**Submitted to *INFORMS Journal on Computing***
**manuscript (Please, provide the manuscript number!)**



# Optimizing over the Closure of Rank Inequalities with a Small Right-Hand Side for the Maximum Stable Set problem via Bilevel Programming*

S. Coniglio
Department of Mathematical Sciences, University of Southampton, University Road, Southampton, UK, s.coniglio@soton.ac.uk

S. Gualandi
Department of Mathematics, University of Pavia, Via Ferrata 1, Pavia, Italy, stefano.gualandi@unipv.it

In the context of the maximum stable set problem, rank inequalities impose that the cardinality of any set of vertices contained in a stable set be, at most, as large as the stability number of the subgraph induced by such a set. Rank inequalities are very general, as they subsume many classical inequalities such as clique, hole, antihole, web, and antiweb inequalities. In spite of their generality, the exact separation of rank inequalities has never been addressed without the introduction of topological restrictions on the induced subgraph and the tightness of their closure has never been investigated systematically.

In this work, we propose a methodology for optimizing over the closure of all rank inequalities with a right-hand side no larger than a small constant without imposing any restrictions on the topology of the induced subgraph. Our method relies on the exact separation of a relaxation of rank inequalities, which we call relaxed $k$-rank inequalities, whose closure is as tight. We investigate the corresponding separation problem, a bilevel programming problem asking for a subgraph of maximum weight with a bound on its stability number, whose study could be of independent interest. We first prove that the problem is $\Sigma_2^{\mathrm{P}}$-hard and provide some insights on its polyhedral structure. We then propose two exact methods for its solution: a branch-and-cut algorithm (which relies on a family of faced-defining inequalities which we introduce in this paper) and a purely combinatorial branch-and-bound algorithm. Our computational results show that the closure of rank inequalities with a right-hand side no larger than a small constant can yield a bound that is stronger, in some cases, than Lovász's Theta function, and substantially stronger than bounds obtained with standard inequalities that are valid for the stable set problem, including odd-cycle inequalities and wheel inequalities.

*Key words*: Integer Programming; Maximum Stable Set Problem; Rank Inequalities; Cutting Plane Generation; Bilevel Programming; Branch-and-Cut; Branch-and-Bound

*History*:

---

* A preliminary version of this paper appeared in Coniglio and Gualandi (2017).

## 1. Introduction

Let $G = (V, E)$ be an undirected graph with vertex set $V$ and edge set $E$ and let $n = |V|$. A subset $S \subseteq V$ is called a *stable set* if its vertices are pairwise non-adjacent, whereas it is said to form a *clique* if its vertices are pairwise adjacent. We call *stability number* of $G$ the size of its maximum-cardinality stable sets and denote it by $\alpha(G)$. The *maximum stable set problem* (MSSP) asks for computing such a number. By complementing the edge set of the graph, the problem is equivalent to the *maximum clique problem* (Padberg 1973).

The MSSP is one of the fundamental problems in combinatorial optimization, with applications in, among others, biochemistry (Butenko and Wilhelm 2006), computer vision (San Segundo and Artieda 2015), combinatorial auctions (Wu and Hao 2016), and social network analysis (Balasundaram et al. 2011). For a more extensive account, we refer the reader to the surveys (Pardalos and Xue 1994, Wu and Hao 2015). The problem was shown to be NP-hard by Karp (1972) and it cannot be approximated in polynomial time to within any factor $O(n^{1-\varepsilon})$ for any $\varepsilon > 0$ unless $\mathsf{P} = \mathsf{NP}$ (Hastad 1999, Zuckerman 2006).

The MSSP is very hard to solve also in practice. While state-of-the-art methods can tackle many structured instances with up to millions of nodes (San Segundo et al. 2016), computing $\alpha(G)$ on very sparse (random) graphs with more than 300 nodes is still beyond the capabilities of even the most efficient algorithms known in the literature, including those proposed by Tomita and Kameda (2007), San Segundo et al. (2011), Held et al. (2012), Maslov et al. (2014), and San Segundo et al. (2019). Remarkably, state-of-the-art branch-and-cut methods based on the polyhedral structure of the problem and on mathematical programming techniques are often outperformed by purely combinatorial branch-and-bound methods such as the ones we mentioned above. For mathematical programming approaches for solving the MSSP and/or computing tight bounds, we refer the reader to the works by Nemhauser and Sigismondi (1992), Bourjolly et al. (1997), Rossi and Smriglio (2001), Amaldi et al. (2010), Rebennack et al. (2011), Giandomenico et al. (2013), Amaldi et al. (2014), Coniglio and Tieves (2015), Giandomenico et al. (2015), Corrêa et al. (2018), and Letchford et al. (2020). An overview of solution algorithms can be found in the surveys authored by Prosser (2012) and Wu and Hao (2015). For exact (either combinatorial or mathematical-programming based) methods for variants of the problem (including the cases with weights on the edges or knapsack constraints), we refer the reader to Gouveia and Martins (2015), Hosseinian et al. (2017, 2018), Shimizu et al. (2018), San Segundo et al. (2019), and Coniglio et al. (2021).

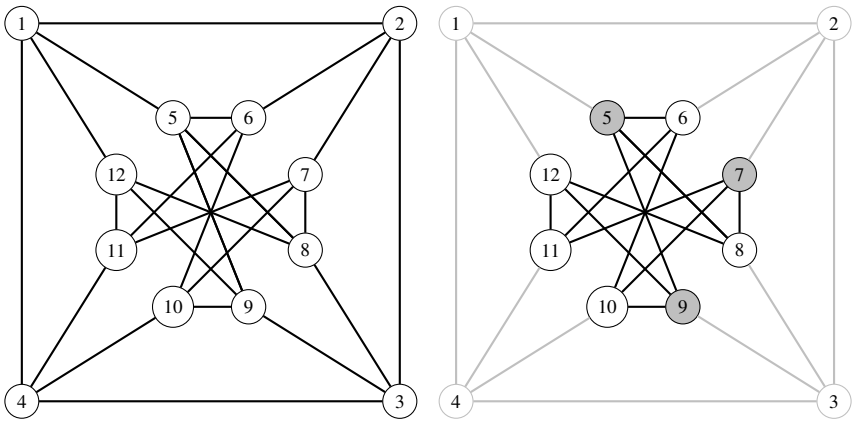


**Figure 1 The Chvàtal Graph.**

*Note.* (a) The smallest triangle-free 4-colorable 4-regular graph (Chvátal 1970). (b) The subgraph induced by $U = \{5,6,7,8,9,10,11,12\}$ (in black) and the vertices forming a maximum stable set (of size $\alpha(G[U]) = 3$) in $G[U]$ (in gray). The corresponding RI is $\sum_{j=5}^{12} x_j \leq 3$.

Let $\text{STAB}(G)$ be the *stable set polytope* of $G$, i.e., the convex hull of the characteristic vectors $x \in \{0,1\}^n$ of all the stable sets of $G$. The polyhedral structure of $\text{STAB}(G)$ has been heavily investigated in the literature starting from the seminal work of Padberg (1973), who introduced the family of *clique* inequalities, defined as $\sum_{j\in C} x_j \leq 1$ for every $C \subseteq V$ inducing a (maximal) clique of $G$, as well as the family of *odd holes*, defined as $\sum_{j\in H} x_j \leq \frac{|H|-1}{2}$ for every $H \subseteq G$ inducing a chordless cycle (a hole). Many other inequalities based on the combinatorial structure of $\text{STAB}(G)$ have since been introduced, including *web* and *antiweb* (Trotter 1975), *wheel* (Cheng and Cunningham 1997), and *antiweb-wheel* (Cheng and de Vries 2002).

