Minimum ratio cover of matrix columns by extreme rays of its induced cone

Alexandre da Silva Freire, Vicente Acuña, Pierluigi Crescenzi, Carlos Eduardo Ferreira, Vincent Lacroix, Paulo Vieira Milreu, Eduardo Moreno, Marie-France Sagot

To cite this version:

Alexandre da Silva Freire, Vicente Acuña, Pierluigi Crescenzi, Carlos Eduardo Ferreira, Vincent Lacroix, et al.. Minimum ratio cover of matrix columns by extreme rays of its induced cone. Proceedings of the Second international Symposium on Combinatorial Optimization (ISCO), Apr 2012, Athens, Greece. pp.165–177, 10.1007/978-3-642-32147-4_16 . hal-00763453

HAL Id: hal-00763453

https://hal.inria.fr/hal-00763453

Submitted on 10 Dec 2012

Minimum ratio cover of matrix columns by extreme rays of its induced cone

A.S. Freire[1,2], V. Acuña[2], P. Crescenzi[3], C.E. Ferreira[1], V. Lacroix[2],
P.V. Milreu[2], E. Moreno[4] and M.-F. Sagot[2]

1 - Instituto de Matemática e Estatística - Universidade de São Paulo, Brazil
2 - INRIA and Université de Lyon, F-69000, Lyon ; Université Lyon 1 ; CNRS, UMR5558, France
3 - Università degli Studi di Firenze, Italy
4 - Faculty of Science and Technology - Universidad Adolfo Ibàñez, Chile

Abstract. Given a matrix $S \in \mathbb{R}^{m\times n}$ and a subset of columns R, we study the problem of finding a cover of R with extreme rays of the cone $\mathcal{F} = \{v \in \mathbb{R}^n \mid Sv = \mathbf{0}, v \geq \mathbf{0}\}$, where an extreme ray v covers a column k if $v_k > 0$. In order to measure how proportional a cover is, we introduce two different minimization problems, namely the MINIMUM GLOBAL RATIO COVER (MGRC) and the MINIMUM LOCAL RATIO COVER (MLRC) problems. In both cases, we apply the notion of the *ratio* of a vector v, which is given by $\frac{\max_i v_i}{\min_{j \mid v_j > 0} v_j}$. These problems are originally motivated by a biological question on metabolic networks. We show that these two problems are NP-hard, even in the case in which $|R| = 1$. We introduce a mixed integer programming formulation for the MGRC problem, which is solvable in polynomial time if all columns should be covered, and introduce a branch-and-cut algorithm for the MLRC problem. Finally, we present computational experiments on data obtained from real metabolic networks.

Keywords: Extreme rays, elementary modes, matrix covering.

1 Introduction

Given a matrix $S \in \mathbb{R}^{m\times n}$, we say that an extreme ray of the cone $\mathcal{F} = \{v \in \mathbb{R}^n \mid Sv = \mathbf{0}, v \geq \mathbf{0}\}$ covers a column k if $v_k > 0$. In this paper, we study the problem of finding a set of extreme rays of $\mathcal{F}$ that cover a subset R of the columns of S.

This problem arises naturally in bioinformatics in the context of metabolic networks, particularly in the study of its *elementary modes* (EMs). Biologically, an EM is a minimal sub-network that enables the metabolic system to operate at steady state, that is, all internal metabolites (chemical compounds) are produced and consumed in equal quantities. Mathematically, an EM corresponds to an extreme ray of the convex cone defined by the "stoichiometric matrix" of the metabolic reactions in the system. In this matrix, each column corresponds to a reaction and each row to a metabolite. Each entry of the matrix indicates the minimum number of molecules of this metabolite that is produced (the entry is then positive) or consumed (the entry is negative) by the reaction. The number of EMs may be extremely large (several millions) even for small networks (hundreds of reactions) [11, 7], therefore, the study of the complexity and of algorithms to enumerate all EMs of a network has been deeply explored [1, 3, 11].

Deciding if the stoichiometric matrix can be covered by elementary modes is a problem that has also been well studied [5], but not the associated optimization problem (among all sets of EMs covering the matrix, which one is the "best"). The optimization criterion we introduce now is based on the key idea that not all EMs are equally interesting from the biological standpoint. In particular, non-proportional EMs, i.e. EMs which use each reaction with extremely different fluxes seem to be less relevant (or at least much harder to exploit for biologists) than proportional EMs.

To account for this, we define the *ratio* of a vector v as the fraction between the maximum and the minimum positive component of v. For a set of extreme rays that cover R, we introduce two different functions in order to measure how proportional a cover is. Namely, we define a *local ratio*, which measures the ratio of each extreme ray of the cover, and a *global ratio*, which measures the ratio of the vector obtained by the combination of all extreme rays of the cover.

We note that these concepts also appear naturally in the context of exact linear programming. In fact, current algorithms for scaling a matrix have a complexity that depends on the ratio of its elements [9, 8]. Hence, obtaining a method to find extreme rays of a cone with minimal ratio is also an interesting problem for the exact optimization community.

