On the Separation of Topology-Free Rank Inequalities for the Max Stable Set Problem

Stefano Coniglio[1] and Stefano Gualandi[2]

1 **Department of Mathematical Sciences, University of Southampton, Southampton, UK**
s.coniglio@soton.ac.uk

2 **Department of Mathematics, University of Pavia, Pavia, Italy**
stefano.gualandi@unipv.it

Abstract

In the context of finding the largest stable set of a graph, rank inequalities prescribe that a stable set can contain, from any induced subgraph of the original graph, at most as many vertices as the stability number of the former. Although these inequalities subsume many of the valid inequalities known for the problem, their exact separation has only been investigated in few special cases obtained by restricting the induced subgraph to a specific topology.

In this work, we propose a different approach in which, rather than imposing topological restrictions on the induced subgraph, we assume the right-hand side of the inequality to be fixed to a given (but arbitrary) constant. We then study the arising separation problem, which corresponds to the problem of finding a maximum weight subgraph with a bounded stability number. After proving its hardness and giving some insights on its polyhedral structure, we propose an exact branch-and-cut method for its solution. Computational results show that the separation of topology-free rank inequalities with a fixed right-hand side yields a substantial improvement over the bound provided by the fractional clique polytope (which is obtained with rank inequalities where the induced subgraph is restricted to a clique), often better than that obtained with Lovász's Theta function via semidefinite programming.

1998 ACM Subject Classification G.1.6 [Optimization] Integer Programming

Keywords and phrases Maximum stable set problem, rank inequalities, cutting planes, integer programming, combinatorial optimization

Digital Object Identifier 10.4230/LIPIcs.SEA.2017.29

1 Introduction

Let $G = (V, E)$ be an undirected graph with vertex set V and edge set E, and let $n := |V|$. Given G as input, the *Maximum Stable Set* (MSS) problem calls for the computation of the size of the largest stable set of G (a subset of V with no pair of vertices sharing an edge). Letting $STAB(G)$ be the set of all characteristic vectors of stable sets in G, i.e., of binary vectors $x \in \{0, 1\}^n$ where, given a stable set $S \subseteq V$ and for all nodes $i \in V$, $x_i = 1$ if and only if $i \in S$, solving MSS boils down to computing $\alpha(G) := \max \left\{\sum_{i \in V} x_i : x \in STAB(G)\right\}$, where $\alpha(G)$ is the so-called *stability number* of the graph.

MSS is one of Karp's 21 $\mathcal{NP}$-hard problems [15] and it cannot be approximated in polynomial time to within $O(n^{1-\epsilon})$ for any $\epsilon > 0$ unless $\mathcal{P} = \mathcal{NP}$ [14]. To date, it is, arguably, among the most challenging "fundamental" problems in combinatorial optimization to tackle with integer programming techniques.

Introduced by Chvàtal in [6], *Rank Inequalities* (RIs) prescribe that, for any subgraph $G[U]$ induced by $U \subseteq V$, at most $\alpha(G[U])$ of its vertices can be part of a stable set of G:

16th International Symposium on Experimental Algorithms (SEA 2017).
Editors: Costas S. Iliopoulos, Solon P. Pissis, Simon J. Puglisi, and Rajeev Raman; Article No. 29; pp. 29:1–29:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

▶ **Definition 1.** The set of all RIs is: $\sum_{i \in U} x_i \leq \alpha(G[U])$, for all $U \subseteq V$.

From a combinatorial perspective, RIs are all the inequalities with binary left-hand side (LHS) coefficients which are valid for $STAB(G)$.[1] These inequalities are very general, as many families of valid inequalities known for $STAB(G)$ are obtained as a special case of RIs when restricting the induced subgraph $G[U]$ to specific topologies (such as cliques, holes, wheels, webs, and antiwebs).

In this work, we propose a novel approach for the separation of RIs where, rather than imposing topological restrictions on the induced subgraph $G[U]$, we assume the right-hand side (RHS) of the inequalities to be fixed to a given (but arbitrary) constant.

To our knowledge, the only methodology that has been developed to separate RIs without topological restrictions is the one proposed [19], which relies on the *edge projection* operator introduced in [16]. Although the method in [19] allows for the generation of RIs without a specific topological restriction, it is heuristic in nature and it can halt before all the violated RIs have been found. See [18] for a study on the impact of those and other (heuristically separated) cuts when solving MSS via branch-and-cut. Recent work on integer programming methods for MSS, partially belonging to the same stream of works, can be found in [9, 10].

The paper is organized as follows. In Section 2, after discussing on the nature of RIs and their separation problem, topology-free RIs with a given RHS are introduced. Our method for their separation is described in Section 3, where we also investigate the polyhedral nature of the corresponding separation problem. Section 4 outlines the main algorithmical aspects of our techniques. Computational results are reported and illustrated in Section 5, while Section 6 draws some concluding remarks.

2 Rank inequalities and topology-free rank inequalities with a fixed right-hand side

Let $RSTAB(G) := \left\{x \in \mathbb{R}^n_+ : \sum_{i \in U} x_i \leq \alpha(G[U]), \forall U \subseteq V\right\}$ be the *closure* of RIs. As it is easy to see, optimizing over this set is, as for MSS, both $\mathcal{NP}$-hard and inapproximable in polynomial time to within $O(n^{1-\epsilon})$ for any $\epsilon > 0$. This is because, for $U = V$, the set of RIs contains the inequality $\sum_{i \in V} x_i \leq \alpha(G)$, whose sole introduction into any relaxation of MSS suffices to obtain $\alpha(G)$.

Due to the equivalence between optimization and separation established in [12], it follows that, given a point x^*, the separation problem of RIs, calling for a subset U of vertices such that $\sum_{i \in U} x^*_i > \alpha(G[U])$, or for a proof that no such subset exists, is also $\mathcal{NP}$-hard.

In integer programming, the $\mathcal{NP}$-hardness of a separation problem is, usually, not an issue *per se*.[2] There are many cases of computationally affordable algorithms in which $\mathcal{NP}$-hard separation problems are routinely solved, often by solving an instance of the very optimization problem being tackled, albeit of smaller size. See, for instance, the pioneering work in [8].