In this work, we focus on *Rank Inequalities* (RIs), a family of valid inequalities introduced by Chvátal (1975). For any $U \subseteq V$, RIs have the form $\sum_{j\in U} x_j \leq \alpha(G[U])$, where $G[U]$ is the subgraph induced by $U$. See Fig. 1 for an example.

RIs are extremely general, as every inequality with binary left-hand side (LHS) coefficients which is valid for $\text{STAB}(G)$ either is a RI or it is dominated by one.[1] Crucially, many of the valid inequalities which are known for $\text{STAB}(G)$ are obtained by imposing *topological restrictions* on $G[U]$—clique, odd hole, web, and antiweb inequalities are indeed special cases of RIs obtained when $G[U]$ is, respectively, a clique, an odd hole, a web, and an antiweb.

[1] By definition, an inequality $ux \leq u_0$ is valid for a set $P \subseteq \mathbb{R}^n$ if and only if it is satisfied by every $x \in P$. This is the same as requiring $u_0 \geq \max\{ux : x \in P\}$. Letting $P = \text{STAB}(G)$ and assuming $u \in \{0,1\}^n$, the condition translates into $u_0 \geq \max\{ux : x \in \text{STAB}(G)\} = \max\{ex : x \in \text{STAB}(G[U])\} = \alpha(G[U])$, where $e$ is the all-one vector. Therefore, $ux \leq u_0$ is a RI if $u_0 = \alpha(G[U])$, whereas it is dominated by the RI $ux \leq \alpha(G[U])$ if $u_0 > \alpha(G[U])$.

Among the special cases of RIs obtained by imposing topological restrictions on $G[U]$, clique inequalities are, arguably, the most important one, as they are separated in almost all of the state-of-the-art branch-and-cut method for the MSSP (such as those that we referenced before). For state-of-the-art approaches for their separation, we refer the reader to (Marzi et al. 2019).

For the general case where topological restrictions are not imposed, it has been shown that RIs have a strong impact when separated (heuristically) within a branch-and-cut solver (Rossi and Smriglio 2001, Rebennack et al. 2012) by relying on the *edge-projection* operator proposed by Mannino and Sassano (1996). For its extension to the *clique-projection* operator and its adoption for the generation of both rank and nonrank inequalities, see (Corrêa et al. 2018).

When aiming at closing a large fraction of the integrality gap, RIs look extremely promising. This is because the introduction of the single RI $\sum_{j\in V} x_j \leq \alpha(G)$ (obtained for $U = V$) into any relaxation of $\text{STAB}(G)$ suffices to calculate $\alpha(G)$, thereby closing 100% of the integrality gap. The obvious consequence is that optimizing over the closure of all RIs (given a family of inequalities, their closure is the set of points that satisfy all of them) is at least as hard as solving the MSSP—we will discuss about this aspect later on in the paper.

*Paper contributions* In this work, we address the separation of RIs from a new angle—separating them with a restriction not on the topology of $G[U]$ but, rather, on the magnitude of their RHS $\alpha(G[U])$—and investigate the tightness of the closure of all RIs with a *small (constant) RHS*. For the purpose, we introduce a closely related family of inequalities (which we call *relaxed k-rank inequalities*) and study their separation problem, which turns out to be a bilevel programming problem asking for a subgraph with bounded stability number of maximum weight and whose investigation could be of independent interest. We prove that the problem is $\Sigma_2^{\text{P}}$-hard, investigate its polyhedral structure, and propose two exact algorithms for its solution (a branch-and-cut algorithm and a purely combinatorial branch-and-bound algorithm). By relying on them, our computational experiments reveal that the closure of all RIs with a small RHS (no larger than 5) can yield bounds that can be, not in general but in a few cases, better than those obtained with Lovász's theta function $\vartheta(G)$, see (Lovász 1979), and also better than those obtained with standard inequalities that are valid for the stable set problem, including odd-cycle inequalities and wheel inequalities. See Figure 2 for a visual illustration of the inequalities obtained with our method.

*Paper outline* The paper is organized as follows. Section 2 introduces relaxed $k$-rank inequalities and investigates their separation problem and its computational complexity. Section 3 furthers the analysis by proposing a formulation for the separation problem which relies on a family of

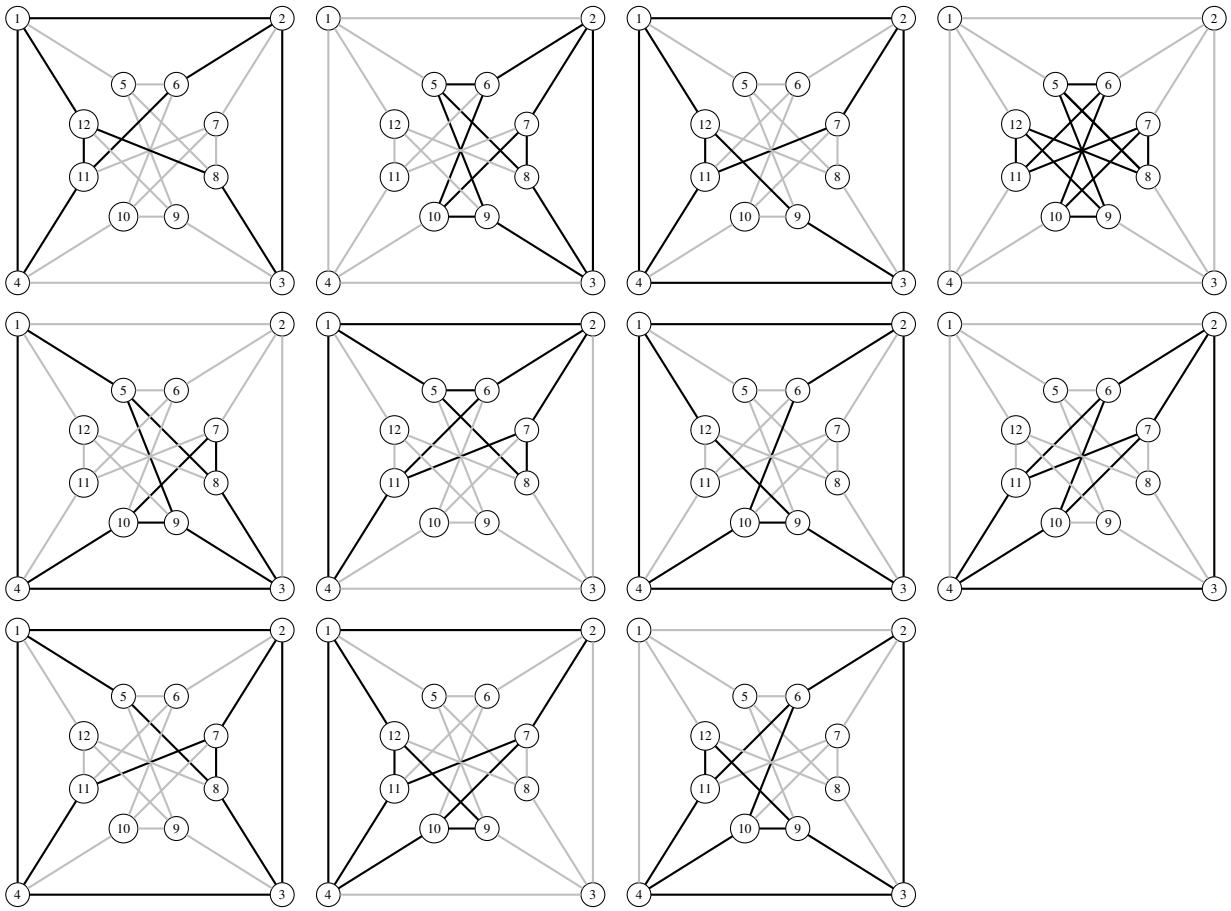

**Figure 2 Rank Inequalities (RIs) with a right-hand size no larger than 5.**

*Note.* The 11 RIs with a right-hand size no larger than 5 generated with our method on the Chvàtal Graph. They yield a bound of 4.5, tighter than the one obtained with Lovász's theta function $\vartheta(G)$, equal to 4.895.

