Discrete Applied Mathematics 164 (2014) 2–12

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Solving the maximum edge biclique packing problem on unbalanced bipartite graphs

V. Acuña[a], C.E. Ferreira[b], A.S. Freire[b,*], E. Moreno[c]

[a] *Université Claude Bernard - Lyon I, France*

[b] *Institute of Mathematics and Statistics, Universidade de São Paulo, Brazil*

[c] *Faculty of Science and Technology, Universidad Adolfo Ibáñez, Chile*

ARTICLE INFO

Article history:
Received 15 October 2010
Received in revised form 31 May 2011
Accepted 27 September 2011
Available online 27 October 2011

Keywords:
Maximum edge biclique packing
Branch-and-price
Metabolic networks
Product bundling

ABSTRACT

A *biclique* is a complete bipartite graph. Given an (L, R)-bipartite graph $G = (V, E)$ and a positive integer k, the *maximum edge biclique packing* (MEBP) problem consists in finding a set of at most k bicliques, subgraphs of G, such that the bicliques are vertex disjoint with respect to a subset of vertices S, where $S \in \{V, L, R\}$, and the number of edges inside the bicliques is maximized. The *maximum edge biclique* (MEB) problem is a special case of the MEBP problem in which $k = 1$.

Several applications of the MEB problem have been studied and, in this paper, we describe applications of the MEBP problem in *metabolic networks* and *product bundling*. In these applications the input graphs are very unbalanced (i.e., $|R|$ is considerably greater than $|L|$), thus we consider carefully this property in our models. We introduce a new formulation for the MEB problem and a branch-and-price scheme, using the classical branch rule by Ryan and Foster, for the MEBP problem. Finally, we present computational experiments with instances that come from the described applications and also with randomly generated instances.

1. Introduction

A *biclique* is a complete bipartite graph. Given an (L, R)-bipartite graph $G = (V, E)$ and a positive integer k, the *maximum edge biclique packing* (MEBP) problem consists in finding a packing containing at most k bicliques, subgraphs of G, such that the bicliques are vertex disjoint with respect to a given subset of vertices S, where $S \in \{V, L, R\}$, and the number of edges inside the bicliques is maximized. If $S = V$, each vertex can be in at most one biclique inside the packing. On the other hand, if $S = L$ or $S = R$, this constraint applies to only one of the classes (L or R). In any case, each edge can be in at most one biclique inside the packing. In this paper, which is an extension of the work presented in [1], we study the MEBP problem and also the *maximum edge biclique* (MEB) problem, a special case of the MEBP problem in which $k = 1$.

As shown in 2003 by Peeters [16], the MEB problem is NP-hard. In 2004, Feige and Kogan [8] conjectured that the MEB problem is hard to approximate within a factor of $O(n^\epsilon)$, for some $\epsilon > 0$. In the cited work, the authors prove that the MEB problem is hard to approximate under the plausible assumption that 3-SAT has no sub-exponential algorithm, and they also show in [7] that the above conjecture is valid under certain other plausible assumptions. Dawande et al. introduced in [5] an algorithm with expected approximation ratio of 2, for sufficiently dense random bipartite graphs. For convex bipartite graphs the problem is polynomially solvable [14]. There are many applications of the MEB problem in biological data analysis and other areas (see [12] for a very comprehensive survey, and see [6] for a slightly different variant of the problem with application in multicast network design).

[*] Corresponding author. Fax: +55 11 72492120.
E-mail address: afreire@ime.usp.br (A.S. Freire).

0166-218X/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2011.09.019

Different variants of the MEB problem can be found in the literature. Consider a function associating a non-negative value with each edge of a bipartite graph G. The *maximum weight biclique* problem consists in finding a biclique, a subgraph of G, with maximum weight, where the weight of a biclique is the sum of the weights of its edges. As shown in 2008 by Jinsong Tan [19], the maximum weight biclique problem is hard to approximate. We say that an (L, R)-bipartite graph B is *balanced* if $|R| = |L|$. Feige and Kogan showed in [8] that the *maximum balanced biclique* problem is hard to approximate. The problem of finding a biclique with maximum number of vertices in a bipartite graph can be solved in polynomial time [9]. Although there is a wide literature on the MEB problem and some variants of it, we could not find any previous result on the MEBP problem.

One of the motivations of this work is data analysis in Bioinformatics. Consider L as a set of *individuals* and R as a set of *conditions*. An edge uv means that individual u satisfies condition v. The aim is to cluster individuals based on the common conditions satisfied by the *entire* cluster. In particular, this method has been used in the study of *metabolic networks* in order to improve the interpretation of the data given by the analysis of *Elementary Flux Modes* (EFMs). The EFMs of a metabolic network correspond to minimal sets of reactions that can be used in steady state (for precise definitions of EFMs see [18]). However, the number of EFMs obtained in real networks is, in general, huge and thus impossible in practice to analyze "by hand". For instance, the central metabolism of *E. coli* corresponds to a network of 106 reactions and about 26 000 000 EFMs (see [20]). In this context, a way to obtain more significant biological information is dividing the metabolic network into k sets of reactions that belong to many EFMs (see [13]). This corresponds to solving the MEBP problem where L is the set of reactions, R is the set of EFMs (with $|L| \ll |R|$) and bicliques are the clusters with non-overlapping reactions (i.e., $S = L$).

Another application of the MEBP problem is in the context of consumer product bundling. The concept of *product bundling* is the sale of two or more separate goods in a single package. This marketing strategy is massively utilized, especially in retail products. We study the problem of selecting an optimal set of k product bundles that maximize the total number of products sold via bundles, subject to client's demand for products. The motivation for this problem came from a major food company that was evaluating the capability to deliver its products directly to small grocers, avoiding its current wholesaler and distributors. This problem can be modeled as an instance of an MEBP problem by constructing a bipartite graph, where vertices in L represent products and vertices in R clients, and an edge uv corresponds to a client v that consumes a particular product u. In this case, a biclique on this graph corresponds to a set of clients (potential consumers) that buy a set of products (product bundle), and the goal is to find k product bundles that maximize the supplied demand (i.e., the number of the edges inside the bicliques). In this application $S = R$ means that each client is supposed to buy at most one bundle. On the other hand, $S = L$ means that one product cannot appear in more than one bundle. Also in this application the input bipartite graphs are very unbalanced (there are much more customers than products).

This paper is organized as follows. In Section 2 we study the MEB problem and in Section 2.1 a new integer linear formulation is introduced for the problem. In Section 3 we present a branch-and-price scheme for the MEBP problem. The computational experiments are shown in Section 4. In Section 4.1 we compare some different approaches for solving the MEB problem and in Section 4.2 we evaluate our implementation of the branch-and-price procedure proposed in Section 3. Finally, we present the concluding remarks and future work in Section 5.

2. Formulations for the MEB problem

Let $G = (V, E)$ be a simple (L, R)-bipartite graph. For simplicity, given an edge $uv \in E$, we assume as a convention that $u \in L$ and $v \in R$ (i.e., the label in the left side corresponds to the vertex in class L, and the label in the right side corresponds to the vertex in class R). We assume, without loss of generality, that $|L| \le |R|$. In particular, we are interested in very unbalanced bipartite graphs (i.e., R is much larger than L).

