manuscript No.
(will be inserted by the editor)

Reformulation of a model for hierarchical divisive graph modularity maximization

Sonia Cafieri · **Alberto Costa** · **Pierre Hansen**

Received: date / Accepted: date

Abstract Finding clusters, or communities, in a graph, or network is a very important problem which arises in many domains. Several models were proposed for its solution. One of the most studied and exploited is the maximization of the so called modularity, which represents the sum over all communities of the fraction of edges within these communities minus the expected fraction of such edges in a random graph with the same distribution of degrees. As this problem is NP-hard, a few non-polynomial algorithms and a large number of heuristics were proposed in order to find respectively optimal or high modularity partitions for a given graph. We focus on one of these heuristics, namely a divisive hierarchical method, which works by recursively splitting a cluster into two new clusters in an optimal way. This splitting step is performed by solving a convex quadratic program. We propose a compact reformulation of such model, using change of variables, expansion of integers in powers of two and symmetry breaking constraints. The resolution time is reduced by a factor up to 10 with respect to the original formulation.

Keywords clustering · compact reformulation · divisive hierarchical heuristic · modularity maximization.

1 Introduction

A graph, or network, $G = (V,E)$ can be represented as a set V of vertices and a set E of edges connecting pairs of vertices. This model has been intensively used in several domains to represent complex systems (Newman 2010). For instance, the metabolic network studied in biology and bioinformatics (Guimerà et Amaral 2004, Palla et al. 2005), social networks

S. Cafieri
Laboratoire MAIAA, École Nationale de l'Aviation Civile, 7 Ave. E. Belin, F-31055 Toulouse, France
E-mail: sonia.cafieri@enac.fr

A. Costa · P. Hansen
LIX, École Polytechnique, F-91128 Palaiseau, France
E-mail: costa@lix.polytechnique.fr

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

(Girvan and Newman 2002) and other applications in informatics, as recommender systems (Adomavicius and Tuzhilin 2005) or the World Wide Web (Flake et al. 2002).

One of the most important tasks is to identify the structure of such graphs, and in particular to find subsets of vertices, called *communities* or *clusters*, where each cluster contains vertices which are more likely to be connected pairwise with its own vertices than to those belonging to other communities. In order to formalize this idea, different definitions were proposed. One of the most known is provided by Radicchi et al. (2004), with the concepts of *strong community* and *weak community*: a strong community contains vertices having more neighbours inside than neighbours outside the community, whereas in a weak community the total number of inner edges (joining two vertices of the same community) must be greater or equal to half of the number of cut edges (with two vertices in different communities).

Given a graph and a partition, another measure of the extent to which the classes of the partition can be considered to be communities is provided by the famous criterion called *modularity* (Girvan and Newman 2002; Newman and Girvan 2004), which represents the fraction of edges within communities minus the expected fraction of such edges in a random graph with the same degree distribution. Alternatively, given a graph, modularity can be maximized to find an optimal partition, together with its number of clusters and their modularities. See e.g. Fortunato (2010), Fortunato and Barthelemy (2007), Cafieri et al. (2010) for a discussion of the strengths and weaknesses of modularity. Given an unweighted graph G, its modularity Q is defined as:

$$Q = \sum_c Q_c = \sum_c \frac{m_c}{m} - \frac{{D_c}^2}{4m^2}, \tag{1}$$

where 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. The extension of this definition to weighted graphs is presented in Fortunato (2010). In order to obtain good quality partitions, one should maximize the modularity. This is a NP-hard problem, as proved by Brandes et al. (2008).

In the literature, several methods have been proposed to find high modularity partitions: a few exact methods, and several heuristics. Among the exact methods, there is a row generation algorithm originally proposed by Grötschel and Wakabayashi (1989) for general clustering problems, which is similar to the one presented in Brandes et al. (2008), a column generation algorithm proposed by Cafieri et al. (2010), and a mixed integer convex quadratic programming formulation due to Xu et al. (2007). Concerning the heuristics, many algorithms have been proposed. They are presented in the survey of Fortunato (2010). In this paper, we analyze one of these in particular, that is the locally optimal hierarchical divisive approach presented in Cafieri et al. (2011).

The rest of the paper is organized as follows: in Section 2 the heuristic proposed in Cafieri et al. (2011) is presented more in detail, while in Section 3 we introduce our reformulations of the bipartition model. Then, in Section 4 we present numerical results and finally Section 5 concludes the paper.

