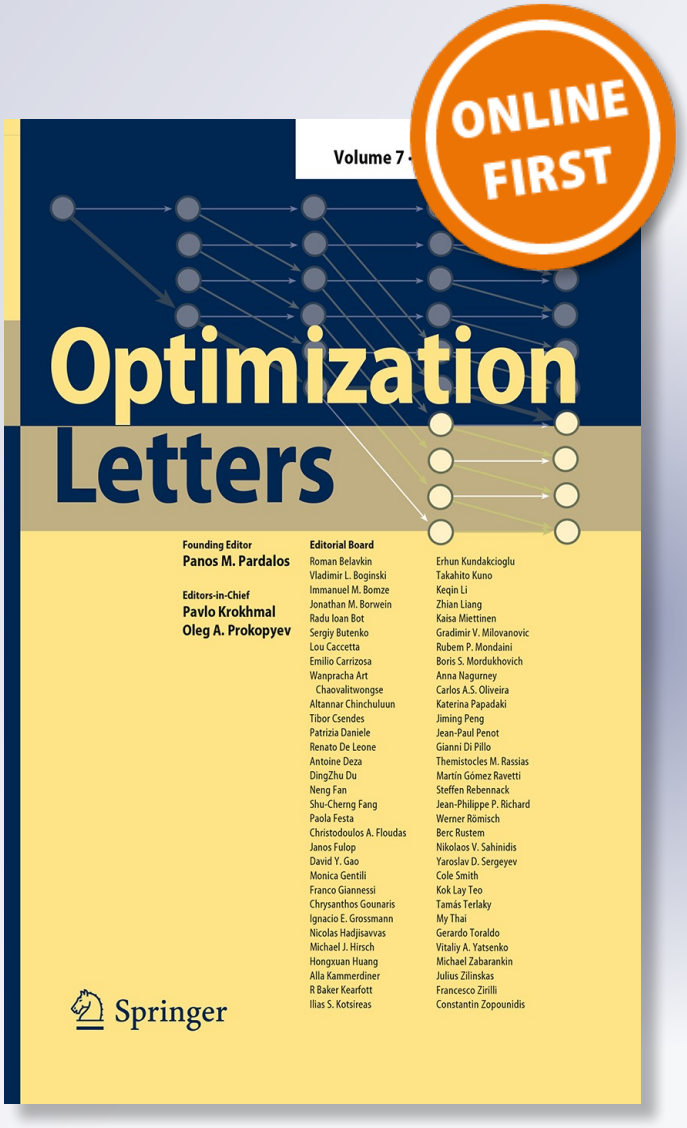
A locally optimal hierarchical divisive heuristic for bipartite modularity maximization

Alberto Costa & Pierre Hansen

Optimization Letters

ISSN 1862-4472

Optim Lett
DOI 10.1007/s11590-013-0621-x



Springer

Optim Lett
DOI 10.1007/s11590-013-0621-x

ORIGINAL PAPER

A locally optimal hierarchical divisive heuristic for bipartite modularity maximization

Alberto Costa · Pierre Hansen

Received: 18 October 2012 / Accepted: 5 February 2013

Abstract Given a set of entities, cluster analysis aims at finding subsets, also called clusters or communities or modules, entities of which are homogeneous and well separated. In the last ten years clustering on networks, or graphs, has been a subject of intense study. Edges between pairs of vertices within the same cluster should be relatively dense, while edges between pairs of vertices in different clusters should be relatively sparse. This led Newman to define the modularity of a cluster as the difference between the number of internal edges and the expected number of such edges in a random graph with the same degree distribution. The modularity of a partition of the vertices is the sum of the modularities of its clusters. Modularity has been extended recently to the case of bipartite graphs. In this paper we propose a hierarchical divisive heuristic for approximate modularity maximization in bipartite graphs. The subproblem of bipartitioning a cluster is solved exactly; hence the heuristic is locally optimal. Several formulations of this subproblem are presented and compared. Some are much better than others, and this illustrates the importance of reformulations. Computational experiences on a series of ten test problems from the literature are reported.

Keywords Bipartite graphs · Clustering · Modularity maximization

A. Costa (✉) · P. Hansen
LIX, École Polytechnique, 91128 Palaiseau, France
e-mail: costa@lix.polytechnique.fr

P. Hansen
GERAD, HEC Montréal, 3000 Chemin de la Côte-Sainte-Catherine,
Montréal H3T 2A7, Canada
e-mail: pierre.hansen@gerad.ca

Published online: 26 February 2013

1 Introduction

Modularity is a famous criterion employed for graph clustering problems. It represents the fraction of edges within clusters minus the expected fraction of such edges in a random graph with the same degree distribution [19,27]. More formally, modularity is expressed as:

$$Q = \sum_{c=1}^{N_c} \left(\frac{m_c}{m} - \frac{{D_c}^2}{4m^2} \right), \tag{1}$$

where N_c is the number of clusters, m is the number of edges, m_c is the number of edges within cluster c and D_c is the sum of the degrees of the vertices which are inside this cluster. Looking more in detail Eq. (1), it appears that modularity is additive, i.e., it is the sum of the modularity of each cluster. Given a cluster c, its modularity can be expressed as the difference between two terms: $\frac{m_c}{m}$, that is the fraction of edges in cluster c, and $\frac{D_c^2}{4m^2}$, that is the expected fraction of edges in cluster c in a random graph whose vertices have the same expected degrees. In order to obtain a good quality partition of a graph into clusters, one should maximize modularity. Although modularity maximization is a very popular criterion, it presents some defects, the main ones being resolution limit and degeneracy. The former refers to the fact that in some cases small clusters may not be detected, and they remain hidden within another cluster, as reported in [18,20]. The latter is related to the possible presence of several high modularity partitions which makes it hard to find the global optimum [20]. Some methods to attenuate these issues are presented in [2,22,29,30]. For a more detailed discussion of the strengths and weaknesses of modularity, see [10,17,18].