Given a subgraph B of G, we denote its vertex set by V_B, where $L_B = V_B \cap L$ and $R_B = V_B \cap R$, and its edge set by E_B. A biclique B of G is said to be *maximum* if $|E_B| \ge |E_{B'}|$, for each biclique B' of G. The MEB problem can be stated as follows.

Problem 2.1. Given a bipartite graph G, find a maximum biclique B of G.

We say that two edges uv and pq of E are *incompatible* if $u \ne p$, $v \ne q$ and $uq \notin E$ or $pv \notin E$. Let $\mathcal{I} = \{\{uv, pq\} \in E \times E \mid uv \text{ and } pq \text{ are incompatible}\}$ be the set of all pairs of incompatible edges of E.

Observation 2.1. An edge set which contains no incompatible edges and has maximum size induces a maximum biclique.

Based on Observation 2.1, Dawande et al. introduced in [5] the following formulation for the MEB problem.

$$\max \sum_{uv \in E} x_{uv}$$

$$\text{(IND SET)}\quad \text{s.t. } x_{uv} + x_{pq} \le 1, \quad \text{for each } \{uv, pq\} \in \mathcal{I} \tag{1}$$

$$x_{uv} \in \{0, 1\}, \quad \text{for each } uv \in E. \tag{2}$$

In (IND SET) we have that $x_{uv} = 1$ if and only if the edge uv is in the biclique (i.e., x is the characteristic vector of the biclique's edge set). Let $H = (V_H, E_H)$ be the graph defined as follows: $V_H = \{v_e \mid e \in E\}$ and $E_H = \{v_e v_f \mid \{e, f\} \in \mathcal{I}\}$. In this construction, there is a one-to-one correspondence between each *independent set* (i.e., a set of pair-wise non-adjacent vertices) in H and a feasible solution to (IND SET). Thus, formulation (IND SET) can be strengthened with well known valid inequalities for the stable set polytope, *odd-hole* and *clique* inequalities, for instance (see [11]). As Berman and Schnitger show in [4], the independent set problem is NP-hard to approximate by a factor of $O(n^{\epsilon})$, for some constant $\epsilon > 0$, where $n = |V_H|$ is the number of vertices in the input graph. Despite the hardness of solving, and even approximating, the independent set problem, there are some methods for solving (in practice) this problem in a reasonable running time, for considerable large graphs (see [2] for an integer programming approach and [10,15,21] for branch-and-bound algorithms). Thus, one approach that we consider in Section 4.1 for solving the MEB problem is to apply these methods for finding a maximum independent set in the graph H (or equivalently, finding a maximum clique in the complement of H).

2.1. A new formulation for the MEB problem

Given a vertex $u \in V$, we denote by $N(u) \subset V$ the set of vertices adjacent to u, and we define $\overline{N}(u) = R \setminus N(u)$, if $u \in L$, otherwise $\overline{N}(u) = L \setminus N(u)$. We denote the degree of a vertex u by $d(u) = |N(u)|$. Below, we introduce a new formulation for the MEB problem.

$$\max \sum_{u \in L} y_u$$

$$(\text{MEB V}) \quad \text{s.t. } z_u + z_v \le 1, \quad \text{for each } u \in L \text{ and } v \in \overline{N}(u) \tag{3}$$

$$y_u \le d(u) z_u, \quad \text{for each } u \in L \tag{4}$$

$$y_u \le \sum_{v \in N(u)} z_v, \quad \text{for each } u \in L \tag{5}$$

$$y_u \ge 0, \quad \text{for each } u \in L \tag{6}$$

$$z_u \in \{0, 1\}, \quad \text{for each } u \in V. \tag{7}$$

Lemma 2.1. (MEB V) *is a formulation for the* MEB *problem.*

Proof. Let (z^*, y^*) be an optimal solution to (MEB V). By constraints (3) we have that z^* is the characteristic vector of a vertex set which induces a biclique B^*. By constraints (4) we have that $y^*_u > 0$ implies $z^*_u = 1$. Since we maximize y, by constraints (5) we have that $y^*_u = |R_{B^*}|$, for each $u \in L_{B^*}$, which implies that $\sum_{u \in L} y^*_v = |E_{B^*}|$. Conversely, given a biclique B, one can easily construct a feasible solution to (MEB V) in which the objective function's value is $|E_B|$. □

In (IND SET) there are $O(|E|)$ variables and $O(|E|^2)$ constraints, while in (MEB V) there are $O(|V|)$ variables and $O(|L \parallel R|)$ constraints. The following observation allows us to adapt the formulation (MEB V) in order to reduce the number of binary variables to $O(|L|)$.

Observation 2.2. Given a subset $L' \subseteq L$, we have that $B = (L_B, R_B)$, where $L_B = L'$ and $R_B = \{v \in R \mid L' \subseteq N(v)\}$, is a biclique and $|E_B| \ge |E_{B'}|$, for each biclique B' such that $L_{B'} = L'$.

From Observation 2.2 follows that we can focus only on finding the appropriate subset of L. Below, we introduce a formulation for the MEB problem based on this observation.

$$\max \sum_{v \in R} y_v$$

$$(\text{MEB L}) \quad \text{s.t. } y_v \le \sum_{u \in N(v)} z_u, \quad \text{for each } v \in R \tag{8}$$

$$y_v \le (1 - z_u) d(v), \quad \text{for each } u \in L \text{ and } v \in \overline{N}(u) \tag{9}$$

$$y_v \ge 0, \quad \text{for each } v \in R \tag{10}$$

$$z_u \in \{0, 1\}, \quad \text{for each } u \in L. \tag{11}$$

The binary variables z and the continuous variables y are interpreted in the same way as in (MEB V). However, now the y variables are defined for the vertices in R instead of L, and z variables are defined only for the vertices in L. A maximum biclique B^* can be retrieved from an optimal solution (z^*, y^*) to (MEB L) in the following way: $L_{B^*} = \{u \in L \mid z^*_u = 1\}$ and $R_{B^*} = \{v \in R \mid y^*_v > 0\}$.

The number of constraints in (MEB L) can be reduced from $O(|L \parallel R|)$ to $O(|V|)$ by aggregating constraints (9) as below.

$$\sum_{v \in \overline{N}(u)} y_v \le \sum_{v \in \overline{N}(u)} d(v)(1 - z_u), \quad \text{for each } u \in L. \tag{12}$$

We denote by (MEB AGG) the formulation which is the same as (MEB L), but replacing constraints (9) by constraints (12). Considering the number of constraints multiplied by the number of variables, (MEB AGG) is the smallest formulation among the ones presented in this section. In practice, the effectiveness of a formulation depends not only on its size, but also it depends on the strength of its linear relaxation. Clearly, (MEB AGG) is weaker than (MEB L). However, it is not clear how good are the bounds given by the linear relaxations of (IND SET), (MEB V), (MEB L) and (MEB AGG). In Section 4.1 we compare the performance of all these formulations in practice.

3. A formulation for the MEBP problem

Given a biclique B, we denote by S_B the set of vertices in $V_B \cap S$, where $S \in \{V, L, R\}$. A *biclique packing* in G is a set $\mathcal{H}$ of bicliques of G, such that for each pair B, B' of distinct bicliques in $\mathcal{H}$, we have that $S_B \cap S_{B'} = \emptyset$. The MEBP problem can be stated as follows.

