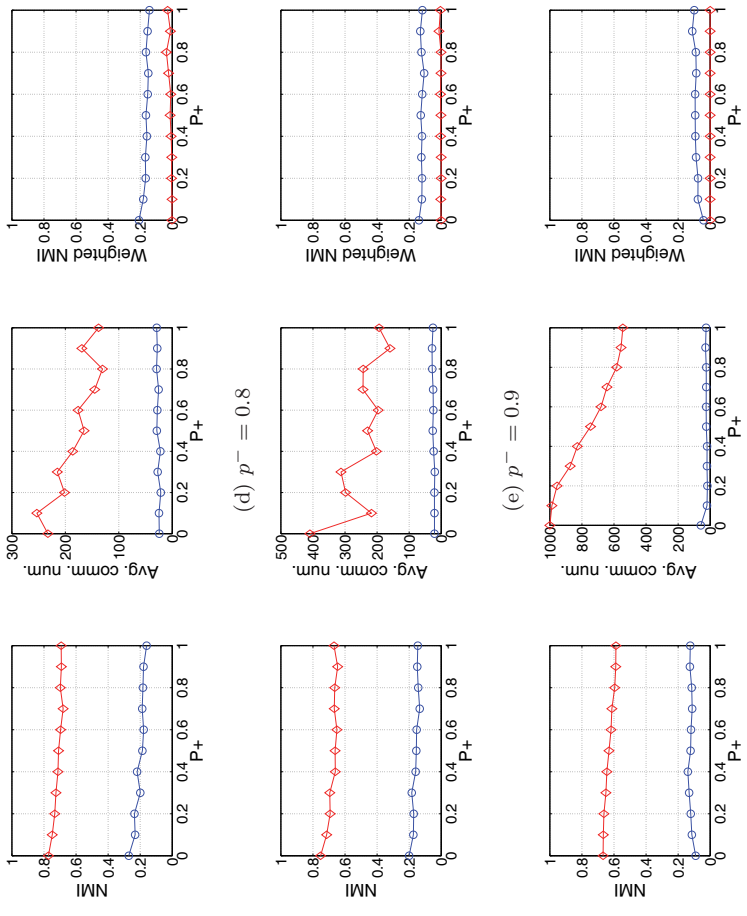


(a) $p^- = 0.5$

(b) $p^- = 0.6$

(c) $p^- = 0.7$

Fig. 20. (Color online) NMI, number of communities, and weighted NMI for all the combinations of $p^+ = \{0, \ldots, 1\}$ and $p^- = \{0, \ldots, 1\}$, when $\mu = 0.3$. The ground truth number of communities is 33. Each row corresponds to a p^- value, starting from $p^- = 0.5$ on the first row, and $p^- = 1$ on the last row.