2 Original model for cluster bipartition

Clustering heuristics are either hierarchical, which aim at finding a set of nested partitions, or partitioning schemes, which aim at finding a single partition or possibly several partitions into given numbers of clusters. Hierarchical heuristics are in principle devised for finding a hierarchy of partitions implicit in the given graph when it corresponds to some situation where hierarchy is observed or postulated. This is often the case, for instance, in social

organization and evolutionary processes. Hierarchical heuristics can be further divided into agglomerative and divisive ones. Hierarchical divisive heuristics (see, e.g. Newman 2006b) proceed from an initial partition containing all the n vertices of the graph and iteratively divide a cluster into two in such a way that the increase in the objective function value is the largest possible, or the decrease in the objective value is the smallest possible. Cluster bipartitions are iterated until a partition into n clusters having each a single entity is obtained. In practice, with an objective function like modularity, bipartitions can be ended once they do not improve the objective function value anymore. A sketch of the divisive algorithm is given in Fig. 1.

Algoritm: Hierarchical divisive algorithm
Input: graph $G = (V,E)$, where $|V| = n$ and $|E| = m$
Output: a hierarchy $H = \{P_1, P_2, \ldots, P_n\}$ of partitions of V

$P_1 \leftarrow C_1 = \{\{v_1, v_2, \ldots, v_n\}\}$
$k \leftarrow 1$
while $k < n$
    do
        select $C_i \in P_k$ with the smallest possible index i
        partition C_i into C_{2i} and C_{2i+1} maximizing the modularity
        $P_{k+1} \leftarrow (P_k \cup \{C_{2k}\} \cup \{C_{2k+1}\}) \setminus \{C_i\}$
end while

Fig. 1: The hierarchical divisive algorithm.

The subproblem of finding a bipartition locally optimizing the modularity criterion is difficult. Brandes et al. (2008) in fact proved that modularity maximization is NP-hard even for two clusters. Cafieri et al. (2011) recently proposed a modularity maximizing divisive heuristic where the optimization subproblem for cluster bipartition is expressed as a quadratic mixed-integer program with a convex relaxation. Binary variables are used to identify to which cluster each vertex and each edge belongs. More precisely, variables $X_{i,j,s}$ for each edge (v_i, v_j) and $s = 1,2$, and variables $Y_{i,1}$ for $i = 1,2,\ldots n$ are defined in such a way that $X_{i,j,s}$ is equal to 1 if the edge (v_i, v_j) is inside the cluster s (i.e., both vertices v_i and v_j are inside the cluster s), and $Y_{i,1}$ is equal to 1 if the vertex v_i is inside the cluster 1, and 0 otherwise. Variables X give rise to two sets of variables, $X_{i,j,1}$ and $X_{i,j,2}$, as an edge may belong to the first cluster, or to the second one, or be a bridge between both of them. As for variables Y, only one set $Y_{i,1}$ suffices as any vertex which does not belong to the first cluster must belong to the second.

Recall the definition of modularity (1). Since a bipartition has to be computed, only two sub-clusters has to be considered, and the sum of degrees of vertices belonging to one of the two sub-clusters can be expressed as a function of the sum of degrees of the other cluster:

$$D_2 = D_c - D_1, \tag{2}$$

where D_1 and D_2 are the sum of the degrees of the vertices inside the two clusters and D_c is a parameter given by the sum of degrees in the cluster c to be bipartitioned (it is equal to $2m$ at the outset). Hence, in the bipartition subproblem the objective function (1) can be rewritten as

$$Q_c = \frac{m_1 + m_2}{m} - \frac{{D_1}^2 + {D_2}^2}{4m^2}, \tag{3}$$

where m_1 and m_2 are respectively the numer of edges inside the two clusters. Using equation (2), equation (3) can be rewritten as

$$Q_c = \frac{m_1+m_2}{m} - \frac{{D_1}^2+(D_c-D_1)^2}{4m^2} = \frac{m_1+m_2}{m} - \frac{{D_1}^2}{2m^2} - \frac{{D_c}^2}{4m^2} + \frac{D_1 D_c}{2m^2}. \tag{4}$$

As for the constraints, the following inequalities are used to impose that any edge (v_i, v_j) with end vertices indiced by i and j can only belong to cluster s if both of its end vertices belong also to that cluster:

$$\begin{array}{l} X_{i,j,1} \leq Y_{i,1} \quad \forall (v_i, v_j) \in E_c \\ X_{i,j,1} \leq Y_{j,1} \quad \forall (v_i, v_j) \in E_c \end{array} \tag{5}$$

and

$$\begin{array}{l} X_{i,j,2} \leq 1 - Y_{i,1} \quad \forall (v_i, v_j) \in E_c \\ X_{i,j,2} \leq 1 - Y_{j,1} \quad \forall (v_i, v_j) \in E_c. \end{array} \tag{6}$$