Problem 3.1. Given a bipartite graph G and a positive integer k, find a biclique packing $\mathcal{H}$ of G such that $|\mathcal{H}| \leq k$ and $\sum_{B\in\mathcal{H}} |E_B|$ is maximum.

If $S = V$, each vertex can be in at most one biclique inside the packing. On the other hand, if $S = L$ or $S = R$, this constraint applies to only one of the classes (L or R). In all cases, each edge can be in at most one biclique inside the packing. Note that if $k \geq |L|$ and $S = L$, the problem becomes trivial, since for each $u \in L$ the star induced by $\{u\} \cup N(u)$ is a biclique (the case in which $S = R$ and $k \geq |R|$ is analogous). On the other hand, if $k = 1$, we have the MEB problem.

Let $\mathcal{B}$ be the set of all bicliques which are subgraphs of G and let $\mathcal{B}(v) = \{B \in \mathcal{B} \mid v \in V_B\}$ be the set of all bicliques containing the vertex v, for each $v \in V$. We define variables $x_B \in \{0, 1\}$, for each $B \in \mathcal{B}$, with the following interpretation: $x_B = 1$ if and only if B is contained in the packing. Now, we introduce a formulation for the MEBP problem.

$$\max \sum_{B\in\mathcal{B}} |E_B| \cdot x_B$$

$$(\mathrm{P_{MEBP}}) \quad \text{s.t.} \sum_{B\in\mathcal{B}} x_B \leq k \tag{13}$$

$$\sum_{B\in\mathcal{B}(u)} x_B \leq 1, \quad \text{for each } u \in S \tag{14}$$

$$x_B \in \{0, 1\}, \quad \text{for each } B \in \mathcal{B}. \tag{15}$$

Constraint (13) guarantees that at most k bicliques are chosen and constraints (14) guarantee that the chosen bicliques are vertex disjoint with respect to S. The objective is to maximize the size of the packing. In [1] a more compact formulation is presented for this problem, in the sense that the number of constraints and variables is polynomial in the input size. We do not expose that formulation here because it has a very poor performance in practice, due to its symmetry.

Let ($\mathrm{L_{MEBP}}$) be the linear relaxation of ($\mathrm{P_{MEBP}}$), where constraints (15) are replaced by $x_B \geq 0$, for each $B \in \mathcal{B}$. The number of variables in ($\mathrm{L_{MEBP}}$) is exponential in the size of the input, thus we propose here a *column generation* algorithm (see [3] for a nice survey on the subject) for solving ($\mathrm{L_{MEBP}}$). For this algorithm, we need to investigate the corresponding so-called *pricing problem* (i.e., find a variable x_B with negative reduced price or give a proof that no such variable exists). First, consider the dual of ($\mathrm{L_{MEBP}}$).

$$\min \sum_{u\in S} \alpha_u + k\beta$$

$$(\mathrm{D_{MEBP}}) \quad \text{s.t. } \beta + \sum_{u\in S_B} \alpha_u \geq |E_B|, \quad \text{for each } B \in \mathcal{B} \tag{16}$$

$$\alpha_u \geq 0, \quad \text{for each } u \in S \tag{17}$$

$$\beta \geq 0. \tag{18}$$

The dual variable β is associated with the constraint (13) and the dual variables α are associated with the constraints (14). For simplicity, when $S \neq V$, we assume that $\alpha \in \mathbb{R}_+^{|V|}$ and $\alpha_v = 0$, for each $v \in V \setminus S$. For a given biclique B in $\mathcal{B}$, the reduced price of a variable x_B is given by $\hat{x}_B = \beta + \sum_{v\in V_B} \alpha_v - |E_B|$. Let $f : \mathcal{B} \times \mathbb{R}_+^{|V|} \to \mathbb{R}$ be the function defined as follows: $f(B, \alpha) = |E_B| - \sum_{v\in V_B} \alpha_v$. The pricing problem is strongly related to the problem of finding a variable with minimum reduced price, which in this case corresponds to solving the following optimization problem.

Problem 3.2. Given a bipartite graph G and a vector $\alpha \in \mathbb{R}_+^{|V|}$, find a biclique B in $\mathcal{B}$ such that $f(B, \alpha)$ is maximum.

The MEB problem is a special case of the Problem 3.2 in which $\alpha_v = 0$, for each $v \in V$. Hence, Problem 3.2 is NP-hard as well. Moreover, adapting the formulations for the MEB problem in order to solve the Problem 3.2 seems a natural idea and, except for formulation (MEB L), it is an easy task. As we show in Section 4.1, (MEB L) performs much better in practice than the other formulations presented in Section 2. Thus, now we focus on how to adapt (MEB L) in order to solve the Problem 3.2.

Observation 3.1. Given a vector $\alpha \in \mathbb{R}_+^{|V|}$ and a subset $L' \subseteq L$, we have that $B = (L_B, R_B)$, where $L_B = L'$ and $R_B = \{v \in R \mid L' \subseteq N(v) \text{ and } \alpha_v < |L'|\}$, is a biclique and $f(B, \alpha) \geq f(B', \alpha)$, for each biclique B' such that $L_{B'} = L'$.

Therefore, for the Problem 3.2 we also have the convenient property that we can focus only on finding the appropriate subset of L in order to find a biclique B in $\mathcal{B}$ which maximizes $f(B, \alpha)$. However, the Problem 3.2 is a little more tricky to formulate using only $O(|L|)$ binary variables, as in (MEB L), when $S \neq L$. We define $\Gamma_{\mathrm{MEB}} = \{(z, y) \in \{0, 1\}^{|L|} \times \mathbb{R}_+^{|R|} \mid (z, y)$

satisfies constraints (8), (9), …, (11) from (MEB L)}. Below, we introduce a formulation for the Problem 3.2 and later on we discuss what can be simplified if $S = L$.

$$\max \sum_{v\in R}(y_v - \alpha_v r_v) - \sum_{u\in L}\alpha_u z_u$$

$$(\mathrm{P_{PRC}})\ \text{s.t.} \sum_{i=1}^{|L|} l_i = 1 \tag{19}$$

$$\sum_{u\in L} z_u = \sum_{i=1}^{|L|} i l_i \tag{20}$$

$$y_v \le |N(v)| \sum_{i=\lfloor\alpha_v+1\rfloor}^{|L|} l_i, \quad \text{for each } v \in R \tag{21}$$

$$\sum_{i=\lfloor\alpha_v+1\rfloor}^{|L|} l_i \le r_v + \sum_{u\in\overline{N}(v)} z_u, \quad \text{for each } v \in R \tag{22}$$

$$(z, y) \in \Gamma_{\mathrm{MEB}} \tag{23}$$

$$r_v \ge 0, \quad \text{for each } v \in R \tag{24}$$

$$l_i \in \{0, 1\}, \quad \text{for } i = 1, \ldots, |L|. \tag{25}$$

The interpretation for the variables is the following: $r_v = 1$ if and only if v is in the biclique; $l_i = 1$ if and only if $|L_B| = i$, where $L_B \subseteq L$ is the set of vertices in L which are in the biclique; variables z and y are interpreted in the same way as in (MEB L).

