A Cooperative Evolutionary Approach to Learn Communities in Multilayer Networks

Alessia Amelio and Clara Pizzuti

National Research Council of Italy (CNR),
Institute for High Performance Computing and Networking (ICAR),
Via P. Bucci 41C, 87036 Rende (CS), Italy
{amelio,pizzuti}@icar.cnr.it

Abstract. In real-world complex systems objects are often involved in different kinds of connections, each expressing a different aspect of object activity. Multilayer networks, where each layer represents a type of relationship between a set of nodes, constitute a valid formalism to model such systems. In this paper a new approach based on Genetic Algorithms to detect community structure in multilayer networks is proposed. The method introduces an extension of the modularity concept and adopts a genetic representation of a multilayer network that allows cooperation and co-evolution of individuals, in order to find an optimal division of the network, shared among all the layers. Moreover, the algorithm relies on a label propagation mechanism and a local search strategy to refine the result quality. Experiments show the capability of the approach to obtain accurate community structures.

1 Introduction

In the last few years complex systems described as networks of nodes connected by different kinds of relationships are receiving a lot of attention. In fact, the approach adopted so far of aggregating the great variety of links connecting objects constituting a network, revealed its weaknesses because of loss of information caused by such a simplified view of a system. Real-life networked systems present multiple ties, each generally playing a different role and exhibiting a different type of strength among objects. Representing such systems by using a single type of interaction is a rough approximation of reality. A more apt modeling of such systems can be obtained by *multilayer* networks[5]. Kivela et al., [5] introduced the concept of multilayer network as the most general notion to model complex networks, including *multiplex* [7], *multirelational* [4], *multidimensional* [9,10,6]. A multilayer network can be viewed as a set of slice networks. Each slice, modeled as a graph, represents an aspect of the object activity, since an object may be involved in distinct activities with variable concern. In multilayer networks grouping actors by considering only one type of interaction may lead to inaccurate community structures because information that could come from all the interactions is discarded. The objective in a multilayer network is to uncover a shared community structure among objects such that a quality function be optimized for all the layers at the same time.

T. Bartz-Beielstein et al. (Eds.): PPSN XIII 2014, LNCS 8672, pp. 222–232, 2014.

Proposals to find groups in multilayer networks can be found in [9,10,6,12,3]. In particular, Tang et al. [9,10] proposed a method, named *Principal Modularity Maximization (PMM)* that for each layer, first the *structural features*, corresponding to the top eigenvectors with positive eigenvalues, are extracted, then these features are combined to obtain latent communities.

In this paper a new method, named *MultiGA* (*Multilayer Genetic Algorithm*), able to detect a shared community structure in a multilayer network, is proposed. *MultiGA* adopts a genetic representation of individuals that allows co-evolution and cooperation among all the network layers. An individual is composed by a number of elements equal to the number of layers. Each element represents a division of the corresponding layer in communities, and it is co-evolved with all the others by *learning* from them their community structure through the optimization of a fitness function that combines the modularity values of each layer. *MultiGA* relies also on a label propagation mechanism and a local search strategy. The former mechanism aggregates nodes having no connection in a layer to the community recurring most often among its neighbors in all the layers. The local search strategy, similarly to the Blondel et al. [1] method for single-layer networks, moves a node to one of its neighboring communities if an increase in modularity is obtained. Experiments on synthetic and real-world networks show that *MultiGA* is able to detect accurate community structures in multilayer networks. The paper is organized as follows. The next section introduces multilayer networks and formalizes the problem of community detection. Section 3 describes the proposed approach. In section 4 the results of the experiments are reported. Finally, Section 5 concludes the paper.

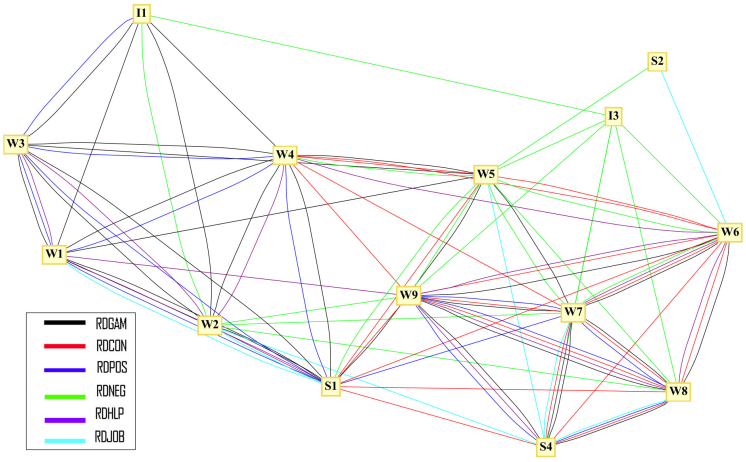


Fig. 1. The Roethlisberger & Dickson Bank Wiring Room of Western Electric multilayer network