In the context of RIs, the situation is even worse. Indeed, not only separating a RI is $\mathcal{NP}$-hard, but even verifying whether a given inequality is a RI is a difficult problem. First, let us define the decision version of MSS (MSS-d) which, given an integer L, asks whether G contains a stable set of size $\geq L$, i.e., whether $\alpha(G) \geq L$. The following holds:

[1] Indeed, $\pi x \leq \pi_0$ is valid for some $P \subseteq \mathbb{R}^n$ if and only if $\pi_0 \geq \max\{\pi x : x \in P\}$. When restricting to π to $\{0,1\}^n$ and $P = STAB(G)$, the definition of RIs follows.

[2] Due to the equivalence between optimization and separation, an $\mathcal{NP}$-hard optimization problem always has at least one family of valid inequalities which is $\mathcal{NP}$-hard to separate.

▶ **Observation 2.** *Given a graph $G = (V, E)$ and a vector $(\pi, \pi_0) \in \mathbb{R}^{n+1}$, it it strongly $\mathcal{NP}$-hard to decide whether $\pi x \leq \pi_0$ is a RI.*

Proof. We can easily establish a Cook-reduction from MSS-d (with input L and G) to the problem of verifying whether $\pi x \leq \pi_0$ is a RI. Indeed, it suffices to call, for all $\pi_0 \in \{L, \dots, n\}$ (thus, $n - L + 1$ times), a routine which solves the problem of membership to the class of RIs with input G and the inequality $\pi x \leq \pi_0$, with $\pi_i = 1$ for all $i \in V$. Since, for the given π, $\pi x \leq \pi_0$ is a RI if and only if $\pi_0 = \alpha(G)$, as soon as the routine returns answer YES for some π_0, we conclude $\pi_0 = \alpha(G)$, thus providing answer YES to MSS-d. If the membership routine returns answer NO for all values of π_0, we conclude that MSS-d has answer NO. ◀

From a cutting plane perspective, especially when cut generation is embedded within a branch-and-cut algorithm, one would arguably look for a small number of inequalities which, jointly, yield the largest bound improvement over the initial relaxation (see [7] for a cutting plane algorithm designed to achieve this via bilevel programming, and [1, 2] for a method which employs cut diversity). With RIs, as we mentioned, the *single* inequality $\sum_{i \in V} x_i \leq \alpha(G)$ always suffices to bring the bound obtained with *any* relaxation of MSS down to $\alpha(G)$. It is thus clear that, if we aim at a practical method relying on the separation of RIs within an efficient algorithm, some restrictions must be introduced.

The restriction that we consider in this paper is not a topological one. Rather, we investigate the problem of separating RIs when their RHS is fixed to an (arbitrary, small) constant $k \in \mathbb{N}$. We refer to such set of RIs as RI$_k$s.

▶ **Definition 3.** The set of all RI$_k$s is: $\sum_{i \in U} x_i \leq k$, for all $U \subseteq V : \alpha(G[U]) = k$.

Note that we can optimize over $RSTAB(G)$ by separating RI$_k$s for all values of $k \in \{1, \dots, n\}$, a feature which cannot be achieved with traditional approaches where topological restrictions are introduced. The assumption on a small k with (in particular) $k \ll \alpha(G)$ is made so as to arrive at a separation problem which is not too hard to solve in practice, as we will better see in the next sections.

From a combinatorial perspective, the following holds:

▶ **Observation 4.** *For any given $k \in \mathbb{N}$, the LHS of a RI$_k$ is the incidence vector of a subgraph $G[U]$ with a K_{k+1}-free complement.*

Proof. By definition, $\sum_{i \in U} x_i \leq k$ is a RI$_k$ if and only if $k = \alpha(G[U])$. If the complement of $G[U]$ is not K_{k+1}-free, then $G[U]$ contains $k + 1$ completely disconnected vertices. Thus $\alpha(G[U]) \geq k + 1 > k$ and $\sum_{i \in U} x_i \leq k$ is not a RI$_k$. ◀

The observation shows that, given any RI$_k$ with vertex set U, $G[U]$ has a K_2-free complement for $k = 1$, a K_3-free complement for $k = 2$, a K_4-free complement for $k = 3$, and so on. See Figure 1 for an illustration.

3 Separation of topology-free rank inequalities

For a given $k \in \mathbb{N}$, let $RSTAB_k(G)$ be the closure of all RI$_k$s, i.e., of all RIs with a RHS equal to k. Throughout the paper, our aim is:

▶ **Aim.** *Given a (reasonably small) upper bound $\bar{k}$ on k, optimize over $\bigcap_{k=1}^{\bar{k}} RSTAB_k(G)$.*

The idea is of investigating the tightness of the bound given by $\bigcap_{k=1}^{\bar{k}} RSTAB_k(G)$ within a pure cutting plane method which, at each iteration, looks for a violated RI$_k$ for each

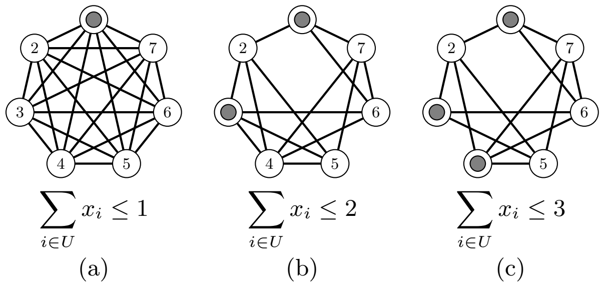

Figure 1 Three induced subgraphs $G[U]$ of the complete graph $G = K_8$ with three RI_ks with $k = 1, 2, 3$, with the corresponding maximum stable set highlighted in gray: (a) an induced subgraph with a K_2-free complement (a clique) and $\alpha(G[U]) = 1$, (b) an induced subgraph with a K_3-free complement and $\alpha(G[U]) = 2$, (c) an induced subgraph with a K_4-free complement and $\alpha(G[U]) = 3$.

$k \in \{1, \ldots, \bar{k}\}$. The overall goal is of assessing whether, even with small values of $\bar{k}$, the bound provided by $\bigcap_{k=1}^{\bar{k}} RSTAB_k(G)$ is stronger than that obtained by computing Lovász's Theta function with semidefinite programming.