With regard to complexity, modularity maximization is a **NP**-hard problem, as shown by Brandes et al. [8]. Many heuristics and a few exact algorithms [1,33] were proposed in the literature to maximize (1). For an in-depth survey with over 450 references, see [17].

Heuristics are usually hierarchical methods, which yield a nested set of partitions (or in other words $2n - 1$ clusters of the n given entities which are pairwise disjoint or included one into another). Hierarchical algorithms can be divided into agglomerative ones, which proceed from an initial partition in which each cluster contains exactly one entity by successive mergings of two clusters at a time, and divisive heuristics which proceed from an initial partition with a single cluster containing all the n entities, by successive bipartitions of a chosen cluster until this is no more profitable.

The most difficult step is the bipartition, which may be **NP**-hard in itself. Again, this was shown to be the case by Brandes et al. for modularity maximization [8]. For moderate size instances the bipartition problem can be solved exactly using tools from combinatorial optimization in a locally optimal heuristic [9,11,13].

In this paper we do the same for bipartite graphs instead of general unipartite graphs. Indeed, bipartite graphs (i.e., graphs with two disjoint sets of red and blue vertices such that every edge joins a red and a blue vertex) appear in many applications (e.g., recommender systems among users and products, actors and films, scientific papers and authors, members of parliament and committees).

Algorithm: Hierarchical divisive heuristic
Input: graph $G = (V, E)$, where $|V| = n$ and $|E| = m$
Output: a partition P of V

```
1  P ← C_1 = {{v_1, v_2, ..., v_n}}
2  k ← 1
3  while k ≤ |P| and ∃ C_i ∈ P not visited
4        do
5           select C_i ∈ P (not visited) with the smallest possible index i
6           partition C_i into C_2i and C_2i+1 maximizing the modularity
7           if Q(C_2i) + Q(C_2i+1) ≥ Q(C_i)
8              then
9                    P ← (P ∪ {C_2i} ∪ {C_2i+1}) \ {C_i}
10                   k ← k + 1
11          end if
12 end while
```

Fig. 1 The hierarchical divisive heuristic

For bipartite graphs Eq. (1) has been modified, leading to the definition of the bipartite modularity [3]:

$$Q_b = \sum_{c=1}^{N_c} \left(\frac{m_c}{m} - \frac{R_c B_c}{m^2} \right), \tag{2}$$

where R_c represents the sum of the degrees of the red vertices in the cluster c, and B_c is the same for the blue vertices. Comparing the bipartite modularity (2) to the general modularity of Eq. (1), it appears that the term $\frac{D_c^2}{4m^2}$ has been replaced by $\frac{R_c B_c}{m^2}$ to take into account the fact that in bipartite graphs each edge joins a red vertex to a blue vertex, unlike the general case where an edge can join any pair of vertices. As in the previous case one wants to maximize bipartite modularity. Some authors stated that bipartite modularity maximization is **NP**-hard [34]. Unfortunately, the proof was not correct, as shown in [14]. Thus, to the best of our knowledge, the complexity of bipartite modularity maximization is still an open problem. The rest of the paper is organized as follows: in Sect. 2 we propose a heuristic obtained by adapting to the bipartite case an existing locally optimal heuristic originally proposed for general graphs [11]. In Sect. 3 we present three reformulations of the model for the bipartition problem. In Sect. 4 we compare these models and discuss the results obtained, and Sect. 5 reports the conclusions.

2 Hierarchical divisive heuristic

The hierarchical divisive heuristic proposed in [11] starts from an initial partitions with one cluster containing all the vertices, and then each cluster is recursively divided into two clusters in an optimal way (i.e., the two new clusters provide the largest possible increase of modularity). The division stops when it does not improve the modularity anymore. A template for hierarchical divisive heuristics is given in Fig. 1.

It turns out that, in order to find the optimal bipartition of a cluster c into two clusters c_1 and c_2, an optimization problem has to be solved at each step. In [11] this

problem has been modeled basing on the quadratic mixed-integer program formulation proposed in [33], with the number of clusters set to 2, and solved using CPLEX [21]. However, some reformulations of this model, obtained by means of techniques similar to these presented in [7,15] have been recently proposed in [9,13], and the best one of them with respect to the computational time needed to obtain the optimal solution is the following:

$$\max \frac{1}{m}\left(\sum_{\{v_i,v_j\}\in E_c}\left(2S_{i,j}-Y_i-Y_j+1\right)-\frac{1}{2m}\left({D_1}^2+\frac{{D_c}^2}{2}-D_1D_c\right)\right) \quad (3)$$

$$\text{s.t. } \forall\{v_i,v_j\}\in E_c \quad S_{i,j}\le Y_i \quad (4)$$

$$\forall\{v_i,v_j\}\in E_c \quad S_{i,j}\le Y_j \quad (5)$$

$$D_1=\sum_{v_i\in V_c}k_iY_i \quad (6)$$

$$Y_g=0, \quad g=\min\{j\,|\,k_j=\max\{k_i,\ \forall v_i\in V_c\}\} \quad (7)$$

$$\forall\{v_i,v_j\}\in E_c \quad S_{i,j}\in\mathbb{R} \quad (8)$$

$$D_1\in\mathbb{R} \quad (9)$$

$$\forall v_i\in V_c \quad Y_i\in\{0,1\}, \quad (10)$$

where E_c and V_c are respectively the set of edges and the set of vertices belonging to the cluster c, Y_i is a binary variable equals to 1 if vertex i belongs to the cluster c_1, and 0 otherwise, D_1 is the sum of degrees of the vertices inside c_1, $S_{i,j}$ is the variable representing the linearization of Y_iY_j, the term $2S_{i,j}-Y_i-Y_j+1$ is equal to 1 if the edge $\{v_i,v_j\}$ has both end vertices either in c_1 or in c_2 (so the sum extended to all edges $\{v_i,v_j\}\in E_c$ corresponds to m_1+m_2 in definition (2)), m is the number of edges of the graph and k_i is the degree of the vertex i. The constraint (7) is a symmetry breaking constraint which fixes the vertex having larger degree (in case of ties that one having smallest index is selected) to belong to cluster c_2, and it is used to avoid the symmetric solution consisting on swapping cluster c_1 with cluster c_2. More details about how to obtain this formulation can be found in [9,13].