Furthermore, the number of edges of each of the two clusters and the sum of vertex degrees of the first cluster are expressed as follows:

$$m_s = \sum_{(v_i,v_j)\in E_c} X_{i,j,s} \quad \forall s \in \{1,2\}, \tag{7}$$

$$D_1 = \sum_{v_i \in V_c} k_i Y_{i,1}, \tag{8}$$

where k_i is the degree of the vertex v_i. Hence, the complete formulation proposed in (Cafieri et al. 2011), and called from now *OB* (Optimal Bipartition), is the following:

$$\max \quad \frac{1}{m}\left(m_1 + m_2 - \frac{1}{2m}\left({D_1}^2 + \frac{{D_c}^2}{2} - D_1 D_c\right)\right) \tag{9}$$

$$\text{s.t.} \quad X_{i,j,1} \leq Y_{i,1} \quad \forall (v_i, v_j) \in E_c \tag{10}$$

$$X_{i,j,1} \leq Y_{j,1} \quad \forall (v_i, v_j) \in E_c \tag{11}$$

$$X_{i,j,2} \leq 1 - Y_{i,1} \quad \forall (v_i, v_j) \in E_c \tag{12}$$

$$X_{i,j,2} \leq 1 - Y_{j,1} \quad \forall (v_i, v_j) \in E_c \tag{13}$$

$$m_s = \sum_{(v_i,v_j)\in E_c} X_{i,j,s} \forall s \in \{1,2\} \tag{14}$$

$$D_1 = \sum_{v_i \in V_c} k_i Y_{i,1} \tag{15}$$

$$Y_{i,1} \in \{0,1\} \quad \forall v_i \in V_c \tag{16}$$

$$X_{i,j,s} \geq 0 \quad \forall (v_i, v_j) \in E_c, \forall s \in \{1,2\}. \tag{17}$$

3 Improved formulation of the bipartition problem

It is possible to obtain a compact and more efficient formulation for the *OB* model. This can be done thanks to 3 considerations, which are discussed in the rest of the section: (i) reduction of the number of variables and constraints; (ii) application of the binary decomposition technique; (iii) addition of a symmetry breaking constraint.

3.1 Reduction of number of variables and constraints

Starting from the *OB* model, some simple considerations can allow to remove half of the variables X, as well as to decrease the number of constraints.

Consider the X variables. Looking at the objective function (9) of the *OB* formulation, we notice that it contains the term $m_1 + m_2$, which represents the number of edges in the first cluster plus the number of edges in the second one. Since we are interested in this sum, we do not actually need to know if an edge is in the cluster 1 or 2, but only if it is within a cluster or not. Hence, we can drop the index s of these variables, moving from the original definition:

$$X_{i,j,s} = \begin{cases} 1, & \text{if edge } (v_i, v_j) \text{ belongs to cluster } s, \\ 0, & \text{otherwise,} \end{cases} \tag{18}$$

to the following one:

$$X_{i,j} = \begin{cases} 1, & \text{if edge } (v_i, v_j) \text{ is within cluster 1 or 2,} \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

In other words, we can define $X_{i,j}$ as:

$$X_{i,j} = \begin{cases} 1, & \text{if } Y_i = Y_j, \\ 0, & \text{otherwise,} \end{cases} \tag{20}$$

where we drop the meaningless index 1 from the Y variables. In order to express the relationship between the X and Y variables, the following constraints can be employed:

$$X_{i,j} \geq Y_i - Y_j + 1 \quad \forall (v_i, v_j) \in E_c \tag{21}$$
$$X_{i,j} \geq Y_j - Y_i + 1 \quad \forall (v_i, v_j) \in E_c \tag{22}$$
$$X_{i,j} \leq Y_i + Y_j - 1 \quad \forall (v_i, v_j) \in E_c \tag{23}$$
$$X_{i,j} \leq 1 - Y_i - Y_j \quad \forall (v_i, v_j) \in E_c. \tag{24}$$

Note that, as in the original model, the Y variables are binary and the X variables are positive and continuous. These considerations allow us to reformulate the *OB* model this way:

$$\max \quad \frac{1}{m} \left(\sum_{(v_i, v_j) \in E_c} X_{i,j} - \frac{1}{2m} \left({D_1}^2 + \frac{{D_c}^2}{2} - D_1 D_c \right) \right) \tag{25}$$
$$\text{s.t.} \quad X_{i,j} \geq Y_i - Y_j + 1 \quad \forall (v_i, v_j) \in E_c \tag{26}$$
$$X_{i,j} \geq Y_j - Y_i + 1 \quad \forall (v_i, v_j) \in E_c \tag{27}$$
$$X_{i,j} \leq Y_i + Y_j - 1 \quad \forall (v_i, v_j) \in E_c \tag{28}$$
$$X_{i,j} \leq 1 - Y_i - Y_j \quad \forall (v_i, v_j) \in E_c \tag{29}$$
$$D_1 = \sum_{v_i \in V_c} k_i Y_i \tag{30}$$
$$Y_i \in \{0, 1\} \quad \forall v_i \in V_c \tag{31}$$
$$X_{i,j} \geq 0 \quad \forall (v_i, v_j) \in E_c. \tag{32}$$

Due to the elimination of the index s from the variables X, their number is now halved.