2 Multilayer Networks

Let V be a set of n objects. A multilayer network is defined as a set $\mathcal{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_d\}$ of *slice networks*. Each slice $\mathcal{N}_s$ can be modeled as a graph $G_s = (V_s, E_s)$ where the

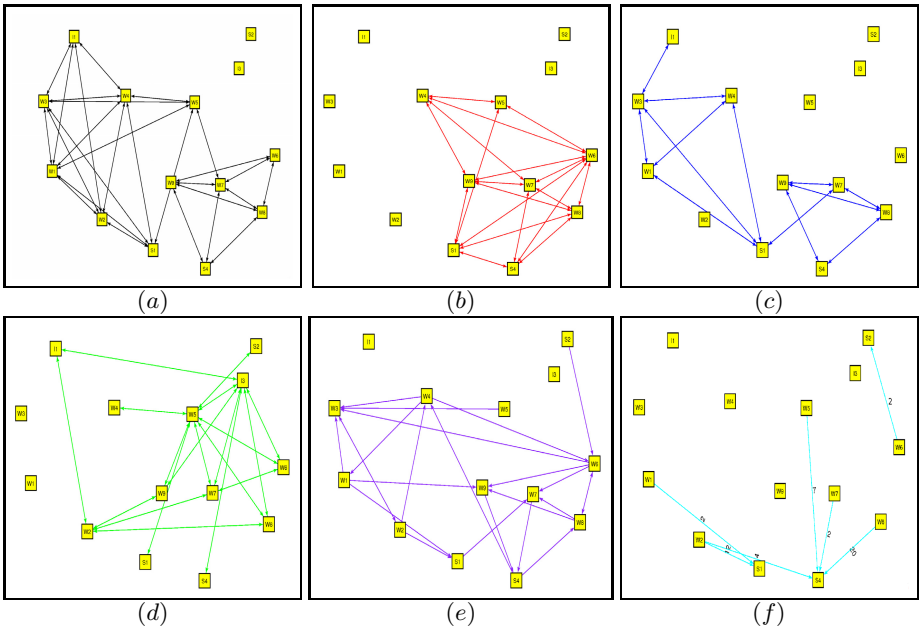


Fig. 2. (a) RDGAM relation, (b) RDCON relation, (c) RDPOS relation, (d) RDNEG relation, (e) RDHLP relation, (f) RDJOB relation

set of nodes $V_s \subseteq V$ is the subset of objects of V appearing in the slice $\mathcal{N}_s$, and E_s is the set of links that connect the objects of V_s in the s-th layer, i.e. an edge $(u, v) \in E_s$ if objects u and v interact in the s-th layer. $\mathcal{N}$ can thus be represented as a set $\mathcal{G} = \{G_1, \ldots, G_d\}$ of graphs, where each $G_s = (V_s, E_s)$, for $s = 1, \ldots, d$, is the graph modeling network $\mathcal{N}_s$ in the s-th layer. A layer thus represents one of the d slices of the network. Given an object $u \in V$, the neighbors of u at layer s are defined as $n_s(u) = \{v \in V_s \mid (u, v) \in E_s\}$, and the neighbors of u in $\mathcal{G}$ as $n(u) = \cup_{s=1}^{d} n_s(u)$.

A clustering, or community structure, $\mathcal{CS}_s = \{C_{s_1}, \ldots C_{s_k}\}$ of a layer $\mathcal{N}_s$ is a partitioning of G_s in groups of nodes that maximizes a quality function. Furthermore, for each couple of communities C_{s_i} and $C_{s_j} \in \mathcal{CS}_s$, $V_{s_i} \cap V_{s_j} = \emptyset$.

Our objective is to *uncover a shared community structure $\mathcal{CS}$ among the objects of the multidimensional network $\mathcal{N}$ such that a quality function is optimized in all the d dimensions*. An example of multilayer network is depicted in Figure 1. The example is taken from [2] and shows the relationships of 14 employees from a bank wiring room of Western Electric (Hawthorne Plant), downloaded from `http://moreno.ss.uci.edu/data.html`. The employees worked in a single room and include two inspectors (I1 and I3), three solderers (S1, S2 and S3), and nine wiremen or assemblers (W1 to W9). There are six different kinds of interactions among the employees: $RDGAM$, participation in horseplay; $RDCON$, participation in arguments about open windows; $RDPOS$, friendship; $RDNEG$, antagonistic behavior; $RDHLP$, helping others with work; and $RDJOB$, the number of times workers traded job assignments. The first

four types of connections are symmetric, while the last two aren't. Figures 2(a-f) show the six networks corresponding to each relation, where an unconnected node in a slice means that this node does not have any interaction of that type. For example, employee $S2$ has no ties with others regarding participation in horseplay ($RDGAM$) and arguments about open windows ($RDCON$), nor any friendship relation ($RDPOS$), i.e. $n_{RDGAM}(S2) = n_{RDCON}(S2) = n_{RDPOS}(S2) = \emptyset$, instead he has an antagonistic behavior with $W5$ ($RDNEG$), he helps $W6$ with work ($RDHLP$), and traded twice job assignment with him, i.e. $n_{RDNEG}(S2) = \{W5\}$, $n_{RDHLP}(S2) = \{W6\}$, but $n_{RDJOB}(S2) = \emptyset$ because there is an edge from $W6$ to $S2$ and not viceversa, thus $n(S2) = \{W5, W6\}$. The figure points out the intrinsic difficulty of grouping nodes in a proper way due to the incompleteness of information about the relations between two employees. In the next section an approach that combines the ties coming from all the layers is presented.