facet-defining inequalities which we introduce in the same section. Section 4 describes the two algorithms we propose for solving the separation problem. Section 5 illustrates a set of computational experiments carried out to assess the tightness of the closure of all RIs with a small RHS (no larger than 5). Concluding remarks are drawn in Section 6.

## 2. (Relaxed) $k$-Rank Inequalities

For a given $k \in \mathbb{N}$, we call *k-rank inequality* ($k$-RI) any RI with a RHS equal to $k$ and we denote by $k$-RSTAB$(G)$ the corresponding closure. For the ease of notation, we assume that $k$-RSTAB$(G)$ also includes the nonnegativity constraints on $x$. Given an integer $\kappa$, the closure of all RIs with a RHS no larger than $\kappa$ is:

$$\bigcap_{k=1}^{\kappa} k\text{-RSTAB}(G). \tag{1}$$

In order to optimize $\sum_{j\in V} x_j$ (the objective function of the MSSP) over (1), we first introduce the family of *relaxed k-rank inequalities* (R-$k$-RI), defined as $\sum_{j\in U} x_j \leq k$ for any $U \subseteq V$ and $k \geq \alpha(G[U])$. We denote their closure by R-$k$-RSTAB$(G)$. As for $k$-RSTAB$(G)$, we assume that R-$k$-RSTAB$(G)$ includes the nonnegativity constraints on $x$. The relationship between R-$k$-RIs and $k$-RIs is quite straightforward: any R-$k$-RI $\sum_{j\in U} x_j \leq k$ is a $k$-RI if $k = \alpha(G[U])$ while, if $k > \alpha(G[U])$, the R-$k$-RI $\sum_{j\in U} x_j \leq k$ is dominated by the $k'$-RI $\sum_{j\in U} x_j \leq k'$ having the same LHS and a RHS $k' = \alpha(G[U])$. Therefore:

$$\bigcap_{k=1}^{\kappa} R\text{-}k\text{-RSTAB}(G) = \bigcap_{k=1}^{\kappa} k\text{-RSTAB}(G).$$

This implies that one can optimize over (1) by separating R-$k$-RIs rather than $k$-RIs. The reason why this is advisable is that the separation problem of the former is a relaxation of that of the latter, as better shown at the end of this section.

Before proceeding to investigate the separation problem of R-$k$-RIs, we point out that verifying the *membership* of a given inequality to the family of either $k$-RIs or R-$k$-RIs is a computationally difficult problem.

PROPOSITION 1. *Given some $u \in \{0,1\}^n$ and $k \in \mathbb{N}$, it is strongly* NP*-hard to verify whether the inequality $ux \leq k$ is a $k$-RI or an R-$k$-RI.*

*Proof.* We show that verifying whether $ux \leq k$ belongs to either family is as hard as solving the MSSP. Let $U = \{j \in V : u_j = 1\}$.

Consider a routine returning the answer YES if and only if $ux \leq k$ is a $k$-RI—by definition of $k$-RI, when called with $u_i = 1$ for all $i \in V$ the routine returns answer YES if and only if $k = \alpha(G[U])$. Calling the routine $O(n)$ times, for each $k = 1,\ldots,n$, solves the MSSP as, whenever it returns the answer YES, it has proven $\alpha(G) = k$.

Consider now a routine returning YES if and only if $ux \leq k$ is a R-$k$-RI—by definition of R-$k$-RI, when called on $ux \leq k$ with $u_i = 1$ for all $i \in V$, the routine returns the answer YES if and only if $\alpha(G) \leq k$. Calling it $O(n)$ times with $k = n,\ldots,1$, the routine returns the answer YES as long as $k$ is an upper bound on $\alpha(G)$, thereby proving, the first time it returns the answer NO for some value of $k$, that $\alpha(G) = k+1$. Q.E.D.

The proposition implies that the separation problem of deciding whether there exists either a $k$-RI or an R-$k$-RI of positive violation does not (unless P = NP) belong to NP as, given an inequality $ux \leq k$, verifying its validity is NP-hard. As a consequence, one may expect the separation problem

for $k$-RIs and R-$k$-RIs to be harder to solve than an NP-hard problem. We show that this is indeed the case, starting from R-$k$-RIs.

Given a vector $x^* \in \mathbb{Q}^n$ and an integer $k$, finding a R-$k$-RI of maximum violation corresponds to solving the following combinatorial optimization problem: [2]

PROBLEM 1. **Maximum Weight/Bounded Stability Subgraph Problem (MW/BSSP):** Given a graph $G = (V,E)$, a weight vector $x^* \in \mathbb{Q}^n$, and an integer $k$, find a subset of vertices $U \subseteq V$ whose induced subgraph $G[U]$ is of maximum weight $\sum_{j\in U} x^*_j$ among all those with stability number $\alpha(G[U])$ no larger than $k$.

We establish the following:

THEOREM 1. *The MW/BSSP is $\Sigma_2^{\mathsf{P}}$-hard.*

*Proof* Let us first introduce the decision version of the MW/BSSP:

PROBLEM 2. **Maximum Weight/Bounded Stability Subgraph Problem (Decision) (MW/BSSP-D):** Given a graph $G = (V,E)$, a weight vector $x^* \in \mathbb{R}^n$, an integer $k$, and a rational $w$, is there a subset of vertices $U \subseteq V$ of weight $\sum_{j\in U} x^*_j \geq w$ whose induced subgraph $G[U]$ has stability number $\alpha(G[U])$ no larger than $k$?

To show that the MW/BSSP-D belongs to $\Sigma_2^{\mathsf{P}}$, it suffices to notice that, given a set $U$ and an oracle which computes $\alpha(G[U])$ in $O(1)$, one can check in polynomial time whether $U$ certifies that the quadruple $(G,x^*,k,w)$ is a YES instance of the MW/BSSP-D (as this only requires computing $\sum_{j\in U} x^*_j$, comparing it to $w$, computing $\alpha(G[U])$ with the oracle, and comparing it to $k$).

We show that the MW/BSSP-D is complete for $\Sigma_2^{\mathsf{P}}$ by reduction from the following decision problem (where $\omega(G)$ is the *clique number* of $G$, i.e., the size of its largest-cardinality clique):

PROBLEM 3. **Generalized Node Deletion Problem (GNDP)** (Rutenburg 1994): Given a graph $G = (V,E)$ and two positive integers $b$ and $c$ with $b \geq 2$, is there a subset $D \subseteq V$ with $|D| \leq b$ such that $\omega(G[V \setminus D]) \leq c$?