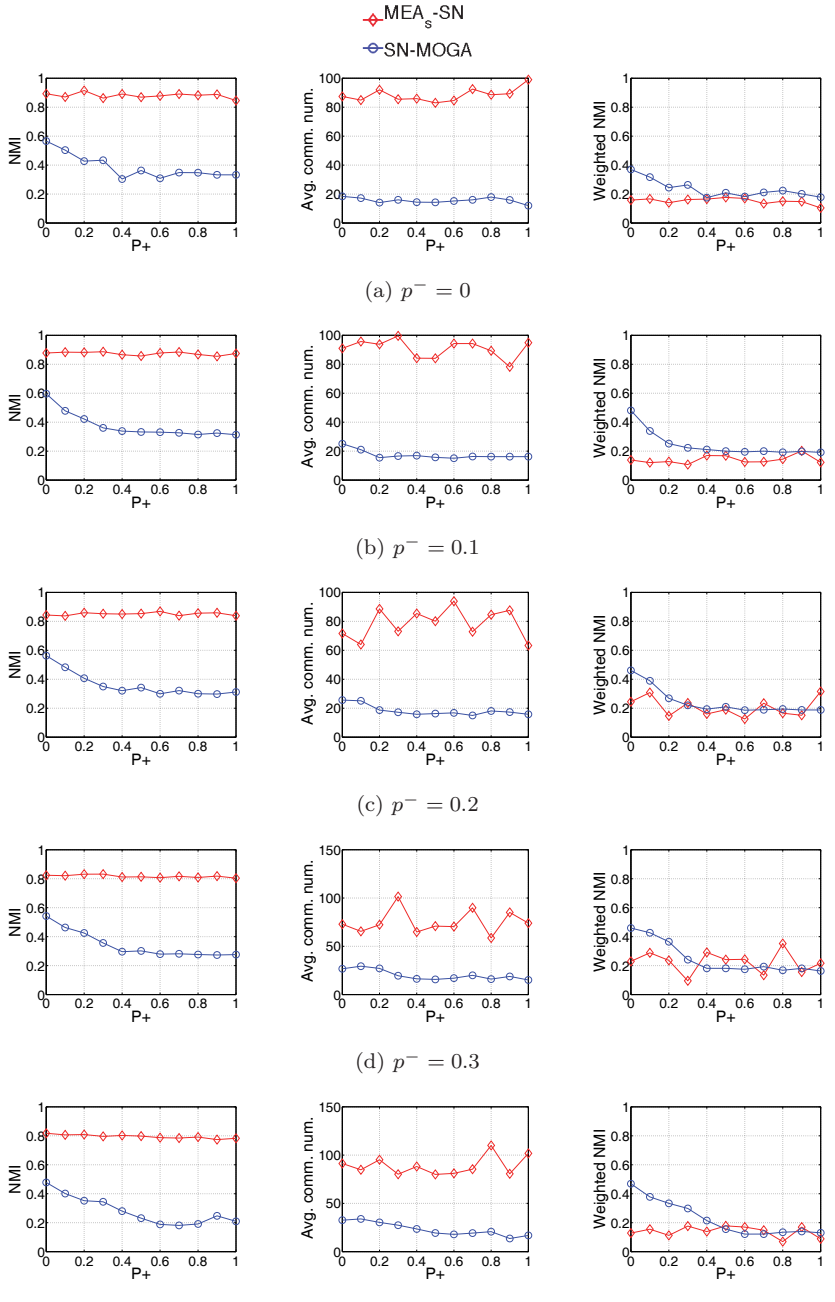


(d) $p^- = 0.8$

(e) $p^- = 0.9$

(f) $p^- = 1$

Fig. 20. *(Continued)*



(a) $p^- = 0$

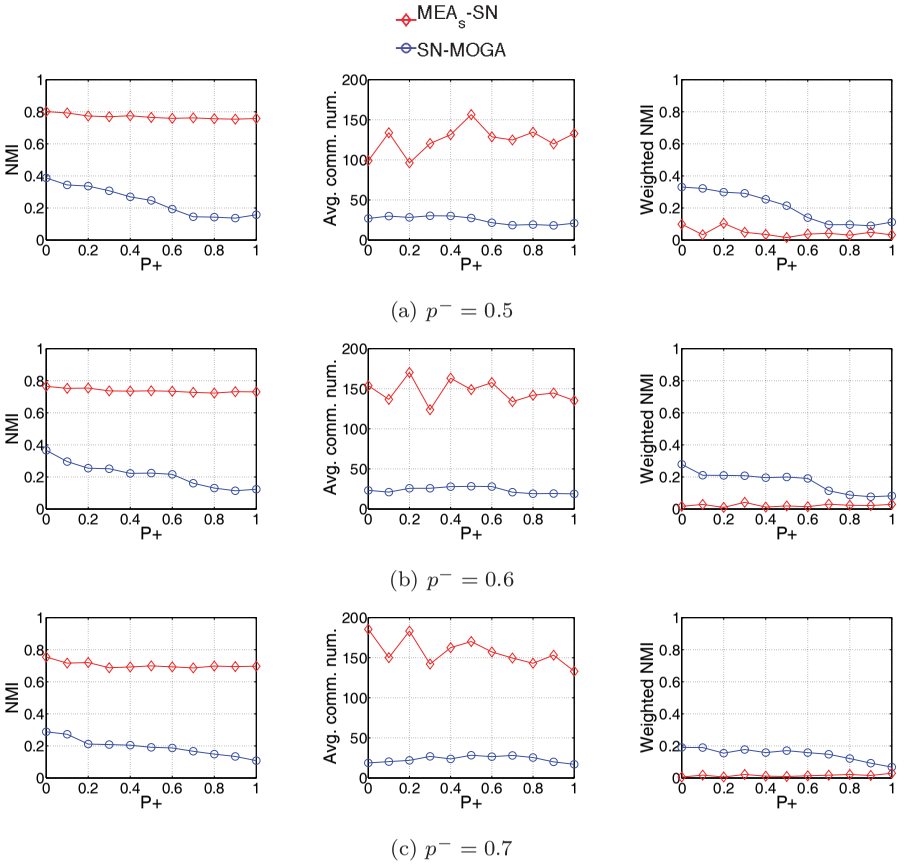
(b) $p^- = 0.1$

(c) $p^- = 0.2$

(d) $p^- = 0.3$

(e) $p^- = 0.4$

Fig. 21. (Color online) NMI, number of communities, and weighted NMI for all the combinations of $p^+ = \{0, \ldots, 1\}$ and $p^- = \{0, \ldots, 1\}$, when $\mu = 0.4$. The ground truth number of communities is 32. Each row corresponds to a p^- value, starting from $p^- = 0$ on the first row, and $p^- = 0.4$ on the last row.