3 Method Description

In this section a detailed description of *MultiGA* is given, along with the genetic representation and operators adopted. Furthermore, the new concept of combined modularity is introduced and used as fitness function to optimize in order to search for a shared community structure in a multilayer network.

Genetic Representation and Operators. The genetic representation used by the approach is an extension of the locus-based adjacency representation. An individual $I = \{I_1, \ldots, I_d\}$ of the population is composed by a set of d elements $I_s, 1 \leq s \leq d$. Each element I_s consists of n genes $g_1, \ldots, g_n$ assuming integer values, corresponding to network nodes, in the range $\{1, \ldots, n\}$. A value v assigned to the u-th gene means that there is a link between nodes u and v in the s-th graph G_s modeling the s-th network layer $\mathcal{N}_s$. If node u has no links in the s-th layer, i.e. $n_s(u) = \emptyset$, then it is assigned a zero value. Thus each element I_s of an individual I of the population gives a division of s-th layer $\mathcal{N}_s$ of $\mathcal{N}$ in a number c_s of communities.

I1	I3	W1	W2	W3	W4	W5	W6	W7	W8	W9	S1	S2	S4
4	0	6	3	3	3	3	9	10	11	14	3	0	10
0	0	0	0	0	7	8	12	8	8	6	11	0	8
5	0	12	0	3	3	0	0	12	11	10	5	0	11
4	8	0	1	0	7	2	7	2	2	2	7	7	2
0	0	11	6	4	5	5	10	14	8	14	9	8	10
0	0	12	12	0	0	14	13	14	14	0	0	0	0

$(a : Parent1)$

I1	I3	W1	W2	W3	W4	W5	W6	W7	W8	W9	S1	S2	S4
6	0	5	3	3	3	3	9	10	11	9	7	0	9
0	0	0	0	0	9	6	10	8	11	10	10	0	8
5	0	12	0	3	3	0	0	10	14	10	3	0	10
4	8	0	1	0	7	2	2	7	2	2	7	7	2
0	0	12	6	4	3	5	10	14	8	14	9	8	10
0	0	12	12	0	0	14	13	14	14	0	0	0	0

$(b : Parent2)$

I1	I3	W1	W2	W3	W4	W5	W6	W7	W8	W9	S1	S2	S4
5	0	6	3	6	3	3	9	10	9	9	4	0	9
0	0	0	0	0	8	8	9	6	8	8	8	0	8
5	0	5	0	3	3	0	0	10	11	14	5	0	10
4	9	0	1	0	7	2	2	2	2	4	7	7	2
0	0	12	6	4	3	5	5	14	8	14	9	8	10
0	0	12	12	0	0	14	13	14	14	0	0	0	0

$(c : before\,mutation)$

I1	I3	W1	W2	W3	W4	W5	W6	W7	W8	W9	S1	S2	S4
0	0	1	1	0	1	1	0	1	1	0	0	0	0
0	0	0	0	0	1	0	1	0	0	1	0	0	1
1	0	1	0	0	0	0	0	0	0	1	1	0	0
1	1	0	1	0	1	1	1	0	1	0	0	1	1
0	0	1	1	1	0	0	1	1	0	0	0	1	1
0	0	0	1	0	0	1	1	0	0	0	0	0	0

$(d : mask)$

I1	I3	W1	W2	W3	W4	W5	W6	W7	W8	W9	S1	S2	S4
6	0	6	3	3	3	3	9	10	11	9	7	0	9
0	0	0	0	0	7	6	12	8	11	6	10	0	8
5	0	12	0	3	3	0	0	10	14	10	5	0	10
4	8	0	1	0	7	2	7	7	2	2	7	7	2
0	0	11	6	4	3	5	10	14	8	14	9	8	10
0	0	12	12	0	0	14	13	14	14	0	0	0	0

$(e : child)$

I1	I3	W1	W2	W3	W4	W5	W6	W7	W8	W9	S1	S2	S4
5	0	6	3	6	3	12	9	10	9	9	4	0	9
0	0	0	0	0	8	12	9	6	8	8	8	0	8
5	0	5	0	3	3	0	0	10	11	14	5	0	10
4	9	0	1	0	7	13	2	2	2	4	7	7	2
0	0	12	6	4	3	5	5	14	8	14	9	8	10
0	0	12	12	0	0	14	13	14	14	0	0	0	0