Letting $U := V \setminus D$, we have $|D| \leq b$ if and only if $|U| \geq |V| - b$. It follows that the GNDP has answer YES if and only if $G$ contains a subset $U \subseteq V$ with $|U| \geq |V| - b$ such that $\omega(G[U]) \leq c$. Letting now $\overline{G} := (V, V \times V \setminus E)$ be the complement graph of $G$ (ignoring loops of type $(i,i), i \in V$), $\omega(G[U]) \leq c$ holds if and only if $\alpha(\overline{G}[U]) \leq c$. Let now $x^*_i := 1$ for all $i \in V$, $w := |V| - b$, and

[2] This problem is similar to the clique interdiction problem studied in Furini et al. (2019). They differ in that, besides considering cliques rather than stable sets, the latter calls for a set of vertices of bounded cardinality to be removed from the graph so to minimize its clique number, whereas the MW/BSSP calls for a set of vertices of maximum weight to be kept in the graph in order for its the stability number to be bounded.

$k := c$, and call an oracle for the solution of the MW/BSSP-D. If the oracle returns answer YES, $\overline{G}$ contains a set $U$ of size $|U| \geq w$ with $\alpha(\overline{G}[U]) \leq c$, which implies that the GNDP has answer YES. If the oracle returns answer NO, than $\overline{G}$ does not contain any set $U$ of size $|U| \geq w$ with $\alpha(\overline{G}[U]) \leq c$ and the GNDP has answer NO. The proof is concluded since, as shown by Rutenburg (1994), the GNDP is $\Sigma_2^{\mathsf{P}}$-complete. Q.E.D.

This result shows that finding a R-$k$-RI of maximum violation is, in general, harder than solving an NP-hard problem (its decision problem is, indeed, one level above the class of NP-complete problems in the polynomial hierarchy) and, in particular, that doing so is harder than solving the MSSP. Theorem 1 also shows that the MW/BSSP cannot be cast as a single-level mathematical program of polynomial size unless the polynomial hierarchy collapses to its first level. This is because, if a polynomially-sized mathematical programming formulation existed, every $\Sigma_2^{\mathsf{P}}$-complete problem would be certifiable in polynomial time via such formulation and, hence, $\Sigma_2^{\mathsf{P}}$ would coincide with NP. Fore more details on the relationship between cutting plane generation, bilevel programming, and the polynomial hierarchy, we refer the reader to (Lodi et al. 2014). For a recent survey on mixed integer multilevel programming (and its relationship with mixed integer multistage programming), we refer the reader to (Bolusani et al. 2020).

Crucially, if $k$ is a constant (as we assume in this paper), the following holds:

COROLLARY 1. *Unless* $\Sigma_2^{\mathsf{P}} = \mathsf{NP}$ *(i.e., unless the polynomial hierarchy collapses to its first level), for a fixed* $k$ *the MW/BSSP is* NP*-hard and not* $\Sigma_2^{\mathsf{P}}$*-hard.*

*Proof.* First, we notice that, if $k$ is fixed, the MW/BSSP-D admits a polynomial-time checkable certificate, which implies that the problem belongs to NP. Indeed, given a set of vertices $\tilde{U} \subseteq V$, $\sum_{j \in \tilde{U}} x_j^* \geq w$ can be checked in linear time and $\alpha(G[\tilde{U}]) \leq k$ can be checked in polynomial time by building all the $O(n^{k+1})$ stable sets $S \subseteq V$ and verifying that $|S \cap \tilde{U}| \leq k$. To see that the MW/BSSP-D is NP-complete for a constant $k$, it suffices to notice that, for $k = 1$, the problem is identical to the decision version of the maximum (weighted) clique problem. Q.E.D.

The corollary implies that, under the assumptions of this work, the problem of separating R-$k$-RIs in computationally as hard as the MSSP (which, due to the equivalence between optimization and separation (Grötschel et al. 1981), is the usual case for at least a family of inequalities for NP-hard optimization problems), but not harder.

We conclude the section by observing that the MW/BSSP is a relaxation of the separation problem for $k$-RIs. The latter is indeed obtained by adding an extra constraint to the MW/BSSP forcing the

induced subgraph $G[U]$ to have a stability number not only no larger than $k$, but also no smaller than $k$. Such a problem can be shown to be $\Sigma_2^{\mathsf{P}}$-hard by a proof similar to that of Theorem 1, in which the oracle for the solution of the problem is called not just a single time but once for each value of $k = 1, \ldots, c$.

## 3. Formulating the MW/BSSP as an MILP

Let $u \in \{0,1\}^n$ be the characteristic vector of $U \subseteq V$. We model the MW/BSSP as the following *bilevel programming problem*:

$$\max_{u \in \{0,1\}^n} \quad \sum_{i \in V} x_j^* u_j \tag{2a}$$

$$\text{s.t.} \quad \underbrace{\max_{z \in \{0,1\}^n} \left\{ \sum_{j \in V} u_j z_j : z \in STAB(G) \right\}}_{\alpha(G[U])} \leq k. \tag{2b}$$

In the upper level, the first player (the leader) picks a subset $U \subseteq V$ of vertices by choosing the corresponding characteristic vector $u \in \{0,1\}^n$. In the lower level, the second player (the follower) chooses a maximum-cardinality stable set in $G[U]$ by choosing its characteristic vector $z \in \{0,1\}^{|U|}$ (or, equivalently, by choosing a maximum-weight stable set in $G$ with weight vector $u$). Assuming the perspective of the leader, their choice of $u$ is made in anticipation of the follower's by choosing a set $U$ with $\alpha(G[U]) \leq k$ (as this guarantees the feasibility of their solution) and, among all such sets $U$, choosing one which maximizes $\sum_{j \in U} x_j^*$.

### 3.1. Single-level reformulation of the MW/BSSP

We introduce the following single-level reformulation of the problem:

PROPOSITION 2. *Let $\mathscr{S}$ be the set of all the stable sets of $G$. The following integer program with exponentially-many constraints is a single-level reformulation of (2):*

$$\max_{u \in \{0,1\}^n} \quad \sum_{j \in V} x_j^* u_j \tag{3a}$$

$$\text{s.t.} \sum_{j \in S} u_j \leq k \quad \forall S \in \mathscr{S} : |S| = k+1. \tag{3b}$$

*Proof.* Since $\max_{z \in \{0,1\}^n} \left\{ \sum_{j \in V} u_j z_j : z \in STAB(G) \right\} = \max_{S \in \mathscr{S}} \left\{ \sum_{j \in S} u_j \right\}$, we can rewrite (2b) as $\max_{S \in \mathscr{S}} \left\{ \sum_{j \in S} u_j \right\} \leq k$. As the inequality associated with every $S$ with $|S| \leq k$ is trivially satisfied, we can w.l.o.g. restrict ourselves to $\max_{S \in \mathscr{S} : |S| \geq k+1} \left\{ \sum_{j \in S} u_j \right\} \leq k$. For any $S \subseteq \mathscr{S}$ with $|S| \geq k+1$, $\sum_{j \in S} u_j \leq k$ is violated if and only if $S$ contains a stable set $S' \subseteq \{j \in V : u_j = 1\}$

with $|S'| = k+1$. Thus, we can further restrict ourselves to $\max_{S \in \mathscr{S}:|S|=k+1} \left\{\sum_{j \in S} u_j\right\} \leq k$. The proof is concluded as $\max_{S \in \mathscr{S}:|S|=k+1} \left\{\sum_{j \in S} u_j\right\} \leq k$ holds if and only if (3b) holds. Q.E.D.