Consider again the definition (20) of the variables X. We can express it by employing the product of the variables Y_i and Y_j this way:

$$X_{i,j} = 2Y_iY_j - Y_i - Y_j + 1. \tag{33}$$

Using this definition, we can replace the constraints (26)-(29) with a new smaller set of inequalities, and replace the variables X with another set of variables S (having the same cardinality), which represent the product of the Y variables in (33), and which are not forced to be defined positive as the X variables. The new variables S are then defined as:

$$S_{i,j} = Y_iY_j, \forall (v_i, v_j) \in E_c, \tag{34}$$

where the inequalities which can be used to describe this relationship are the followings:

$$S_{i,j} \geq 0 \quad \forall (v_i, v_j) \in E_c \tag{35}$$
$$S_{i,j} \geq Y_j + Y_i - 1 \quad \forall (v_i, v_j) \in E_c \tag{36}$$
$$S_{i,j} \leq Y_i \quad \forall (v_i, v_j) \in E_c \tag{37}$$
$$S_{i,j} \leq Y_j \quad \forall (v_i, v_j) \in E_c. \tag{38}$$

We can put now in the objective function (25) the definition (33), using the S variables, at the place of $X_{i,j}$, and we can replace the constraints (26)-(29) with the new set (35)-(38). Actually, only half of these constraints are useful: as explained in Adams and Dearing (1994), since the coefficient of the variables S is positive in the objective function, and we are considering a maximization problem, we can drop the constraints (35) and (36). Thus, the new model, called OB_1, is the following:

$$\max \quad \frac{1}{m}\left(\sum_{(v_i,v_j)\in E_c}(2S_{i,j} - Y_i - Y_j) + |E_c| - \frac{1}{2m}\left({D_1}^2 + \frac{{D_c}^2}{2} - D_1D_c\right)\right) \tag{39}$$
$$\text{s.t.} \quad S_{i,j} \leq Y_i \quad \forall (v_i, v_j) \in E_c \tag{40}$$
$$S_{i,j} \leq Y_j \quad \forall (v_i, v_j) \in E_c \tag{41}$$
$$D_1 = \sum_{v_i \in V_c} k_iY_i \tag{42}$$
$$Y_i \in \{0,1\} \quad \forall v_i \in V_c, \tag{43}$$
$$\tag{44}$$

where in the objective function we use the fact that $\sum_{(v_i,v_j)\in E_C} 1 = |E_c|$.

3.2 Binary decomposition

The objective function of OB presents the term ${D_1}^2$, which is a sum of integer variable Y multiplied for integer values, i.e., the degrees of the vertices. Hence, it is possible to apply the binary decomposition technique, recently employed for mixed-integer quadratic programming in Billionnet et al. (2010), which consists on writing the term D_1 in this way:

$$D_1 = \sum_{l=0}^{t} 2^l a_l, \tag{45}$$

where a_l are binary variables, and t is a parameter which will be computed later. Using this definition of D_1, we can express ${D_1}^2$ as:

$$ {D_1}^2 = \sum_{l=0}^{t} 2^l a_l \cdot \sum_{h=0}^{t} 2^h a_h = \sum_{l=0}^{t}\sum_{h=0}^{t} 2^{l+h} a_l a_h = \sum_{l=0}^{t}\sum_{h=0}^{t} 2^{l+h} R_{lh} = \sum_{l=0}^{t} 2^{2l} a_l + \sum_{l=0}^{t}\sum_{h<l} 2^{l+h+1} R_{lh}, \tag{46} $$

where R are the variables used to replace the products between the variables a. The constraints used to express this relationship are the following:

$$ R_{l,h} \geq 0 \quad \forall l \in \{0,\dots,t\}, \forall h \in \{0,\dots,l-1\} \tag{47} $$
$$ R_{l,h} \geq a_l + a_h - 1 \quad \forall l \in \{0,\dots,t\}, \forall h \in \{0,\dots,l-1\} \tag{48} $$
$$ R_{l,h} \leq a_l \quad \forall l \in \{0,\dots,t\}, \forall h \in \{0,\dots,l-1\} \tag{49} $$
$$ R_{l,h} \leq a_h \quad \forall l \in \{0,\dots,t\}, \forall h \in \{0,\dots,l-1\}. \tag{50} $$

Again, as for constraints (35)-(38), only half of the inequalities can be adjoined. This time, since the variables R appear in the objective function with a negative sign, we should add (47) and (48) to the model.