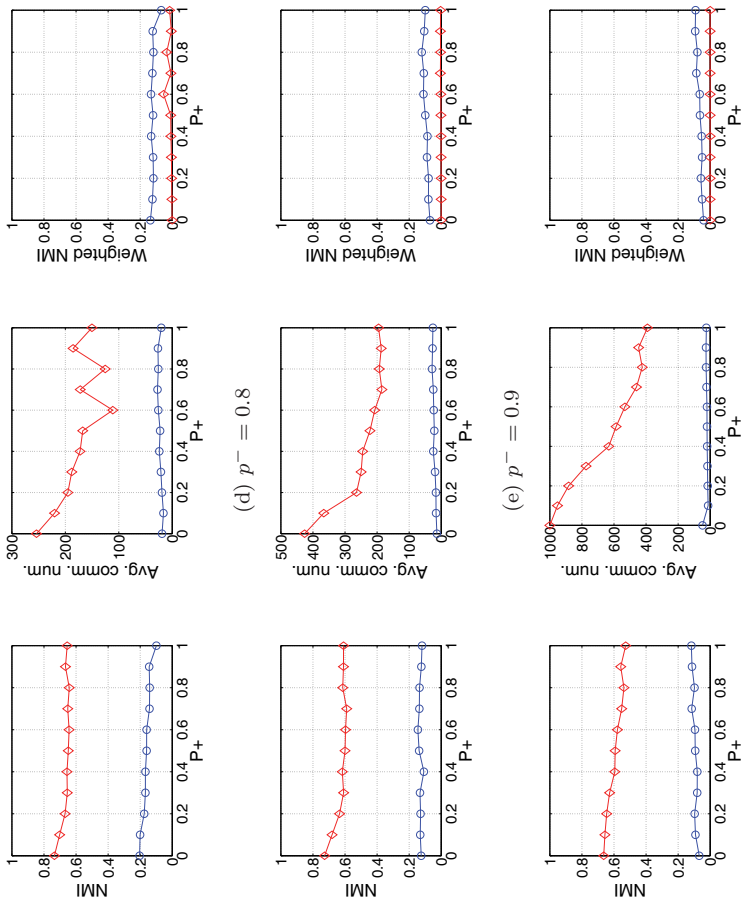


(a) $p^- = 0.5$

(b) $p^- = 0.6$

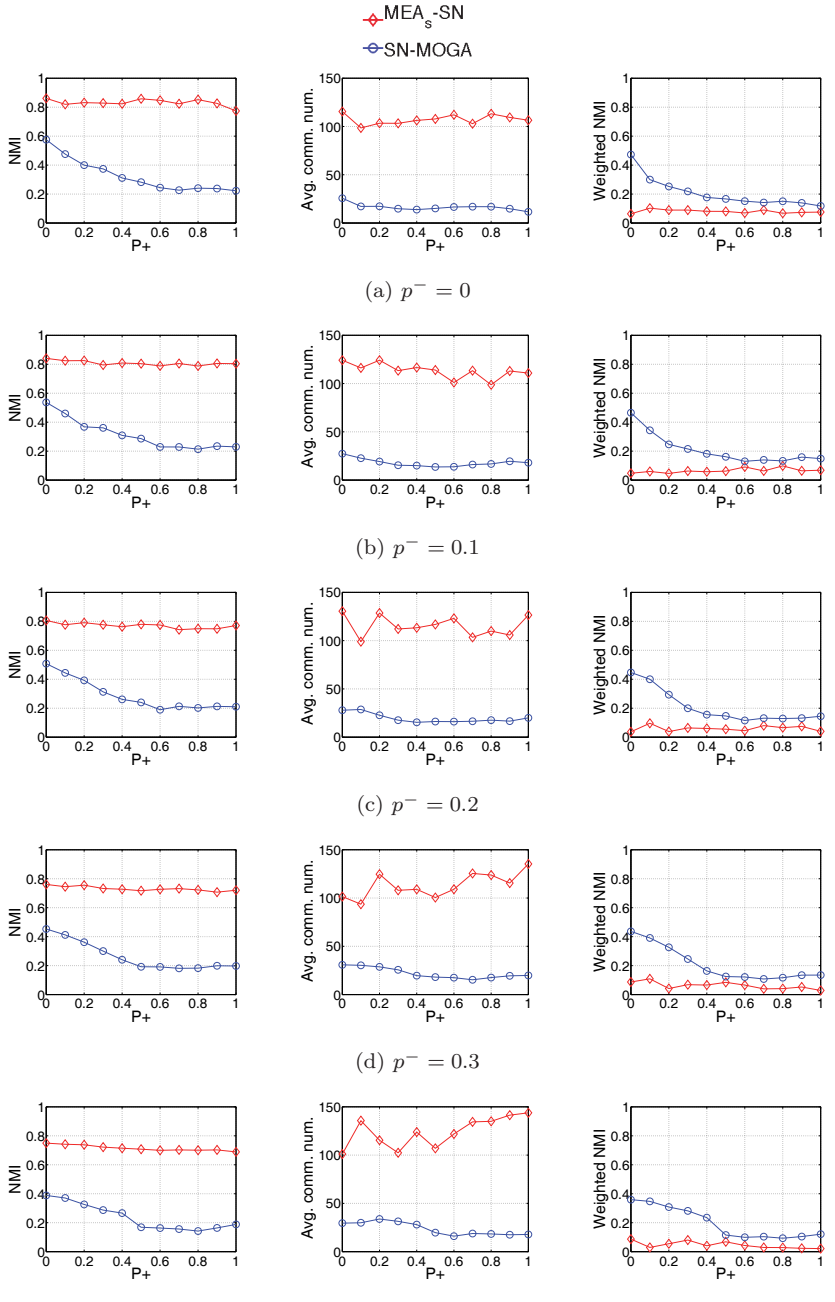
(c) $p^- = 0.7$

Fig. 22. (Color online) NMI, number of communities, and weighted NMI for all the combinations of $p^+ = \{0, \ldots, 1\}$ and $p^- = \{0, \ldots, 1\}$, when $\mu = 0.4$. The ground truth number of communities is 32. Each row corresponds to a p^- value, starting from $p^- = 0.5$ on the first row, and $p^- = 1$ on the last row.