Given an optimal solution (z^*, y^*, r^*, l^*) to $(\mathrm{P_{PRC}})$, let B^* be the biclique such that $L_{B^*} = \{u \in L \mid z^*_u = 1\}$ and $R_{B^*} = \{v \in R \mid y^*_v > 0\}$. Note that $v \in R_{B^*}$ if and only if (i) $\alpha_v < \sum_{u\in L} z^*_u$ and (ii) $u \in N(v)$, for each $u \in L$ such that $z^*_u = 1$. Thus, the vertices in R are chosen correctly, according to Observation 3.1. More precisely, constraints (19) and (20) are introduced in order to "keep track" of the cardinality of L^*_B, while constraints (21) are introduced for preventing the vertices in R, which do not satisfy the property of Observation 3.1, from being included in the biclique. Constraints (22) guarantee that if v is chosen to be in the biclique, then $r_v = 1$. The r variables are defined in order to calculate the value of $f(B^*, \alpha)$ correctly in the objective function.

Note that we define in $(\mathrm{P_{PRC}})$ only $2|L| + 1$ binary variables. But, since l and z are integral, then in any optimal solution r and y are integral as well.

If $S = L$, we can remove the variables r and l, and also remove all the constraints, except (23) (in this case we have the same formulation as (MEB L), except for the objective function).

One can solve $(\mathrm{L_{MEBP}})$ by column generation, using a MIP solver for solving $(\mathrm{P_{PRC}})$ at each pricing step. Alternatively, one can use heuristics for finding columns with negative reduced price (not necessarily minimum) and solve $(\mathrm{P_{PRC}})$ only if the heuristics fail. Since we cannot expect the optimal solution found to $(\mathrm{L_{MEBP}})$ to be integral, we propose now a *branch-and-price* scheme for solving $(\mathrm{P_{MEBP}})$.

3.1. *A branch-and-price scheme for the* MEBP *problem*

Given an optimal solution x to $(\mathrm{L_{MEBP}})$, let $\lambda_{uv} = \sum_{B\in\mathcal{B}(u)\cap\mathcal{B}(v)} x_B$ be an implicit variable, for each pair of distinct vertices $\{u, v\} \subset S$, with the following interpretation: $\lambda_{uv} = 1$ if and only if the vertices u and v are in the same biclique. In this section, we introduce a *branch-and-price* algorithm which uses these variables for branching. The branching rule proposed here is similar to the one introduced by Ryan and Foster in [17] for the scheduling problem. For more general branching rules we refer to the work of Vanderbeck and Wolsey [22].

Given a variable λ_{uv} with fractional value μ, one can branch into the following two possibilities: $\lambda_{uv} \le \lfloor\mu\rfloor = 0$ or $\lambda_{uv} \ge \lceil\mu\rceil = 1$. Thus, we are interested in how to solve the following problem.

Problem 3.3. Given two sets $F_1 \subset S \times S$ and $F_0 \subset S \times S$, find a solution to $(\mathrm{L_{MEBP}})$ subject to $\lambda_{uv} = 1$, for each $\{u, v\} \in F_1$, and $\lambda_{uv} = 0$, for each $\{u, v\} \in F_0$.

We define $\mathcal{B}_{F_0} = \{B \in \mathcal{B} \mid \{u, v\} \not\subseteq V_B$, for each $\{u, v\} \in F_0\}$ and $X_{\mathrm{MEBP}} = \{x \in \mathbb{R}^{\mathcal{B}}_+ \mid x$ satisfies (13) and (14) from $(\mathrm{P_{MEBP}})$ and $x \ge 0\}$. Consider the following formulation for the Problem 3.3.

$$\max \sum_{B\in\mathcal{B}_{F_0}} |E_B| \cdot x_B$$

$$(\mathrm{L^2_{MEBP}})\ \text{s.t.} \sum_{B\in\mathcal{B}_{F_0}(u)\cap\mathcal{B}_{F_0}(v)} x_B \ge 1, \quad \text{for each } \{u, v\} \in F_1 \tag{26}$$

$$x \in X_{\mathrm{MEBP}}. \tag{27}$$

Since we include in ($\mathrm{L}^2_{\mathrm{MEBP}}$) only columns in $\mathcal{B}_{F_0}$, then we have that $\lambda_{uv} = 0$, for each $\{u, v\} \in F_0$, and by constraints (26) and (27) we have that $\lambda_{uv} = 1$, for each $\{u, v\} \in F_1$. Thus ($\mathrm{L}^2_{\mathrm{MEBP}}$) is a formulation for the Problem 3.3.

Given an optimal solution x to ($\mathrm{L}^2_{\mathrm{MEBP}}$), we define $\omega(x) = \sum_{B\in\mathcal{B}} |E_B|x_B$. If x is integral, thus we do not need to branch on the corresponding node. Otherwise, we have to select a variable λ_{uv} with fractional value (a candidate for branching). In Lemma 3.3 we prove that if x is fractional, then there exists a variable λ_{uv} with fractional value. First, we prove the following two auxiliary lemmas.

Lemma 3.1. *Given an optimal solution x to ($\mathrm{L}^2_{\mathrm{MEBP}}$), such that x is fractional, we have that x has at least two entries with fractional value.*

Proof. Since the right hand sides of all constraints in ($\mathrm{L}^2_{\mathrm{MEBP}}$) are integral, then the result follows immediately. □

Lemma 3.2. *Let x be an optimal solution to ($\mathrm{L}^2_{\mathrm{MEBP}}$), such that for each pair of distinct variables x_{B_1} and x_{B_2} with fractional values we have that either $S_{B_1} \cap S_{B_2} = \emptyset$ or $S_{B_1} = S_{B_2}$. Thus, there exists an optimal solution x^* to ($\mathrm{L}^2_{\mathrm{MEBP}}$), such that x^* is integral and $\omega(x) = \omega(x^*)$.*

Proof. Let $\Psi : \mathbb{R}_+^{|\mathcal{B}|} \times \mathcal{B} \times \mathcal{B} \times \mathbb{R}_+ \Rightarrow \mathbb{R}_+^{|\mathcal{B}|}$ be the function defined as follows: $\Psi(x, B_1, B_2, \epsilon) = x'$, where $x'_{B_1} = x_{B_1} + \epsilon$, $x'_{B_2} = x_{B_2} - \epsilon$ and $x'_B = x_B$, for each $B \in \mathcal{B} \setminus \{B_1, B_2\}$. We prove the lemma by induction on n, where n is the number of variables with fractional value. If $n = 0$, then the result follows immediately. Now, suppose that $n > 0$. By Lemma 3.1, we have that $n \geq 2$. Let $x_{B_{\max}}$ be a variable with fractional value, such that $|E_{B_{\max}}|$ is maximum. Note that one of the following two cases must occur: (i) $S_{B_{\max}} = S_{B_{\mathrm{eq}}}$, for some other variable $x_{B_{\mathrm{eq}}}$ with fractional value; (ii) $S_{B_{\max}} \cap S_B = \emptyset$, for each other variable x_B with fractional value.

