Annals of Operations Research 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 one obtained with 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 best 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. Given an unweighted graph G, its modularity Q is defined as:

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

where m is the number of edges of the graph, 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. Note that the modularity of the graph can be seen as the sum of the modularities of each cluster. The extension of this definition to weighted graphs is presented in Fortunato (2010). The maximization of modularity is a NP-hard problem, as proved by Brandes et al. (2008).

Although modularity maximization is a very popular criterion, a few criticisms have been recently raised, the most important of them being the resolution limit and the degeneracy of the modularity function. The former refers to the fact that in some cases some clusters, smaller than a certain size which depends on the number of edges of the network, may not be detected and they remain hidden within larger clusters, as reported in Fortunato and Barthelemy (2007), Good et al. (2010). The latter is related to the possible presence of several high modularity partitions, even very different from each other, which makes hard to identify the global optimum (Good et al. 2010). Some approaches to address these criticisms are presented in Arenas et al. (2008), Kumpula et al. (2007), Reichardt and Bornholdt (2006) and Sales-Pardo et al. (2007), though they do not solve the problems in a fully satisfactory manner. Strengths and weaknesses of modularity are also discussed in Fortunato (2010) and Cafieri et al. (2010). Despite these criticisms, modularity still remains an interesting criterion for network clustering, which is widely employed in the literature. Clear advantages of this criterion include its intuitive meaning and simple mathematical formulation, as well as its independence on any parameter to be decided arbitrarely.

Several methods have been proposed in the literature to find high modularity partitions: a few exact methods, and several heuristics. Among the exact methods, there is a clique partitioning algorithm originally proposed by Grötschel and Wakabayashi (1989), which is similar to the one presented in Brandes et al. (2008), column generations algorithms proposed by Aloise et al. (2010), and a mixed-integer convex quadratic programming formulation due to Xu et al. (2007). Concerning the heuristics, many methods have been proposed. They are presented in the survey of Fortunato (2010). 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. The aim of this paper is to investigate mathematical programming reformulations of a clustering bipartition problem formulated as a quadratic mixed-integer program, arising in the locally optimal hierarchical divisive approach for modularity maximization proposed in Cafieri et al. (2011). We present several reformulations of the original model. They include reformulations aiming to reduce the dimension of the problem (number of variables and constraints), reformulations aiming to linearize nonlinear terms by binary decompositions and a reformulation based on symmetry breaking constraints. The path of reformulations leading to the *best formulation* for the bipartition problem appears to be the one passing through a few steps whose effect is to reduce the number of variables and constraints and to adjoin symmetry breaking constraints to the obtained compact formulation. Note that with best formulation we mean the one that provide the optimal solution in less CPU time. Hence, we do not improve the quality of the results, in terms of modularity, provided by the hierarchical divisive heuristic in Cafieri et al. (2011). Rather, we propose some techniques to decrease the computational time required to obtain the solution.

The rest of the paper is organized as follows: in Section 2 the heuristic proposed in Cafieri et al. (2011) is presented more in details, 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

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.

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 $i = 1,2,\dots n$ 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 have to be considered, and the sum of degrees of vertices belonging to one of

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

$P \leftarrow C_1 = \{\{v_1, v_2, \dots, v_n\}\}$
$k \leftarrow 1$
while $k \leq |P|$ and $\exists C_i \in P$ not visited **do**
    select $C_i \in P$ (not visited) with the smallest possible index i
    partition C_i into C_{2i} and C_{2i+1} maximizing the modularity
    if $Q(C_{2i}) + Q(C_{2i+1}) \geq Q(C_i)$
      then
        $P \leftarrow (P \cup \{C_{2i}\} \cup \{C_{2i+1}\}) \setminus \{C_i\}$
        $k \leftarrow k+1$
    end if
end while

Fig. 1: The hierarchical divisive algorithm.

the two sub-clusters can be expressed as a function of the sum of degrees of the other cluster:

$$D_2 = D_c - D_1, \quad (2)$$

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, the objective function (1) of the bipartition subproblem :