(d) $p^- = 0.8$ (e) $p^- = 0.9$ (f) $p^- = 1$

Fig. 22. *(Continued)*



(a) $p^- = 0$

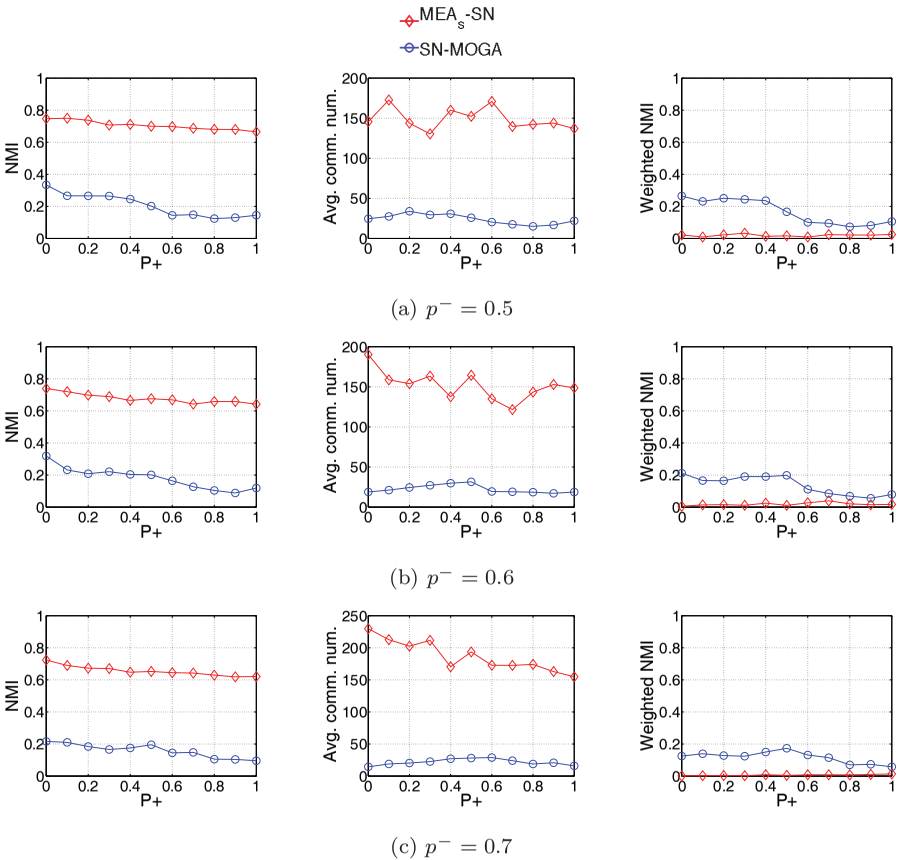
(b) $p^- = 0.1$

(c) $p^- = 0.2$

(d) $p^- = 0.3$

(e) $p^- = 0.4$

Fig. 23. (Color online) NMI, number of communities, and weighted NMI for all the combinations of $p^+ = \{0, \ldots, 1\}$ and $p^- = \{0, \ldots, 1\}$, when $\mu = 0.5$. The ground truth number of communities is 32. Each row corresponds to a p^- value, starting from $p^- = 0$ on the first row, and $p^- = 0.4$ on the last row.

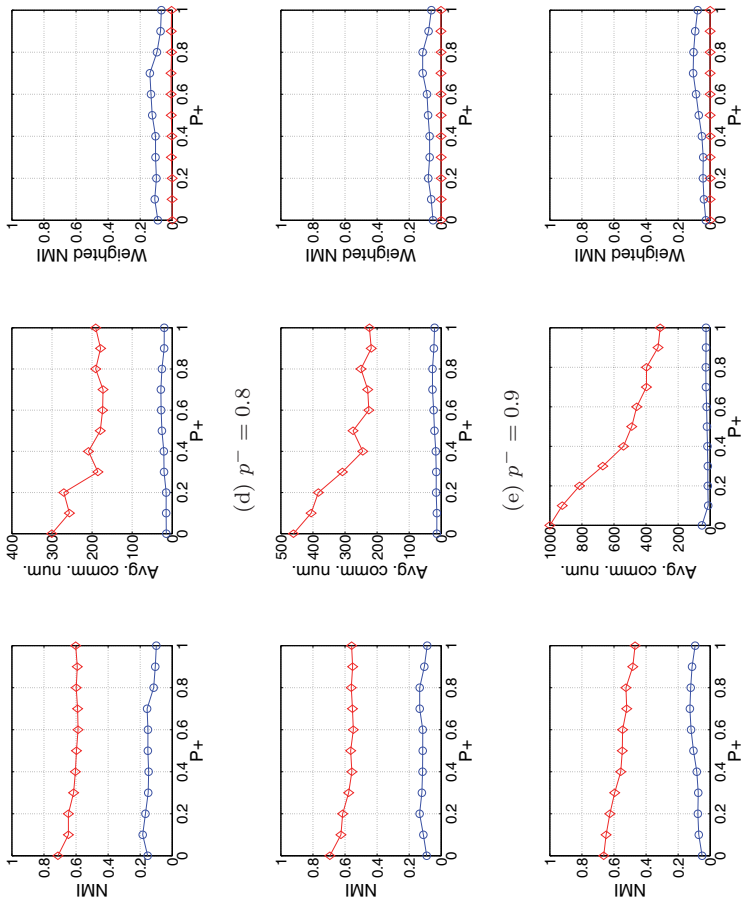


(a) $p^- = 0.5$

(b) $p^- = 0.6$

(c) $p^- = 0.7$

Fig. 24. (Color online) NMI, number of communities, and weighted NMI for all the combinations of $p^+ = \{0, \ldots, 1\}$ and $p^- = \{0, \ldots, 1\}$, when $\mu = 0.5$. The ground truth number of communities is 32. Each row corresponds to a p^- value, starting from $p^- = 0.5$ on the first row, and $p^- = 1$ on the last row.