Finally, to estimate t, recall that the maximum value which can be assumed by D_1 is the sum of the degrees of all the vertices in the current cluster D_c. Moreover, from (45) the maximum possible value for D_1 is $2^{t+1}-1$. Hence, t can be computed as:

$$ 2^{t+1} - 1 \geq D_c \quad \Rightarrow \quad t = \lceil \log_2(D_c + 1) - 1 \rceil. \tag{51} $$

Now we can define the formulation OB_{2a}:

$$ \max \quad \frac{1}{m}\left(m_1 + m_2 - \frac{1}{2m}\left(\sum_{l=0}^{t} 2^{2l} a_l + \sum_{l=0}^{t}\sum_{h<l} 2^{l+h+1} R_{lh} + \frac{{D_c}^2}{2} - D_1 D_c \right)\right) \tag{52} $$
$$ \text{s.t.} \quad X_{i,j,1} \leq Y_{i,1} \quad \forall (v_i, v_j) \in E_c \tag{53} $$
$$ X_{i,j,1} \leq Y_{j,1} \quad \forall (v_i, v_j) \in E_c \tag{54} $$
$$ X_{i,j,2} \leq 1 - Y_{i,1} \quad \forall (v_i, v_j) \in E_c \tag{55} $$
$$ X_{i,j,2} \leq 1 - Y_{j,1} \quad \forall (v_i, v_j) \in E_c \tag{56} $$
$$ R_{l,h} \geq a_l + a_h - 1 \quad \forall l \in \{0,\dots,t\}, \forall h \in \{0,\dots,l-1\} \tag{57} $$
$$ m_s = \sum_{(v_i,v_j)\in E_c} X_{i,j,s} \forall s \in \{1,2\} \tag{58} $$
$$ D_1 = \sum_{v_i \in V_c} k_i Y_{i,1} \tag{59} $$
$$ Y_{i,s} \in \{0,1\} \quad \forall v_i \in V_c, \forall s \in \{1,2\} \tag{60} $$
$$ X_{i,j,s} \geq 0 \quad \forall (v_i, v_j) \in E_c, \forall s \in \{1,2\}, \tag{61} $$
$$ R_{l,h} \geq 0 \quad \forall l \in \{0,\dots,t\}, \forall h \in \{0,\dots,l-1\} \tag{62} $$

3.2.1 Compact binary decomposition

It is possible to reduce the number of variables R in the previous model. The variable $R_{l,h}$ is the linearization of the term $a_l a_h$, used in the objective function (52). We can write the term

of this objective function which involves the variables $R_{l,h}$ in this way:

$$\sum_{l=0}^{t}\sum_{h<l} 2^{l+h+1}R_{lh} = \sum_{l=0}^{t}\sum_{h<l} 2^{l+h+1}a_l a_h = \sum_{l=0}^{t} 2^{l+1}a_l \sum_{h<l} 2^h a_h = \sum_{l=0}^{t} 2^{l+1}a_l b_l = \sum_{l=0}^{t} 2^{l+1}R_l, \tag{63}$$

where $R_l = a_l b_l$ and b_l is a new variable defined as $\sum_{h<l} 2^h a_h$. Since the upper bound for b_l is $U_{b_l} = \sum_{h<l} 2^h = 2^l - 1$, the constraints to add to the model are the following:

$$b_l = \sum_{h<l} 2^h a_h, \quad \forall l \in \{0,\ldots,t\} \tag{64}$$

$$R_l \geq 0 \quad \forall l \in \{0,\ldots,t\} \tag{65}$$

$$R_l \geq U_{b_l} a_l + b_l - U_{b_l} \quad \forall l \in \{0,\ldots,t\}. \tag{66}$$

With respect to the previous formulation, we have now $2(t+1)$ continuous variables instead of t^2+t ones, and we have adjoined $t+1$ constraints. Actually, we can notice that $b_0 = 0$ and $b_1 = a_0$, but avoiding to define these variables does not change significatively the computation time. This formulation will be addressed as OB_{2b}.

3.2.2 Compact binary decomposition 2

Consider again the objective function (52) obtained after the transformation proposed in the previous section. In order to have a more compact representation of it, we can put together the term containing the variables a_l and R_l in this way:

$$\sum_{l=0}^{t} 2^{2l}a_l + \sum_{l=0}^{t} 2^{l+1}R_l = \sum_{l=0}^{t} 2^{2l}a_l + \frac{2^{2l}}{2^{l-1}}R_l = \sum_{l=0}^{t} 2^{2l}\left(a_l + \frac{a_l b_l}{2^{l-1}}\right). \tag{67}$$

Hence, we can write

$$\sum_{l=0}^{t} 2^{2l}\left(a_l + \frac{a_l b_l}{2^{l-1}}\right) = \sum_{l=0}^{t} \frac{2^{2l}}{2^{l-1}}a_l\left(b_l + 2^{l-1}\right) = \sum_{l=0}^{t} 2^{l+1}a_l z_l = \sum_{l=0}^{t} 2^{l+1}T_l, \tag{68}$$

where the new variable z_l is equal to $b_l + 2^{l-1}$ and T_l is the linearization of $a_l z_l$. Then, we should remove the variables R and b from our formulation (and all the related constraints), and adjoin the new variables z and T, as well as these constraints:

$$z_l = \sum_{h<l} 2^h a_h + 2^{l-1}, \quad \forall l \in \{0,\ldots,t\} \tag{69}$$

$$T_l \geq 0 \quad \forall l \in \{0,\ldots,t\} \tag{70}$$

$$T_l \geq U_{z_l} a_l + b_l - U_{z_l} \quad \forall l \in \{0,\ldots,t\}, \tag{71}$$

where U_{z_l} is the upper bound of the variable z_l, and it is equal to 2^l. The number of variables and constraints is the same as in the previous section (again, we could omit to define z_0 and z_1, since $z_0 = 2^{-1}$ and $z_1 = a_0 + 1$). The corresponding reformulation is called OB_{2c}.