In Section 2, we present the definitions and notation used throughout this paper, as well as the formal definition of the MINIMUM GLOBAL RATIO COVER (MGRC) problem and of the MINIMUM LOCAL RATIO COVER (MLRC) problem. In Section 3, we show that the MGRC and MLRC problems are both NP-hard, even in the case that $|R| = 1$. In Section 4, we introduce a mixed integer programming formulation for the MGRC problem, which is solvable in polynomial time if all columns should be covered, and introduce a branch-and-cut algorithm for the MLRC problem. Finally, we present in Section 5 computational experiments on data obtained from real metabolic networks.

2 Notation and definitions

Given a matrix $S \in \mathbb{R}^{m \times n}$, we define the cone $\mathcal{F} = \{v \in \mathbb{R}^n \mid Sv = \mathbf{0},\ v \geq \mathbf{0}\}$. The *support* of a vector $v \in \mathcal{F}$, denoted by $sup(v)$, is the set of indexes of all nonzero entries of v. A nonzero vector $v \in \mathcal{F}$ is an *extreme ray* (ER) of $\mathcal{F}$ if its support is minimal, in the sense that there is no other nonzero vector $v' \in \mathcal{F}$ such that $sup(v') \subset sup(v)$. Let $S' \in \mathbb{R}^{m \times n'}$ be the matrix S without some columns. Note that an ER of $\mathcal{F}' = \{v \in \mathbb{R}^{n'} \mid S'v = \mathbf{0}, v \geq \mathbf{0}\}$ is also an ER of $\mathcal{F}$. Two vectors u and v of $\mathcal{F}$ are *equivalent* if $u = \gamma v$, for some real number $\gamma > 0$. Given two ERs u and v of $\mathcal{F}$, we have that $sup(u) = sup(v)$ if and only if u and v are equivalent. Since $\mathcal{F}$ is a cone, each vector of $\mathcal{F}$ can be obtained by a conical combination of ERs of $\mathcal{F}$. Moreover, given two vectors u and v in $\mathcal{F}$ such that $u \leq v$, we have that the vector $v' = v - u$ is in $\mathcal{F}$.

For simplicity, given an index $i \in \{1, 2, \ldots, n\}$, we use the term "column i" instead of "column indexed by i". A column i of S is *covered* by a vector $v \in \mathcal{F}$ if $v_i > 0$. Given a set $R \subseteq \{1, 2, \ldots, n\}$, we say that a set C of ERs of $\mathcal{F}$ *covers* R if each column in R is covered by at least one ER in C. Equivalently, we say that C is a *cover* of R. In [1] V. Acuña *et al.* introduced a polynomial time algorithm for finding an ER of $\mathcal{F}$ which covers a column k. Using this algorithm as a subroutine, we can design a polynomial time algorithm for solving the problem of finding a cover of a set $R \subseteq \{1, 2, \ldots, n\}$.

The *ratio* of a nonzero vector $v \in \mathcal{F}$ is given by $r(v) = \frac{\max_i v_i}{\min_{j|v_j>0} v_j}$. The *global ratio* of a cover C is given by $\psi(C) = r(\sum_{v \in C} v)$, and the *local ratio* of C is given by $\phi(C) = \max_{v \in C} r(v)$. We investigate the following two problems.

Problem 1. [MGRC] Given a matrix $S \in \mathbb{R}^{m \times n}$ and a set $R \subseteq \{1, 2, \ldots, n\}$, the MINIMUM GLOBAL RATIO COVER problem consists in finding a cover of R with minimum global ratio.

Problem 2. [MLRC] Given a matrix $S \in \mathbb{R}^{m \times n}$ and a set $R \subseteq \{1, 2, \ldots, n\}$, the MINIMUM LOCAL RATIO COVER problem consists in finding a cover of R with minimum local ratio.

We say that a vector $v \in \mathcal{F}$ is *normalized* if $\max_i v_i = 1$. Note that $r(v) = r(\gamma v)$, for any $\gamma > 0$. Thus, for any vector $v \in \mathcal{F}$, there exists an equivalent normalized vector $\gamma v \in \mathcal{F}$ such that $r(v) = r(\gamma v)$, where $\gamma = \frac{1}{\max_i v_i}$. Considering only normalized vectors, we have that a vector $v \in \mathcal{F}$ with minimum ratio maximizes $x = \min_{i|v_i>0} v_i$. Since we are seeking for vectors with minimum ratio, this concept of normalization helps us to design our formulations. Note that scaling the ERs of a cover does not affect its local ratio, but it can affect its global ratio, unless we use the same scalar for all ERs in the cover. Thus, for the MLRC problem, we can work only with normalized ERs. For the MGRC problem, we normalize the vector obtained by the combination of the ERs in the cover, but the ERs themselves can be non-normalized.

We assume that S is *consistent*, in the sense that Problems 1 and 2 have feasible solutions. The problem of recognizing whether S is consistent or not can be solved in polynomial time [1].