For the case (i), consider the vector $x' = \Psi(x, B_{\max}, B_{\mathrm{eq}}, \epsilon)$, where $\epsilon = \min\{1 - x_{B_{\max}}, x_{B_{\mathrm{eq}}}\}$. Since $S_{B_{\max}} = S_{B_{\mathrm{eq}}}$, then x' does not violate any constraint of ($\mathrm{L}^2_{\mathrm{MEBP}}$). Moreover, we have that $x'_{B_{\max}} = 1$ or $x'_{B_{\mathrm{eq}}} = 0$ (i.e., x' has less than n variables with fractional values), and also $\omega(x') \geq \omega(x)$, which implies $\omega(x') = \omega(x)$, since x is optimal. Thus, by the induction hypothesis, we have that there is a feasible solution x^* such that $\omega(x^*) = \omega(x') = \omega(x)$ and x^* is integral.

For the case (ii), let $x_{B_{neq}}$ be a variable with fractional value, such that $B_{\max} \neq B_{neq}$. Since constraints (26) are satisfied with equality, then we have that $\{u, v\} \not\subseteq S_{B_{\max}}$, for each $\{u, v\} \in F_1$. By (ii), we have that the value of the slack variable in row corresponding to u in constraints (14) is $\tau_u = 1 - x_{B_{\max}}$, for each $u \in S_{B_{\max}}$. Therefore, $x' = \Psi(x, B_{\max}, B_{neq}, \epsilon)$ does not violate any constraint of ($\mathrm{L}^2_{\mathrm{MEBP}}$), where $\epsilon = \min\{1 - x_{B_{\max}}, x_{B_{neq}}\}$. As in case (i), we have that $x'_{B_{\max}} = 1$ or $x'_{B_{neq}} = 0$ and, by the induction hypothesis, there is a feasible solution x^* such that $\omega(x^*) = \omega(x') = \omega(x)$ and x^* is integral. □

Lemma 3.3. *Given an optimal solution x to ($\mathrm{L}^2_{\mathrm{MEBP}}$), such that x is fractional, we have that λ_{uv} is fractional for some $u, v \in S$ or there exists a feasible solution x^* to ($\mathrm{L}^2_{\mathrm{MEBP}}$), such that $\omega(x) = \omega(x^*)$ and x^* is integral.*

Proof. By Lemma 3.1, there are at least two entries in x with fractional value, thus one of the following two cases must occur: (i) for each pair of distinct variables x_{B_1} and x_{B_2} with fractional values, we have that either $S_{B_1} \cap S_{B_2} = \emptyset$ or $S_{B_1} = S_{B_2}$; (ii) there are two distinct variables x_{B_1} and x_{B_2} with fractional values, such that $u \in S_{B_1} \cap S_{B_2}$ and $v \in S_{B_1} \nabla S_{B_2}$, for some $u, v \in S$, where $S_{B_1} \nabla S_{B_2} = (S_{B_1} \cup S_{B_2}) \setminus (S_{B_1} \cap S_{B_2})$.

In case (i), by Lemma 3.2, we have that there is a feasible solution x^* such that $\omega(x^*) = \omega(x)$ and x^* is integral. In case (ii), without loss of generality, suppose that $v \in S_{B_1}$ and $v \notin S_{B_2}$. Since $B_1 \in \mathcal{B}(u) \cap \mathcal{B}(v)$, then $\sum_{B\in\mathcal{B}(u)\cap\mathcal{B}(v)} x_B \geq x_{B_1} > 0$. Since $B_2 \notin \mathcal{B}(u) \cap \mathcal{B}(v)$, $B_2 \in \mathcal{B}(u)$ and $\sum_{B\in\mathcal{B}(u)} x_B \leq 1$ (by constraints (14)), then we have that $\sum_{B\in\mathcal{B}(u)\cap\mathcal{B}(v)} x_B \leq \sum_{B\in\mathcal{B}(u)} x_B - x_{B_2} < 1$. Thus, we have that $0 < \lambda_{uv} < 1$. □

Now, we investigate the pricing problem with respect to ($\mathrm{L}^2_{\mathrm{MEBP}}$). The reduced price of a variable x_B is given by

$$\hat{x}_B = \beta + \sum_{v\in V_B} \alpha_v - |E_B| - \sum_{\{u,v\}\subseteq S_B\cap F_1} \gamma_{uv},$$

where γ is the vector of dual variables associated with constraints (26). Let $\phi(B, \alpha, \gamma) : \mathcal{B}_{F_0} \times \mathbb{R}_+^{|V|} \times \mathbb{R}_+^{|F_1|} \to \mathbb{R}$ be the function defined as follows: $\phi(B, \alpha, \gamma) = |E_B| + \sum_{\{u,v\}\subseteq S_B\cap F_1} \gamma_{uv} - \sum_{u\in V_B} \alpha_u$. The problem of finding a minimum reduced price variable x_B with respect to ($\mathrm{L}^2_{\mathrm{MEBP}}$) can be stated as follows.

Problem 3.4. Given a bipartite graph $G = (V, E)$, two sets $F_1 \subseteq S \times S$ and $F_0 \subseteq S \times S$, and two vectors $\alpha \in \mathbb{R}_+^{|V|}$ and $\gamma \in \mathbb{R}_+^{|F_1|}$, find a biclique B, subgraph of G, such that $B \in \mathcal{B}_{F_0}$ and $\phi(B, \alpha, \gamma)$ is maximum.

We define binary variables z_u, not only for each $u \in L$ as before, but also for each $u \in R_F$, where $R_F = \{v \in R \mid v \in \{u, v\}$, for some $\{u, v\} \in F_1 \cup F_0\}$, and we denote by $z_{|L}$ the vector z restricted to the elements in L. On the other hand, we define variables r_v only for each $v \in \overline{R_F}$, where $\overline{R_F} = R \setminus R_F$. The variables y and l are defined exactly as in ($\mathrm{P}_{\mathrm{PRC}}$). Finally, we define

$\Upsilon_{\text{PRC}} = \{(z_{|L}, l, y, r) \in \{0,1\}^{|L|} \times \{0,1\}^{|L|} \times \mathbb{R}_+^{|R|} \times \mathbb{R}_+^{|\overline{R_F}|} \mid (z_{|L}, y, l, r)$ satisfies (19)–(25) from (P_{PRC}), where (21), (22) and (24) are defined only for each $v \in \overline{R_F}\}$. Consider the following formulation for the Problem 3.4.

$$\max \sum_{v\in R} y_v + \sum_{\{u,v\}\in F_1} \gamma_{uv} z_u - \sum_{v\in\overline{R_F}} \alpha_v r_v - \sum_{u\in L\cup R_F} \alpha_u z_u$$

$$(\text{P}^2_{\text{PRC}}) \text{ s.t. } z_u + z_v \le 1, \quad \text{for each } \{u,v\} \in F_0 \tag{28}$$

$$z_u = z_v, \quad \text{for each } \{u,v\} \in F_1 \tag{29}$$

$$y_v \le d(v) z_v, \quad \text{for each } v \in R_F \tag{30}$$

$$\sum_{u\in\bar{N}(v)} z_u \le (1 - z_v)|\overline{N}(v)|, \quad \text{for each } v \in R_F \tag{31}$$