(d) $p^- = 0.8$ (e) $p^- = 0.9$ (f) $p^- = 1$

Fig. 24. *(Continued)*

Table 1. Comparison between *SN-MOGA* and *MEA_s-SN* on a network of 10000 nodes.

Algorithm	NMI	WNMI	NC	NS
SN-MOGA $\min F(C)$	0.6766	0.5376	123	0
SN-MOGA $\max Q_S$	0.6766	0.5376	123	0
MEA_s-SN Liu *et al.*[20]	0.9916	0.3797	196	64

In such a case the algorithm is not able to find any community structure, since it obtains almost 1000 groups, which is the number of nodes, i.e it finds singleton communities. Nevertheless, the NMI is around 0.6. As μ increases, the number of communities obtained by *MEA_s-SN* increases too, while *SN-MOGA* is stable, finding, on average, a number of communities close to the number of the ground-truth division. Thus, while the WNMI values of *MEA_s-SN* are drastically lower than the corresponding NMI values, the differences between the NMI and WNMI values that *SN-MOGA* obtains are minimal. The behavior of *MEA_s-SN* is exacerbated as μ increases. On the contrary, *SN-MOGA* continues to be rather stable as regards the average number of communities it obtains, with close NMI and WNMI values, proportionally decreasing, as expected, when μ augments.

The figures also point out that the NMI value obtained by *MEA_s-SN* is above 0.8 in almost all the experiments. Thus *MEA_s-SN* outperforms *SN-MOGA*, even if it splits the network in many groups of small size, often singleton. When computing the weighted NMI, however, the values obtained by *SN-MOGA* slightly diminish with respect to NMI, and are always higher than that obtained by *MEA_s-SN*. In fact, the WNMI values of *MEA_s-SN* drastically reduce, due to the too high number of communities it obtains. This experiment highlights the characteristic of the weighted NMI to better discriminate solutions far from the true network division, by assigning them a lower and fairer value.

It is worth pointing out that a correlation analysis of the two objectives employed by *MEA_s-SN* revealed a positive Pearson correlation value of 0.0517. According to the observation of Shi *et al.*[8] the multiobjective method becomes, actually, a single objective community detection method. Thus *SN-MOGA* effectively exploits the multiobjective approach by trying to obtain the best tradeoff between the two objective functions.

In order to more deeply investigate the differences between *SN-MOGA* and *MEA_s-SN*, both methods have been executed on a synthetic network of 10000 nodes, with average node degree 64, exponent of degree distribution -2, community size distribution -1, mixing parameter $\mu = 0.1$, $p_- = p_+ = 0.5$. The number of clusters of the ground truth division is 100. Table 1 reports the NMI and weighted NMI (in the table denoted as WNMI), the number of clusters NC and the number of singletons NS obtained by the two methods. The behavior of *MEA_s-SN* on this network is similar to the previous experimentation, i.e. it has the tendency of finding many clusters constituted by a single node, in this case it obtains 64

Table 2. Error obtained by *SN-MOGA*, *MEA_s-SN* and Chiang's method on the Wikipedia network, having number of nodes 7118, number of positive edges $E^+ = 83953$, and negative edges $E^- = 23118$.

Algorithm	Error	NC
SN-MOGA min $F(C)$	(0.0009868) (0.00005728)	106
SN-MOGA max Q_S	0.0016 (0.0001142)	115
MEA_s-SN Liu *et al.*[20]	0.0020 (0.0001609)	3341
k-way Chiang *et al.*[16]	0.2186	3-30

Note: The error has been computed as Chiang *et al.*[16]

singletons out of 196 clusters. Because of the selection bias discussed is Sec. 5, the normalized mutual information value assigned to the clustering of *MEA_s-SN* is 0.9916, i.e. it should be almost a perfect match. Clearly this result is not reliable because of the presence of the 64 singletons. *SN-MOGA* obtains 123 clusters and an NMI value of 0.6766. The weighted NMI values are reduced to 0.5376 for *SN-MOGA* and 0.3797 for *MEA_s-SN*, thus with the corrected measure the partition of *SN-MOGA* is considered better than that of *MEA_s-SN*. As regards the execution times of the two methods, a comparison is difficult because *SN-MOGA* has been written in MATLAB, while *MEA_s-SN* in C++. It is worth to point out that a fair comparison should consider the computational complexity of the methods. As reported in Sec. 4, the complexity of *SN-MOGA* is $O((gp\ log\ p) \times (n\ log\ n + m))$, but that of *MEA_s-SN* has not been reported by the authors.

6.3. *Comparison on Wikipedia network*

In this section we consider a real life signed network, namely the English Wikipedia network for admin elections, studied by Leskovec *et al.*,[39] downloadable from http://konect.uni-koblenz.de/networks/elec. The network is constituted by 7118 nodes, and has number of positive edges $E^+ = 83953$, and negative edges $E^- = 23118$. This network has also been tested by Chiang *et al.*[16] by applying their *k-way* multilevel algorithm. In Ref. 16 the authors reported the error they computed by applying formula (13), where the denominator, however, is substituted by n^2, i.e. the square of the number of nodes. We executed 10 times both *SN-MOGA* and *MEA_s-SN* on this network and in Table 2 the error, computed like Chiang et al., obtained by the two methods, with the standard deviation in parenthesis, and that obtained by Chiang *et al.*,[16] are reported. Moreover, also the average number NC of obtained clusters is shown. For *SN-MOGA* two results are shown: when the solution having minimum frustration $F(C)$ and maximum modularity Q_S are chosen from the Pareto front. The table points out that *SN-MOGA* obtains lower errors