Starting from the model (3)–(10), we adapt it to the bipartite case obtaining a formulation called *BOB* (Bipartite Optimal Bipartition), from which we will derive some reformulations in Sect. 3. As notation, V_{R_c} and V_{B_c} are respectively the set of red vertices and blue vertices belonging to the cluster c, R_1 and B_1 are respectively the sum of degrees of red and blue vertices in c_1, while R_c and B_c are two parameters, known before the splitting, corresponding to the sum of degrees of red and blue vertices in the cluster c. More formally, the new variables and parameters of the model *BOB* are presented in the following.

Since the first p vertices are red, and the other $n-p$ vertices are blue, we can define the two sets of blue and red vertices as:

$$V_{R_c}=\{v_1,\ldots,v_p\}$$
$$V_{B_c}=\{v_{p+1},\ldots,v_n\}.$$

We should also define these two parameters:

$$R_c = \sum_{v_i \in V_{R_c}} k_i \tag{11}$$

$$B_c = \sum_{v_j \in V_{B_c}} k_j. \tag{12}$$

Moreover, we shall define the following variables:

$$R_1 = \sum_{v_i \in V_{R_c}} k_i Y_i \tag{13}$$

$$B_1 = \sum_{v_j \in V_{B_c}} k_j Y_j. \tag{14}$$

Furthermore, the following relationships hold:

$$R_c = R_1 + R_2$$
$$B_c = B_1 + B_2,$$

where the variables R_2 and B_2 represent, respectively, the sum of degrees of red and blue vertices in the clusters 2 obtained by splitting the cluster c.

These new parameters and variables are related to the corresponding ones of the model (3)–(10) by means of these relationships:

$$V_c = V_{R_c} \cup V_{B_c} \tag{15}$$

$$D_c = R_c + B_c \tag{16}$$

$$D_1 = R_1 + B_1. \tag{17}$$

Thus, the *BOB* model is the following:

$$\max \frac{1}{m} \left(\sum_{\{v_i, v_j\} \in E_c} \left(2S_{i,j} - Y_i - Y_j + 1\right) - \frac{1}{m} \left(2R_1 B_1 - B_c R_1 - R_c B_1 + R_c B_c\right) \right) \tag{18}$$

$$\text{s.t. } \forall \{v_i, v_j\} \in E_c \quad S_{i,j} \leq Y_i \tag{19}$$

$$\forall \{v_i, v_j\} \in E_c \quad S_{i,j} \leq Y_j \tag{20}$$

$$R_1 = \sum_{v_i \in V_{R_c}} k_i Y_i \tag{21}$$

$$B_1 = \sum_{v_j \in V_{B_c}} k_j Y_j \tag{22}$$

$$Y_g = 1, \quad g = \min\{j \mid k_j = \max\{k_i, \forall v_i \in V_c\}\} \tag{23}$$
$$R_1 \in \mathbb{R} \tag{24}$$
$$B_1 \in \mathbb{R} \tag{25}$$
$$\forall \{v_i, v_j\} \in E_c \quad S_{i,j} \in \mathbb{R} \tag{26}$$
$$\forall v_i \in V_c \quad Y_i \in \{0, 1\}. \tag{27}$$

The objective function corresponds to (2), when $N_c = 2$. Note that the variables R_2 and B_2 do not appear, since they can be expressed respectively as $R_c - R_1$ and $B_c - B_1$. Note that Y_g is fixed to 1 in the constraint (23), whereas it was fixed to 0 in the constraint (7) of the unipartite model, since computational experiments show this choice is better.

3 Reformulations of the optimal bipartition model

The formulation (3)–(10) presented in [9,13], from which we derive the *BOB* model, can be solved efficiently by CPLEX as it is a convex Mixed Quadratic Programming (cMIQP) problem. Unfortunately, the *BOB* model does not belong to this class, as it is a Mixed Integer Nonlinear Programming (MINLP) problem, thus to solve it directly general MINLP solvers should be used, but this would be too much time demanding. Hence, we can reformulate the *BOB* model, in order to obtain Mixed Integer Linear Programming (MILP) or cMIQP models which can be solved more efficiently by CPLEX.

3.1 Fortet reformulation

Since the nonlinearity of the *BOB* formulation comes from the term $R_1 B_1$ in the objective function (18), which involves products of binary variables (due to definitions of R_1 and B_1 provided by constraints (21)–(22)), one can employ the Fortet inequalities [16] to obtain an exact reformulation (i.e., a reformulation which preserves all local and global optima [23]) of the model *BOB*, called BOB_{1a}:

$$\max \frac{1}{m}\left(\sum_{\{v_i,v_j\}\in E_c}\left(2S_{i,j} - Y_i - Y_j + 1\right)\right.$$
$$\left. -\frac{1}{m}\left(R_c B_c - B_c R_1 - R_c B_1 + \sum_{v_i \in V_{R_c}} \sum_{v_j \in V_{B_c}} 2k_i k_j W_{i,j}\right)\right) \tag{28}$$
$$\text{s.t. } \forall \{v_i, v_j\} \in E_c \quad S_{i,j} \leq Y_i \tag{29}$$
$$\forall \{v_i, v_j\} \in E_c \quad S_{i,j} \leq Y_j \tag{30}$$
$$R_1 = \sum_{v_i \in V_{R_c}} k_i Y_i \tag{31}$$
$$B_1 = \sum_{v_j \in V_{B_c}} k_j Y_j \tag{32}$$