3 Complexity of the MGRC and MGLC problems

Given a column k, we denote by MGRC^k and MLRC^k, respectively, the special case of the MGRC and MLRC problems in which $R = \{k\}$. We show that the MGRC^k and the MLRC^k problems are NP-hard by reducing the THREE DIMENSIONAL MATCHING problem to them. The 3DM problem is NP-complete [4] and can be stated as follows:

Problem 3. [3DM] Given a set of triples $E \subseteq W \times X \times Y$, where $|W| = |X| = |Y| = t$ and the sets W, X and Y are disjoint, determine if there exists a subset $E^* \subseteq E$ of non-overlapping triples such that $|E^*| = t$ (E^* is called a *perfect matching*).

Theorem 1. *The* MGRC^k *and the* MLRC^k *problems are NP-hard.*

Proof. Given an instance $\mathcal{I}_{3\text{DM}} = (E, W, X, Y)$ of the 3DM problem, where $|W| = |X| = |Y| = t$, we construct S and R in the following way. For simplicity, we assume that $E = \{e_1, e_2, \ldots, e_{|E|}\}$ and $W \cup X \cup Y = \{1, 2, \ldots, 3t\}$. Let $S \in \mathbb{R}^{m \times n}$ be a matrix, where $m = 3t$ and $n = |E| + 1$, such that, for each triple $e_i = (w, x, y)$ in E we have that $S_{wi} = S_{xi} = S_{yi} = 1$, for each $j \in \{1, 2, \ldots, m\} \setminus \{w, x, y\}$ we have that $S_{ji} = 0$, and for $j = 1, 2, \ldots, m$ we have that $S_{jn} = -1$. Let $R = \{n\}$ be the set of columns that we want to cover. Figure 1 illustrates an example of this construction. We now show that (1) $\mathcal{I}_{3\text{DM}}$ contains a perfect matching if and only if there exists a cover C of R with $\phi(C) \leq 1$ and (2) $\mathcal{I}_{3\text{DM}}$ contains a perfect matching if and only if there exists a cover C of R with $\psi(C) \leq 1$.

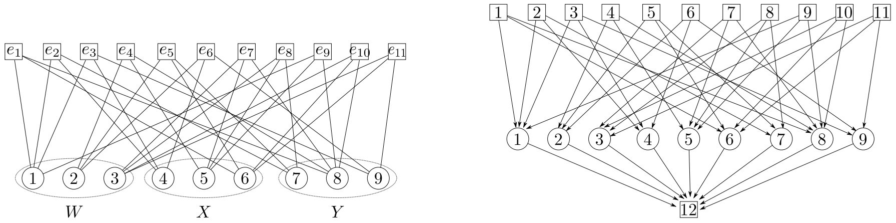

Fig. 1. An instance of the 3DM problem and the corresponding (S, R) are drawn at the left and right side, respectively. For each triple $e_i = (w, x, y)$ in E, there are edges linking the square e_i to circles w, x and y. For each $S_{ij} = 1$ there is arc (i, j) going from a square i to a circle j and each $S_{uv} = -1$ there is an arc (u, v) going from a circle u to a square v.

Assume that $\mathcal{I}_{3\text{DM}}$ contains a perfect matching E^*. Let $v \in \mathbb{R}^n$ be a vector such that $v_n = 1$ and, for $i = 1, 2, \ldots, n - 1$, if $e_i \in E^*$ then $v_i = 1$, otherwise $v_i = 0$. Since in this construction $Sv = \mathbf{0}$ is satisfied and $v \geq \mathbf{0}$, we have that $v \in \mathcal{F}$. Since column n is the only one with negative coefficients, we have that all nonzero vectors in $\mathcal{F}$ must cover n. Observe that any nonzero vector w such that $\text{sup}(w) \subset sup(v)$ does not satisfy $Sw = \mathbf{0}$. Thus, v is an ER of $\mathcal{F}$ and $C = \{v\}$ is a cover of R such that $\phi(C) = \psi(C) = 1$.

Conversely, assume that there exist a cover C of R such that $\psi(C) \leq 1$. Let $v = \sum_{w \in C} w$ and let $E^* = \{e_i \in E \mid v_i > 0\}$. Since column n is the only one with negative coefficients and $Sv = \mathbf{0}$ is satisfied, we have that $v_n > 0$. Moreover, since column n has coefficient -1 in all rows, we have that each element of $W \cup X \cup Y$ is contained in at least one triple of E^*. Suppose, by contradiction, that E^* contains two triples e_i and e_j that overlap each other at an element

$h \in W \cup X \cup Y$. By definition of the function ψ, we have that $\psi(C) \geq 1$ and thus $\psi(C) = 1$. This implies that all nonzero entries of v have the same value. As a consequence, we have that $S_{h*}v \geq v_i + v_j - v_n > 0$, where S_{h*} is the sub-matrix of S containing only row h, which is a contradiction. Thus E^* is a perfect matching. Assume now that there exist a cover C' of R such that $\phi(C') \leq 1$. Applying the same arguments we can construct the perfect matching E^*. Figure 2 illustrates an example of this construction.