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

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

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

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 also belong to that cluster:

$$\forall (v_i, v_j) \in E_c \quad X_{i,j,1} \leq Y_{i,1} \quad (5)$$
$$\forall (v_i, v_j) \in E_c \quad X_{i,j,1} \leq Y_{j,1} \quad (6)$$
$$\forall (v_i, v_j) \in E_c \quad X_{i,j,2} \leq 1 - Y_{i,1} \quad (7)$$
$$\forall (v_i, v_j) \in E_c \quad X_{i,j,2} \leq 1 - Y_{j,1}. \quad (8)$$

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:

$$\forall s \in \{1,2\} \quad m_s = \sum_{(v_i,v_j) \in E_c} X_{i,j,s} \quad (9)$$
$$D_1 = \sum_{v_i \in V_c} k_i Y_{i,1}, \quad (10)$$

where k_i is the degree of the vertex v_i and V_c and E_c are respectively the set of vertices and the set of edges of the cluster c to be bipartitioned. Hence, the complete formulation proposed in Cafieri et al. (2011), and called from now *OB* (Optimal Bipartition), is the following:

$$\max \quad \frac{1}{m}\left(m_1+m_2-\frac{1}{2m}\left({D_1}^2+\frac{{D_c}^2}{2}-D_1D_c\right)\right) \tag{11}$$

$$\text{s.t.} \quad \forall (v_i,v_j)\in E_c \quad X_{i,j,1}\le Y_{i,1} \tag{12}$$

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

$$\forall (v_i,v_j)\in E_c \quad X_{i,j,2}\le 1-Y_{i,1} \tag{14}$$

$$\forall (v_i,v_j)\in E_c \quad X_{i,j,2}\le 1-Y_{j,1} \tag{15}$$

$$\forall s\in\{1,2\} \quad m_s=\sum_{(v_i,v_j)\in E_c} X_{i,j,s} \tag{16}$$

$$D_1=\sum_{v_i\in V_c} k_iY_{i,1} \tag{17}$$

$$\forall s\in\{1,2\} \quad m_s\in\mathbb{R} \tag{18}$$

$$D_1\in\mathbb{R} \tag{19}$$

$$\forall v_i\in V_c \quad Y_{i,1}\in\{0,1\} \tag{20}$$

$$\forall (v_i,v_j)\in E_c, \forall s\in\{1,2\} \quad X_{i,j,s}\in\mathbb{R}_0^+. \tag{21}$$

Note that the variable D_1 is not defined to be integer and non-negative in the *OB* model, since this is automatically implied by constraint (17).

3 Improved formulations 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 reformulations, which are discussed separately 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, half of the variables X can be removed and the number of constraints can be reduced on the basis of the following considerations.

Consider the X variables. Looking at the objective function (11) of the *OB* formulation, we notice that it contains the term m_1+m_2, which represents the sum of the number of edges in the first and the second cluster. 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 variables X, moving from the original definition:

$$X_{i,j,s}=\begin{cases}1, & \text{if edge } (v_i,v_j) \text{ belongs to cluster } s,\\ 0, & \text{otherwise,}\end{cases} \tag{22}$$

to a new set of variables $X_{i,j}$, where $X_{i,j}$ is equal to 1 if the edge (v_i,v_j) is within the cluster 1 or 2, and 0 otherwise. In other words, we can define $X_{i,j}$ as:

$$X_{i,j}=\begin{cases}1, & \text{if } Y_i=Y_j,\\ 0, & \text{otherwise,}\end{cases} \tag{23}$$

where we also drop the meaningless index 1 from the Y variables. Since $X_{i,j}$ can be seen as the negation of the XOR operation between Y_i and Y_j variables, the following constraints can be employed (Brown and Dell, 2007) to express the relationship between X and Y:

$$\forall (v_i, v_j) \in E_c \quad X_{i,j} \leq Y_i - Y_j + 1 \tag{24}$$
$$\forall (v_i, v_j) \in E_c \quad X_{i,j} \leq Y_j - Y_i + 1 \tag{25}$$
$$\forall (v_i, v_j) \in E_c \quad X_{i,j} \geq Y_i + Y_j - 1 \tag{26}$$
$$\forall (v_i, v_j) \in E_c \quad X_{i,j} \geq 1 - Y_i - Y_j. \tag{27}$$