$$\forall v_i \in V_{R_c},\ \forall v_j \in V_{B_c} \quad W_{i,j} \geq Y_i + Y_j - 1 \tag{33}$$
$$Y_g = 1, \quad g = \min\{j \mid k_j = \max\{k_i,\ \forall v_i \in V_c\}\} \tag{34}$$
$$R_1 \in \mathbb{R} \tag{35}$$
$$B_1 \in \mathbb{R} \tag{36}$$
$$\forall \{v_i, v_j\} \in E_c \quad S_{i,j} \in \mathbb{R} \tag{37}$$
$$\forall v_i \in V_{R_c},\ \forall v_j \in V_{B_c} \quad W_{i,j} \in \mathbb{R}_0^+ \tag{38}$$
$$\forall v_i \in V_c \quad Y_i \in \{0, 1\}. \tag{39}$$

Each term $Y_i Y_j$ which would appear in the objective function inside the term $R_1 B_1$ has been replaced by a new variable $W_{i,j}$. Equation (33) and the nonnegativity of variables $W_{i,j}$ expressed in (38) are the Fortet inequalities needed to define $W_{i,j}$. Note that only two constraints have been adjoined, even if the Fortet inequalities are four: since the variables $W_{i,j}$ are minimized in the objective function, two inequalities can be discarded.

3.2 Compact Fortet reformulation

We can actually obtain a more compact model. First, using Eqs. (11)–(14) the objective function (28) can be rewritten as:

$$\frac{1}{m}\left(\sum_{\{v_i,v_j\}\in E_c}\left(2S_{i,j}-Y_i-Y_j+1\right)-\sum_{v_i\in V_{R_c}}\sum_{v_j\in V_{B_c}}\frac{k_i k_j}{m}\left(2W_{i,j}-Y_i-Y_j+1\right)\right). \tag{40}$$

At this point, let $a_{i,j}$ be the generic term of G's adjacency matrix, which is equal to 1 if there exists the edge (i, j), and 0 otherwise, and let $H_{i,j}$ be the parameter defined as:

$$\forall v_i \in V_{R_c},\ \forall v_j \in V_{B_c} \quad H_{i,j} = a_{i,j} - \frac{k_i k_j}{m}.$$

It is not difficult but tedious to show that these considerations let us to rewrite the model BOB_{1a} in a more compact way, obtaining the following BOB_{1b} model:

$$\max \frac{1}{m}\sum_{v_i\in V_{R_c}}\sum_{v_j\in V_{B_c}} H_{i,j}\left(2W_{i,j} - Y_i - Y_j + 1\right) \tag{41}$$
$$\text{s.t. } \forall v_i \in V_{R_c},\ \forall v_j \in V_{B_c} : H_{i,j} < 0 \quad W_{i,j} \geq 0 \tag{42}$$
$$\forall v_i \in V_{R_c},\ \forall v_j \in V_{B_c} : H_{i,j} < 0 \quad W_{i,j} \geq Y_i + Y_j - 1 \tag{43}$$
$$\forall v_i \in V_{R_c},\ \forall v_j \in V_{B_c} : H_{i,j} > 0 \quad W_{i,j} \leq Y_i \tag{44}$$
$$\forall v_i \in V_{R_c},\ \forall v_j \in V_{B_c} : H_{i,j} > 0 \quad W_{i,j} \leq Y_j \tag{45}$$
$$Y_g = 1, \quad g = \min\{j \mid k_j = \max\{k_i,\ \forall v_i \in V_c\}\} \tag{46}$$
$$\forall v_i \in V_{R_c},\ \forall v_j \in V_{B_c} \quad W_{i,j} \in \mathbb{R} \tag{47}$$
$$\forall v_i \in V_c \quad Y_i \in \{0, 1\}. \tag{48}$$

3.3 Square reformulation

It is possible to reformulate the *BOB* model to have only squares as nonlinearities in the objective function, and thus obtaining a cMIQP which can be solved by CPLEX as done for the unipartite case. Consider this part of the objective function (18):

$$2R_1B_1 - B_cR_1 - R_cB_1. \tag{49}$$

First at all, we can write the last two terms as:

$$\begin{aligned} B_cR_1 &= (B_c + R_c)R_1 - R_cR_1 \\ R_cB_1 &= (B_c + R_c)B_1 - B_cB_1, \end{aligned}$$

therefore we can rewrite (49) as:

$$2R_1B_1 - (B_c + R_c)(R_1 + B_1) + B_cB_1 + R_cR_1.$$

If we are able to introduce the terms ${B_1}^2$ and ${R_1}^2$, we can replace them and $2R_1B_1$ with $(R_1 + B_1)^2$. To do that, consider first the term R_cR_1. Using definitions (11) and (13), we can write it this way:

$$R_cR_1 = \sum_{v_i \in V_{R_c}} k_i \sum_{v_j \in V_{R_c}} k_jY_j = \sum_{v_i \in V_{R_c}} {k_i}^2Y_i + \sum_{v_i \in V_{R_c}} \sum_{v_j \in V_{R_c}: j<i} k_ik_j \left(Y_i + Y_j\right). \tag{50}$$