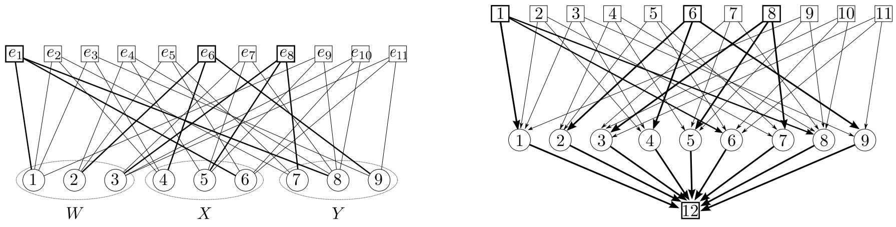

Fig. 2. At the left side a perfect matching in $\mathcal{I}_{\text{3DM}}$ and at the right side a cover $C = \{v\}$ of R such that $\phi(C) = \psi(C) = 1$.

Clearly this reduction is made in polynomial time, considering the size of $\mathcal{I}_{\text{3DM}}$. As we show above, solving the MGRC^k problem or the MLRC^k problem leads to a solution to the 3DM problem, therefore the MGRC^k and MLRC^k problem are NP-hard. □

Corollary 1. *The* MGRC *and* MLRC *problems are NP-hard.*

We are interested in solving the MLRC^k problem because, as we show later on, one can solve Problem 2 by solving MLRC^k for each column k. As we show in the next section, the MGRC problem can be solved in polynomial time if $R = \{1, 2, \dots, n\}$. However, by adapting the proof of Theorem 1 we can prove the following theorem.

Theorem 2. *The* MLRC *problem remains NP-hard even if* $R = \{1, 2, \dots, n\}$.

We achieve this by adding a column $n + i$ to S, for each triple $e_i = (w, x, y)$ in E, such that $S_{j,n+i} = -S_{j,i}$, for $j = 1, 2, \dots, m$. In this way it is possible to cover each pair of columns $\{i, n + i\}$ with an ER of ratio 1, except for column n, which can be covered by an ER of ratio 1 if and only if E contains a perfect matching.

4 MIP approaches for the MGRC and MLRC problems

In this section we present Mixed Integer Programming (MIP) approaches for the MGRC and MLRC problems.

4.1 A MIP formulation for the MGRC problem

In this section, we introduce a MIP formulation for finding a normalized vector $v \in \mathcal{F}$ which covers R and has minimum ratio. Then, by decomposing v into ERs of $\mathcal{F}$, we obtain an optimal cover.

$$
(\mathrm{P_{MGRC}}) \qquad
\begin{array}{rlll}
\max x & & & \\
\text{s.t.} & Sv = \mathbf{0} & & (1)\\
& x \leq v_i - s_i + 1, & \text{for } i = 1, 2, \ldots, n & (2)\\
& 0 \leq v_i \leq s_i, & \text{for } i = 1, 2, \ldots n & (3)\\
& s_k = 1, & \text{for each } k \in R & (4)\\
& s \in \{0, 1\}^n & & (5)
\end{array}
$$

The decision variables $s \in \{0, 1\}^n$ represent the support of v. In [3], K. Fukuda and A. Prodon prove the following theorem.

Theorem 3. *Any vector $v \in \mathcal{F}$ can be expressed as a convex combination of $m - n$ extreme rays of $\mathcal{F}$.*

In [6], R. M. Jungers *et. al.* introduce a polynomial time algorithm for finding a such decomposition with minimum cardinality. After solving $(\mathrm{P_{MGRC}})$, we run this decomposition algorithm on v. Then, we obtain $C = \{w^1, w^2, \ldots, w^t\}$, where $w^1, w^2, \ldots, w^t$ is the decomposition found by the algorithm. Observe that $\psi(C) = r(v)$. Since S is consistent and x is maximized, we have that in an optimal solution, $x > 0$ and v is a normalized vector which covers all columns in R. Moreover, since $x = \min_{i|v_i>0} v_i$ is maximum, we have that v has minimum ratio. We now argue that C has minimum global ratio. Suppose, by contradiction, that $\psi(C)$ is not minimum. Let C' be a cover of R such that $\psi(C') < \psi(C)$. Thus, the vector $h = \sum_{w \in C} w$ covers all columns in R and $r(h) < r(v)$. Thus, there is a normalized vector γh, such that γh is a feasible solution of $(\mathrm{P_{MGRC}})$. Moreover, γh leads to a greater value in the objective function, which is a contradiction. Therefore, this procedure solves the MGRC problem.

Observe that if $R = \{1, 2, \ldots, n\}$, then all variables s are fixed (i.e., we have no decision variables in $(\mathrm{P_{MGRC}})$). In this case, $(\mathrm{P_{MGRC}})$ is a linear program and, thus, can be solved in polynomial time [10]. We denoted this special case of the MGRC problem by MGRC*.