$(f : after\,mutation)$

Fig. 3. (a) First parent, (b) second parent, (c) individual before mutation, (d) binary mask, (e) individual after crossover, (f) individual after mutation

The initialization process, for every individual $I = \{I_1, \ldots, I_d\}$, considers all the elements I_s, and assigns to a node u one of its neighbors v at random, where $v \in n_s(u)$, i.e. it is one of the neighboring nodes of u relative to the graph G_s corresponding to the s-th layer. If u has no neighbors in G_s, then the corresponding gene $g_u = 0$. Consider Figure 3(a) representing a generic individual $I = \{I_1, I_2, \ldots, I_6\}$ of the Bank Wiring Room example consisting of six elements, i.e the number of different relations. Employees have been numbered from 1 to 14, thus employee $I1$ is node 1, employee $I3$ is node 2, employee $W1$ is node 3, and so on. Every element I_s of I corresponds to a layer s and represents the graph G_s. For example I_2, second row of Figure 3(a), represents the connections among employees with respect to relation $RDCON$. Nodes $I1, I3, W1, W2, W3, S2$ have no connections of type $RDCON$, thus the value of their neighbor node in I_2 is set to zero. $W4$ has neighbors $\{W5, W6, W7, W9\}$, as can be seen from Figure 2(b), thus I_2 at position 6 (corresponding to $W4$) has value 7, corresponding to $W5$, which is one of its neighbors.

The crossover operator is executed on each layer by applying uniform crossover. Given two parents $I = \{I_1, \ldots, I_d\}$ and $J = \{J_1, \ldots, J_d\}$, and a randomly generated binary vector, for each couple (I_s, J_s) uniform crossover selects the genes where the vector is a 1 from the first parent I_s, and the genes where the vector is a 0 from the second parent J_s, and combines the genes to form a child IJ_s. The crossover operator is showed in Figure 3. Consider, for example, the $RDGAM$ layer, and the corresponding parents I_1, first row of Figure 3(a), and J_1 first row of Figure 3(b). The mask is that reported in the first row of Figure 3 (d). Thus the child IJ_1 generated by I_1 and J_1, first row of Figure 3(e), is such that $IJ_1(1) = J_1(1) = 6$ and $IJ_1(2) = J_1(2) = 0$ because the mask is zero in the first two positions, while $IJ_1(3) = I_1(3) = 6$ because the mask value is 1 in the third position, and so on.

The mutation operator for every element I_s of $I = \{I_1, \ldots, I_d\}$ assigns to each node u one of its neighbors $v \in n_s(u)$ at random. An example of mutation can be seen in Figure 3(c),(f). For example, consider relation $RDNEG$ (fourth row in Figure 3(c)). The neighbor of $W5$ is changed from 2 (I3) to 13 (S2), as can be seen in Figure 3(f).

Fitness Function. The choice of an appropriate fitness function is a key point to obtain a good solution for the problem to solve. As regards single-layer networks, the well known concept of *modularity* introduced by Girvan and Newman [8] is generally considered the one that at the best interprets the intuitive idea of dense group of nodes. The definition of modularity Q for single-layer networks is the following: $Q = \frac{1}{2m}\sum_{ij}(A_{ij} - \frac{k_i k_j}{2m})\delta(C_i, C_j)$ where A is the adjacency matrix of the associated graph, m is the number of edges of the graph, k_i and k_j are the degrees of nodes i and j respectively. δ is the Kronecker function and yields 1 if i and j are in the same community (i.e. $C_i = C_j$), zero otherwise. Values approaching 1 indicate high quality clustering.

We propose to extend the concept of modularity to multilayer networks by combining the modularity values computed for each layer in such a way that the value for each layer is influenced by the values of all the other layers. Let s and r be two slices of a multilayer network, and $\mathcal{CS}_s$, $\mathcal{CS}_r$ be the clustering obtained on networks $\mathcal{N}_s$ and $\mathcal{N}_r$ respectively. Then the combined modularity Q_{sr} between slices s and r is defined as follows:

$$Q_{sr} = \frac{1}{2m_s + 2m_r} \sum_{ij} \left[(A_{ijs} - \frac{k_{is}k_{js}}{2m_s})\delta(C_{is}, C_{js}) + (A_{ijr} - \frac{k_{ir}k_{jr}}{2m_r})\delta(C_{is}, C_{js}) \right] \quad (1)$$

where A_s and A_r are the adjacency matrices of graphs G_s and G_r, respectively, k_{is} and k_{js} are the degrees of nodes i and j in G_s, while k_{ir} and k_{jr} are the degrees of nodes i and j in G_r, respectively. The Kronecker function δ yields 1 if i and j are in the same community C_s (i.e. $C_{is} = C_{js}$), zero otherwise. The meaning of Q_{sr} is that, while computing a community structure $\mathcal{CS}_s$ on slice $\mathcal{N}_s$, this community structure is also checked on slice $\mathcal{N}_r$. Thus, if $\mathcal{CS}_s$ does not determine a good grouping of nodes also in $\mathcal{N}_r$, it is penalized because the second term of Formula (1) is low.