As stated earlier, we are interested in adding the term ${R_1}^2$. We can express it as:

$${R_1}^2 = \sum_{v_i \in V_{R_c}} k_iY_i \sum_{v_j \in V_{R_c}} k_jY_j = \sum_{v_i \in V_{R_c}} {k_i}^2Y_i + \sum_{v_i \in V_{R_c}} \sum_{v_j \in V_{R_c}: j<i} 2k_ik_jY_iY_j, \tag{51}$$

where we use the fact that $Y_i = {Y_i}^2$, since these are binary variables. Comparing (50) and (51), it appears that we can write R_cR_1 in terms of ${R_1}^2$ in this way:

$$\begin{aligned} R_1R_c &= {R_1}^2 - \sum_{v_i \in V_{R_c}} \sum_{v_j \in V_{R_c}: j<i} 2k_ik_jY_iY_j + \sum_{v_i \in V_{R_c}} \sum_{v_j \in V_{R_c}: j<i} k_ik_j \left(Y_i + Y_j\right) \\ &= {R_1}^2 + \sum_{v_i \in V_{R_c}} \sum_{v_j \in V_{R_c}: j<i} k_ik_j \left(Y_i + Y_j - 2Y_iY_j\right) = {R_1}^2 \\ &\quad + \sum_{v_i \in V_{R_c}} \sum_{v_j \in V_{R_c}: j<i} k_ik_j \left(Y_i - Y_j\right)^2. \end{aligned}$$

We can obtain a similar result for the term B_1B_c. More precisely, we can write:

$$B_1B_c = {B_1}^2 + \sum_{v_i \in V_{B_c}} \sum_{v_j \in V_{B_c}: j<i} k_ik_j \left(Y_i - Y_j\right)^2.$$

Finally, Eq. (49) can be reformulated as:

$$\begin{aligned}
&2R_1B_1 - (B_c + R_c)(R_1 + B_1) + B_cB_1 + R_cR_1 = 2R_1B_1 - (B_c + R_c)(R_1 + B_1) \\
&\quad + {R_1}^2 + {B_1}^2 + \sum_{v_i \in V_{R_c}} \sum_{v_j \in V_{R_c}: j<i} k_i k_j \left(Y_i - Y_j\right)^2 + \sum_{v_i \in V_{B_c}} \sum_{v_j \in V_{B_c}: j<i} k_i k_j \left(Y_i - Y_j\right)^2 \\
&= (R_1 + B_1)^2 - (B_c + R_c)(R_1 + B_1) + \sum_{v_i \in V_{R_c}} \sum_{v_j \in V_{R_c}: j<i} k_i k_j \left(Y_i - Y_j\right)^2 \\
&\quad + \sum_{v_i \in V_{B_c}} \sum_{v_j \in V_{B_c}: j<i} k_i k_j \left(Y_i - Y_j\right)^2.
\end{aligned}$$

Using relationships (15)–(17), we can now write the model BOB_2:

$$\max \frac{1}{m}\left(\sum_{\{v_i,v_j\}\in E_c}\left(2S_{i,j} - Y_i - Y_j\right) + |E_c| - \frac{1}{m}\left({D_1}^2 - D_1 D_c + \sum_{v_i \in V_{R_c}} \sum_{v_j \in V_{R_c}: j<i} k_i k_j \left(Y_i - Y_j\right)^2 + \sum_{v_i \in V_{B_c}} \sum_{v_j \in V_{B_c}: j<i} k_i k_j \left(Y_i - Y_j\right)^2 + R_c B_c\right)\right) \tag{52}$$

$$\text{s.t. } \forall \{v_i, v_j\} \in E_c \quad S_{i,j} \le Y_j \tag{53}$$

$$\forall \{v_i, v_j\} \in E_c \quad S_{i,j} \le Y_i \tag{54}$$

$$D_1 = \sum_{v_i \in V_c} k_i Y_i \tag{55}$$

$$Y_g = 1, \quad g = \min\{j \mid k_j = \max\{k_i,\ \forall v_i \in V_c\}\} \tag{56}$$

$$\forall \{v_i, v_j\} \in E_c \quad S_{i,j} \in \mathbb{R} \tag{57}$$

$$D_1 \in \mathbb{R} \tag{58}$$

$$\forall v_i \in V_c \quad Y_i \in \{0, 1\}. \tag{59}$$

In this model the objective function is similar to the objective function of the unipartite model (3)–(10), and the set of constraints is the same, thus underlying the strong relationship between these problems.

3.4 Binary decomposition

Another way to reformulate the *BOB* model to obtain a MILP is to employ the binary decomposition technique, which has been also employed for general graph partitioning problems in [7]. The first step is to define the variables R_1 and B_1 as:

$$R_1 = \sum_{v_i \in V_{R_c}} k_i Y_i = \sum_{h=0}^{t_R} 2^h a_h \tag{60}$$

$$B_1 = \sum_{v_j \in V_{B_c}} k_j Y_j = \sum_{l=0}^{t_B} 2^l b_l, \tag{61}$$

where a_h and b_l are binary variables. The maximum value which can be taken by R_1 (B_1) is equal to R_c (B_c). Furthermore, the maximum possible value for R_1 (B_1) is $2^{t_R+1}-1$ ($2^{t_B+1}-1$). Thus, parameters t_R and t_B are defined respectively as:

$$2^{t_R+1}-1 \geq R_c \Rightarrow t_R = \lceil \log_2(R_c+1)-1 \rceil \tag{62}$$

$$2^{t_B+1}-1 \geq B_c \Rightarrow t_B = \lceil \log_2(B_c+1)-1 \rceil. \tag{63}$$

The product $R_1 B_1$ can then be expressed as:

$$R_1 B_1 = \sum_{h=0}^{t_R} 2^h a_h \sum_{l=0}^{t_B} 2^l b_l = \sum_{h=0}^{t_R}\sum_{l=0}^{t_B} 2^{l+h} a_h b_l. \tag{64}$$