4.2 An algorithm for the MLRC problem

The MLRC problem is considerably harder to formulate as a MIP than the MGRC problem. In MGRC we solve a MIP in which the optimal solution can be a non-ER and, after that, we decompose this vector regardless of the ratio of each ER obtained in the decomposition. We cannot apply the same idea for the MLRC, because in this case we have to consider the ratio of each ER inside the cover individually. In this section, we introduce an algorithm for solving the MLRC problem, which works as follows. At each iteration, it chooses an uncovered column $k \in R$ and solves the MLRC^k problem, i.e. finds an ER v of $\mathcal{F}$ which covers k and has minimum ratio. Since the MLRC^k problem is NP-hard, we introduce a MIP formulation in which an optimal solution must be an ER. This requirement is not easy to be described with simple linear inequalities and, hence, we propose a branch-and-cut algorithm to solve the problem.

Let $\mathcal{H}(k)$ be the set of all ER's of $\mathcal{F}$ which cover column k and let $\overline{\mathcal{H}}(k)$ be the set of all ER's of $\mathcal{F}$ which do not cover column k. Below we introduce a formulation for solving the MLRC^k problem.

$$
(\text{P}_{\text{MLRC}^k}) \quad
\begin{array}{lrll}
\max x & & & \\
\text{s.t.} & Sv = \mathbf{0} & & (6)\\
& x \le v_i - s_i + 1, & \text{for } i = 1,2,\dots,n & (7)\\
& 0 \le v_i \le s_i, & \text{for } i = 1,2,\dots n & (8)\\
& s_k = 1 & & (9)\\
& \sum_{i \in sup(h)} s_i + s_j \le |sup(h)|, & \forall h \in \mathcal{H}(k), \forall j \notin sup(h) & (10)\\
& \sum_{i \in sup(h)} s_i \le |sup(h)| - 1, & \text{for each } h \in \overline{\mathcal{H}}(k) & (11)\\
& s \in \{0,1\}^n & & (12)
\end{array}
$$

Formulation (P_{MLRC^k}) without constraints (10) and (11) is the same as (P_{MGRC}) for $R = \{k\}$. In this case, an optimal solution to (P_{MGRC}) is a vector $v \in \mathcal{F}$ which covers k and has minimum ratio. Since v can be a non-ER, we include constraints (10) and (11) in order to guarantee that v is an ER. Constraint (10) eliminates all vectors which strictly contain the support of an ER which covers k. We could apply inequalities (10) also for the ERs which do not cover k, but in this case we introduce (11), which is more tight, in the sense that it eliminates all vectors whose support contains (not necessarily strictly) the support of an ER which does not cover k. On the other hand, if we apply constraints (11) to the ERs which cover k we have no feasible solutions.

Since $|\mathcal{H}(k)|$ and $|\overline{\mathcal{H}}(k)|$ can be huge, instead of including constraints (10) and (11) a priori, we solve (P_{MLRC^k}) by applying the so-called *branch-and-cut* method [10]. To this purpose, we have to solve the *separation problem* for inequalities (10) and (11). Let (L_{MLRC}) be the linear relaxation of (P_{MLRC^k}), where constraints (12) are replaced by $0 \le s_i \le 1$, for $i = 1,2,\dots,n$. The separation problem for inequalities (10) and (11) consist in, given a feasible solution of (L_{MLRC}), prove that (10) and (11) are satisfied, or find an ER which does not satisfy (10) or (11). One can solve these separation problems by solving their optimization versions, which are stated as follows.

Problem 4. Given a feasible solution of (L_{MLRC}), find an ER h in $\mathcal{H}(k)$ such that $\sum_{i \in sup(h)}(1 - s_i) - s_j$, for some $j \notin sup(h)$, is minimum.

Problem 5. Given a feasible solution of (L_{MLRC}), find an ER h in $\overline{\mathcal{H}}(k)$ such that $\sum_{i \in sup(h)}(1 - s_i)$ is minimum.

We denote Problems 4 and 5 by $\text{SEP}^{(10)}$ and $\text{SEP}^{(11)}$, respectively. In [1] the authors prove the following theorem.

Theorem 4. *Given an integer t, deciding the existence of an ER v of $\mathcal{F}$ such that $|sup(v)| \leq t$ is NP-complete.*

Let S' be the matrix S without column k. For simplicity, we assume that $k = n$. By finding an ER v^* of $\mathcal{F}' = \{v \in \mathbb{R}^{n-1} \mid S'v = \mathbf{0}$ and $v \geq \mathbf{0}\}$ which minimizes $\sum_{i \in sup(v^*)}(1 - s'_i)$, where $s' = \mathbf{0}$, we can decide the existence of an ER h of $\mathcal{F}'$ such that $|sup(h)| \leq t$. Thus, the $\textsc{sep}^{(11)}$ problem is NP-hard. If we fix a column $j \neq k$, then the $\textsc{sep}^{(10)}$ problem becomes equivalent to finding an ER v^* of $\mathcal{F}'' = \{v \in \mathbb{R}^n \mid Sv = \mathbf{0}, v \geq \mathbf{0}$ and $v_j = 0\}$ which minimizes $\sum_{i \in sup(v^*)}(1 - s'_i)$, where $s' = \mathbf{0}$, and such that $k \in sup(v^*)$. If we solve this problem for each $k \neq j$, we can decide the existence of an ER h of $\mathcal{F}''$, such that $|sup(h)| \leq t$. Thus, we have that the $\textsc{sep}^{(10)}$ problem is NP-hard. Therefore, from Theorem 4 follows Corollary 2.