Formulation (3) imposes $\alpha(G[U]) \leq k$ by interdicting every set of vertices $U$ forming a stable set $S$ of size $|S| = k+1$. In line with what we discussed before on the complexity of the MW/BSSP, if $k$ is fixed, Formulation (3) contains only a polynomial number of constraints of type (3b).

### 3.2. A tighter family of inequalities

Let $\mathscr{S}_M^{\geq k+1}$ be the collection of all *inclusion-wise maximal* stable sets of $G$ of cardinality at least $k+1$. Consider the inequalities:

$$\sum_{i \in S} u_i \leq k, \quad \forall S \in \mathscr{S}_M^{\geq k+1}. \tag{4}$$

The following holds:

PROPOSITION 3. *Constraints* (4) *dominate Constraints* (3b).

*Proof.* Let $\sum_{i \in S} u_i \leq k$ be a constraint of type (3b). If $S$ is an inclusion-wise maximal stable set, the inequality is also of type (4) and the claim is proven. If $S$ is not maximal, the claim is proven by letting $S'$ be a maximal stable set containing $S$, as $\sum_{i \in S'} u_i \leq k$ dominates $\sum_{i \in S} u_i \leq k$. Q.E.D.

In the remainder of the subsection, we show that Constraints (4) are facet defining for the convex hull of the MW/BSSP formulated in the $x$ space as done in (3).

LEMMA 1. *Let $\tilde{S}$ be a stable set of size $|\tilde{S}| = k+1$. When restricting the MW/BSSP to $G[\tilde{S}]$, i.e., to the subspace where $u_j = 0$ for all $j \in V \setminus \tilde{S}$, the constraint of type* (4) *associated with $\tilde{S}$ (which, in that subspace, is identical to the constraint of type* (3b) *associated with $\tilde{S}$), is facet defining.*

*Proof.* Let $e$ be the all-one vector and $e_j$ be the binary vector with a single one in position $j$. The $k+1$ vectors $\tilde{u}_j := e - e_j$ belong to the face $F := \{u \in \mathbb{R}^{|\tilde{S}|} : \sum_{j \in \tilde{S}} u_j = k\}$ and are affinely independent. Thus, $F$ has dimension $k$ (it is a facet). Q.E.D.

THEOREM 2. *Constraints* (4) *are facet defining for the convex hull of the MW/BSSP.*

*Proof.* For every $S \subseteq V$, let $\mathscr{U}^S$ be the collection of all integer vectors which are feasible for the MW/BSSP when restricted to the subspace where $u_j = 0$ for all $j \in V \setminus S$. Let $\tilde{S}$ be a stable set of size $k+1$ and let $\hat{S} \supseteq \tilde{S}$ be a maximal stable set containing $\tilde{S}$. We employ a sequential lifting argument, lifting first the coefficients with indices in $\hat{S} \setminus \tilde{S}$, and then all those with indices in $V \setminus \hat{S}$.

Due to Lemma 1, the inequality $\sum_{j\in\tilde{S}} u_j \leq k$ is facet defining for $\text{conv}(\mathscr{U}^{\tilde{S}})$. Let $j_1$ be the first index in $\hat{S}\setminus\tilde{S}$ in some (arbitrary) order. Its lifted coefficient is:

$$\lambda_{j_1} = k - \max_{u\in\{0,1\}^{|\tilde{S}|}} \left\{ \sum_{j\in\tilde{S}} u_j : \alpha(G[\{j\in\tilde{S} : u_j = 1\}\cup\{j_1\}]) \leq k \right\}.$$

As $\tilde{S}\cup\{j_1\}$ is a stable set of size $k+1$, $\alpha(G[\tilde{S}\cup\{j_1\}]) = k+1$. Since $\alpha(G[\{j\in\tilde{S} : u_j = 1\}\cup\{j_1\}]) \leq k$ implies that we can have $u_j = 1$ for no more than $k-1$ vertices in $\tilde{S}$, we deduce $\lambda_{j_1} = k-(k-1) = 1$. This shows that $\sum_{j\in\tilde{S}} u_j + u_{j_1} \leq k$ is facet defining for $\text{conv}(\mathscr{U}^{\tilde{S}\cup\{j_1\}})$. Iterating this reasoning for each $j\in\hat{S}\setminus\tilde{S}$, we obtain that the inequality $\sum_{j\in\hat{S}} u_j \leq k$ is facet defining for $\text{conv}(\mathscr{U}^{\hat{S}})$.

We now lift all the vertices in $V\setminus\hat{S}$. Let $j_1'$ be the first vertex in $V\setminus\hat{S}$ in some (arbitrary) order. Its lifted coefficient is:

$$\lambda_{j_1'} = k - \max_{u\in\{0,1\}^{|\hat{S}|}} \left\{ \sum_{j\in\hat{S}} u_j : \alpha(G[\{j\in\hat{S} : u_j = 1\}\cup\{j_1'\}]) \leq k \right\}.$$

Since $\hat{S}$ is maximal and $j_1'\notin\hat{S}$, there is at least an edge in $G$ containing $j_1'$ and a vertex $i\in\hat{S}$. Therefore, $\alpha(G[\{j\in\hat{S} : u_j = 1\}\cup\{j_1'\}]) \leq k$ implies that, setting $u_i = 1$, we can have $u_j = 1$ for a total of $k$ vertices in $\hat{S}$, from which we deduce $\lambda_{j_1'} = k-k = 0$. This shows that $\sum_{j\in\tilde{S}} u_j \leq k$ is facet defining for $\text{conv}(\mathscr{U}^{\hat{S}\cup\{j_1'\}})$. Iterating this reasoning for each $j\in V\setminus\hat{S}$, we obtain that the inequality $\sum_{j\in\hat{S}} u_j \leq k$ is facet defining for $\text{conv}(\mathscr{U}^{V})$, i.e., for the convex hull of all feasible solutions of the MW/BSSP. Q.E.D.

With Constraints (4), we obtain a stronger single-level formulation for the MW/BSSP. We rely on it in the next section to develop a branch-and-cut algorithm for solving the problem.

## 4. Algorithms for Separating R-$k$-RIs by Solving the MW/BSSP

We describe two algorithms for the solution of the MW/BSSP, to be embedded in a cutting plane algorithm for the separation of R-$k$-RIs. The first algorithm is of branch-and-cut type, and it is based on the results of the previous section. The second one is a purely-combinatorial branch-and-bound algorithm which does not rely on mathematical programming techniques. As we will see in the computational results section, the former performs better when $k$ is large, whereas the latter is faster for smaller values of $k$.

### 4.1. Branch-and-cut algorithm

The branch-and-cut algorithm that we propose is based on solving Formulation (3) by separating, rather than Constraints (3b), the tighter Constraints (4). W.l.o.g., we separate such constraints only when an incumbent solution $u^* \in \{0,1\}^n$ is found.

We approach the separation problem in two steps, by first separating a constraint where $S$ has size $|S| \geq k+1$ but is not necessarily maximal and, then, lifting the constraint into one of type (4) *a posteriori* via a greedy algorithm.