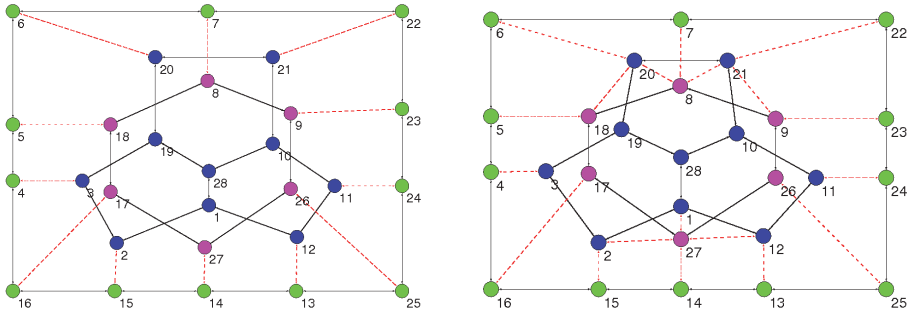


Fig. 25. (Color online) Synthetic networks reported in Yang *et al.* For each network, colors correspond to the true partitioning, red dotted lines to the negative edges and black lines to the positive edges.

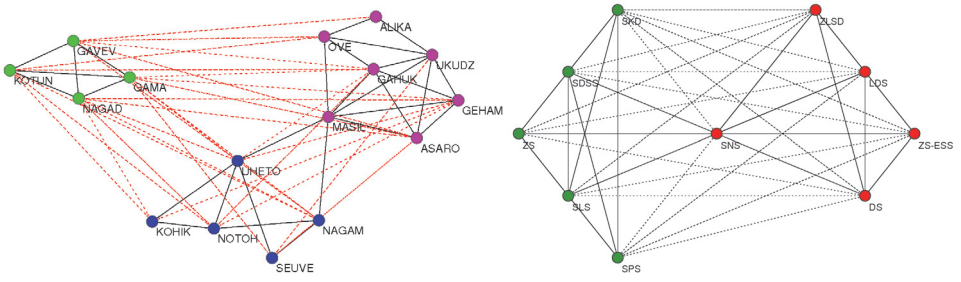
and standard deviations, both when minimum frustration and maximum modularity are chosen. In the former case the error value is the lowest, being 0.0009868. *MEA_s-SN* on this network is not able to detect meaningful groups of nodes. In fact, it finds 3341 communities, where 3303 are singleton nodes. Thus it does not assign almost half of the nodes to any community. The number of clusters found by *SN-MOGA*, instead, has been, on average, 106 for minimum frustration, and 115 for maximum modularity.

As regards the method of Chiang *et al.*, since the number of clusters must be given as input parameter, the authors computed the empirical error for values of k ranging from 3 to 30. The value they reported is 0.2186, which is much higher than that obtained by *SN-MOGA*. They observed that, for each k, the errors are very close. Since *SN-MOGA* finds around 100 clusters, the range of values used by Chiang *et al.* was perhaps insufficient to obtain a reasonable partitioning of the Wikipedia network. This result confirms the advantage of applying *SN-MOGA*, which is capable of finding meaningful divisions with small frustration values, without any knowledge on the network structure and no need of fixing the number of communities in advance.

7. Comparison with Particle Swarm Optimization

In this section a comparison between *SN-MOGA* and the Particle Swarm Optimization based method of Gong *et al.*,[9] on two artificial signed networks and two real-life networks analyzed by Yang *et al.*,[11] is presented.

The two artificial signed networks, illustrated in Fig. 25, show the difference between balanced and partitionable networks. The network *Network 1*, consisting of 28 nodes, 30 positive edges and 12 negative edges and displayed in Fig. 25(a), is partitionable and can be divided into the three groups {4, 5, 6, 7, 22, 23, 24, 25, 13, 14, 15, 16}, {8, 9, 26, 27, 17, 18}, and {20, 21, 10, 11, 12, 1, 2, 3, 19, 28}. *Network 2*, having 28 nodes, 30 positive edges and 19 negative edges (Fig. 25(b)),



(a) (b)

Fig. 26. (Color online) Networks representing the political alliances and oppositions among 16 Gahuku-Gama subtribes (left) and the Slovene Parliamentary Party (right). Colors correspond to the true partitioning, red dotted lines to negative edges and black solid lines to positive edges.

Table 3. Comparison between *SN-MOGA* and the *MODPSO* method of Gong *et al.*

Network	Algorithm	Modularity	NMI
Network 1	*SN-MOGA* min $F(C)$	0.5612 (0.5612)	1 (1)
	SN-MOGA max Q_S	0.5612 (0.5612)	1(1)
	MODPSO	0.5213 (0.5112)	1 (0.9742)
Network 2	*SN-MOGA* min $F(C)$	0.5257 (0.5257)	1 (1)
	SN-MOGA max Q_S	0.5257 (0.5257)	1(1)
	MODPSO	0.5643 (0.5634)	1 (0.9959)
GGS	*SN-MOGA* min $F(C)$	0.4310 (0.4310)	1 (1)
	SN-MOGA max Q_S	0.4310 (0.4310)	1 (1)
	MODPSO	0.4310 (0.4310)	1 (1)
SPP	*SN-MOGA* min $F(C)$	0.4556 (4556)	1 (1)
	SN-MOGA max Q_S	0.4556 (0.4556)	1(1)
	MODPSO	0.4547 (0.4532)	1 (0.9949)

is also partitionable in the same three groups of *Network 1*. The main difference between these two networks is that the former is also balanced since it has a two-way partitioning constituted by the first group and the union of the other two groups, while the latter is not balanced.