Actually, only half of these constraints are useful: as explained in Adams and Dearing (1994), since the coefficient of the variables X is positive in the objective function, and we are considering a maximization problem, we can drop the constraints (26) and (27). Note that, as in the original model, the Y variables are binary and the X variables are non-negative and continuous. On the basis of these considerations, the *OB* model can be reformulated this way:

$$\max \quad \frac{1}{m}\left(\sum_{(v_i,v_j)\in E_c} X_{i,j} - \frac{1}{2m}\left({D_1}^2 + \frac{{D_c}^2}{2} - D_1 D_c\right)\right) \tag{28}$$
$$\text{s.t.} \quad \forall (v_i, v_j) \in E_c \quad X_{i,j} \leq Y_i - Y_j + 1 \tag{29}$$
$$\forall (v_i, v_j) \in E_c \quad X_{i,j} \leq Y_j - Y_i + 1 \tag{30}$$
$$D_1 = \sum_{v_i \in V_c} k_i Y_i \tag{31}$$
$$D_1 \in \mathbb{R} \tag{32}$$
$$\forall v_i \in V_c \quad Y_i \in \{0,1\} \tag{33}$$
$$\forall (v_i, v_j) \in E_c \quad X_{i,j} \in \mathbb{R}. \tag{34}$$

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

Consider again the definition (23) of the variables X. We can express it by employing the product of the variables Y_i and Y_j this way:

$$X_{i,j} = 2Y_iY_j - Y_i - Y_j + 1. \tag{35}$$

Using this definition, we can replace the constraints (24)-(27) with a new set of inequalities, and replace the set of variables X with another set of variables S (having the same cardinality), which represent the product of the Y variables in (35). The new variables S are then defined as:

$$\forall (v_i, v_j) \in E_c \quad S_{i,j} = Y_iY_j, \tag{36}$$

where the inequalities needed to describe this relationship can be obtained after applying (35) to (24)-(27) (note that these constraints correspond to the Fortet inequalities (Fortet, 1960), which provide an exact linearization of a product of binary variables):

$$\forall (v_i, v_j) \in E_c \quad S_{i,j} \geq 0 \tag{37}$$
$$\forall (v_i, v_j) \in E_c \quad S_{i,j} \geq Y_j + Y_i - 1 \tag{38}$$
$$\forall (v_i, v_j) \in E_c \quad S_{i,j} \leq Y_i \tag{39}$$
$$\forall (v_i, v_j) \in E_c \quad S_{i,j} \leq Y_j. \tag{40}$$

We can put now in the objective function (28) the definition (35), using the S variables, in place of $X_{i,j}$, and we can replace the constraints (29)-(30) with the new set (39)-(40) (again, only half of the constraints are needed). Thus, the new model, called OB_1, is the following:

$$\max \quad \frac{1}{m}\left(\sum_{(v_i,v_j)\in E_c}(2S_{i,j}-Y_i-Y_j)+|E_c|-\frac{1}{2m}\left(D_1{}^2+\frac{D_c{}^2}{2}-D_1D_c\right)\right) \tag{41}$$

$$\text{s.t.} \quad \forall (v_i,v_j)\in E_c \quad S_{i,j}\le Y_i \tag{42}$$

$$\forall (v_i,v_j)\in E_c \quad S_{i,j}\le Y_j \tag{43}$$

$$D_1=\sum_{v_i\in V_c}k_iY_i \tag{44}$$

$$\forall (v_i,v_j)\in E_c \quad S_{i,j}\in\mathbb{R} \tag{45}$$

$$D_1\in\mathbb{R} \tag{46}$$

$$\forall v_i\in V_c \quad Y_i\in\{0,1\}, \tag{47}$$

where in the objective function we use the fact that $\sum_{(v_i,v_j)\in E_c} 1 = |E_c|$. Computational experiments show that the formulation using the S variables outperforms the one with the X variables. Intuitively, the constraints (42) and (43), which involve separately variables Y_i and Y_j, give rise to a more sparse matrix constraints than the one associated with constraints (29) and (30) involving both Y_i and Y_j variables.