$$(z_{|L}, l, y, r) \in \Upsilon_{\text{PRC}} \tag{32}$$

$$z_u \in \{0,1\}, \quad \text{for each } u \in L \cup R_F. \tag{33}$$

Given an optimal solution (z^*, y^*, r^*, l^*) to (P^2_{PRC}), let B^* be the biclique such that $u \in L_{B^*}$ if and only if $z^*_u = 1$, and $v \in R_{B^*}$ if and only if $y^*_v > 0$. For simplicity, we assume that if $v \in R_F$ and $z_v = 1$, then $y^*_v > 0$ (if it does not hold we would have $L_{B^*} = \emptyset$ and $R_{B^*} \neq \emptyset$, but the formulation works for this case as well). Constraints (28) guarantee that $B^* \in \mathcal{B}_{F_0}$. From constraints (26) in (L^2_{MEBP}) follows that $x_B = 0$ in any optimal solution, for each $B \in \mathcal{B}$ such that $u \in S_B$ and $v \notin S_B$, for some $\{u, v\} \in F_1$. Thus, we introduce constraints (29) in order to prevent the generation of such variables. Constraints (30) guarantee that if $z^*_v = 0$, then $y^*_v = 0$ as well. On the other hand, constraints (31) guarantee that $z^*_v = 1$ only if $N(v) \cap L_{B*} = L_{B*}$, for each $v \in R_F$. Since the value of the objective function for (z^*, l^*, y^*, r^*) is precisely $\phi(B^*, \alpha, \gamma)$, thus ($\text{P}^2_{\text{PRC}}$) is a formulation for the Problem 3.4.

Again, we can simplify the formulation of the pricing problem if $S = L$. In this case we have that $R_F = \emptyset$, thus we have no constraints (30) and (31). If for a given instance our branch-and-price procedure needs to investigate a huge number of nodes in the branch-and-bound tree, then we do not expect the program to stop in reasonable running time. Thus, we are interested in the cases in which the number of investigated nodes is relatively small (i.e., F_1 and F_0 are small and, consequently, R_F is small as well). Considering this assumption, the formulation (P^2_{PRC}) has almost the same number of binary variables as (P_{PRC}).

3.2. Considerations about our implementation

There are many details that have to be carefully taken into account in order to make a good implementation of a branch-and-price algorithm. In this section we comment the issues that have more impact in our implementation, as evaluated in various experiments (not all of them are reported in this paper):

- *How to construct a first set of columns for the master LP*: we use a greedy algorithm which takes a maximum biclique B in G, add x_B in the master LP and repeat (k times) recursively the process for $G[V \setminus S_B]$. Note that this procedure also produces a feasible solution to (P_{MEBP}) and, consequently, a lower bound on the optimal value. We refer to this procedure as the *greedy heuristic* (GH).
- *Heuristics for solving the pricing problem*: usually, MIP solvers provide callback routines for getting feasible solutions found during the optimization process, thus we include all collected solutions with negative reduced price, not only the optimal one. Moreover, if a certain number of solutions with negative reduced price are found, we stop the solver before finding a variable with minimum reduced price.
- *Heuristics for getting primal bounds*: at any time we can obtain a feasible solution by solving the master IP restricted to the columns added in the master LP up to the moment and, as we noted empirically, it takes a very short time to do. Before doing that, we carry out the following pre-processing: for each pair of variables x_{B_1} and x_{B_2} with positive value in the current solution, we add to the master LP the columns $x_{B'_1}$ and $x_{B'_2}$, where $V_{B'_1} = V_{B_1} \setminus (S_{B_1} \cap S_{B_2})$ and $V_{B'_2} = V_{B_2} \setminus (S_{B_1} \cap S_{B_2})$. We refer to this procedure applied at the first node of the branch-and-bound tree as the *post-processing heuristic* (PPH).
- *How to select the next node to be explored in the branch-and-bound tree and how to choose a variable with fractional value for branching*: since we are leading with a maximization problem, the node's selection policy affects more directly the upper bound. As is usually done in this case, we select a node with the biggest upper bound and branch in a more fractional variable.

4. Computational experiments

In this section we present computational experiments with randomly generated instances and also with instances that come from the applications in metabolic networks and product bundling. We used *CPLEX*©12.2 as the MIP solver and the machine configurations are the following: eight processors Intel©Xeon©E5440 (2.83 GHz) and 32 GB of RAM.

Table 1
Randomly generated graphs ($|L| = 10$ and $|R| = 3000$).

Dns	(MEB V)			(MEB L)			(MEB AGG)			(JK)
	Time (s)	Nd	Gap (%)	Time	Nd	Gap (%)	Time (s)	Nd	Gap (%)	Time
0.1	20	75	361	25 s	0	361	4	71	361	1 s
0.2	38	82	376	1 m 06 s	0	376	9	105	376	16 s
0.3	47	88	381	1 m 25 s	13	381	11	109	381	1 m 16 s
0.4	31	140	383	1 m 14 s	175	383	12	197	383	4 m 37 s
0.5	28	234	372	53 s	406	371	13	372	371	13 m 14 s
0.6	24	351	298	50 s	585	296	14	647	296	33 m 58 s
0.7	20	597	228	39 s	1333	221	24	1256	223	TLE
0.8	17	1207	148	23 s	1897	131	23	1499	135	TLE
0.9	4	0	58	5 s	278	27	10	35	38	TLE

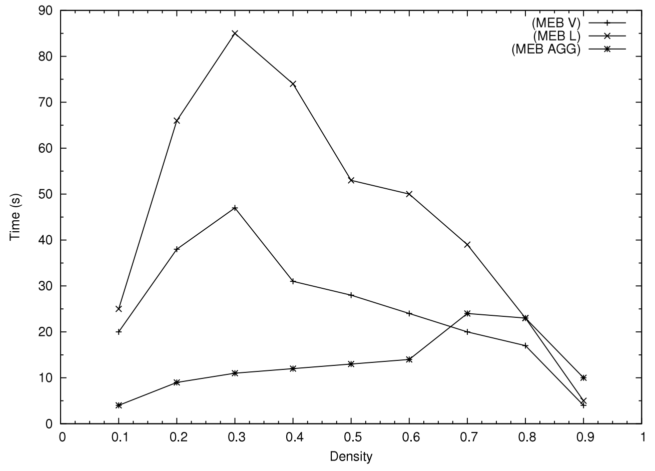

Fig. 1. Running time of formulations (MEB V), (MEB L) and (MEB AGG).

4.1. *Computational experiments with* MEB *problem*

We consider two approaches for solving the MEB problem. The first one is to solve (MEB V), (MEB L) and (MEB AGG) in a Mixed Integer Program (MIP) solver, and the second one is to use algorithms for finding a maximum clique in the complement of the graph H, as described in Section 2. We do not report the experiments made with formulation (IND SET), since its performance is really poor, even for solving only its linear relaxation. For solving the maximum clique problem we used the branch-and-bound algorithm introduced by Janez Konc [10], which we denote by (JK) (the source code is available at http://www.sicmm.org/~konc/maxclique/).

We present the results of the experiments with randomly generated instances in Table 1. In each row of Table 1 we show the arithmetic means between the instances solved to optimality in less than 1 h, where the number inside brackets indicates how many instances, among the 10 generated, were solved to optimality. Column "Dns" has the density of the input graph (i.e., $\frac{|E|}{|R|\cdot|L|}$), column "Nd" contains the number of nodes explored by cplex in the branch-and-bound tree and column "Gap" contains the *gap* between the linear relaxation (before cplex adding general purpose cutting-planes) and the optimal integer solution.