3.1 Different separation problems

Given $k \in \mathbb{N}$, the separation problem (in its optimization version) of RI_ks corresponds to the following combinatorial optimization problem:

▶ **Problem 1** (Maximum Weighted Subgraph with Given Stability Number (MWS-GSN)). *Given a graph $G = (V, E)$, a weight vector $x^* \in \mathbb{R}^n$, and an integer k, find a subset of vertices $U \subseteq V$ of maximum weight inducing a subgraph $G[U]$ with stability number equal to k.*

The restriction of RIs to a given RHS does not yield an easier separation problem, at least not from a theoretical perspective:

▶ **Observation 5.** *MWS-GSN is strongly $\mathcal{NP}$-hard.*

Proof. Consider the decision version of MWS-GSN, which asks whether G contains an induced subgraph $G[U]$ of weight $\geq M$ and $\alpha(G[U]) = k$. Letting $M = 0$ and $x^* \in \mathbb{R}^n_+$, the problem has answer YES/NO if and only if MSS-d with input $L = k$ and G has answer YES/NO. ◀

For computational ease but without loss of generality, we introduce an alternative way to optimize over $\bigcap_{k=1}^{\bar{k}} RSTAB_k(G)$, which only requires to solve a relaxation of MWS-GSN. Consider the following inequalities, which we refer to as $\mathrm{RI}_k^{\leq}$s:

▶ **Definition 6.** The set of all $\mathrm{RI}_k^{\leq}$s is: $\sum_{i \in U} x_i \leq k$, for all $U \subseteq V : \alpha(G[U]) \leq k$.

The relationship between RI_ks and $\mathrm{RI}_k^{\leq}$s is as follows:

▶ **Proposition 7.** *For any $k \in \mathbb{N}$, a $\mathrm{RI}_k^{\leq}$ is either a RI_k or it is dominated by a $\mathrm{RI}_{k'}^{\leq}$ for some $k' < k$.*

Proof. Let $\sum_{i \in U} x_i \leq k$ be a $\mathrm{RI}_k^{\leq}$. If $\alpha(G[U]) = k$, it is a RI_k. If $\alpha(G[U]) < k$, the inequality $\sum_{i \in U} x_i \leq k'$ is a $\mathrm{RI}_{k'}$ with $k' = \alpha(G[U]) < k$. It also dominates $\sum_{i \in U} x_i \leq k$: the two inequalities have the same left-hand side, while the second one has a strictly smaller right-hand side. ◀

Proposition 7, when applied recursively, implies that, by iteratively separating $\text{RI}_k^{\leq}$s *in lieu* of RI_ks for increasing values of $k \in \{1, \ldots, \bar{k}\}$, the only inequalities that will be generated are RI_ks, thus showing that the adoption of $\text{RI}_k^{\leq}$ is without loss of generality.

The separation problem (in optimization version) for $\text{RI}_k^{\leq}$s is:

▶ **Problem 2** (Maximum Weighted Subgraph with Bounded Stability Number (MWS-BSN)). *Given a graph $G = (V, E)$, a weight vector $x^* \in \mathbb{R}^n$, and an integer k, find a subset of vertices $U \subseteq V$ of maximum weight inducing a subgraph $G[U]$ with stability number smaller than or equal to k.*

Observe that MWS-BSN is a relaxation of MWS-GSN.

3.2 MWS-BSN: the separation problem of $\text{RI}_k^{\leq}$s

We will now investigate the separation problem for $\text{RI}_k^{\leq}$s: MWS-BSN. Previous work on a closely related problem can be found in [3, 4].

The aim of this section is to show how MWS-BSN can be solved via branch-and-cut. For the purpose, we will introduce a set of inequalities which are necessary to correctly formulate it in the vertex-space. We remark that those inequalities are valid for MWS-BSN only, and *not* for MSS.

Observe that, for any $U \subseteq V$, $\alpha(G[U]) \leq k$ if and only if, for all stable sets S of G with $|S| = k+1$, $|S \cap U| \leq k$. We deduce that, letting $u \in \{0,1\}^n$ be the characteristic vector of U, the following constraints are both necessary and sufficient for u to be a feasible solution to MWS-BSN. We refer to them as *Cover Inequalities* (CIs) (as they play the same role as cover inequalities for the 0-1 knapsack problem):

▶ **Definition 8.** Let $\mathcal{S}^{=k+1}$ be the collection of all stable sets of G of cardinality equal to $k+1$. The set of CIs is: $\sum_{i \in S} u_i \leq k$, for all $S \in \mathcal{S}^{=k+1}$.

Note that, as much as RI_ks for $k > 1$ can be seen as a generalization of clique inequalities, CIs can be regarded as a generalization of edge inequalities which, in the separation problem of clique inequalities (the max clique problem), prevent the presence of stable sets of size 2 in the induced subgraph.

From a polyhedral perspective, the following holds:

▶ **Proposition 9.** *CIs are not facet defining for MWS-BSN.*

Proof. Consider a CI $\sum_{i \in S} u_i \leq k$. If S is not an inclusion-wise maximal stable set, there is a larger stable set S' containing it. It follows that the inequality $\sum_{i \in S'} u_i \leq k$ dominates $\sum_{i \in S} u_i \leq k$, as it is obtained from the latter by lifting each variable u_j with $j \in S' \setminus S$ with a unit coefficient. ◀

Consider now the following constraints, which we call *Lifted Cover Inequalities* (LCIs):

▶ **Definition 10.** Let $\mathcal{S}_M^{\geq k+1}$ be the collection of *maximal* stable sets of G of cardinality $\geq k+1$. The set of LCIs is: $\sum_{i \in S} u_i \leq k$, for all $S \in \mathcal{S}_M^{\geq k+1}$.