The reformulation of OB into OB_1 proposed in this section is the result of a few reformulation steps aiming at reducing the number of variables and constraints. These reformulations, though proposed in the framework of modularity maximizing graph clustering, can be more generally applied to a mathematical program exhibiting objective and constraints with the same mathematical expressions. More precisely, consider variables of type $X_{i,j,s}$, $s\in\{1,2\}$, where $X_{i,j,1}+X_{i,j,2}\le 1$, coupled with binary variables Y such that $\forall s\in\{1,2\}\ X_{i,j,s}=1$ if $Y_i=Y_j=1$, and 0 otherwise. If in the problem the variables $X_{i,j,s}$ only appear in the form $\sum_{s\in\{1,2\}} X_{i,j,s}$, the following symbolic algorithm can be applied to automatically obtain the described reformulations:

1. replace expression $\sum_{s\in\{1,2\}} X_{i,j,s}$ with $X_{i,j}$
2. replace variables $X_{i,j}$ with $2Y_iY_j-Y_i-Y_j+1$
3. replace products Y_iY_j with variables $S_{i,j}$
4. adjoin constraints (37)-(40) (only (37)-(38) if the variables S are minimized in the objective function, (39)-(40) if they are maximized).

As remarked, the $X_{i,j}$ represent the negation of the XOR operation between Y_i and Y_j. In case of other relationships holding between X and S, these relationships have to be expressed by a different set of constraints. An extended version of the proposed symbolic algorithm can take into account binary operators other than the negation of XOR.

3.2 Binary decomposition

The objective function of OB involves the term $D_1{}^2$, which is the square of a sum of binary variables Y multiplied by 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 in writing the term D_1 this way:

$$D_1=\sum_{l=0}^{t}2^l a_l, \tag{48}$$

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:

$$ {D_1}^2 = \sum_{l=0}^{t} 2^l a_l \cdot \sum_{h=0}^{t} 2^h a_h = \sum_{l=0}^{t}\sum_{h=0}^{t} 2^{l+h} a_l a_h = \sum_{l=0}^{t}\sum_{h=0}^{t} 2^{l+h} R_{lh} = \sum_{l=0}^{t} 2^{2l} a_l + \sum_{l=0}^{t}\sum_{h<l} 2^{l+h+1} R_{lh}, \tag{49} $$

where R are the variables used to replace the products between the variables a. The Fortet inequalities can be used to express this relationship:

$$ \forall l \in \{0,\dots,t\}, \forall h \in \{0,\dots,l-1\} \quad R_{l,h} \geq 0 \tag{50} $$
$$ \forall l \in \{0,\dots,t\}, \forall h \in \{0,\dots,l-1\} \quad R_{l,h} \geq a_l + a_h - 1 \tag{51} $$
$$ \forall l \in \{0,\dots,t\}, \forall h \in \{0,\dots,l-1\} \quad R_{l,h} \leq a_l \tag{52} $$
$$ \forall l \in \{0,\dots,t\}, \forall h \in \{0,\dots,l-1\} \quad R_{l,h} \leq a_h. \tag{53} $$

Again, as for constraints (37)-(40), only half of the inequalities need to be adjoined. This time, since the variables R appear in the objective function with a negative sign, we should adjoin (50) and (51) to the model.