For the first step, we need to find a stable set $S$ which has size at least $k+1$ and satisfies $\sum_{j\in S} u^*_j > k$. In principle, this first step requires the solution of a generalization of the MSSP to the case with weights on the vertices and an extra cardinality constraint. Nevertheless, as, with $u^* \in \{0,1\}^n$, $\sum_{j\in S} u^*_j > k$ translates into $|\{j \in S : u^*_j = 1\}| \geq k+1$, which is a tighter constraint than $|S| \geq k+1$, the problem is solved by finding a stable set of cardinality at least $k+1$ in the subgraph induced by $\{j \in V : u^*_j = 1\}$. As, in the worst case, finding such a stable set takes $O(n^{k+1})$, if $k$ is very small, solving this problem is computationally easier than solving the MSSP.

For the second step, after a (not necessarily maximal) stable set $S$ has been found, we can lift the corresponding inequality into a constraint of type (4) by iteratively adding (in an arbitrary order) to $S$ any vertex $j \in V \setminus S$ whose neighborhood has an empty intersection with $S$ (this guarantees that $S$ remains a stable set after $j$ is added to it). The procedure runs in $O(n^2)$.

### 4.2. Combinatorial branch-and-bound algorithm

We now present a combinatorial branch-and-bound algorithm for the solution of the MW/BSSP. The algorithm relies on a binary branching operation by which a vertex $v$ is selected and two children nodes are created, one with $v$ in the solution and one without it. Each node of the branch-and-bound tree is characterized by three subsets: $U$, $P$, and $\mathscr{T}$.

- $U$ is the set of vertices corresponding to the partial solution that has been built from the root node to the current node via branching operations. By construction, the subgraph $G[U]$ induced by $U$ always satisfies the constraint $\alpha(G[U]) \leq k$.
- $P$ is the set of vertices which are candidate for branching; it contains all the vertices $w$ which can be individually added to $U$ so that the stability number of $G[U \cup \{w\}]$ does not exceed $k$. Formally:

$$P := \{w \in V \setminus U : \alpha(G[U \cup \{w\}]) \leq k\}. \tag{5}$$

- $\mathscr{T}$ is the set of all stable sets of $G$ of cardinality $k-1$ containing the whole of $U$ and a subset of vertices in $P$ (that is, for each $S \in \mathscr{T}$, $S \subseteq U \cup P$ and $S \supseteq U$). $\mathscr{T}$ can be computed in $O(n^{k-1})$. We will rely on it for updating $P$.

Globally, we also keep track of the best solution found with the set $B \subseteq V$.

**4.2.1. Updating $P$** Assume branching has just taken place on vertex $v \in P$, which is about to be added to the current solution $U$. We scan all vertices $w \in P$ in order to remove all those which, after adding $v$ to $U$, would violate (5), i.e., all the vertices $w \in P$ such that $\alpha(G[U] \cup \{v,w\}]) > k$. Recall that, since $v,w \in P$, both $\alpha(G \cup \{v\}) \leq k$ and $\alpha(G \cup \{w\}) \leq k$ hold.

Let $R$ be the common anti-neighborhood of $v$ and $w$ in $U$. We consider three conditions, increasingly harder to check computationally. They are designed in such a way that, if any of them is satisfied, we have a certificate that $w$ should remain in $P$ and we can move on to checking the next vertex.

a) Assume $\{v,w\} \in E$. In this case, the increase in stability number due to adding both $v$ and $w$ to $U$ is no larger than that due to adding either $v$ or $w$ individually. As $\alpha(G[U \cup \{v\}]) \leq k$ and $\alpha(G[U \cup \{w\}]) \leq k$ hold, we deduce $\alpha(G[U \cup \{v,w\}]) \leq k$. Hence, $w$ does not violate (5) after $v$ is added to $U$. The condition is checked in $O(1)$.

b) Assume $\{v,w\} \notin E$. Since $v$ and $w$ do not share an edge, they could form a stable set of size up to $|R|+2$ with the vertices in $R$. Therefore, $|R|+2 \leq k$ implies $\alpha(G[U \cup \{v,w\}]) \leq k$ which, in turn, implies that $w$ does not violate (5) after $v$ is added to $U$. The condition is checked in $O(n)$.

c) Assume $\{v,w\} \notin E$ and $|R|+2 > k$. Let $S^* \in \mathscr{T}$ be a stable set maximizing $|R \cap S|$. Adding $v$ and $w$ to $U$ results in a stable set of size $|R \cap S^*|+2$. If $|R \cap S^*|+2 \geq k+1$, $\alpha(G[U \cup \{v,w\}]) \geq k+1$. As $w$ violates (5), we let $P := P \setminus \{w\}$. If $|R \cap S|+2 \leq k$ holds for all $S \in \mathscr{T}$, $w$ does not violate (5) after $v$ is added to $U$. Overall, the condition is checked in $O(n \cdot n^{k-1})$.

See Figure 3 for an illustration of each of the three conditions.

**4.2.2. Description of the algorithm** The algorithm visits the enumeration tree recursively in a depth-first fashion. It starts from the root node with $U := \emptyset$ and $P := V$, and with an empty incumbent $B := \emptyset$ with $LB = 0$. The incumbent is updated by setting $B := U$ whenever either $x^*(U) > x^*(B)$ or $x^*(U) = x^*(B)$ and $|U| > |B|$, as this leads to a denser R-$k$-RI. Besides halting the recursion whenever the candidate set $P$ is empty, we rely on a simple fathoming condition and prune any node where $x^*(U) + x^*(P) < x^*(B)$. If the recursion is not halted, we create a left child node by adding to $U$ a vertex in $P$ of maximum weight (and remove it from $P$), following a *best bound* criterion. Ties