Finally, to linearize the products $a_h b_l$, we introduce the variables $R_{l,h}$, and the corresponding Fortet inequalities:

$$\forall l \in \{0,\ldots,t_B\},\ \forall h \in \{0,\ldots,t_R\} \quad R_{l,h} \geq 0 \tag{65}$$

$$\forall l \in \{0,\ldots,t_B\},\ \forall h \in \{0,\ldots,t_R\} \quad R_{l,h} \geq a_h + b_l - 1. \tag{66}$$

This leads to the model BOB_3:

$$\max \frac{1}{m}\left(\sum_{\{v_i,v_j\}\in E_c}\left(2S_{i,j}-Y_i-Y_j\right)+|E_c| - \frac{1}{m}\left(R_c B_c - B_c R_1 - R_c B_1 + 2\sum_{h=0}^{t_R}\sum_{l=0}^{t_B} R_{l,h}\right)\right) \tag{67}$$

$$\text{s.t. } \forall\{v_i,v_j\}\in E_c \quad S_{i,j} \leq Y_i \tag{68}$$

$$\forall\{v_i,v_j\}\in E_c \quad S_{i,j} \leq Y_j \tag{69}$$

$$R_1 = \sum_{v_i \in V_{R_c}} k_i Y_i \tag{70}$$

$$B_1 = \sum_{v_j \in V_{B_c}} k_j Y_j \tag{71}$$

$$R_1 = \sum_{h=0}^{t_R} 2^h a_h \tag{72}$$

$$B_1 = \sum_{l=0}^{t_B} 2^l b_l \tag{73}$$

$$\forall l \in \{0,\ldots,t_B\},\ \forall h \in \{0,\ldots,t_R\} \quad R_{l,h} \geq a_h + b_l - 1 \tag{74}$$

$$Y_g = 1, \quad g = \min\{j \mid k_j = \max\{k_i,\ \forall v_i \in V_c\}\} \tag{75}$$

$$R_1 \in \mathbb{R} \tag{76}$$

$$B_1 \in \mathbb{R} \tag{77}$$

$$\forall \{v_i, v_j\} \in E_c \quad S_{i,j} \in \mathbb{R} \tag{78}$$

$$\forall l \in \{0, \ldots, t_B\},\ \forall h \in \{0, \ldots, t_R\} \quad R_{l,h} \in \mathbb{R}_0^+ \tag{79}$$

$$\forall v_i \in V_c \quad Y_i \in \{0, 1\} \tag{80}$$

$$\forall h \in \{0, \ldots, t_R\} \quad a_h \in \{0, 1\} \tag{81}$$

$$\forall l \in \{0, \ldots, t_B\} \quad b_l \in \{0, 1\}. \tag{82}$$

4 Computational results

We present the comparison of the numerical results obtained by the proposed reformulations on a 2.4GHz Intel Xeon CPU of a computer with 24 GB RAM running Linux and CPLEX 12.2 [21], with the best configuration we found, that is MIP cutting plane generation disabled and branching based on pseudo reduced costs. Table 1 presents the details about the bipartite graphs used in our tests: p represents the number of red vertices, n is the total number of vertices (that is, red and blue vertices), while m refers to the number of edges. Note that the instance graph product contains some unconnected components, since $m < n - 1$. The datasets can be found on Pajek [5].

Table 2 reports the details on the size and type of the various formulations presented in this paper, namely the number of binary and continuous variables, the number of constraints and the kind of formulation. The original formulation *BOB* is a MINLP

Table 1 Informations about the graphs used in tests

ID	Graph	p	n	m
1	Southern women	18	32	89
2	Supreme Court voting (yes)	26	35	147
3	Supreme Court voting (not)	26	35	86
4	Social work	18	36	99
5	Wafa-CEO	26	41	98
6	Divorces	50	59	225
7	Hollywood movies	62	102	192
8	Scotland interlocks	108	244	358
9	Graph product	314	674	613
10	Network science	960	2,549	2,580

Table 2 Details of the size and kind of the formulations (note that $|V_c| = |V_{R_c}| + |V_{B_c}|$)

Formulation	Binary variables	Continuous variables	Constraints	Type
BOB	$\|V_c\|$	$\|E_c\|+2$	$2\|E_c\|+3$	MINLP
BOB_{1a}	$\|V_c\|$	$\|E_c\|+\|V_{R_c}\|\|V_{B_c}\|+2$	$2\|E_c\|+2\|V_{R_c}\|\|V_{B_c}\|+3$	MILP
BOB_{1b}	$\|V_c\|$	$\|V_{R_c}\|\|V_{B_c}\|$	$2\|V_{R_c}\|\|V_{B_c}\|+1$	MILP
BOB_2	$\|V_c\|$	$\|E_c\|+1$	$2\|E_c\|+2$	cMIQP
BOB_3	$\|V_c\|+t_R+t_B+2$	$\|E_c\|+(t_R+1)(t_B+1)+2$	$2\|E_c\|+2(t_R+1)(t_B+1)+5$	MILP

Table 3 Comparison between the different reformulations

ID	N_c	Q	BOB_{1a}		BOB_{1b}		BOB_2		BOB_3	
			Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time
1	4	0.3409	437	0.30	**72**	**0.19**	3,372	1.32	670	0.39
2	2	0.2704	154	0.19	**10**	**0.09**	1,074	1.39	618	0.43
3	2	0.4538	45	0.14	**6**	**0.07**	132	0.14	183	0.19
4	5	0.2883	2,169	1.46	**1,360**	1.24	67,364	13.11	1,854	**0.93**
5	4	0.3329	1,963	1.25	**276**	0.44	117,997	23.84	647	**0.39**
6	3	0.1876	1,123	0.77	**27**	**0.16**	2,497,924	646.78	2,521	2.12
7	8	0.4939	1,223,370	4440.04	407,104	3038.06	–	–	**38,910**	**5.26**
8	13	0.7153	–	–	–	–	–	–	**3,793**	**5.81**
9	139	0.9363	–	–	–	–	–	–	**71,927,548**	**15450.40**
10	414	0.9696	–	–	–	–	–	–	**91,917**	**38.49**