LCIs can be shown to be facet defining for MWS-BSN. For the purpose, we first introduce the following lemma:

▶ **Lemma 11.** *Let $S \in \mathcal{S}_M^{\geq k+1}$. LCIs are facet defining for MWS-BSN when restricted to $G[S]$, i.e., to the subspace where $u_i = 0$ for all $i \in V \setminus S$.*

Proof. Since $G[S]$ is a stable set, any subset $S' \subseteq S$ of at most k vertices yields a feasible solution to MWS-BSN. The convex hull of such solutions is thus given by three groups of constraints: $\sum_{i \in S} u_i \leq k$; $u_i \geq 0$ for all $i \in S$; and $u_i \leq 1$ for all $i \in S$. Together, they form a totally unimodular system. Since, by definition of LCIs, $|S| \geq k+1$, the inequality $\sum_{i \in S} u_i \leq k$ is not implied nor dominated by any of the constraints in the other two groups and, thus, it is facet defining. ◀

The following can now be established:

▶ **Theorem 12.** *LCIs are facet defining for MWS-BSN.*

Proof. Let $j_1, \ldots, j_{|V \setminus S|}$ be an ordering of $V \setminus S$. Let M be the set of integer solutions to MWS-BSN and let M^ℓ be the subset of M restricted to $u_{j_k} = 0$ for all $k \in \{\ell+1, \ldots, |V \setminus S|\}$, where $\{\ell+1, \ldots, |V \setminus S|\}$ is considered equal to $\emptyset$ if $\ell + 1 > |V \setminus S|$. We employ a sequential lifting argument. Starting from the inequality $\sum_{i \in S} u_i \leq k$ which, as of Lemma 11, is facet defining for $\text{conv}(M^0)$, at each lifting iteration ℓ we obtain a facet of $\text{conv}(M^\ell)$ and, for $\ell = |V \setminus S|$, a facet of $\text{conv}(M)$.

At iteration ℓ, given the lifted inequality $\sum_{i \in S} u_i + \sum_{k \in \{1,\ldots,\ell-1\}} \lambda_{j_k} u_{j_k} \leq k$, valid for $\text{conv}(M^{\ell-1})$ for some $\lambda_{j_1}, \ldots, \lambda_{j_{\ell-1}} \in \mathbb{R}^+$, we compute the (largest) coefficient λ_{j_ℓ} for which the new inequality $\sum_{i \in S} u_i + \sum_{k \in \{1,\ldots,\ell-1\}} \lambda_{j_k} u_{j_k} + \lambda_{j_\ell} u_{j_\ell} \leq k$ is valid for $\text{conv}(M^\ell \cap \{u_{j_\ell} = 1\})$ (and thus for $\text{conv}(M^\ell)$). This lifting problem reads:

$$\Lambda_\ell = \max_{u \in \{0,1\}^n} \sum_{i \in S} u_i + \sum_{k \in \{1,\ldots,\ell-1\}} \lambda_{j_k} u_{j_k} \tag{1a}$$

$$\text{s.t. } u_{j_\ell} = 1 \tag{1b}$$

$$u_{j_k} = 0 \ \ \forall k \in \{\ell+1, \ldots, |V \setminus S|\} \tag{1c}$$

$$\alpha(G[\{i \in V : u_i = 1\}]) \leq k. \tag{1d}$$

Since S is maximal by definition of LCIs and $j_\ell \notin S$, $\exists i \in S : \{i, j_\ell\} \in E$. Let then S' be a subset of S containing vertex i, of cardinality $|S'| = k$. Since S' is a stable set and $\{i, j_\ell\} \in E$, $\alpha(G[S' \cup \{j_\ell\}]) = \alpha(G[S']) = |S'| = k$. By letting $u_{j_\ell} = 1$ and $u_i = 1$ for all $i \in S'$ we thus obtain a feasible solution to the lifting problem of value k. This shows that $\Lambda_\ell \geq k$. Since the lifted inequality is valid if and only if $\Lambda_\ell + \lambda_{j_\ell} \leq k$, we deduce $\lambda_{j_\ell} \leq 0$.

To show that $\lambda_{j_\ell} = 0$ for all $\ell \in \{1, \ldots, |V \setminus S|\}$, first note that, if $\lambda_{j_k} = 0$ for all $k \in \{1, \ldots, \ell - 1\}$, then $\Lambda_\ell \leq k$. Due to the previous argument, this implies $\Lambda_\ell = k$ and, hence, $\lambda_{j_\ell} = 0$. Also note that, for $\ell = 1$, no terms $\lambda_{j_k} u_{j_k}$ appear in the objective function and, hence, $\lambda_{j_1} = 0$. The claim then follows by induction (if $\lambda_{j_1}, \ldots, \lambda_{j_{\ell-1}} = 0$, then $\lambda_{j_\ell} = 0$), proving that, at the end of the lifting procedure, any LCI is lifted back to itself, and, therefore, is facet defining. ◀

Letting $u^* \in [0,1]^n$ (corresponding to a, possibly infeasible, solution to MWS-BSN), the separation problem for LCIs (in search version) reads:

▶ **Problem 3** (SEParation problem for LCIs (LCI-SEP))**.** *Given a graph $G = (V, E)$, a vector of vertex weights $u^* \in \mathbb{R}^n$, and an integer k, find a maximal stable set S of G with both weight and cardinality greater than or equal to $k+1$, or prove that none exists.*

Not surprisingly, the following holds:

▶ **Proposition 13.** *LCI-SEP is $\mathcal{NP}$-hard.*

Algorithm 1: Exact algorithm for the optimization over $\bigcap_{k=1}^{\bar{k}} RSTAB_k(G)$.

Solve the (current) relaxation of MSS; let x^* be its solution;
Let $k := 1$;
while $k \leq \bar{k}$ **do**
  solve MWS-BSN via branch-and-cut, separating LCIs;
  if *the corresponding* $RI_k^{\leq}$ *is violated* **then**
    add it to the relaxation of MSS;
    let $k := 1$;
  else
    let $k := k + 1$;
  end
end

Proof. MSS-d with input L and G has answer YES if and only if LCI-SEP with input $k = L - 1$ admits a feasible solution. ◀

Note that, due to the equivalence between optimization and separation [11], the facet-definingness of LCIs and their $\mathcal{NP}$-hardness imply, *en passant*, the $\mathcal{NP}$-hardness of MWS-BSN.

We remark that, since CIs are necessary to formulate MWS-BSN in the vertex space and there is an exponential number of them, solving MWS-BSN in that space via branch-and-bound requires a cut generation procedure.

4 Algorithmic aspects

In this section, we provide an outline of our algorithm for optimizing over $\bigcap_{k=1}^{\bar{k}} RSTAB_k(G)$ and then discuss a few of its aspects.