Corollary 2. *Problems* $\textsc{sep}^{(10)}$ *and* $\textsc{sep}^{(11)}$ *are NP-hard.*

Despite the drawback of Corollary 2, we introduce a MIP formulation for solving the separation of inequalities (10) and (11) at the same time, which works quite well in practice, as we show in Section 5.

$$(\textsc{P}_{\textsc{sep}}) \qquad \begin{aligned} \min\ & z = (2 - s_k)b_k + \textstyle\sum_{i \neq k}(1 - s_i)b_i - \sum_{j \neq k} s_j w_j \\ \text{s.t.}\ & Sh = \mathbf{0} && (13) \\ & 0 \leq h_i \leq b_i, \quad \text{for } i = 1, 2, \ldots, n && (14) \\ & \textstyle\sum_{i=1}^{n} h_i \geq 1 && (15) \\ & \textstyle\sum_{j \neq k} w_j \leq b_k && (16) \\ & w_j + b_j \leq 1, \quad \text{for each } j \neq k && (17) \\ & w_j \in \{0, 1\}, \quad \text{for each } j \neq k && (18) \\ & b \in \{0, 1\}^n && (19) \end{aligned}$$

By constraints (13), (14) and (15), we have that h is a nonzero vector in $\mathcal{F}$. The decision variables w are introduced in order to choose a column j which is not in the support of h. By constraints (16) and (17), if $k \in sup(h)$, we have that at most one column outside $sup(h)$ is chosen, otherwise no columns outside $sup(h)$ is chosen. As we explain later on, we are interested only in solutions such that $z - 1 < 0$. Thus, we can apply formulation ($\textsc{P}_{\textsc{sep}}$) to the sub-matrix of S which contains only columns in $\{i \mid s_i > 0\}$. From a practical point of view, this is a very important property, because this sub-matrix can be much smaller than the original one.

In an optimal solution, if $b_k = 1$, since z is minimum and the variables w have negative coefficients in the objective function, we have that $\sum_{j \neq k} w_j = 1$ and thus $z = \sum_{i \in sup(h)}(1 - s_i) - s_j$, for some $j \notin sup(h)$. Let h' be any ER such that $sup(h') \subseteq sup(h)$ and $k \in sup(h')$. Since z is minimum and $(1 - s_i)$ is non-negative, for $i = 1, 2, \ldots, n$, we have that h' is an ER in $\mathcal{H}(k)$ such that $\sum_{i \in sup(h')}(1 - s_i) - s_j$ is minimum, where $j \notin sup(h')$. In [1], the authors introduce a procedure, namely $\textsc{FindER}(S, k)$, which receives a matrix $S \in \mathbb{R}^{m \times n}$ and a column $k \in \{1, 2, \ldots, n\}$, and returns an ER of $\mathcal{F}$ which covers column k. We use this algorithm in order to find h'. Therefore, in this case we solve the $\textsc{sep}^{(10)}$ problem . If $b_k = 0$, we have that $\sum_{j \neq k} s_j w_j = 0$ and $z = (2 - s_k) + \sum_{i \in sup(h) \setminus \{k\}} (1 - s_i) = 1 + \sum_{i \in sup(h)} (1 - s_i)$. In the same way as we mention above, we obtain an ER h' in $\overline{\mathcal{H}}(k)$, such that $sup(h') \subseteq sup(h)$ and $\sum_{i \in sup(h')} (1 - s_i)$ is minimum. Thus, in this case we solve the $\textsc{sep}^{(11)}$ problem. In both cases, we have that if $z - 1 < 0$, then s violates either (10) or (11), and thus we obtain a cutting-plane from $sup(h')$ to be included explicitly in ($\textsc{l}_{\textsc{mlrc}}$). Otherwise, constraints (10) and (11) are satisfied and thus (s, v, x) is a feasible solution of ($\textsc{l}_{\textsc{mlrc}}$). From a practical point of view, since h' is an ER, we can use $r(h')$ as an upper bound on the optimal solution of the MLRC^k problem for each $i \in sup(h')$. This trick can speed up quite a lot the whole process, because as we show in the algorithm below, we have to solve the MLRC^k problem for $i = 1, 2, \ldots, n$.

We now introduce an algorithm for the MLRC problem. Correctness of Algorithm 1 follows from the fact that $r(v)$ is a lower bound for the local ratio of an optimal cover of any set R' such that $k \in R'$, where v is ER obtained in line 4.