Finally, to estimate t, recall that the maximum value which can be taken by D_1 is the sum D_c of the degrees of all the vertices in the current cluster. Moreover, from (48) the maximum possible value for D_1 is $2^{t+1}-1$. Hence, t can be computed as:

$$ 2^{t+1} - 1 \geq D_c \quad \Rightarrow \quad t = \lceil \log_2(D_c+1) - 1 \rceil. \tag{54} $$

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

$$ \max \quad \frac{1}{m}\left(m_1 + m_2 - \frac{1}{2m}\left(\sum_{l=0}^{t} 2^{2l} a_l + \sum_{l=0}^{t}\sum_{h<l} 2^{l+h+1} R_{lh} + \frac{{D_c}^2}{2} - D_1 D_c \right)\right) \tag{55} $$
$$ \text{s.t.} \quad \forall (v_i,v_j) \in E_c \quad X_{i,j,1} \leq Y_{i,1} \tag{56} $$
$$ \forall (v_i,v_j) \in E_c \quad X_{i,j,1} \leq Y_{j,1} \tag{57} $$
$$ \forall (v_i,v_j) \in E_c \quad X_{i,j,2} \leq 1 - Y_{i,1} \tag{58} $$
$$ \forall (v_i,v_j) \in E_c \quad X_{i,j,2} \leq 1 - Y_{j,1} \tag{59} $$
$$ \forall l \in \{0,\dots,t\}, \forall h \in \{0,\dots,l-1\} \quad R_{l,h} \geq a_l + a_h - 1 \tag{60} $$
$$ \forall s \in \{1,2\} \quad m_s = \sum_{(v_i,v_j)\in E_c} X_{i,j,s} \tag{61} $$
$$ D_1 = \sum_{v_i \in V_c} k_i Y_{i,1} \tag{62} $$
$$ D_1 = \sum_{l=0}^{t} 2^l a_l \tag{63} $$
$$ \forall s \in \{1,2\} \quad m_s \in \mathbb{R} \tag{64} $$
$$ D_1 \in \mathbb{R} \tag{65} $$
$$ \forall v_i \in V_c, \forall s \in \{1,2\} \quad Y_{i,s} \in \{0,1\} \tag{66} $$
$$ \forall (v_i,v_j) \in E_c, \forall s \in \{1,2\} \quad X_{i,j,s} \in \mathbb{R}_0^+ \tag{67} $$
$$ \forall l \in \{0,\dots,t\}, \forall h \in \{0,\dots,l-1\} \quad R_{l,h} \in \mathbb{R}_0^+. \tag{68} $$

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 (55). We can write the term of this objective function which involves the variables $R_{l,h}$ in this way:

$$\sum_{l=0}^{t}\sum_{h<l} 2^{l+h+1} R_{lh} = \sum_{l=0}^{t}\sum_{h<l} 2^{l+h+1} a_l a_h = \sum_{l=0}^{t} 2^{l+1} a_l \sum_{h<l} 2^h a_h = \sum_{l=0}^{t} 2^{l+1} a_l b_l = \sum_{l=0}^{t} 2^{l+1} R_l, \tag{69}$$

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:

$$\forall l \in \{0,\ldots,t\} \quad b_l = \sum_{h<l} 2^h a_h \tag{70}$$

$$\forall l \in \{0,\ldots,t\} \quad R_l \geq 0 \tag{71}$$

$$\forall l \in \{0,\ldots,t\} \quad R_l \geq U_{b_l} a_l + b_l - U_{b_l}. \tag{72}$$

With respect to the previous formulation, we have now $t+1$ variables R_l instead of $\frac{t^2+t}{2}$ variables $R_{l,h}$, and we have adjoined $t+1$ variables b and $t+1$ constraints (70). 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 (55) 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:

$$\sum_{l=0}^{t} 2^{2l} a_l + \sum_{l=0}^{t} 2^{l+1} R_l = \sum_{l=0}^{t} 2^{2l} a_l + \frac{2^{2l}}{2^{l-1}} R_l = \sum_{l=0}^{t} 2^{2l}\left(a_l + \frac{a_l b_l}{2^{l-1}}\right). \tag{73}$$

Hence, we can write

$$\sum_{l=0}^{t} 2^{2l}\left(a_l + \frac{a_l b_l}{2^{l-1}}\right) = \sum_{l=0}^{t} \frac{2^{2l}}{2^{l-1}} a_l \left(b_l + 2^{l-1}\right) = \sum_{l=0}^{t} 2^{l+1} a_l z_l = \sum_{l=0}^{t} 2^{l+1} T_l, \tag{74}$$