Bold values indicate the best values in terms of number of nodes and execution time

(due to the term $R_1 B_1$ in the objective function), hence it is difficult to solve. The MILP formulations BOB_{1a} and BOB_{1b} are those based on Fortet inequalities; the latter is more compact than the former, both in terms of number of variables and constraints, and this has an impact on the computational results. The formulation BOB_2 is the most compact. Nevertheless, it is the one which provides the worst results in terms of computational time, as reported in Table 3, due to the complexity of its objective function. Finally, the formulation BOB_3 is based on the binary decomposition. The terms t_R and t_B introduced respectively by Eqs. (62) and (63) are related to the binary variables needed to define the binary decomposition. Nonnegativity of a variable is considered as a constraint in Table 2.

In Table 3 we compare the results obtained by the different reformulations for the graphs tested. N_c and Q are respectively the number of clusters and the modularity of the partition obtained as solution. Since each model represents an exact reformulation of the *BOB* formulation, the solutions in terms of number of clusters and modularity value are the same. For each formulation, the total number of Branch-and-Bound nodes and the computational time are reported. It turns out that the compact formulation BOB_{1b} outperforms the original Fortet linearization BOB_{1a}, and it yields the best results for small graphs. For larger graphs, however, the formulation BOB_3 which employs the binary decomposition provides the best results. As a matter of fact, for the last three instances only BOB_3 was able to solve the problem, whilst the Fortet linearizations failed because of memory space overhead. The formulation BOB_2 is the worst in terms of computing time. We do not provide the results obtained using the *BOB* formulation, since the computational time is too large. For example, solving the instance 1 (Southern Women) using the divisive heuristics with the *BOB* model and the MINLP solver COUENNE [6] takes about 40 seconds, whereas using the worst reformulation (BOB_2) the solution is obtained in 1.32 seconds.

Finally, as for the quality of the partitions provided by our heuristic, we compare our results with the values of modularity obtained by other heuristics. More precisely, we consider the label propagation algorithm LPAb proposed by Barber and Clark

Table 4 Comparison between different algorithms for bipartite modularity maximization on three instances: 1 (Southern women), 8 (Scotland interlocks) and 10 (network science)

Algorithm	Graph ID 1		Graph ID 8		Graph ID 10	
	N_c	Q	N_c	Q	N_c	Q
Adaptive-BRIM	4	**0.3455**	13	0.6861	107	0.8894
LPA-BRIM	4	**0.3455**	17	0.7141	500	0.9363
CNM	3	0.3430	32	0.7008	414	0.9695
MSG	3	0.3411	30	0.7004	414	0.9687
LPAb	4	0.3192	60	0.5783	691	0.7807
LPAb-MSG	4	**0.3455**	16	**0.7194**	414	0.9695
LPAb+	4	**0.3455**	16	**0.7194**	415	**0.9696**
Divisive	4	0.3409	13	0.7153	414	**0.9696**

Bold values indicate the best values in terms of modularity (Q)

[4], which is an adaptation to bipartite case of LPA (Label Propagation Algorithm), proposed by Raghavan, Albert, and Kumara in [28]. LPA works by initially assigning a unique label to each vertex, then vertices are visited in a random order and the label of a visited vertex becomes the label shared by the majority of its neighbors. This process is iterated until convergence. Another heuristic considered is the adaptive BRIM (Bipartite Recursively Induced Modules) proposed by Barber [3]: starting from an arbitrary partition of the blue vertices, the corresponding partition of the red vertices which maximizes the modularity is easy to find. Then, the partition of the red vertices is used to update that of blue vertices and the process continues until convergence. Some greedy algorithms are proposed as well. For example, the extension to bipartite graphs of the greedy agglomerative algorithm CNM of Clauset, Newman, and Moore [12], that is an algorithm which improves the performances of the heuristic proposed by Newman in [26] by using more sophisticated data structures. The multistep greedy agglomerative algorithm MSG by Schuetz and Caflish [31,32] proposes some tricks to solve one of the problems of CNM, i.e., the identification of very large communities. Liu and Murata proposed some extensions of label propagation algorithms: in [24], they presented a combination of LPA and BRIM (LP-BRIM), while in [25] they proposed a combination of LPAb and MSG, as well as LPAb+, that is a combination of a modified version LPAb, called LPAb' (where labels of blue and red vertices are not updated randomly as for LPAb, but by turn) and MSG.

For bipartite graphs there are not many instances of the literature that are extensively used. In [25], the results are available only for three instances among the ones we tested (e.g., Southern women, Scotland interlocks and network science), and they are reported in Table 4. Our algorithm appears in the last row of the table, and it is referred as "Divisive". It appears that our heuristic presents results comparable with those provided by the other algorithms. More precisely, it provides the best result (together with LPAb+) for network science graph, the second best result for Scotland interlocks, while many other heuristics find slightly better results for the small Southern women graph. Note that our version of the network science graph consists of 2,580 edges, whereas the one of [25] has 2,579 edges.

5 Conclusions

This paper presents the extension to bipartite graphs of a previously proposed heuristic for modularity maximization in general graphs. We first propose some reformulations of the mathematical programming model used by this heuristic to compute, at each step, the optimal splitting of a cluster into two new clusters, starting from the best model presented in [9,13] for the general case. A compact model using Fortet linearization is the best one for small instances, and the model using binary decomposition is more suitable for larger instances. Indeed, it could solve an instance with 102 nodes and 192 edges in about 600 times less computing time than the second best formulation. Note that formulation BOB_2 was not introduced for its performances in terms of computing time, but in order to understand better which reformulations are efficient in terms of size of instances solved, as well as the close relationship between the models for unipartite graphs and for bipartite graphs.