Algorithm 1: Solves the MLRC problem.

```
Input: A matrix $S \in \mathbb{R}^{m \times n}$ and a set $R \subseteq \{1, 2, \ldots, n\}$.
Output: A cover of $R$ with minimum local ratio.
1: $C \leftarrow \emptyset$
2: while $C$ does not cover $R$ do
3:     Choose an uncovered column $k \in R$
4:     Let $v$ be an ER obtained by solving ($\textsc{p}_{\textsc{mlrc}^k}$)
5:     $C \leftarrow C \cup \{v\}$
6: return $C$
```

In our implementation, we use some tricks to speed up Algorithm 1. We reuse the cuts found in previous iterations as soon as they violate inequalities (10) or (11). We use FindER as a heuristic for finding upper bounds on the minimum ratio needed to cover each column. Moreover, since each cut corresponds to an ER, all ERs found during the branch-and-cut procedure are used in order to update these upper bounds. In this way, we profit from the computational effort spent for finding violated inequalities, in order to generate feasible solutions of the problem. Since, at each iteration, we can choose arbitrarily the next column to be covered, we take one with highest upper bound and, if this upper bound is lesser than or equal to the maximum ratio among the ERs already included in C, we can stop the algorithm. With this strategy, we aim to close the gap between the lower and upper bounds on the minimum local ratio of the cover as soon as possible.

In the separation step, before solving ($\textsc{p}_{\textsc{sep}}$), we try to separate inequalities (10) and (11) by applying a heuristic which considers only the integral entries of support s. In this case, the separation problems can be solved using a simple modification of the FindER algorithm. If we do not succeed, then we solve ($\textsc{p}_{\textsc{sep}}$) and collect all ERs which violate (10) or (11) found during this process, not only the optimal ones (most solvers provide callback routines to this purpose).

5 Computational experiments

We obtained our data set by downloading metabolic networks from MetExplore [2]. We restricted the networks to the small-molecule metabolism, meaning that reactions involving macromolecules such as nucleic acids or proteins were removed. We set the filters to exclude pairs of co-factors and common compounds, which otherwise would connect unrelated reactions. Pairs of co-factors include NAD / NADH, NADP / NADPH, ADP / ATP (for the full list, see the MetExplore documentation). Common compounds include water, proton, CO2, phosphate, diphosphate, NH3, H2O2 and O2. In order to make the matrices consistent, we removed all columns which are not covered by any ER. We used *CPLEX*$^{\copyright}$ *12.2* as the MIP solver and the machine configurations are the following: 1 single processor 3.8GHz and 4GB of RAM. In our tests, we did not consider the time for reading the input files.

In Table 1, we show the results of the computational experiments made with our MIP approaches for the MGRC* and the MGRC problems. In the case of the MGRC problem, we made tests with several different sizes of R. We solved more than 100 different instances and report here the results that we consider more relevant. We observed that the MGRC problem becomes harder when $|R|$ is very small, thus in the results presented here we chose small subsets of columns to be covered. Each row of Table 1 shows one execution of the MGRC* problem and the arithmetic mean of 100 executions of the MGRC problem, where in each execution a set R was randomly chosen, such that $1 \leq |R| \leq 5$. The acronyms "NzCf" and "BBN" stand for, respectively, "nonzero entries in the input matrix" and "branch-and-bound nodes explored during the execution".

Table 1: Computational experiments with the MGRC* and the MGRC problems.

				MGRC*		MGRC		
Instance	n	m	# NzCf	$\psi(C)$	Time	$\psi(C)$	# BBN	Time
PSEAB608	10718	925	27777	23.4	1s	1.0	378	18s
RHICF157	11269	922	28538	27.7	1s	1.0	306	16s
MOUSE	12479	2215	32250	38.8	1s	1.1	478	26s
CHLAMY	11144	2149	29011	40.1	1s	1.1	765	27s
ARA	14009	2251	35984	35.7	1s	1.0	323	22s

As Table 1 shows, our approach for the MGRC problem is very effective. We selected the hardest instances among all that we tested, and nevertheless each instance was solved in a few seconds. In the case of the MGRC* problem, the running time was quite short.

In Table 2, we present the results of the computational experiments made with our branch-and-cut algorithm for the MLRC problem. In our tests, we chose $R = \{1, 2, \ldots, n\}$. The acronyms "MS', '"NO" and "Cuts" stand for, respectively, "MIPs solved during the procedure", "columns covered by ERs that can be non-optimal" and "cuts generated during the branch-and-cut procedure".

Table 2: Computational experiments with the MLRC problem, where $R = \{1, 2, \ldots, n\}$.