4.1 Algorithm outline

The overall algorithm by which the function $\sum_{i \in V} x_i$ is maximized over $RSTAB_k(G)$ can be summarized as follows:

4.2 Domination aspects of RIs: connectedness of $G[U]$

An easy condition under which a RI is dominated is the following one:

▶ **Observation 14.** *Any RI corresponding to a disconnected $G[U]$ is dominated.*

Proof. Assuming that $G[U]$ contains ℓ connected components $G[U_1], \dots, G[U_\ell]$, $\alpha(G[U]) = \sum_{j=1}^{\ell} \alpha(G[U_\ell])$. Hence, $\sum_{i \in U} x_i \leq \alpha(G[U])$ is the linear combination with unit weights of the ℓ inequalities $\sum_{i \in U_j} x_i \leq \alpha(G[U_j])$, for $j \in \{1, \dots, \ell\}$. ◀

To prevent the introduction of RI_ks with a disconnected $G[U]$, we identify (in linear time) the connected components $G[U_1], \dots, G[U_k]$ of $G[U]$ after each RI_k is generated. We then introduce a RI for each component, *in lieu* of the original one. For that, we recompute the RHS of each new inequality as $\alpha(G[U_j])$ (which is an easy task, provided that $|U|$ is reasonably small). Note that, since, for all $j \in \{1, \dots, k\}$, $\alpha(G[U_j]) \leq \alpha(G[U]) = k$, all the inequalities obtained after the decomposition of $G[U]$ are $\mathrm{RI}_{k'}$s with $k' < k$, thus being in $\bigcap_{k=1}^{\bar{k}} RSTAB_k(G)$.

4.3 Practical separation of LCIs

First, we note that, in the context of a branch-and-cut algorithm for MWS-BSN, LCIs can be separated on the *incumbent* solution. This allows to consider only the case where u is a binary vector. If this is the case, LCI-SEP becomes *exactly* an instance of MSS-d with $L = k+1$ due to the weight of the stable set becoming equal to its cardinality. Note also that, conveniently, LCIs can be obtained by separating CIs and, then, making the corresponding stable set maximal *a posteriori* via a greedy algorithm, in $O(n^2)$.

We remark that the separation of $\mathrm{RI}_k^{\leq}$s for the MSS problem entails, via the separation of CIs/LCIs, the solution of, yet again, MSS. Two things must be noted though: 1) the separation problem for CIs/LCIs can be solved on the subgraph induced by the incumbent solution u of MWS-BSN, which is much smaller, in practice, than G; 2) assuming $k \ll \alpha(G)$ for a sufficiently small k, finding a stable set of size k is, in practice, a computationally more affordable task than computing $\alpha(G)$.

In our computations, we will carry out the separation of CIs/LCIs with the exact solver Cliquer [17], which implements a combinatorial branch-and-bound algorithm not relying on mathematical programming relaxations.

4.4 Separating $\mathrm{RI}_k^{\leq}$s on the support of x^*

We will restrict ourselves to the subgraph induced by the solution vector being separated, x^* in this case, also when solving MWS-BSN. For this problem, a simple argument also allows to fix $u_i = 0$ for all $i \in V$ where $x_i^* = 1$. This is because, if $x_i^* = 1$, assuming that the LP relaxation of MSS contains, at least, all edge inequalities (which is always the case in our implementation), we have that, for all $j \in V : \{i,j\} \in E, x_j^* = 0$. As a consequence, when the aforementioned restriction is in place, vertex i is isolated. Since we are looking for inequalities where $G[U]$ is connected, node i can thus be safely discarded.

4.5 Heuristic procedure

To speedup the cutting plane algorithm for RI_ks, we also introduce a simple greedy heuristic for their separation. After sorting the vertices of V in nonincreasing order of x^*, we add them to U one at a time, until a maximal clique is formed (this way, only stable sets of cardinality 1 are introduced). Then, we add, in the previously found order, the next $k-1$ nodes. After this operation, the stability number of $G[U]$ is, at most, k. Then, for each vertex currently not in of U, we add it to U only if it does not form a stable set of cardinality $k+1$. If it does, we skip it and continue to the next vertex.

The algorithm runs in $O(n \log n + n^{k+1})$, where $O(n \log n)$ accounts for sorting and $O(n^k)$ is the number of operations needed to check whether a new vertex increases the stability number of the current subgraph past the upper bound of k. The latter operations are executed $O(n)$ times. Note that, by construction, any solution found by this heuristic is maximal. If, after the exploration of a given amount of nodes, the heuristic terminates without finding a violated inequality (the amount is set to 2 millions in our experiments), we resort to branch-and-cut.

5 Computational study

We now report on a set of results obtained with the algorithm that we described in the previous sections for the separation of topology-free RIs with a given RHS.

We remark that computational efficiency is not our primary concern here. Rather, we focus on assessing the *quality of the bounds* obtained with $\bigcap_{k=1}^{\bar{k}} RSTAB_k(G)$ for increasing values of $\bar{k}$. We will compare those bounds to those obtained when optimizing over $QSTAB(G)$ (the relaxation containing all clique inequalities) and when employing Lovász's Theta function $\vartheta(G)$, which yields one of the tightest upper bounds to MSS known in the literature (always at least as tight as that obtained with $QSTAB(G)$). We refer to the latter two bounds as $\alpha_{QSTAB}(G)$ and $\alpha_{\vartheta}(G)$. Throughout our experiments, we adopt $QSTAB(G)$ as the initial relaxation of MSS. Given an upper bound UB, we will measure its quality in terms of the fraction of gap that it closes w.r.t. $\alpha_{QSTAB}(G)$. Formally, we define the *closed gap* as:

$$\text{Closed Gap } \% := \left(1 - \frac{UB - \alpha(G)}{\alpha_{QSTAB}(G) - \alpha(G)}\right) 100.$$

5.1 Instances

We consider three groups of instances, all corresponding to sparse graphs (we recall that sparse graphs are usually much harder to solve than dense ones):

1. The first group contains uniform random graphs, generated with `rudy` [20]. They have 60, 70, and 80 vertices and an edge density between 5% and 25%. Those instances are particularly useful to measure the impact of $\text{RIs}_{\bar{k}}^{\leq}$ with $\bar{k} > 3$.
2. The second group is a subset of the largest instances among those used in [13] to solve MSS via SDP techniques. They are very sparse, with a density between 1% and 5%.
3. The third group is a small subset of sparse graphs taken from the DIMACS challenge on the max clique problem. All the instances for which either $\alpha_{QSTAB}(G) = \alpha(G)$ or for which $\alpha_{QSTAB}(G)$ cannot be computed *exactly* within the time limit are discarded.