3.3 Symmetry breaking constraint

At each step of the algorithm a cluster is split in two new clusters. This problem presents some symmetries, which could be avoided in order to decrease the computational time. A simple way to do it is to fix one of the vertex to belong to one of the clusters.

Some tests show that the best is to fix the vertex with highest degree, probably because its role is more relevant in the various constraints.

Hence, the model OB_3 is obtained by adding the following constraint to the model OB:

$$Y_g = 0, \quad g = \arg\max\{k_i, \forall v_i \in V_c\}. \tag{72}$$

4 Numerical results

In this section we present the comparison of the numerical results obtained by using the proposed reformulations. Results have been obtained on a 2.8GHz Intel Core i7 CPU of a computer with 8 GB RAM running Linux and CPLEX 12.2 (IBM 2010), where we performed a fine tuning of the parameters (more precisely, we disabled the MIP cutting plane generation, and we used as branching variable selection strategy the branch based on pseudo costs). Results are obtained on a set of instances of the literature, presented in Table 1. In Tables 2-4 we show the comparison of the performaces of the divisive hierarchical heuristic algorithm when the different proposed formulations for the bipartition model are used. M denotes the number of clusters, and Q the modularity. The total number of Branch-and-Bound nodes, as well as the gap at the root node for the first bipartition (that is $\left(\frac{100\cdot|f^*-f_{UB}|}{|f^*+10^{-10}|}\right)\%$, where f_{UB} is the best upper bound found in the case of maximization problems, and f^* is the objective function value of the incumbent), are also provided. Computing times are in seconds. Note that slight discrepancies may arise in the values of M and Q; they are due to the fact that optimal bipartitions are not necessarily unique.

Table 1: Informations about the used graphs.

ID	Graph	n	m	Reference
1	Karate	34	78	Zachary (1977)
2	Dolphin	62	159	Lusseau et al. (2003)
3	Les misérables	77	254	Hugo (1951), Knuth (1993)
4	A00 main	83	135	Batagelj and Mrvar (2006)
5	P53 protein	104	226	Dartnell at al. (2005)
6	Political books	105	441	Krebs (2008)
7	Football	115	613	Girvan and Newman (2002)
8	A01 main	249	635	Batagelj and Mrvar (2006)
9	USAir97	332	2126	Batagelj and Mrvar (2006)
10	Netscience main	379	914	Newman (2006a)
11	S838	512	819	Milo et al. (2004)
12	Power	4941	6594	Watts and Strogatz (1998)

It appears from Table 2 that the proposed reformulations of the original quadratic model instantly impact the resolution time. OB_1 outperforms OB_3 in terms of computational time. As expected OB_3 allows to reduce the number of Branch-and-Bound nodes.

Table 2: Comparison between the original formulation *OB* proposed in Cafieri et. al. (2011) and recalled in Section 2, the reformulation OB_1 with less variables and constraints proposed in Section 3.1, and OB_3 obtained by adding the symmetry breaking constraint to the original formulation, as proposed in Section 3.3.

ID	*OB*					OB_1					OB_3				
	M	*Q*	nodes	gap (%)	time	*M*	*Q*	nodes	gap (%)	time	*M*	*Q*	nodes	gap (%)	time
1	4	0.4188	45	34.60	0.14	4	0.4188	41	**34.48**	**0.06**	4	0.4188	**18**	83.65	0.07
2	4	0.5265	207	33.66	0.59	4	0.5265	157	**31.00**	**0.19**	4	0.5265	**98**	34.71	0.49
3	8	0.5468	205	65.37	1.09	8	0.5468	185	**45.73**	**0.40**	8	0.5468	**102**	80.13	0.58
4	7	0.5281	76	557.89	0.35	7	0.5281	56	557.89	0.11	7	0.5278	**27**	**0.45**	**0.08**
5	7	0.5284	275	70.45	1.10	7	0.5284	201	**58.72**	**0.53**	7	0.5284	**135**	824.60	0.59
6	4	0.5263	313	26.76	3.04	4	0.5263	294	26.76	**1.00**	4	0.5263	**145**	**3.15**	1.36
7	10	0.6009	8853	106.41	307.66	10	0.6009	5410	96.40	**56.69**	10	0.6009	**3014**	**77.95**	118.24
8	15	0.6288	1119	**58.02**	47.83	15	0.6288	1010	60.87	**16.85**	15	0.6288	**997**	81.02	**45**.85
9	8	0.3596	16682	684.12	4585.04	8	0.3596	17811	**282.62**	**1041.89**	8	0.3596	**9446**	923.23	2510.81
10	20	0.8470	291	2.39	3.64	20	0.8470	267	2.73	**1.44**	20	0.8470	**108**	**0.61**	1.82
11	15	0.8166	392	5.21	5.26	15	0.8166	304	3.54	**1.26**	15	0.8166	**197**	**0.74**	2.15
12	41	0.9396	1459	**37.00**	708.51	41	0.9396	1449	62.88	**217.61**	41	0.9394	**815**	62.93	417.26