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:

$$\forall l \in \{0,\ldots,t\} \quad z_l = \sum_{h<l} 2^h a_h + 2^{l-1} \tag{75}$$

$$\forall l \in \{0,\ldots,t\} \quad T_l \geq 0 \tag{76}$$

$$\forall l \in \{0,\ldots,t\} \quad T_l \geq U_{z_l} a_l + z_l - U_{z_l}, \tag{77}$$

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 hierarchical divisive algorithm a cluster is split in two new clusters, if this division improves the total modularity. It is easy to see that, given a solution of the bipartition problem, the vertices in the first and second cluster can be swapped to obtain a symmetric solution. Since the problem of the optimal bipartitioning is solved exactly by the Branch-and-Bound Mixed Integer Linear Programming (MILP) algorithm of CPLEX, removing symmetric optima may reduce the size of the Branch-and-Bound tree, and consequently reduce the time to reach the leaves of the tree (i.e., the optimal solutions). A simple way to avoid this is to fix one of the vertex to belong to one of the two clusters.

Some tests show that the best results are obtained by fixing the vertex with highest degree. Intuitively, this happens because that vertex is involved in more constraints. Hence, the model OB_3 is obtained by adding the following constraint to the model OB:

$$Y_g = 0, \quad g = \arg\max\{k_i, \forall v_i \in V_c\}. \tag{78}$$

4 Numerical results

In this section we present a comparison of numerical results provided by the hierarchical divisive heuristic with 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 found experimentally that the best configuration is the one where the MILP cutting plane generation is disabled, and the branching variable selection strategy is the branch based on pseudo reduced costs). The stopping rule of CPLEX is the default, i.e., no limitation on the number of nodes or execution time has been used, so CPLEX stops when the optimal solution of the bipartition problem is found. 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 performances of the divisive hierarchical heuristic algorithm when the different proposed formulations for the bipartition model are used. M denotes the number of clusters, Q the modularity and *nodes* the total number of Branch-and-Bound nodes. 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.

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

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 based on binary decomposition is that they yield MILP models which can be solved by other integer linear programming solvers which cannot be employed to solve convex mixed-integer quadratic problems. However, when using CPLEX, best results can be obtained by a suitable parameter setting with the quadratic reformulated model, as shown in Table 4. Note that with different settings 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.

In Table 4 we present the results of the best formulation found obtained by merging OB_1 and OB_3, that is the compact reformulation of the original quadratic model with the

Table 1: Informations about the used graphs.

ID	Graph	n	m	Reference
1	Karate	34	78	Zachary (1977)
2	Dolphins	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	Protein p53	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)

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

			OB		OB_1		OB_3	
ID	M	Q	*nodes*	*time*	*nodes*	*time*	*nodes*	*time*
1	4	0.4188	45	0.14	41	**0.06**	**18**	0.07
2	4	0.5265	207	0.59	157	**0.19**	**98**	0.49
3	8	0.5468	205	1.09	185	**0.40**	**102**	0.58
4	7	0.5281	76	0.35	56	0.11	**27**	**0.08**
5	7	0.5284	275	1.10	201	**0.53**	**135**	0.59
6	4	0.5263	313	3.04	294	**1.00**	**145**	1.36
7	10	0.6009	8853	307.66	5410	**56.69**	**3014**	118.24
8	15	0.6288	1119	47.83	1010	**16.85**	**997**	45.85
9	8	0.3596	16682	4585.04	17811	**1041.89**	**9446**	2510.81
10	20	0.8470	291	3.64	267	**1.44**	**108**	1.82
11	15	0.8166	392	5.26	304	**1.26**	**197**	2.15
12	41	0.9396	1459	708.51	1449	**217.61**	**815**	417.26

symmetry breaking constraint adjoined. 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 instances, that is the number 9 (USAir97). For the sake of completeness, we also report the optimal results (M_{opt} and Q_{opt}) obtained by the column generation method presented in Aloise et al. (2010) for the tested graphs except the number 12, i.e., Power, since the largest instance solved by this approach is S838.