Analogously, the *combined modularity* Q_{rs} between slices r and s is defined as

$$Q_{rs} = \frac{1}{2m_s + 2m_r} \sum_{ij} \left[(A_{ijr} - \frac{k_{ir}k_{jr}}{2m_r})\delta(C_{ir}, C_{jr}) + (A_{ijs} - \frac{k_{is}k_{js}}{2m_s})\delta(C_{ir}, C_{jr}) \right] \quad (2)$$

Finally, the *total combined modularity* Q_{ml} is computed on all the d slices by considering the sum of combined modularities Q_{sr}, for each couple of slices s and r:

$$Q_{ml} = \sum_{s,\, r \; s \neq r} \Big(Q_{sr} + Q_{rs} \Big) \quad (3)$$

In the next section a genetic algorithm that discovers shared community structure in multilayer networks by optimizing Q_{ml} is presented.

Algorithm Description. The evolutionary method we propose is based on the idea that, while detecting a community structure $\mathcal{CS}_s$ on a layer s, it must be taken into account how much $\mathcal{CS}_s$ is *similar* to the community structures $\mathcal{CS}_r$ of the other layers, $r = 1, \ldots, d, r \neq s$. The intuitive idea of similarity is that $\mathcal{CS}_s$ contains groups of nodes that also appear in $\mathcal{CS}_r$, i. e. layers share communities. In order to pursue this objective, the algorithm *MultiGA*, thanks to the genetic representation that consists of individuals $I = \{I_1, \ldots, I_d\}$ composed by a number d of elements, one for each layer, evolves individuals I by exchanging information among the layers. In fact, while searching for the division of a generic layer s in groups of nodes, by evolving the corresponding element $I_s \in I$, it *learns* from the other elements I_r $r = 1, \ldots, d, r \neq s$ how much its clustering is shared with the other layers. This exchange of information is made possible by computing the total combined modularity value, that guides the search by exploiting the knowledge coming from all the slices. The *MultiGA* algorithm is described in Figure 4. It receives in input a multilayer network $\mathcal{N}$ and gives a vector L containing a cluster labeling for each node of $\mathcal{N}$. *MultiGA* creates a random population of individuals $I = \{I_1, \ldots, I_d\}$ (step 1) and evolves it for a fixed number of generations (step 2). For each individual in the population (step 3) the fitness function is calculated by using Formula (3) of the total combined modularity Q_{ml}. To this end, for each element I_s of I, the combined modularity of I_s with all the other layers is computed (steps 4-9). Then variation operators are applied and a new population created. At the end of the evolutionary process, a node label vector L_s is generated for each layer by assigning to each node the label of the community it belongs to, as determined by the clustering $\mathcal{CS}_s$ (steps 13-17). If a node u in the layer s has not been assigned to any cluster because it has no links with nodes of that layer, then the *LabelAssignment* method (Figure 4(b)) considers its neighbors $n(u) = \cup_{s=1}^{d} n_s(u)$ in $\mathcal{G}$, and assigns to u the majority cluster

MultiGA **Method:**
Input: A multilayer network $\mathcal{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_d\}$ of d dimensions, the set of graphs $\mathcal{G} = \{G_1, \ldots, G_d\}$ modeling it
Output: A node cluster labeling L that partitions $\mathcal{N}$ in the optimal shared community structure
1 **create** an initial population of random individuals $I = \{I_1, \ldots, I_d\}$
2 **while** not maxGen
3 **for each** individual $I = \{I_1, \ldots, I_d\}$
4 **decode** I and obtain partitionings $\mathcal{CS}_s = \{C_{s1}, \ldots, C_{sk}\}$ for $s = 1, \ldots, d$
5 $Q_{ml} = 0$
6 **for** $s = 1, \ldots, d$
7 **compute** $Q_s = \sum Q_{sr} + Q_{rs}$, for $r = 1, \ldots, d, r \neq s$
8 $Q_{ml} = Q_{ml} + Q_s$
9 **end for**
10 **end for**
11 **create** a new population by applying the variation operators
12 **end while**
13 **for** s=1,...,d
14 **initialize** the labeling vector L_s to null values
15 **for each** node v_j of $\mathcal{G}$ appearing in $C_{si} \subset \mathcal{CS}_s$
16 **assign** cluster label si to v_j, i.e. $L_s(v_j) = si$
17 **end for**
18 **Perform *LabelAssignment*** on L_s
19 **end for**
20 **compute** the modularity value Q for each partitioning determined by L_s $s = 1, \ldots, d$
22 **choose** as node label vector L the label vector L_s returning the maximum Q value;
21 **let** $\overline{\mathcal{G}} = \cup_{s=1}^{d} G_s$ be the graph obtained by joining all layers, where $\overline{A}_{ij} = 1$ if $\exists s$ such that $A_{ijs} = 1$
22 **Perform** *LocalSearch* on $\overline{\mathcal{G}}$ starting from solution L to improve modularity value Q