Instance	n	m	#NzCf	#MS	#Cuts	#BBN	$\phi(C)$	#NO	Time
ONYPE335	838	174	1839	95691	13	1581	8.0	0	27m08s
YERYP364	955	409	2393	57	33584	128	8.0	0	7m26s
SHIFL233	1046	489	2636	10	49647	171	8.0	0	16m14s
BUCAP86	886	282	2079	18	89772	875	8.0	0	26m31s
DESPS65	629	297	1533	13	62580	900	11.0	0	15m32s
BUCBP85	1137	290	2691	46	94608	512	[8 .. 9]	1	27m25s
HELPY117	684	264	1710	15	168630	3097	[8 .. 10]	5	47m19s
PVIVAX	1618	346	3878	130	263111	1906	[2 .. 9]	72	1h25m51s
PLASMO	2563	411	6387	311	193590	1253	[2 .. 12]	133	2h13m51s

In our implementation, we used three parameters, namely TOTAL_TL=1h, IT1_TL=5m and IT2_TL=15s. We observed that different choices of these parameters can change dramatically the running time of the algorithm, as well as the number of uncovered columns and the quality of the gap, in the case that the solution found is not optimal. Moreover, some choices are good for some instances, but worse for others. At each step in which we solve the MLRC^k problem, we set the time limit of either IT1_TL or IT2_TL, where IT2_TL is used only if the total time spent is greater than TOTAL_TL. In the case that no optimal solution is found, we leave the corresponding column temporarily uncovered and use the lower bound obtained by the unterminated MIP as a lower bound on the minimum local ratio of the cover. If after the last iteration there are uncovered columns, we cover these columns with the best ERs that cover them and report, in column NO, how many columns were covered in this way.

As Table 2 shows, our branch-and-cut algorithm was able to solve instances of reasonable size. In most of the cases, the number of columns covered by possibly non-optimal ERs was small if compared to the total number of columns. In some cases, the gap between the solution found and an optimal one is tight. Depending on the structure of S, some columns are much harder than others to be covered. In general, in the biological applications that we are interested in, R is a small subset of columns. Thus, depending on the choice of R, our algorithm can solve instances with about 2500 columns. However, if R intersects the "hardest" columns, the algorithm may take a long time and still not solve the problem. Since the application that initially motivated this problem comes from biology, where the input can be very large, one future work is to improve the method to enable solving larger instances. In the next section, we suggest some directions that can be explored in order to improve our method.

6 Conclusion and future work

We showed that the MGRC and MLRC problems are NP-hard even when $|R| = 1$. We then presented a mixed integer programming formulation for the MGRC problem, which is solvable in polynomial time if all columns should be covered, and a branch-and-cut algorithm for the MLRC problem. We experimentally showed that our approach for the MGRC problem is very effective for solving large scale instances of the problem. In the case of the MLRC problem, we were able to solve instances of reasonable size.

As future work, we suggest some directions that can be explored in order to improve our method for the MLRC problem. Our method has the following three key points which can be explored in order to achieve a better performance: (1) the order in which we try to cover the columns; (2) the configuration of the time outs; (3) the strength of the formulation for the MLRC^k problem. With respect to (1), approximation algorithms for the MLRC^k problem may lead to a better estimative of the upper bounds of the minimum local ratio needed to cover each column. This could help to find a better order to iterate over the columns. With respect to (2), since a given choice of the time outs can be good for some instances and worse for others, probably we can obtain better results by making this choice dynamically. Finally, with respect to (3), it is still not clear in which cases we can strengthen inequalities (10). Moreover, new classes of valid inequalities could also improve the performance of the method.

References

1. V. Acuna, F. Chierichetti, V. Lacroix, A. Marchetti-Spaccamela, M.-F. Sagot, and L. Stougie. Modes and cuts in metabolic networks: Complexity and algorithms. *Biosystems*, 95(1):51 – 60, 2009.
2. L. Cottret, D. Wildridge, F. Vinson, M. P. Barrett, H. Charles, M.-F. Sagot, and F. Jourdan. MetExplore: a web server to link metabolomic experiments and genome-scale metabolic networks. *Nucleic Acids Research*, 38:W132-7(suppl 2):W132–W137, July 2010.
3. K. Fukuda and A. Prodon. Double description method revisited. In *Combinatorics and Computer Science*, volume 1120 of *Lecture Notes in Computer Science*, pages 91–111. Springer Berlin / Heidelberg, 1996.
4. M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, 1979.
5. M. Heiner and I. Koch. Petri net based model validation in systems biology. In *ICATPN*, pages 216–237, 2004.
6. R. M. Jungers, F. Zamorano, V. D. Blondel, A. V. Wouwer, and G. Bastin. Fast computation of minimal elementary decompositions of metabolic flux vectors. *Automatica - Special Issue on Systems Biology*, 47(6):1255 – 1259, 2011.
7. S. Klamt and J. Stelling. Combinatorial complexity of pathway analysis in metabolic networks. *Molecular Biology Reports*, 29:233–236, 2002.
8. A. Nemirovski and U. Rothblum. On complexity of matrix scaling. *Linear Algebra and its Applications*, 302-303:435–460, 1999.
9. G. Rote and M. Zachariasen. Matrix scaling by network flow. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '07, pages 848–854, 2007.
10. A. Schrijver. *Theory of Linear and Integer Programming.* Wiley-Interscience Series in Discrete Mathematics and Optimization.
11. M. Terzer and J. Stelling. Large-scale computation of elementary flux modes with bit pattern trees. *Bioinformatics*, 24(19):2229–2235, 2008.