5.2 Implementation details

Our algorithm is coded in C, using Gurobi 7.0 as MILP solver. We adopt the `parallel` setting, with 8 threads and default parameters. In all the separation problems, we set `solutionlimit=1`, imposing a violation cutoff of 0.01. For the separation of LCIs, we use Cliquer 1.21. The value $\vartheta(G)$ is obtained with DSDP 5.8. All the results are produced within a time limit of 7200 seconds (two hours) on an Intel i7-3770 CPU @ 3.40GHz desktop computer with 8 cores, with 16GB RAM.

5.3 A small example: the Chvàtal graph

As an illustrative example, we report the results obtained over the Chvàtal graph, the smallest triangle free 4-colorable 4-regular graph, see [5].

Figure 2 shows 11 RIs with $\bar{k} = 3$ generated by our topology free cutting plane algorithm, assuming $QSTAB(G)$ as the initial relaxation. Apart from the fourth inequality, which is isomorphic to the web inequality $W(8, 3)$, none of the remaining RIs corresponds to any of the valid inequalities with a given topology that are known in the literature. While the bound obtained with $QSTAB(G)$ is $\alpha_{QSTAB}(G) = 6$ (corresponding to the solution $x_i = \frac{1}{2}$ for all $i \in V$) and that obtained with Lovász's Theta function is $\alpha_{\vartheta}(G) = 4.895$, with RIs_ks and $k = 3$ we obtain a better bound equal to 4.5.

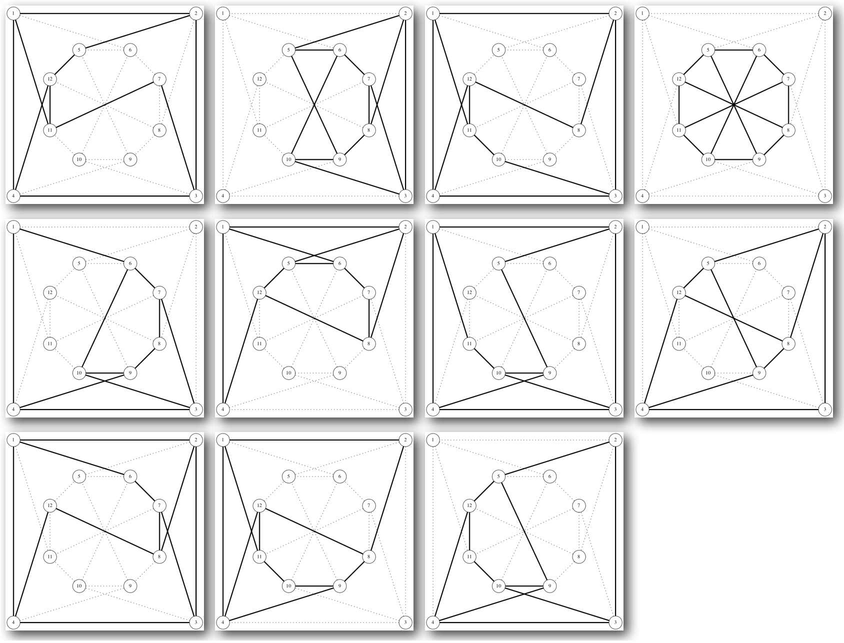

Figure 2 The set of the 11 RI_ks which are obtained when optimizing over $RSTAB_k(G)$ with $k = 3$ on the Chvàtal graph. They yield a bound of 4.5, as opposed to $\alpha_{QSTAB}(G) = 6$ and $\vartheta(G) = 4.895$.

5.4 Computational Results

Figure 3 reports the percentage of closed gap plotted against the running time for instance `r-70-10` in group 1, obtained when executing Algorithm 1 with $\bar{k} = 5$. This plot clearly shows that, in the very first iterations of the algorithm, RIs are already able to close a large percentage of the gap, closing 90% of it in only 40 seconds. After 250 seconds, nearly 100% of the gap is closed. The additional 250 seconds are only necessary to prove (computationally) that the upper bound that has been obtained cannot be improved any further. The plot in the figure illustrates a behaviour which can be observed in all the results that we will discuss in the next paragraph.

The results obtained on the three groups of instances are summarized in Table 1. For each value of $\bar{k} = \{2, 3, 4, 5\}$, we report the Upper Bound (UB) that has been found, the running time in seconds (Time), and the number of cuts that were generated (Cuts). We also report the average closed gap (Avg ClGap), as computed over the instances belonging to each group.

On the first group of instances, our algorithm manages to close, on average, more than 50% of the open gap already with $\bar{k} = 2$. Larger values of $\bar{k}$ yield a larger closed gap, up to more than 80% with $\bar{k} = 5$. Note though that this result is counterbalanced by an increase of running time as, for $\bar{k} = 5$, most instances hit the time limit of 2 hours.

We remark that, in the first two groups of instances, RI_ks with $\bar{k} = 3$ suffice to obtain stronger bounds than those achieved with Lovász's Theta function $\vartheta(G)$. On the instances in group 1 we register, on average, 67.7% of gap closed with RI_k, as opposed to 67.6% with $\vartheta(G)$, a value which increases to 73.2% for group 2 as opposed, for that group, to the 65.8% obtained with $\vartheta(G)$.