(a)

LabelAssignment **Method:**
Input: the sequence of graphs $\mathcal{G} = \{G_1, \ldots, G_d\}$ modeling $\mathcal{N}$ and the node cluster labeling L_s of s-th layer
Output: A node cluster labeling L_s where each node has been assigned a cluster label
1 **for each** node $u \in V$
2 **if** $(L_s(u) == 0)$
3 **let** $n(u) = \{v_{n_1}, \ldots, v_{n_t}\}$ be the neigh. of u in $\mathcal{G}$ and $L_s(v_{n_1}), \ldots L_s(v_{n_t})$ be the clust. label of v_{n_i} in $\mathcal{CS}_s$
4 $L_s(u) = argmax\,\{L_s(v_{n_1}), \ldots, L_s(v_{n_t})\}$
5 **end if**
6 **end for**

(b)

Fig. 4. The pseudo-code of the *MultiGA* algorithm

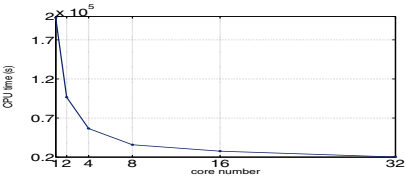
label in $\mathcal{CS}_s$ of these neighbors, i.e. u is assigned the cluster label that occurs most often in $\mathcal{CS}_s$ among its overall neighbors (steps 2-5 of *LabelAssignment* method). After that, for each layer s, the modularity value Q of Girvan and Newman [8] with respect to the partitioning determined by L_s is computed, and the labeling L_s giving the maximum Q value is chosen as final solution (steps 20-21). Finally, a post-processing local search, analogous to that proposed by Blondel et al. [1], is performed on the graph $\overline{\mathcal{G}} = \cup_{s=1}^{d} G_s$ obtained by the union of all the slices G_s where edges between two nodes are counted once, in order to improve modularity value. The local search, only once a time, moves a node to one of its neighboring communities if an increase in modularity, computed on the total graph $\overline{\mathcal{G}}$, is obtained. In the next section experiments on multilayer networks will show the feasibility of the approach in finding shared community structure.

Table 1. Average NMI values together with the standard deviation and best NMI values (in parenthesis) of *MultiGA* and a standard genetic algorithm using one slice at a time. Population size=$\{100, 500\}$, $\nu = 0.1, 0.3, 0.5$, $\mu = 0.5$. nc is the number of communities found.

Pop.	*Strategy*	$\nu = 0.1$	nc	$\nu = 0.3$	nc	$\nu = 0.5$	nc
100	A1	0.7629 ± 0.071 (0.8751)	7 (34)	0.4620 ± 0.124 (0.6684)	18 (145)	0.3696 ± 0.116 (0.5560)	5 (22)
	A2	0.7277 ± 0.110 (0.9116)	9 (20)	0.5345 ± 0.101 (0.6267)	4 (8)	0.3939 ± 0.158 (0.7063)	30 (262)
	A3	0.7997 ± 0.067 (0.9303)	3 (5)	0.5461 ± 0.151 (0.7092)	4 (6)	0.4451 ± 0.177 (0.6681)	10 (43)
	A4	0.6421 ± 0.167 (0.9097)	17 (53)	0.5179 ± 0.118 (0.7080)	9 (30)	0.4801 ± 0.157 (0.6667)	4 (11)
	MultiGA	0.9778 ± 0.024 (1)	3 (3)	0.7793 ± 0.139 (1)	2 (3)	0.7513 ± 0.084 (0.8335)	2 (2)
500	A1	0.8568 ± 0.088 (0.9498)	6 (35)	0.6663 ± 0.200 (0.9498)	17 (145)	0.6076 ± 0.105 (0.7529)	3 (10)
	A2	0.8307 ± 0.087 (0.9498)	5 (11)	0.7617 ± 0.087 (0.8846)	3 (3)	0.5883 ± 0.141 (0.8134)	29 (262)
	A3	0.9048 ± 0.081 (0.9707)	3 (3)	0.7627 ± 0.072 (0.9383)	3 (5)	0.5833 ± 0.174 (0.7371)	7 (43)
	A4	0.7237 ± 0.174 (0.9414)	14 (44)	0.6653 ± 0.088 (0.7795)	7 (36)	0.6562 ± 0.064 (0.7304)	4 (9)
	MultiGA	0.9808 ± 0.021 (1)	3 (3)	0.8376 ± 0.139 (1)	2 (3)	0.7530 ± 0.054 (0.7700)	2 (2)