Acknowledgments The authors would like to thank Sonia Cafieri and Leo Liberti for the precious suggestions and comments. Financial support by grants: Digiteo 2009-14D "RMNCCO", Digiteo 2009-55D "ARM" is gratefully acknowledged.

References

1. Aloise, D., Cafieri, S., Caporossi, G., Hansen, P., Perron, S., Liberti, L.: Column generation algorithms for exact modularity maximization in networks. Phys. Rev. E **82**(4), 046112 (2010)
2. Arenas, A., Fernández, A., Gómez, S.: Analysis of the structure of complex networks at different resolution levels. New J. Phys. **10**(5), 053039 (2008)
3. Barber, M.J.: Modularity and community detection in bipartite networks. Phys. Rev. E **76**(6), 066102 (2007)
4. Barber, M.J., Clark, J.W.: Detecting network communities by propagating labels under constraints. Phys. Rev. E **80**(2), 026129 (2009)
5. Batagelj, V., Mrvar, A.: Pajek datasets. http://vlado.fmf.uni-lj.si/pub/networks/data (2006)
6. Belotti, P., Lee, J., Liberti, L., Margot, F., Wächter, A.: Branching and bounds tightening techniques for non-convex MINLP. Optim. Methods Softw. **24**(4–5), 597–634 (2009)
7. Boulle, M.: Compact mathematical formulation for graph partitioning. Optim. Eng. **5**(3), 315–333 (2004)
8. Brandes, U., Delling, D., Gaertler, M., Görke, R., Hoefer, M., Nikoloski, Z., Wagner, D.: On modularity clustering. IEEE Trans. Knowl. Data Eng. **20**(2), 172–188 (2008)
9. Cafieri, S., Costa, A., Hansen P.: Reformulation of a model for hierarchical divisive graph modularity maximization. Ann. Operat. Res. (accepted)
10. Cafieri, S., Hansen, P., Liberti, L.: Loops and multiple edges in modularity maximization of networks. Phys. Rev. E **81**(4), 046102 (2010)
11. Cafieri, S., Hansen, P., Liberti, L.: Locally optimal heuristic for modularity maximization of networks. Phys. Rev. E **83**(5), 056105 (2011)
12. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Phys. Rev. E **70**(6), 066111 (2004)
13. Costa A.: Applications of reformulation in mathematical programming. PhD thesis, École Polytechnique (2012)
14. Costa, A., Hansen, P.: Comment on "Evolutionary method for finding communities in bipartite networks". Phys. Rev. E **84**(5), 058101 (2011)
15. Fan, N., Pardalos, P.M.: Linear and quadratic programming approaches for the general graph partitioning problem. J. Glob. Optim. **48**(1), 57–71 (2010)
16. Fortet, R.: Applications de l'algèbre de Boole en recherche opérationelle. Revue Française de Recherche Opérationelle **4**(14), 17–26 (1960)

17. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**(3–5), 75–174 (2010)
18. Fortunato, S., Barthélemi, M.: Resolution limit in community detection. Proc. Nat. Acad. Sci. USA **104**(1), 36–41 (2007)
19. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proc. Nat. Acad. Sci. USA **99**(12), 7821–7826 (2007)
20. Good, B.H., de Montjoye, Y.-A., Clauset, A.: Performance of modularity maximization in practical contexts. Phys. Rev. E **81**(4), 046106 (2010)
21. IBM. ILOG CPLEX 12.2 User's Manual. IBM (2010)
22. Kumpula, J.M., Saramäki, J., Kaski, K., Kertész, J.: Limited resolution and multiresolution methods in complex network community detection. Fluctuations Noise Lett. **7**(3), 209 (2007)
23. Liberti, L.: Reformulations in mathematical programming: definitions and systematics. RAIRO-OR **43**(1), 55–86 (2009)
24. Liu, X., Murata, T.: Community detection in large-scale bipartite networks. In: IEEE/WIC/ACM international conference on web intelligence and Intelligent Agent Technologies, pp. 50–57 (2009)
25. Liu, X., Murata, T.: An efficient algorithm for optimizing bipartite modularity in bipartite networks. J. Adv. Comput. Intell. Intell. Inform. **14**(4), 408–415 (2010)
26. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Phys. Rev. E **69**(6), 066133 (2004)
27. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2), 026113 (2004)
28. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. Phys. Rev. E **76**(3), 036106 (2007)
29. Reichardt, J., Bornholdt, S.: Statistical mechanics of community detection. Phys. Rev. E **74**(1), 016110 (2006)
30. Sales-Pardo, M., Guimerà, R., Moreira, A.A., Amaral, L.A.N.: Extracting the hierarchical organization of complex systems. Proc. Nat. Acad. Sci. USA **104**(39), 15224–15229 (2007)
31. Schuetz, P., Caflisch, A.: Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement. Phys. Rev. E **77**(4), 046112 (2008)
32. Schuetz, P., Caflisch, A.: Multistep greedy algorithm identifies community structure in real-world and computer-generated networks. Phys. Rev. E **78**(2), 026112 (2008)
33. Xu, G., Tsoka, S., Papageorgiou, L.G.: Finding community structures in complex networks using mixed integer optimisation. Eur. Phys. J. B **60**(2), 231–239 (2007)
34. Zhan, W., Zhang, Z., Guan, J., Zhou, S.: Evolutionary method for finding communities in bipartite networks. Phys. Rev. E **83**(6), 066120 (2011)