The *Gahuku-Gama Subtribes* (*GGS*) social network (Fig. 26(a)) describes the political alliances (29 positive edges) and oppositions (29 negative edges) among 16 sub-tribes. The Slovene Parliamentary Party (*SPP*) network (Figure 26(b)) shows the relation among 10 parties of the Slovene Parliament in 1994. It has 18 positive edges and 27 negative edges. Table 3, for each network, shows the maximum modularity and NMI values obtained by *MODPSO*, as reported in Gong *et al.*,[9] while for *SN-MOGA* the modularity and NMI values when the solution having minimum frustration is chosen from the Pareto front, and the solution with maximum modularity is chosen from the Pareto front with the corresponding NMI value. Average

values for both the methods are in parenthesis. The table points out that both methods obtain the ground truth division for all the networks. However, *MODPSO* does not find the best solution for all the executions, as average values in parenthesis show. Thus *SN-MOGA* has a more stable behavior than *MODPSO*. Moreover, the modularity values obtained by *SN-MOGA* are higher on *Network 1* and *SPP*, while lower on *Network 2* and the same on *GGS*. Thus we can observe that both methods are able to properly divide these networks.

8. Running Time Analysis

One of the main criticisms to evolutionary based methods is the high execution time required to obtain a solution. However, it is known that genetic algorithms are naturally parallelizable.[40] Since *SN-MOGA* has been written in Matlab, we could exploit the Parallel Computing Toolbox implemented in Matlab to allow multicore processing. We executed *SN-MOGA* on a computer cluster of 32 nodes, with 4 Gbyte of RAM and a 16-core Intel Xeon CPU at 2.6 GHz each.

In order to show the drastic reduction of execution times that can be obtained when the network size is large, Fig. 14 shows the time in seconds required by *SN-MOGA* to find a solution for the Wikipedia network (recall it has 7118 nodes, 83953 positive edges, and 23118 negative edges), when the number of cores used varies as 1, 2, 4, 8, 16, and 32. Population size and number of generations have been fixed to 100.

The figure points out that *SN-MOGA* presents a superlinear speedup when using two cores instead of one. In fact the running time reduces from 5 hours and an half to almost 2 hours. Moreover, the speedup is linear from 2 to 16 cores, since doubling the number of cores, the time required to execute the method becomes the half, and almost linear for 32 cores. In this latter case 20 minutes required with 16 cores reduce to 14 minutes on 32 cores. This experiment shows that *SN-MOGA* has a very good scalability. Thus, having at disposal sufficient computational resources, the method is able to deal with networks of very large size.

9. Conclusions

The paper proposed a multiobjective approach to detect communities in signed networks. The method optimizes two objectives in order to find network divisions such that intra-connections are dense and most edges within clusters are positive, and inter-connections between clusters are sparse, and most of these edges are negative. In order to evaluate the method, selection bias of the most used evaluation measures, namely the normalized mutual information, has been pointed out, and a corrected measure, the Weighted NMI, that avoids this bias adopted. An extensive experimental evaluation on randomly generated networks for which the ground-truth division is known, proved the ability of the method to find solutions having low frustration and high NMI and WNMI values. Furthermore, community structure found on the real life network Wikipedia showed that the error obtained by

SN-MOGA is lower than that obtained by *MEA_s-SN* and the *k-way* method of Chiang *et al.* A comparison with the *MODPSO* method, based on Particle Swarm Optimization, showed that the two methods are comparable, though *SN-MOGA* has a more stable behavior than *MODPSO*. Because of the genetic representation, the method cannot assign a node to multiple communities, thus generating overlapped community structures. Future work will investigate extensions to locus-based representation to obtain overlapping communities.

References

1. F. Heider, Attitudes and cognitive organization, *J. Psycology* **21** (1946) 107–112.
2. D. Cartwright and F. Harary, Structure balance: A generalization of Heider's theory, *Psycological Review* **63**(5) (1956) 277–293.
3. M. E. J. Newman and M. Girvan, Finding and evaluating community structure in networks, *Physical Review E* **69** (2004) 026113.
4. S. Gómez, P. Jensen and A. Arenas, Analysis of community structure in networks of correlated data, *Physical Review E* **80** (2009) 016114.
5. P. Doreian and A. Mrvar, A partitioning approach to structural balance, *Social Networks* **18** (1996) 149–168.
6. D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley, 1989).
7. C. Coello, G. B. Lamont and D. A. van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems* (Springer, 2007).
8. C. Shi, P. S. Yu, Z. Yan, Y. Huang and B. Wang, Comparison and selection of objective functions in multiobjective community detection, *Computational Intelligence* **30**(3) (2014) 562–582.
9. M. Gong, Q. Cai, X. Chen and L. Ma, Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition, *IEEE Transactions on Evolutionary Computation* **18** (February 2014) 82–97.
10. J. Davis, Clustering and structural balance in graphs, *Human Relations* **20** (1967) 181–187.
11. B. Yang, W. K. Cheung and J. Liu, Community mining from signed social networks, *IEEE Transactions on Knowledge and Data Engineering* **19**(10) (2007) 1333–1348.
12. J. Shi and J. Malik, Normalized cuts and image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(8) (2000) 888–905.
13. V. Traag and J. Bruggeman, Community detection in networks with positive and negative links, *Physical Review E* **80**(3) (2009) 036115.
14. J. Reichardt and S. Bornholdt, Statistical mechanics of community detection, *Physical Review E* **74** (2006) 016110.
15. J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E. W. D. Luca and S. Albayrak, Spectral analysis of signed graphs for clustering, prediction and visualization, in *Proc. of the SIAM Int. Conf. on Data Mining* (*SDM 2010*) (Columbus, Ohio, 2010), pp. 559–570.
16. K.-Y. Chiang, J. J. Whang and I. S. Dhillon, Scalable clustering of signed networks using balance normalized cut, in *Proc. of the 21st ACM Int. Conf. on Information and Knowledge Management* (*CIKM'12*) (Maui, Hi, USA, 2012), pp. 615–624.
17. P. Anchuri and M. Magdon-Ismail, Communities and balance in signed networks: A spectral approach, in *Proc. of the IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining* (*ASONAM 2012*) (Istanbul, Turkey, 2012), pp. 235–242.