Table 3: Comparison between the different binary decomposition reformulations proposed in Section 3.2

ID	OB_{2a}					OB_{2b}					OB_{2c}				
	M	*Q*	nodes	gap (%)	time	*M*	*Q*	nodes	gap (%)	time	*M*	*Q*	nodes	gap (%)	time
1	4	0.4188	**123**	769.79	0.52	4	0.4188	137	**423.80**	0.44	4	0.4188	148	776.17	**0.13**
2	4	0.5265	505	180.71	1.29	4	0.5265	**466**	726.01	1.92	4	0.5265	498	**148.91**	**0.59**
3	8	0.5468	577	121.96	2.16	8	0.5468	563	371.05	1.97	8	0.5468	**559**	**45.21**	**0.80**
4	7	0.5281	**251**	**43.35**	0.74	7	0.5278	272	229.49	0.46	7	0.5278	345	52.08	**0.35**
5	7	0.5284	**678**	154.01	3.22	7	0.5284	815	528.00	1.85	7	0.5284	1052	**52.51**	**1.38**
6	5	0.5270	**1284**	**61.84**	9.17	5	0.5270	1407	217.21	4.19	5	0.5270	1670	73.13	**3.99**
7	10	0.6009	**25406**	421.16	**252.96**	10	0.6009	40922	458.33	340.23	10	0.6009	38910	**210.41**	331.50
8	15	0.6288	**4395**	224.27	61.49	15	0.6288	5912	767.07	66.04	15	0.6288	5783	**150.85**	**58.73**
9	8	0.3596	**63687**	931.13	**3074.09**	8	0.3596	89520	739.82	4295.85	8	0.3596	91917	**716.74**	4610.60
10	20	0.8470	**931**	**50.36**	14.53	20	0.8470	1206	123.17	9.46	20	0.8470	1359	61.45	**7.17**
11	15	0.8167	**1348**	**68.06**	22.46	15	0.8167	2032	530.84	24.08	15	0.8167	2317	173.52	**11.31**
12	41	0.9395	**11289**	**28.94**	**2029.63**	40	0.9395	16940	73.87	2605.25	40	0.9395	19672	102.68	3071.16

From Table 3 we note that when using the binary decomposition reformulations we obtain the best computational time by employing OB_{2c}, except for the largest instances (i.e., 7 (Football), 9 (USAir97), and 12 (Power)) where the best one is OB_{2a}. The interest of reformulations compared in Table 3 based on binary decomposition is that they allow to obtain linear models which can be solved by other integer linear programming solvers. However, when using CPLEX, a suitable setting of the parameters allows to obtain best results with the quadratic reformulated model, as showed in Table 4. Note that with different setting of the parameters and earlier versions of CPLEX, the best results (but still worse than the ones presented in Table 4) were obtained by the binary decomposition reformulation OB_{2c} merged with OB_1 and OB_3.

Table 4: Results obtained by the formulation with less variables and constraints OB_1 together with the symmetry breaking constraint of formulation OB_3.

ID			$OB_1 + OB_3$		
	M	Q	nodes	gap (%)	time
1	4	0.4188	17	77.60	0.04
2	4	0.5265	93	36.20	0.16
3	8	0.5468	105	80.13	0.35
4	7	0.5278	26	9.11	0.04
5	7	0.5284	119	824.60	0.26
6	4	0.5263	152	3.15	0.51
7	10	0.6009	3822	65.69	44.38
8	15	0.6288	726	93.06	9.72
9	8	0.3596	8665	640.90	446.06
10	20	0.8470	94	8.42	0.85
11	15	0.8166	186	0.74	1.18
12	41	0.9396	891	62.93	123.85

In Table 4 we present the best results obtained by merging OB_1 and OB_3, that is the compact reformulation of the original quadratic model plus the symmetry breaking constraint. The computing time is significantly reduced with respect to the original formulation. It is reduced by a factor up to 10 for one of the largest instance, that is the number 9 (USAir97).

5 Conclusions

In this paper we analyze the impact of reformulating the mathematical programming formulation of the bipartition problem arising in a hierarchical divisive algorithm for graph clustering. The original quadratic model is reformulated in such a way that the number of variables and constraints is reduced and a symmetry breaking constraint is added. An alternative linear formulation, obtained by employing a binary decomposition, is also proposed. Numerical results show that the proposed reformulations of the quadratic model significantly reduce the computational time.