	Strategy	*Evolutionary*			*Spectral*
		pop=100	pop=300	pop=500	
1-D	A1	0.7629 ± 0.071	0.8484 ± 0.082	0.8568 ± 0.088	0.7237 ± 0.1924
	A2	0.7277 ± 0.110	0.7969 ± 0.088	0.8307± 0.087	0.6798± 0.1888
	A3	0.7997 ± 0.067	0.8338 ± 0.111	0.9048 ± 0.081	0.6672 ± 0.1848
	A4	0.6421 ± 0.167	0.7285 ± 0.169	0.7237 ± 0.174	0.6906± 0.1976
4-D	*MultiGA*	0.9778 ± 0.024	0.9785 ± 0.022	0.9808 ± 0.021	-
	P M M	-	-	-	0.9351 ± 0.1059

(a)



(b)

Fig. 5. (a) Comparison of the NMI values between the evolutionary computation approaches and spectral approaches of Tang et al. [9]; (b) computation times required by *MultiGA* for increasing number of cores.

4 Experimental Results

This section provides a thorough experimentation for assessing the capability of *MultiGA* in detecting shared community structure in multilayer networks. The *MultiGA* algorithm has been written in MATLAB 7.14 R2012a, using the Genetic Algorithms and Direct Search Toolbox 2. A trial and error procedure has been adopted for fixing the parameter values. Thus the crossover rate has been fixed to 0.8, mutation rate to 0.2, elite reproduction 10% of the population size, number of generations is 150. We first present the results *MultiGA* obtained on randomly generated synthetic data sets. The networks have been generated as proposed by Tang et al. [9]. Each network is composed by 350 objects divided into three different clusters: the first one contains 50 objects, the second one 100 and the last one 200 objects. The objects are involved in $d = 4$ relations. A within-group probability μ connects the objects inside the same cluster. This probability value changes between groups at different slices. Any two nodes are connected to each other with probability ν, providing a controlled noise to the network. A clear network structure is obtained when the μ value is high and the ν value is low. 50 different synthetic networks have been randomly generated for different combinations of the parameters μ and ν and the average and standard deviation computed from the 50 runs. Since the ground truth partitioning of the nodes in communities is known a priori, in order to evaluate *MultiGA*, the Normalized Mutual Information (NMI) has been computed between the ground truth division in communities and the partitioning found

by *MultiGA*. Table 1 reports the average NMI values obtained by *MultiGA* together with standard deviation, and best NMI, in parenthesis, reached among all the runs. In order to show the superiority of *MultiGA* with respect to a naive approach that uses only one layer, the NMI values are compared with those returned by a standard genetic algorithm that optimizes Newman's modularity by using only one layer at a time. For this experiment the within-group interaction parameter μ has been fixed to 0.5, while the noise parameter ν has been varied as 0.1, 0.3 and 0.5. Furthermore, we report the results for increasing values of population size, namely 100 and 500. In the table, nc denotes the average and the maximum, in parenthesis, number of communities found. The table clearly shows the very good results obtained by *MultiGA* that simultaneously evolves all the layers, with respect to running a naive method that finds a solution by using only one slice. This confirms the superiority of our technique with respect to single dimension based methods to discover community structure. It is worth to note that increasingly high NMI values are obtained by *MultiGA* at increasing values of population size, and often *MultiGA*, among the executions, is able to detect the ground truth division of the network. Moreover, when the noise is high, $\nu = 0.5$, the NMI values of *MultiGA* are never less than 0.75. Figure 5 (a) compares *MultiGA* with the method proposed by Tang et al. [9,10]. The table reports the results appearing in [9] for the spectral approach PMM that uses all the layers, and the spectral approach that uses a single layer at a time. The first observation is that the evolutionary approach always obtains higher NMI values with respect to the spectral approaches proposed by Tang et al. In particular, *MultiGA* reaches 0.97 with population size 100, and an even higher value of 0.98 when population size is 500, while the NMI value of PMM is 0.93. Analogously, the evolutionary strategy on single layers obtains better results than the spectral approach. It is worth pointing out that the spectral approach needs the number of communities as input parameter, while *MultiGA* automatically determines the number by optimizing the objective function. We also experiments *MultiGA* by fixing within-group interaction parameter $\mu = 0.8$. For lack of space we cannot report the overall results. However, for instance, in such a case, with population size 300, the NMI values are 0.75 for noise $\nu = 0.5$, 0.88 for $\nu = 0.3$ and 0.99 for $\nu = 0.1$. We performed experiments also on two real-life multilayer networks. The former is the Bank Wiring Room of Western Electric network of Figure 1. The grouping obtained consists of two communities, {W1,W2,W3,W4,I1,S1} and {W5,W6,W7,W8,W9,I3,S4} which seems rather plausible by observing Figure 1. The second one, is the famous multilayer network consisting of marriage and business ties among 16 Florentine families in the 15th century [11], depicted in Figure 6. The figure shows the division we obtained in two groups (cyan and magenta respectively), and the isolated node $Pucci$ in green, which has no connections in any of the two layers (as can be seen from Figure 6(b) and (c)). The division in two communities reflects the sharing of business and marriage relations. Moreover, it is worth to note that $Strozzi$ and $Ridolfi$ families have no marriage relationships, and they have been joined to the communities composed by {*Lamberteschi, Guadagni, Castellani, Peruzzi, Bischeri*} and {*Acciaiuoli, Albizzi, Barbadori, Ginori, Medici, Pazzi, Salviati,Tornabuoni*} respectively, because they effectively have more business ties with the corresponding group. Thus *MultiGA* was able to properly capture the information coming from both types of relations. It is known