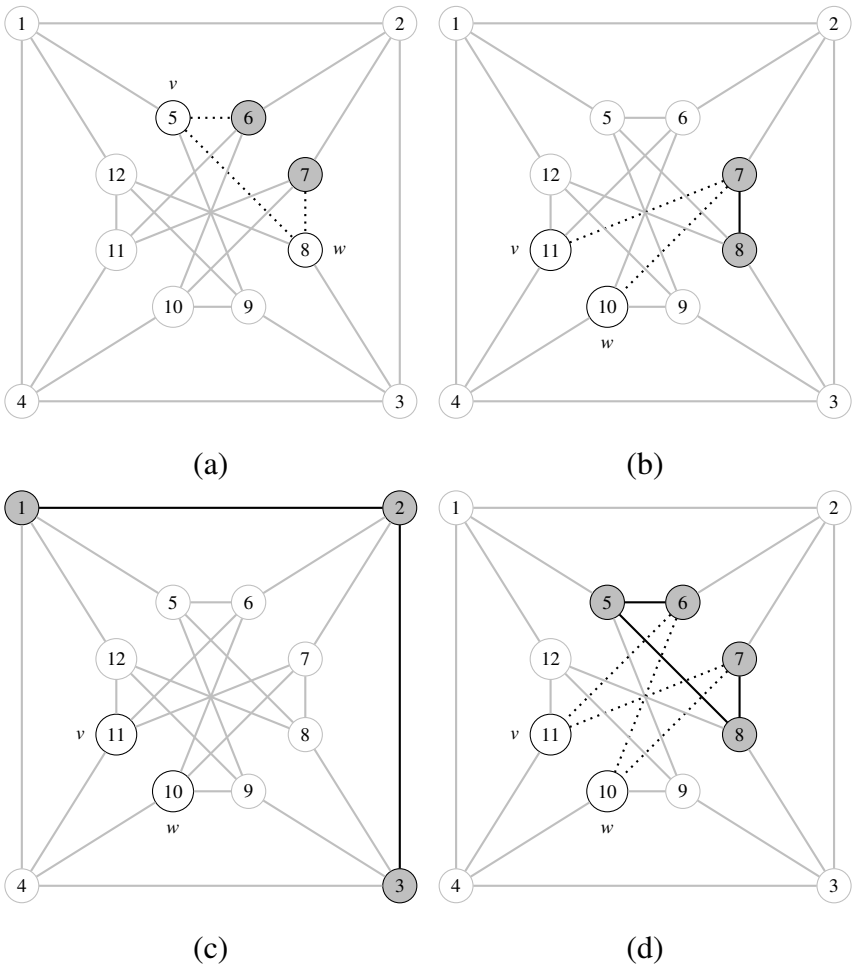


**Figure 3 Illustration of the update of $P$ with $k = 3$.**

*Note.* The vertices $v$ and $w$ are drawn in white and with a solid line. The vertices in $U$ are drawn in gray and with a solid line. The edges of $G[U]$ are drawn with a solid line. The edges belonging to $G[U \cup \{v\} \cup \{w\}]$ but not to $G[U]$ are drawn with a dashed line. (a) A case where $\{v,w\} \in E$ and $w$ is not removed from $P$. (b) A case where $\{v,w\} \notin E$ and $R = \{8\}$. Since $|R|+2 = 3 \leq 3$, $w$ is not removed from $P$. (c) A case where $\{v,w\} \notin E$, $R = \{1,2,3\}$, and $|R|+2 = 5 \geq 3$. $\mathscr{T} = \{\{1,3\}\}$. Since $|R \cap \{1,3\}| = |\{1,3\}| = 2$, we have $|R \cap S^*| \geq k-1 = 2$. Thus, $w$ is removed from $P$. (d) A case where $\{v,w\} \notin E$, $R = \{5,8\}$, and $|R|+2 = 4 \geq 3$. $\mathscr{T} = \{\{5,7\},\{6,7\},\{6,8\}\}$. Since $|R \cap \{5,6\}| = |\{5\}| = 1$, $|R \cap \{6,7\}| = |\{\}| = 0$, and $|R \cap \{6,8\}| = |\{8\}| = 1$, we have $|R \cap S^*| < k-1 = 2$. Thus, $w$ is not removed from $P$.

are broken by choosing a vertex of maximum degree. After updating $P$ according to the previously described procedure and updating the set $\mathscr{T}$ to be consistent with $U := U \cup \{v\}$ and the new $P$, the algorithm is called recursively on the left child node. Stepping out of the recursion, we revert the updates on $U$, $P$, and $\mathscr{T}$ (due to $v$ not being in $U$ anymore) and call the algorithm again on the right child node with partial solution set $U$ and candidate set $P := P \setminus \{v\}$.

## 5. Experimental Setup and Computational Study

In this section, we investigate the tightness of $\bigcap_{k=1}^{\kappa} k\text{-RSTAB}(G)$ for increasing values of $\kappa = 1,\ldots,5$ by relying on the two algorithms we proposed for solving the MW/BSSP. We compare the bounds corresponding to $\bigcap_{k=1}^{\kappa} k\text{-RSTAB}(G)$ to those obtained when optimizing the function $\sum_{j\in N} x_j$ exactly over four different polyhedral LP relaxations for the MSSP (employing an exact separation procedure in all cases except for one—see further) and to the bound corresponding to Lovász's Theta function $\vartheta(G)$. In more detail, the relaxations we consider are:

- The clique polytope $\text{QSTAB}(G)$, defined by the non-negativity constraints and the set of all (maximal) clique inequalities; since every clique inequality coincides with a $k$-RI with $k = 1$, the bound obtained with $\text{QSTAB}(G)$ coincides with the one obtained with $\bigcap_{k=1}^{\kappa} k\text{-RSTAB}(G)$ for $\kappa = 1$.
- The odd-cycle polytope $\text{CSTAB}(G)$, defined by the non-negativity constraints, the edge constraints, and the odd-cycle inequalities $\sum_{j\in C} x_j \leq \frac{|C|-1}{2}$ defined for each $C \subseteq V$ such that the set of edges with both endpoints in it forms a cycle (Grötschel et al. 1988). These inequalities are facet defining only if $G[C]$ is *chordless*—if this is the case, they are called odd-hole inequalities. We solve the corresponding separation problem (exactly) by solving a *shortest odd-cycle problem* following the procedure in Section 8.3.6 and Lemma 9.1.11 of (Grötschel et al. 1988); we adapt the procedure so to maximize, rather than the cut violation, the difference between the cut violation and the cardinality of $C$ multiplied by some $\varepsilon > 0$; if $\varepsilon$ is small enough, by optimizing this objective function one obtains an inequality of maximum violation which contains the smallest number of vertices—as one can show, this suffices to guarantee that $C$ induces a chordless cycle.
- The wheel polytope $\text{WSTAB}(G)$, obtained by further restricting $\text{CSTAB}(G)$ with the wheel inequalities $\sum_{j\in W\setminus\{w\}} x_j + \frac{|W|-2}{2} x_w \leq \frac{|W|-2}{2}$, where $W \subseteq V$ induces a wheel centered at $w \in W$. We carry out the (exact) separation for wheel inequalities via the algorithm reported in Theorem 9.5.6 in (Grötschel et al. 1988), only after checking that all the other inequalities in $\text{CSTAB}(G)$ are satisfied.
- The bound obtained by solving the MSSP with Gurobi 9.1 halting the execution at the root node and deactivating every family of cutting planes besides $\{0,\frac{1}{2}\}$ Chvàtal-Gomory (CG-) cuts, for which we adopt the `aggressive` setting; while it is well-known that the rank-1 closure of the $\{0,\frac{1}{2}\}$ CG cuts coincides with $\text{CSTAB}(G)$ (Campêlo and Cornuéjols 2009), with the aggressive setting Gurobi goes beyond the first closure, often (but not always) obtaining a stronger bound. The bound we obtain is an approximation of the bound that corresponds to the closure (of arbitrary rank) of the $\{0,\frac{1}{2}\}$ Chvàtal-Gomory (CG-) cuts.

- The function $\vartheta(G)$, originally proposed by Lovász (1979) and further studied in a series of papers culminating in (Grötschel et al. 1988); its value coincides with the maximum of $\sum_{j\in V} x_j$ over the Theta body $\mathrm{TH}(G)$, which is defined by the non-negativity constraints and the constraints of type $\sum_{j\in V}(c\,u_j)^2 x_j \leq 1$, where $c \in \mathbb{R}^{\bar{n}}$ with $||c||_2 = 1$ and $(u_j \mid j \in V)$ with $u_j \in \mathbb{R}^{\bar{n}}$ for some $\bar{n} \in \mathbb{N}$ is an *orthonormal representation* of $G$ (i.e., a set of vectors satisfying $||u_j||_2 = 1$ and $u_j u_{j'} = 0$ for all $j, j' \in V$ with $\{j, j'\} \notin E$). The tightness of this bound has been in assessed in many papers, including, e.g., (Dukanovic and Rendl 2007). $\vartheta(G)$ can be computed by semidefinite programming—see (Grötschel et al. 1988); we compute it with the semidefinite programming solver DSDP 5.8 (Benson et al. 2000).