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

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

			OB_{2a}		OB_{2b}		OB_{2c}	
ID	*M*	*Q*	*nodes*	*time*	*nodes*	*time*	*nodes*	*time*
1	4	0.4188	**123**	0.52	137	0.44	148	**0.13**
2	4	0.5265	505	1.29	**466**	1.92	498	**0.59**
3	8	0.5468	577	2.16	563	1.97	**559**	**0.80**
4	7	0.5281	**251**	0.74	272	0.46	345	**0.35**
5	7	0.5284	**678**	3.22	815	1.85	1052	**1.38**
6	5	0.5270	**1284**	9.17	1407	4.19	1670	**3.99**
7	10	0.6009	**25406**	**252.96**	40922	340.23	38910	331.50
8	15	0.6288	**4395**	61.49	5912	66.04	5783	**58.73**
9	8	0.3596	**63687**	**3074.09**	89520	4295.85	91917	4610.60
10	20	0.8470	**931**	14.53	1206	9.46	1359	**7.17**
11	15	0.8167	**1348**	22.46	2032	24.08	2317	**11.31**
12	41	0.9395	**11289**	**2029.63**	16940	2605.25	19672	3071.16

Table 4: Optimal solutions (M_{opt} and Q_{opt}) obtained by the column generation approach presented in Aloise et al. (2010), and results obtained by the formulation with less variables and constraints OB_1 together with the symmetry breaking constraint of formulation OB_3.

					OB		$OB_1 + OB_3$	
ID	M_{opt}	Q_{opt}	*M*	*Q*	*nodes*	*time*	*nodes*	*time*
1	4	0.4198	4	0.4188	45	0.14	**17**	**0.04**
2	5	0.5285	4	0.5265	207	0.59	**93**	**0.16**
3	6	0.5600	8	0.5468	205	1.09	**105**	**0.35**
4	9	0.5309	7	0.5278	76	0.35	**26**	**0.04**
5	7	0.5351	7	0.5284	275	1.10	**119**	**0.26**
6	5	0.5272	4	0.5263	313	3.04	**152**	**0.51**
7	10	0.6046	10	0.6009	8853	307.56	**3822**	**44.38**
8	14	0.6329	15	0.6288	1119	47.83	**726**	**9.72**
9	6	0.3682	8	0.3596	16682	4585.04	**8665**	**446.06**
10	19	0.8486	20	0.8470	291	3.64	**94**	**0.85**
11	12	0.8194	15	0.8166	392	5.26	**186**	**1.18**
12	-	-	41	0.9396	1459	708.51	**891**	**123.85**

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 The authors would like to thank the anonymous referees for their preciuos comments and suggestions. 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.

Arenas, A., Fernández, F., & Gómez, S. (2008). Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics*, 10(5), 053039.

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

Billionnet, A., Elloumi, S., & Lambert, A. (2012). Extending the QCR method to general mixed-integer programs. *Mathematical Programming*, 131, 381-401.

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.

Brown, G. G., & Dell, R. F. (2007). Formulating integer linear programs: A rogues' gallery. *Informs Transactions on Education*, 7(2), 1-13.

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.

Fortet, R. (1960). Applications de l'algèbre de Boole en recherche opérationelle. *Revue Française de Recherche Opérationelle*, 4, 17-26.

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/`.

Kumpula, J. M., Saramaki, J., Kaski, K., & Kertesz, J. (2007). Limited resolution and multiresolution methods in complex network community detection. *Fluctuations and Noise Letters*, 7(3), 209-214.

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 of the U.S.A.*, 101(9), 2658-2663.

Reichardt, J., & Bornholdt, S. (2006). Statistical mechanics of community detection. *Physical Review E*, 74(1), 016110.

Sales-Pardo, M., Guimerà, R., Moreira, A., & Amaral, L. A. N. (2007). Extracting the hierarchical organization of complex systems. *Proceedings of the National Academy of Sciences of the U.S.A.*, 104(39), 15224-9.

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.