The improvement w.r.t. $\vartheta(G)$ further increases when considering $\bar{k} = 4$ and $\bar{k} = 5$. The quality of the bound improvement becomes hard to assess though on the third group of

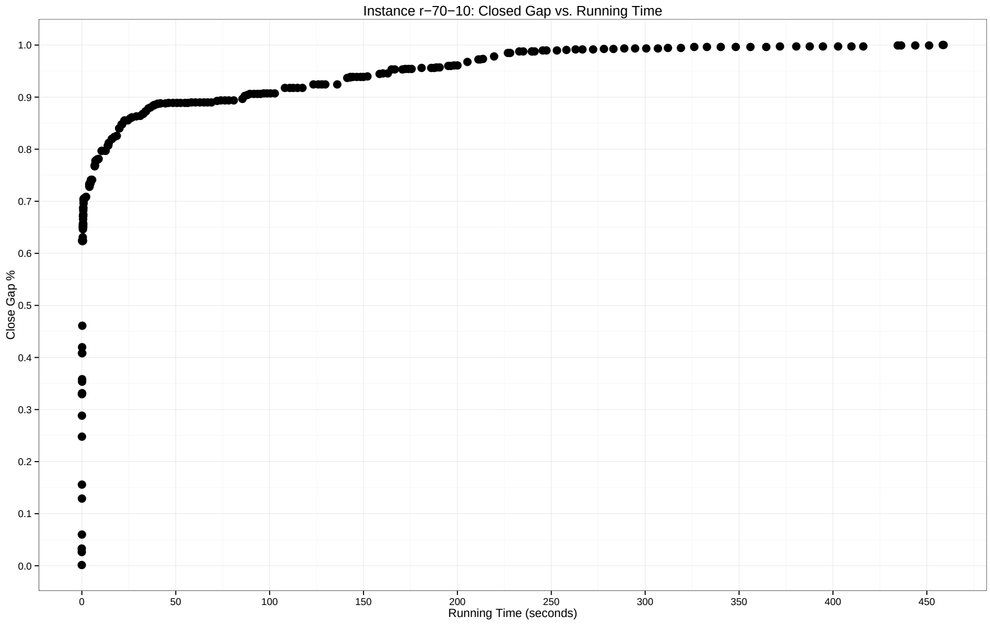

Figure 3 Percentage of closed gap plotted against the running time for instance `r-70-10` in group 1, obtained when executing Algorithm 1 with $\bar{k} = 5$.

instances where, already with $\bar{k} = 2$, our algorithm hits the time limit in three cases out of seven.

We remark that the cuts that we generate are quite sparse. As an example, consider instance `r-70-10` from group 1 (containing 70 nodes). On average, we generate inequalities with $|U|$ (corresponding to the number of nonzeros in the LHS) equal to 5.2 for $k = 2$, 8.01 for $k = 3$, 10.7 for $k = 4$, and 12.9 for $k = 5$.

To conclude, we highlight the results on the instance `hamming6-4`: with only 11 cuts with $\bar{k} = 2$, generated on top of those in $QSTAB(G)$, our algorithm achieves the optimal bound equal to 4, while both $QSTAB(G)$ and $\vartheta(G)$ yield a larger upper bound equal to 5.33.

6 Concluding remarks

We have addressed the separation of topology-free rank inequalities with a fixed (arbitrary) right-hand side (RI$_k$s). We have proposed a methodology to optimize over the closure of RI$_k$s for all $k \in \{1, \dots, \bar{k}\}$, investigating the arising separation problem and its polyhedral structure. For its solution, we have proposed a branch-and-cut method which separates facet defining inequalities belonging to an exponentially large family of inequalities that are needed to correctly model the problem.

Overall, RI$_k$s with a small right-hand side $k \ll \alpha(G)$ yield a substantial bound improvement over the bound provided by the fractional clique polytope $QSTAB(G)$. In a number of cases, such bound is also tighter than $\vartheta(G)$, the bound obtained with Lovász's Theta function via semidefinite programming.

Future work includes the development of *ad hoc* algorithms for the separation of RI$_k$s with a small right-hand side k. Due to the bound improvement that, in our experiments, RIs have shown to yield, the effectiveness of such algorithms could allow to add RI$_k$s with $k = 2$ and $k = 3$ to the set of cutting planes that are routinely generated to solve the maximum stable set problem to optimality.

Table 1 Bounds (UB) obtained with RI_ks with $k \in \{2,3,4,5\}$, compared to $\alpha(G)$, $\alpha_{QSTAB}(G)$, and $\vartheta(G)$. Computing times in seconds (Time) and total number of generated cutting planes (Cuts) are also reported. Bounds which are tighter than those obtained with $\vartheta(G)$ are highlighted in bold. Closed Gap, averaged in geometric mean (Avg ClGap), is reported for the three classes of instances also considering instances for which the time limit is met: small uniform random graphs generated with rudy, large uniform random graphs taken from [13], and structured instances from the DIMACS challenge.