In Fig. 1 we draw a graph of the running time of our formulations. We do not include the (JK) algorithm in the graphic because it is much slower than the others and for graphs with density greater than 0.6 it was *time limit exceeded* (TLE). In the case of algorithm (JK) and formulation (MEB AGG), the smaller is the density of the graph, the easier it becomes to solve the problem. On the other hand, in the case of formulations (MEB L) and (MEB V), the gap seems to have more influence on the results than the density of the graph. All the formulations have almost the same values of gap. Since (MEB V) has more binary variables than (MEB L), it gives more information for the solver to generate good cuts and to make a better pre-processing in the constraint matrix. Although (MEB AGG) has the same variables as (MEB L), its performance is better than the other ones, for graphs with density up to 0.6, due to its reduced size. But this trick does not work with the "aggregated version" of (MEB V). We made tests with it and we decided not to report them, since the results are far from being reasonable, due to its very weak linear relaxation. For graphs with density 0.7 or greater, (MEB V) performed slightly better than (MEB AGG).

In Table 2 we show the experiments with instances that come from applications in product bundling (instances PB1 and PB2) and in metabolic networks (instances MN1 and MN2). For these instances, (MEB AGG) performed rather better than the other methods. For instances that come from the application in metabolic networks the algorithm (JK) performed very poorly (in instance MN2 it got a *memory limit exceeded*—MLE).

Table 2
Instances that come from the applications.

Inst.	$\|L\|$	$\|R\|$	Dns	(MEB V)			(MEB L)			(MEB AGG)			(JK)
				Time	Nd	Gap (%)	Time	Nd	Gap (%)	Time	Nd	Gap	Time
PB1	32	760	0.4	1 m 02 s	3 038	179	52 s	2 295	178	20 s	966	188	4 m 09 s
PB2	77	760	0.3	1 h 00 m 00 s	128 285	358	24 m 52 s	29 961	358	1 m 51 s	11 146	361	6 m 10 s
MN1	16	4637	0.6	3 m 25 s	391	190	3 m 17 s	779	190	1 m 29 s	613	192	12 h 36 m 31 s
MN2	40	4637	0.4	46 m 59 s	1 198	255	1 h 00 m 00 s	2 508	255	4 m 34 s	935	255	MLE

4.2. *Computational experiments with* MEBP *problem*

In this section we show the experiments with MEBP problem. We present the results of the experiments with randomly generated instances in Table 3. In each row of Table 3 we show the arithmetic means between the instances solved to optimality in less than 1 h, among the 10 generated. The first two columns show the parameters k and S (we tested several values for these parameters and selected the ones we consider more significant), while the third column shows the density of the input graph. The next four columns have the time spent to run the greedy heuristic "GH", to solve the master linear program "LP", to run the post-processing heuristic "PPH" and to solve the integer program with branch-and-price "IP". The next two columns show the number of pricing problems solved "Prc" and the number of explored nodes in the branch-and-bound tree "Nd". The next two columns "GH" and "PPH" show the gap between heuristic solutions and the optimal solution, respectively. Column "OPT" shows the number of instances solved to optimality. The next two columns show the number of tests which were stopped by timeout, where column "LP" corresponds to instances for which even the LPs were not solved and column "IP" corresponds to instances for which the timeout occurred after solving the LP. Finally, the last column "Gap" shows the gap between the upper and lower bounds when the timeout occurred.

As shown in Table 3, the results are quite good for $S = L$. In this case, all instances were solved to optimality at the root node of the branch-and-bound tree and the number of pricing problems solved as well as the total time needed for solving these instances was quite small. Intuitively, since $S = L$ is small, the number of intersecting bicliques tends to be smaller than in other cases. In all choices of S, the smaller is the density of the graph, the easier it becomes to solve the problem. The case in which $S = V$ is slightly more difficult than when $S = R$. This can be better seen if we consider the last three columns. Some values in Table 3 should be carefully analyzed to not be misinterpreted. For example, when $k = 6$ (and $k = 9$), dens $= 0.5$ and $S = V$, just one instance was solved to optimality and in those cases the time spent to solve the IP (after column generation) was 0 s. In cases like these the last three columns express better what happened in the tests.

The gap between GH and the optimal value is reasonable in most cases, while the solutions produced by PPH are in all cases very close to optimal. On the other hand, the time needed by GH is rather shorter than the time needed by PPH (note that PPH can be executed only after column generation terminates). Moreover, in most cases the time needed to prove that the solution found by PPH is optimal is much larger than the time needed to find the solution itself. The value of k and the density of the input graph seem to have an influence on the quality of the solutions found by GH. In all cases, the bigger k is, the worse the gap is. When $S = L$, GH obtains worse gaps when the input graph is more dense. On the other hand, when $S = V$ or $S = R$, GH obtains worse gaps when the input graph is less dense.

In Table 4 we show the experiments with instances that come from applications in product bundling and metabolic networks. We solved these instances for $k = 3, \dots, 9$ and calculated the arithmetic means of the results. In instances with $S = R$ or $S = V$ we could not solve even the master linear program for any value of k (we stopped the program after 48 h of processing). Thus, we consider only the case in which $S = L$.

As shown in Table 4, the number of pricing problems solved in all instances was very small and the optimal solutions were found at the root node (except in instance PB1 for which it was necessary to explore only one more node). The hardest instance was PB2, in which $|L|$ is the biggest one.

The data for the application in product bundling comes from a major food company in Chile. It was constructed from the historical weekly demand of 760 small retailers over 497 SKU. After preprocessing the data, we obtain an instance of 77 products and 760 clients (instance PB2). The interest of the company was to construct at most five product bundles. The resulting five bundles correspond to 44% of the total demand.

5. Concluding remarks and future work

We presented new formulations for MEB and MEBP problems and evaluated them with computational experiments with randomly generated instances. Moreover, we described two important applications of the MEBP problem and solved instances that come from these applications. Our formulations were specially designed for instances in which the input graphs are very unbalanced and, as shown in the computational experiments, we obtained good results for this case. Our experiments also show that pure combinatorial algorithms for the maximum clique problem can be successfully applied in order to solve small instances of the MEB problem.

Future work can be done on the MEB problem by developing pure combinatorial algorithms for solving it (adapting the algorithms for the clique problem seems a promising idea). Perhaps, such an algorithm can be adapted in order to improve

Table 3
Randomly generated bipartite graphs ($|L| = 10$ and $|R| = 100$).