Acknowledgements Financial support by Grants Digiteo 2009-14D "RMNCCO" and Digiteo 2009-55D "ARM" is gratefully acknowledged. P.H. was partially supported by FQRNT (Fonds de recherche du Québec – Nature et technologies) team grant PR-131365.

References

Adams, W. P., & Dearing, P. M. (1994). On the equivalence between roof duality and Lagrangian duality for unconstrained 0-1 quadratic programming problems. *Discrete Applied Mathematics*, 48(1), 1-20.

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734-749.

Agarwal, G., & Kempe, D. (2008). Modularity-maximizing graph communities via mathematical programming. *The European Physical Journal B - Condensed Matter and Complex Systems*, 66(3), 409-418.

Aloise, D., Cafieri, S., Caporossi, G., Hansen, P., Perron, S., & Liberti, L. (2010). Column generation algorithms for exact modularity maximization in networks. *Physical Review E*, 82(4), 046112.

Batagelj, V. & Mrvar, A. (2006). Pajek Datasets. `http://vlado.fmf.uni-lj.si/pub/networks/data/`.

Billionnet, A., Elloumi, S., & Lambert, A. (2010). Extending the QCR method to general mixed-integer programs. *Mathematical Programming*, doi:10.1007/s10107-010-0381-7.

Brandes, U., Delling, D., Gaertler, M., Görke, R., Hoefer, M., Nikoloski, Z., & Wagner, D. (2008). On Modularity Clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2), 172-188.

Cafieri, S., Hansen, P., & Liberti, L. (2010). Loops and multiple edges in modularity maximization of networks. *Physical Review E*, 81(4), 046102.

Cafieri, S., Hansen, P., & Liberti, L. (2011). Locally optimal heuristic for modularity maximization of networks. *Physical Review E*, 83(5), 056105.

Clauset, A., Newman, M. E. J., & Moore, C. (2004). Finding and evaluating community structure in very large networks. *Physical Review E*, 70(6), 066111.

Dartnell, L., Simeonidis E., Hubank M., Tsoka S., Bogle I. D. L., & Papageorgiou, L. G. (2005). Robustness of the p53 network and biological hackers. *Federation of European Biochemical Societies (FEBS) Letters*. 579(14), 3037-3042.

Flake, G. W., Lawrence S., Lee Giles C., & Coetzee F. M. (2002). Self-Organization and Identification of Web Communities. *IEEE Computer*, 35(3), 66-71.

Fortunato, S., & Barthelemy, M. (2007). Resolution limit in community detection. *Proceedings of the National Academy of Sciences of the U.S.A.*, 104(1), 36-41.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3-5), 75-174.

Girvan, M., & Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the U.S.A.*, 99(12), 7821-7826.

Grötschel, M., & Wakabayashi, Y. (1989). A cutting plane algorithm for a clustering problem. *Mathematical Programming*, 45(1), 59-96.

Guimerà, R., & Amaral, L. A. N. (2004). Functional cartography of complex metabolic networks. *Nature*, 433, 895-900.

Hugo, V. (1951). *Les Misérables*. Gallimard, Bibliothèque de la Pleiade, Paris.

IBM. ILOG CPLEX 12.2 *User's Manual*, IBM, 2010.

Knuth, D. E. (1993). *The Stanford GraphBase: a platform for combinatorial computing*. Addison-Wesley.

Krebs, V. (2008). `http://www.orgnet.com/`.

Lusseau, D., Schneider, K., Boisseau, O. J., Haase, P., Slooten, E., & Dawson, S. M. (2003). The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4), 396-405.

Milo, R., Itzkovitz, S., Kashtan, N., Levitt, R., Shen-Orr, S., Ayzenshtat, I., Sheffer, M., & Alon, U. (2004). Superfamilies of evolved and designed networks. *Science*, 303, 1538-1542.

Newman, M. E. J., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, 69(2), 026113.

Newman, M. E. J. (2006a). Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3), 036104.

Newman, M. E. J. (2006b). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the U.S.A.*, 103(23), 8577-8582.

Newman, M. E. J. (2010). *Networks: An Introduction*. Oxford University Press.

Palla, G., Derényi, I., Farkas, I., & Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435, 814-818.

Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., & Parisi, D. (2004). Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences*, 101(9), 2658-2663.

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393, 440-442.

Xu, G., Tsoka, S., & Papageorgiou, L. G. (2007). Finding community structures in complex networks using mixed integer optimisation. *The European Physical Journal B - Condensed Matter and Complex Systems*, 60(2), 231-239.

Zachary, W. W. (1977). An Information Flow Model for Conflict and Fission in Small Groups. *Journal of Anthropological Research*, 33(4), 452-473.