that genetic algorithms are naturally parallelizable. To this end, we used the Parallel Computing Toolbox of Matlab that allows multicore processing to deal with computationally intensive problems. We computed the times required by *MultiGA* on a computer cluster of 24 nodes, with 4 Gbyte of RAM and a 24-core Intel Xeon CPU at 2.6 GHz each, for a synthetic network of 5000 nodes with population size fixed to 300, when the number of cores used varies as 1, 2, 4, 8, 16, and 32. We obtained a linear speedup of the parallel implementation when 2 or 4 cores are employed (see Figure 5 (b)). In such a case, in fact, doubling the number of cores doubles the algorithm speed. When the number of cores increases to 8, 16, and 32 the speedup is almost linear, due to the times needed for communication. However, the time reduction is notably, going from 55 hours on one processor, to 5 hours when using 32 cores, showing that parallel implementation can give a valuable help in dealing with large networks.

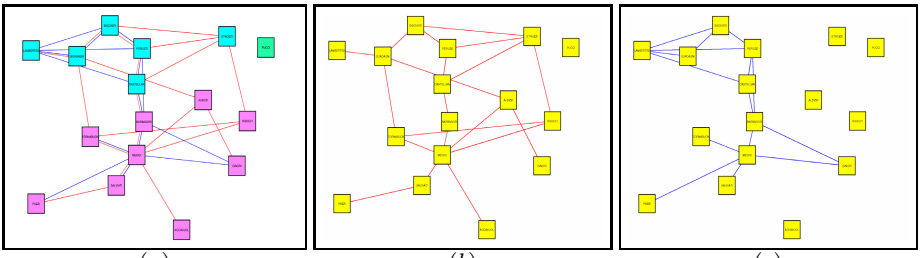


Fig. 6. (a) The Florentine families, (b) business relation (red), (c) marriage relation (blue)

5 Conclusions

The paper proposed a genetic algorithm to find shared community structure in multilayer networks, based on the extension of genetic representation from single to multilayer networks, and on the definition of total combined modularity concept. It employs two strategies, one to aggregate isolated nodes, and another to improve quality results by performing local search. Experiments on synthetic and real-life networks proved the capability of the approach to detect meaningful shared community structure.

References

1. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefevre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, P10008 (2008)
2. Breiger, R.R., Boorman, S.A., Arabie, P.: An algorithm for clustering relational data with applications to social network analysis and comparison with multidimensional scaling. Journal of Mathematical Psychology 12, 328–383 (1975)
3. Comar, P.M., Tan, P.-N., Jain, A.K.: A framework for joint community detection across multiple related networks. Neurocomputing 76(1), 93–104 (2012)

4. Harrer, A., Schmidt, A.: Blockmodelling and role analysis in multi-relational networks. Social Netw. Analys. Mining 3(3), 701–719 (2013)
5. Kivela, M., Arenas, A., Barthelemy, M., Gleeson, J.P., Moreno, Y., Porter, M.A.: Multilayer networks. arXiv:1309.7233v3 (2014)
6. Li, X., Ng, M.K., Ye, Y.: Multicomm: Finding community structure in multi-dimensional networks. In: IEEE Trans. on Knowl. and Data Eng. (2013) (in press)
7. Mucha, P.J., Richardson, T., Macon, K., Porter, M.A., Onnela, J.-P.: Community structure in time-dependent, multiscale, and multiplex networks. Science 328(5980), 876–878 (2010)
8. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Review E69, 026113 (2004)
9. Tang, L., Wang, X., Liu, H.: Uncoverning groups via heterogeneous interaction analysis. In: The Ninth IEEE Int. Conf. on Data Mining, ICDM 2009, pp. 503–512 (2009)
10. Tang, L., Wang, X., Liu, H.: Community detection via heterogeneous interaction analysis. Data Mining and Knowledge Discovery 25(1), 1–33 (2012)
11. Wasserman, S., Faust, K.: Social Network Analysis Methods and Applications. Cambridge University Press (2009)
12. Zhang, Z., Li, Q., Zeng, D., Gao, H.: User community discovery from multi-relational networks. Decision Support Systems 54(2), 870–879 (2013)