				RI_k with $\bar{k}=2$			RI_k with $\bar{k}=\{2,3\}$			RI_k with $\bar{k}=\{2,3,4\}$			RI_k with $\bar{k}=\{2,3,4,5\}$		
	$\alpha(G)$	$\alpha_Q(G)$	$\vartheta(G)$	**Cuts**	**UB**	**Time**	**Cuts**	**UB**	**Time**	**Cuts**	**UB**	**Time**	**Cuts**	**UB**	**Time**
r-60-5	31	31.50	31.07	5	**31.00**	0	5	**31.00**	0	5	**31.00**	0	5	**31.00**	0
r-60-10	23	25.63	23.67	42	23.47	1	96	23.14	28	124	**23.00**	77	124	**23.00**	77
r-60-15	18	21.20	19.54	70	19.70	2	189	19.19	77	315	18.97	521	455	**18.81**	2090
r-60-20	7	9.25	7.5	437	8.15	1254	946	7.79	tlim	946	7.79	tlim	947	**7.47**	tlim
r-60-25	14	16.50	14.67	112	15.24	12	348	14.70	413	569	14.38	2814	756	**14.13**	tlim
r-70-5	35	36.00	35.53	5	35.50	0	6	35.50	0	6	35.50	0	8	**35.00**	0
r-70-10	26	28.66	26.86	63	26.78	2	139	26.29	72	230	26.01	417	236	**26.00**	459
r-70-15	21	23.82	21.91	103	22.16	8	265	21.63	289	423	21.37	2199	550	**21.22**	tlim
r-70-20	17	20.52	18.23	119	19.18	30	337	18.46	985	510	18.16	tlim	510	**18.15**	tlim
r-70-25	14	18.13	15.72	144	16.66	39	421	16.04	2039	607	15.71	tlim	607	**15.64**	tlim
r-80-5	39	39.50	39.02	3	**39.00**	0	3	**39.00**	0	3	**39.00**	0	3	**39.00**	0
r-80-10	27	30.50	28.55	69	29.02	8	194	28.38	399	350	27.95	4220	399	**27.76**	tlim
r-80-15	22	26.74	23.65	120	24.76	26	328	23.97	1874	448	23.67	tlim	448	**23.59**	tlim
r-80-20	18	22.78	20.05	145	21.06	41	422	20.40	3335	512	20.22	tlim	512	**20.07**	tlim
r-80-25	16	19.85	17.07	178	18.19	85	477	17.61	tlim	478	17.61	tlim	478	**17.55**	tlim
Avg ClGap			**67.6%**		**53.0%**			**67.7%**			**73.4%**			**80.3%**	
g150.4	59	67.00	61.8	99	62.09	50	250	60.80	tlim	250	60.80	tlim	250	**60.67**	tlim
g150.5	55	64.00	58.73	152	58.56	72	304	57.75	tlim	304	57.75	tlim	304	**57.67**	tlim
g170.3	71	78.50	73.34	76	73.53	44	181	72.16	6861	182	72.15	7415	182	**72.14**	tlim
g200.2	96	100.00	97.17	21	97.00	11	46	**96.00**	378	49	**96.00**	437	50	**96.00**	439
g200.3	83	94.50	86.52	123	86.61	221	202	85.21	tlim	202	85.21	tlim	202	**85.02**	tlim
g300.2	122	141.00	129.47	144	130.43	861	169	**130.07**	tlim	169	**130.07**	tlim	169	**130.07**	tlim
g350.2	133	161.00	143.43	273	146.11	4996	274	145.99	tlim	274	145.99	tlim	274	**145.87**	tlim
g400.1	191	201.00	194.79	33	195.50	131	60	**193.73**	tlim	60	**193.73**	tlim	60	**193.73**	tlim
Avg ClGap			**65.8%**		**61.6%**			**73.2%**			**73.2%**			**73.3%**	
brock200_1	21	38.02	27.46	267	35.59	tlim	267	35.59	tlim	267	35.59	tlim	267	35.59	tlim
C125.9	34	43.06	37.81	188	39.75	409.2	322	39.20	tlim	320	39.21	tlim	322	39.21	tlim
C250.9	44	71.37	56.24	375	66.05	tlim	375	66.05	tlim	375	66.05	tlim	375	66.05	tlim
hamming6-4	4	5.33	5.33	11	**4.00**	1.5	11	**4.00**	1.5	11	**4.00**	1.5	11	**4.00**	1.5
keller4	11	14.83	14.01	314	**13.80**	tlim	318	**13.80**	tlim	314	**13.80**	tlim	314	**13.80**	tlim
MANN_a9	16	18.00	17.48	1	18.00	0.1	1	18.00	0.7	1	18.00	2.7	1	18.00	13.1
sanr200_0.9	45	59.82	49.27	366	55.14	tlim	366	55.13	tlim	366	55.14	tlim	366	55.14	tlim
Avg ClGap			**26.5%**		**19.5%**			**19.9%**			**19.9%**			**19.9%**	

References

1. Edoardo Amaldi, Stefano Coniglio, and Stefano Gualandi. Improving cutting plane generation with 0-1 inequalities by bi-criteria separation. *Experimental Algorithms*, pages 266–275, 2010.
2. Edoardo Amaldi, Stefano Coniglio, and Stefano Gualandi. Coordinated cutting plane generation via multi-objective separation. *Mathematical Programming*, 143(1-2):87–110, 2014.
3. Chitra Balasubramaniam and Sergiy Butenko. The maximum s-stable cluster problem. In *INFORMS 2015 Annual Meeting*. INFORMS, 2015.
4. Chitra Balasubramaniam and Sergiy Butenko. The maximum s-stable cluster problem. Working paper, 2017.
5. Václav Chvátal. The smallest triangle-free 4-chromatic 4-regular graph. *Journal of Combinatorial Theory*, 9(1):93–94, 1970.
6. Vašek Chvátal. On certain polytopes associated with graphs. *Journal of Combinatorial Theory, Series B*, 18(2):138–154, 1975.
7. Stefano Coniglio and Martin Tieves. On the generation of cutting planes which maximize the bound improvement. In *Experimental Algorithms (14th International Symposium, SEA 2015, Paris, France, June 29 – July 1, 2015, Proceedings)*, volume 9125, pages 97–109. Springer International Publishing, 2015.
8. Harlan Crowder, Ellis L. Johnson, and Manfred Padberg. Solving large-scale zero-one linear programming problems. *Operations Research*, 31(5):803–834, 1983.
9. Monia Giandomenico, Adam N Letchford, Fabrizio Rossi, and Stefano Smriglio. Ellipsoidal relaxations of the stable set problem: theory and algorithms. *SIAM Journal on Optimization*, 25(3):1944–1963, 2015.
10. Monia Giandomenico, Fabrizio Rossi, and Stefano Smriglio. Strong lift-and-project cutting planes for the stable set problem. *Mathematical Programming*, 141(1-2):165–192, 2013.
11. Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.
12. Martin Grötschel, Laszlo Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, 1988.
13. Gerald Gruber and Franz Rendl. Computational experience with stable set relaxations. *SIAM Journal on Optimization*, 13(4):1014–1028, 2003.
14. J. Hastad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999.
15. Richard M. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Proceedings of a Symposium on the Complexity of Computer Computations*, The IBM Research Symposia Series. Plenum Press, 1972.
16. Carlo Mannino and Antonio Sassano. Edge projection and the maximum cardinality stable set problem. *DIMACS series in discrete mathematics and theoretical computer science*, 26:205–219, 1996.
17. Patric R. J. Östergård. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, 120(1):197–207, 2002.
18. Steffen Rebennack, Marcus Oswald, Dirk Oliver Theis, Hanna Seitz, Gerhard Reinelt, and Panos M. Pardalos. A branch and cut solver for the maximum stable set problem. *Journal of combinatorial optimization*, 21(4):434–457, 2011.
19. Fabrizio Rossi and Stefano Smriglio. A branch-and-cut algorithm for the maximum cardinality stable set problem. *Operations Research Letters*, 28(2):63–74, 2001.
20. Yinyu Ye. Rudy random graph generator. `http://web.stanford.edu/~yyye/yyye/Gset`.