18. P. Bogdanov, N. D. Larusso and A. K. Singh, Towards community discovery in signed collaborative interaction networks, in *Proc. of the ICDM Workshops* (Sidney, Australia, 2010), pp. 288–295.
19. Y. Li, J. Liu and C. Liu, A comparative analysis of evolutionary and memetic algorithms for community detection from signed networks, *Soft Computing* **18**(2) (2014) 329–348.
20. C. Liu, J. Liu and Z. Jiang, A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks, *IEEE Transactions on Cybernetics* (2014).
21. J. Huang, H. Sun, Y. Liu, Q. Song and T. Weninger, Towards online multiresolution community detection in large-scale networks, *PLOS ONE* **8** (2011) e23829.
22. V. D. Blondel, J.-L. Guillaume, R. Lambiotte and E. Lefevre, Fast unfolding of communities in large networks, *Journal of Statistical Mechanics: Theory and Experiment* **P10008** (2008).
23. A. Lancichinetti, S. Fortunato and J. Kertész, Detecting the overlapping and hierarchical community structure of complex networks, *New Journal of Physics* **11**(3) (2009) 033015.
24. N. Srinivas and K. Deb, Multiobjective optimization using nondominated sorting in genetic algorithms, *Evol. Comp.* **2**(3) (1994) 221–248.
25. Y. Park and M. Song, A genetic algorithm for clustering problems, in *Proc. of 3rd Annual Conf. on Genetic Algorithms* (Fairfax, Virginia USA, 1989), pp. 2–9.
26. M. Ehrgott, *Multicriteria Optimization* (Springer, Berlin, 2005).
27. M. T. Jensen, Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms, *IEEE Transactions on Evolutionary Computation* **7**(5) (2003) 503–515.
28. T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms*, 2nd edn. (MIT Press, 2007).
29. L. Danon, A. Díaz-Guilera, J. Duch and A. Arenas, Comparing community structure identification, *Journal of Statistical Mechanics* **P09008** (2005).
30. T. Cover and J. Thomas, *Elements of Information Theory* (Wiley, 1991).
31. T. O. Kvalseth, Entropy and correlation: Some comments, *IEEE Transactions on Systems, Man and Cybernetics* **17**(3) (1987) 517–519.
32. A. Strehl and J. Ghosh, Cluster ensembles — A knowledge reuse framework for combining multiple partitions, *Journal of Machine Learning Research* **3** (2002) 583–617.
33. S. Romano, J. Bailey, V. Nguyen and K. Verspoor, Standardized mutual information for clustering comparisons: One step further in adjustment for chance, in *Proc. of the 31st Int. Conf. on Machine Learning* (Beijing, China, 2014), JMLR: W&CP Vol. 32, pp. 1143–1151.
34. N. X. Vinh, J. Epps and J. Bailey, Information theoretic measures for clusterings comparison: Is a correction for chance necessary?, in *Proc. of the 26th Annual Int. Conf. on Machine Learning* (*ICML '09*) (Montreal, Quebec, Canada, 2009), pp. 1073–1080.
35. N. X. Vinh, J. Epps and J. Bailey, Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance, *J. Mach. Learn. Res.* **11** (2010) 2837–2854.
36. A. Amelio and C. Pizzuti, Is normalized mutual information a fair measure for comparing community detection methods?, in *Proc. of the IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining*, (*ASONAM 2015*) (Paris, France, 2015).

37. A. Lancichinetti, S. Fortunato and F. Radicchi, Benchmark graphs for testing community detection algorithms, *Physical Review E* **78**(046110) (2008).
38. M. Girvan and M. E. J. Newman, Community structure in social and biological networks, in *Proc. National Academy of Science, USA 99* (Washington, USA, 2002), pp. 7821–7826.
39. J. Leskovec, D. P. Huttenlocher and J. M. Kleinberg, Predicting positive and negative links in online social networks, in *Proc. of the 19th Int. Conf. on World Wide Web* (*WWW 2010*) (Raleigh, NC, USA, 2010), pp. 641–650.
40. M. Tomassini, *Parallel and Distributed Evolutionary Algorithms: A Review* (in Evolutionary Algorithms in Engineering and Computer Science), eds. Chichester *et al.* (J. Wiley and Sons, 1999).