S	k	Dns	Time				Number of		Gap		#OPT	Timeout		
			GH (s)	LP	PPH (s)	IP	Prc	Nd	GH (%)	PPH (%)		#LP	#IP	Gap (%)
L	3	0.3	0	0 s	0	0 s	2	0	0	0	10	–	–	–
		0.5	1	1 s	0	0 s	3	0	0	0	10	–	–	–
		0.8	0	5 s	0	0 s	17	0	15	0	10	–	–	–
	6	0.3	1	0 s	0	0 s	2	0	0	0	10	–	–	–
		0.5	1	1 s	0	0 s	5	0	1	0	10	–	–	–
		0.8	0	2 s	0	0 s	21	0	56	0	10	–	–	–
	9	0.3	1	0 s	0	0 s	2	0	0	0	10	–	–	–
		0.5	1	1 s	0	0 s	7	0	34	0	10	–	–	–
		0.8	2	2 s	0	0 s	21	0	75	0	10	–	–	–
R	3	0.3	0	41 s	1	4 m 19 s	522	15	9	1	80	20	–	–
		0.5	1	1 m 57 s	1	7 m 43 s	365	13	8	1	90	10	–	–
		0.8	1	13 m 14 s	1	9 m 15 s	373	2	8	0	90	–	10	1
	6	0.3	0	1 m 11 s	1	5 m 14 s	586	21	32	2	60	10	30	2
		0.5	1	5 m 18 s	3	14 m 46 s	452	14	25	0	30	–	70	1
		0.8	1	9 m 42 s	2	24 m 27 s	555	14	23	1	30	–	70	1
	9	0.3	1	1 m 12 s	2	8 m 10 s	495	35	47	2	80	–	20	2
		0.5	1	6 m 56 s	4	18 m 40 s	512	20	41	0	40	–	60	2
		0.8	3	15 m 8 s	3	22 m 18 s	569	21	31	1	60	10	30	1
V	3	0.3	0	36 s	1	2 m 01 s	387	11	9	1	80	20	–	–
		0.5	1	1 m 33 s	1	10 m 37 s	464	15	7	1	90	–	10	2
		0.8	0	37 m 49 s	14	16 m 31 s	883	2	3	0	50	10	40	3
	6	0.3	0	1 m 15 s	3	7 m 36 s	633	24	27	3	50	10	40	4
		0.5	0	3 m 55 s	1	0 s	270	0	7	0	10	–	90	6
		0.8	0	37 m 57 s	18	9 m 17 s	941	0	1	0	30	20	50	3
	9	0.3	0	1 m 21 s	44	8 m 15 s	521	9	25	3	80	–	20	5
		0.5	1	3 m 25 s	1	0 s	283	0	9	0	10	–	90	6
		0.8	0	43 m 13 s	21	7 m 18 s	1006	0	1	0	30	30	40	3

Table 4
Experiments with instances that come from the applications.

Inst.	Input Size			Time				Number of		Gap	
	$\|L\|$	$\|R\|$	Dns	GH	LP	PPH (s)	IP (s)	Prc	Nd	GH (%)	PPH (%)
PB1	32	760	0.4	5 m 24 s	19 m 59 s	0	23	31	1	27	0
PB2	77	760	0.3	6 h 26 m 24 s	30 h 58 m 37 s	0	0	16	0	8	0
MN1	16	4637	0.6	28 m 50 s	1 h 10 m 59 s	0	0	11	0	17	0
MN2	40	4637	0.4	1 h 26 m 28 s	11 h 29 m 9 s	0	0	12	0	5	0

the performance of our branch-and-price algorithm, since the MEB problem is strongly related with the pricing problem of our formulation to the MEBP problem.

Acknowledgments

We thank the *STIC-Amsud* project, Anillo project ACT-88, CAPES and CNPq (Proc. No. 302736/2010-7 and 201036/2010-0) for supporting part of this research.

References

[1] V. Acuña, C. Ferreira, A. Freire, E. Moreno, The biclique k-clustering problem in bipartite graphs and its application in bioinformatics, Electronic Notes in Discrete Mathematics 36 (2010) 159–166. ISCO 2010—International Symposium on Combinatorial Optimizatin.
[2] E. Balas, S. Ceria, G. Cornuéjols, G. Pataki, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, in: Cliques, Coloring and Satisfiability: Second DIMACS Implementation Challenge, vol. 26, 1993, (Chapter) Polyhedral Methods for the Maximum Clique Problem.
[3] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W. Savelsbergh, P.H. Vance, Branch-and-price: column generation for solving huge integer programs, Operations Research 3 (1998) 316–329.
[4] P. Berman, G. Schnitger, On the complexity of approximating the independent set problem, Information and Computation 96 (1) (1992) 77–94.
[5] M. Dawande, P. Keskinocak, S. Tayur, On the biclique problem in bipartite graphs, Tech. Rep., Carnegie-Mellon University, 1997.
[6] N. Faure, P. Chrétienne, E. Gourdin, F. Sourd, Biclique completion problems for multicast network design, Discrete Optimization 4 (3–4) (2007) 360–377.
[7] U. Feige, Relations between average case complexity and approximation complexity, in: Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing, STOC'02, ACM, New York, NY, USA, 2002, pp. 534–543.
[8] U. Feige, S. Kogan, Hardness of approximation of the balanced complete bipartite subgraph problem, Tech. Rep., 2004.
[9] M.R. Garey, D.S. Johnson, Computers and Intractability, A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, New York, 1979.
[10] J. Konc, D. Janezic, An improved branch and bound algorithm for the maximum clique problem, MATCH Communications in Mathematical and in Computer Chemistry 58 (2007) 569–590.

[11] L. Lovász, A. Schrijver, Matrix cones, projection representations, and stable set polyhedra, DIMACS Series in Discrete Mathematics and Theoretical Computer Science 1 (1990) 1–17.
[12] S.C. Madeira, A.L. Oliveira, Biclustering algorithms for biological data analysis: a survey, IEEE/ACM Transactions on Computational Biology and Bioinformatics 1 (1) (2004) 24–45.
[13] R.A. Notebaart, B. Teusink, R.J. Siezen, B. Papp, Co-regulation of metabolic genes is better explained by flux coupling than by network distance, PLoS Comput. Biol. (4) (2008) 157–163.
[14] D. Nussbaum, S. Pu, J.-R. Sack, T. Uno, H. Zarrabi-Zadeh, Finding maximum edge bicliques in convex bipartite graphs, in: M. Thai, S. Sahni (Eds.), Computing and Combinatorics, in: Lecture Notes in Computer Science, vol. 6196, Springer, Berlin, Heidelberg, 2010, pp. 140–149.
[15] P.R.J. Östergård, A fast algorithm for the maximum clique problem, Discrete Applied Mathematics 120 (2002) 197–207.
[16] R. Peeters, The maximum edge biclique problem is NP-complete, Discrete Applied Mathematics (2003) 651–654.
[17] D.M. Ryan, B.A. Foster, An integer approach to scheduling, Computer Scheduling of Public Transport (1981) 269–280.
[18] S. Schuster, C. Hilgetag, On elementary flux modes in biochemical reaction systems at steady state, Journal of Biological Systems 2 (2) (1994) 165–182.
[19] J. Tan, Inapproximability of maximum weighted edge biclique and its applications, in: Proceedings of the 5th International Conference on Theory and Applications of Models of Computation, TAMC'08, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 282–293.
[20] M. Terzer, J. Stelling, Large-scale computation of elementary flux modes with bit pattern trees, Bioinformatics 24 (19) (2008) 2229–2235.
[21] E. Tomita, T. Seki, Discrete Mathematics and Theoretical Computer Science, in: Lecture Notes in Computer Science, vol. 2731, Springer, Berlin, Heidelberg, 2003, pp. 279–289. (Chapter) An Efficient Branch-and-Bound Algorithm for Finding a Maximum Clique.
[22] F. Vanderbeck, L.A. Wolsey, An exact algorithm for IP column generation, Operations Research Letters 19 (4) (1996) 151–159.