Given an upper bound $UB$, we measure its strength in terms of the multiplicative gap with respect to $\alpha(G)$, defined as $gap := \frac{UB}{\alpha(G)}$. [3]

### 5.1. Algorithm setup

Preliminary experiments show that, for a given $k$, running either of the two separation algorithms we proposed is very time consuming when no violated R-$k$-RI exists. We have also observed that, when that is the case, decreasing $k$ leads to spending a large amount of time in the generation of R-$k$-RIs which do not lead to any bound improvements, whereas, if $k$ is increased, the two algorithms manage to find a bound-improving R-$k$-RI in a much shorter computing time.

The cutting plane method that we adopt circumvents both of the aforementioned issues by employing an adaptive cut-off mechanism. The method starts with $k = 1$ and a violation cut-off $\tau = 0.5$. The separation algorithm it employs (either of the two we proposed) is run until a solution corresponding to a R-$k$-RI of violation $\tau$ is found. If the global upper bound falls below $\tau$, the algorithm is halted and $\tau$ is reduced to the largest value in $\{0.5, 0.25, 0.1, 0.05, 0.01\}$ which is strictly smaller than the current value of $\tau$ and the separation algorithm is run again. If $\tau$ falls below 0.01, it is reset to 0.5 and the value of $k$ is increased by one.

To solve an easier separation problem defined on smaller graphs, we rely on the following property of RIs:

PROPOSITION 4. *Assume that $x^*$ satisfies all edge inequalities. When looking for a violated RI, we can w.l.o.g. restrict ourselves to the subgraph induced by the set of vertices $\{j \in V : 0 < x^*_j < 1\}$.*

[3] While the multiplicative definition of "gap" is not as commonplace as $gap := \frac{UB-\alpha(G)}{\alpha(G)} \cdot 100$, we opt for the multiplicative one as it leads to easier to read tables with fewer decimal figures.

*Proof.* All vertices of index $j$ with $x^*_j = 0$ can be ignored as, having 0 weight in the separation problem, do not contribute to the cut violation. Assume $x^*_{j'} = 1$ for some $j' \in V$. When the LP relaxation of the MSSP contains, at least, all edge inequalities (which is always the case in our setup), $x^*_{j'} = 1$ implies $x^*_j = 0$ for all $j \in V : \{j', j\} \in E$ due to the edge inequality $x^*_{j'} + x^*_j \leq 1$. Therefore, when considering only vertices with $x^*_j > 0$ in the separation procedure, every vertex $j'$ with $x_{j'} = 1$ is isolated. Given any $G[U]$ containing an isolated vertex $j'$, $\alpha(G[U]) = \alpha(G[U \setminus \{j'\}]) + 1$. Thus, the RI $\sum_{j \in U} x_j \leq \alpha(G[U])$ is dominated by the two inequalities $x_{j'} \leq 1$ and $\sum_{j \in U \setminus \{j'\}} x_j \leq \alpha(G[U \setminus \{j'\}])$. This implies that, in the separation problem, any vertex $j'$ with $x^*_{j'} = 1$ can be discarded. Q.E.D.

We remark that the result in Proposition 4 dates back to the work of Hoffman and Padberg (1993), and its applicability goes beyond rank inequalities. We opted for proposing a proof tailored to rank inequalities for the sake of clarity.

Our branch-and-cut algorithm is based on Gurobi 9.1, which we also use for solving the different LP relaxations of the MSSP. We rely on Gurobi's `parallel` setting, using up to 10 threads, while we leave all the other parameters to their default values. In our computations, we carry out the pre-lifting separation of Constraints (4) by resorting to the exact solver `Cliquer` 1.21 proposed by Östergård (2002). Since `Cliquer`, which implements a combinatorial branch-and-bound algorithm, is designed to find cliques rather than stable sets, we apply it to the complement graph $\overline{G}$ of $G$. We halt `Cliquer`'s execution as soon as a stable set of size $k+1$ is found, after which we greedily expand it into a maximal stable set.

The implementation of our branch-and-bound algorithm relies on efficient bit-set data structures and bit-wise operations for storing the graph, the (anti)neighborhoods of the vertices, and the sets $U$, $V$, $\mathscr{T}$, and $B$.

All the results are obtained within a time limit of 7200 seconds using a Dell workstation equipped with an Intel Xeon W-2155 running at 3.3 GHz with 10 cores and 32 GB RAM. All our algorithms are coded in standard ANSI C, compiled with Visual Studio Community 2019, and run in Windows 10. The code and the graph instances are available at the URL `https://github.com/stegua/mss-k-rank`.

### 5.2. Instances

We consider three groups of instances:

1) *Small*: Erdős-Rényi graphs with 50, 75, 100 vertices and edge density in $\{10\%,\dots,90\%\}$, produced with the instance generator Rudy. [4] For each graph dimension, we generate 10 different instances of connected graphs.

These instances are particularly useful to measure the gap closed as a function of the density of the graph, as well as to assess how the upper bounds scale in terms of graph size and density.

2) *Sparse*: very sparse random graphs with up to 400 vertices, with an edge density in $\{1\%,\dots,5\%\}$. They are a subset of the largest instances among those used by Gruber and Rendl (2003) to solve the MSSP via semidefinite programming techniques.

3) *DIMACS*: A subset of the DIMACS dataset (Johnson and Trick 1996), complemented to obtain MSSP instances.

We discard all the instances which correspond to a perfect graph as well all those on which we run out of time while optimizing over the closure of all R-$k$-RIs with $k = 1$ (i.e., of all clique inequalities).

### 5.3. Experimental results: tightness of the bound

We now rely on the methodology that we introduced to investigate the tightness of $\bigcap_{k=1}^{\kappa} k\text{-RSTAB}(G)$ on the three datasets. On each instance, the results that we report are those obtained by employing as separation algorithm the one that allowed for closing the largest portion of the integrality gap within the time limit. For a more thorough discussion on the efficiency of the two separation algorithms, we refer the reader to Subsection 5.4.

Table 1 summarizes the results obtained on a subset of *small* random instances with $n = 50$ vertices reporting precise values of the upper bounds for $\kappa = 1,\dots,5$ as well as the value of $\alpha(G)$. The missing entries are due to the fact that, whenever we achieve a multiplicative gap of 1 (indicating that the bound we have computed coincides with $\alpha(G)$) for a certain value of $\kappa$, the execution of the cutting plane algorithm is halted. The table shows that, on these instances, the bound obtained by optimizing over $\bigcap_{k=1}^{\kappa} k\text{-RSTAB}(G)$ for $\kappa = 5$ is very tight. In particular, by optimizing over $\bigcap_{k=1}^{\kappa} k\text{-RSTAB}(G)$ for $\kappa = 5$ we manage to achieve a multiplicative gap of 1 (which results in $\alpha(G)$ being computed exactly) for 13 instances out of 16. Moreover, in 12 of these instances the solution $x^*$ obtained by optimizing over $\bigcap_{k=1}^{\kappa} k\text{-RSTAB}(G)$ for $\kappa = 5$ is component-wise integer (the last column of the table reports a check mark every time this is the case).

By closely inspecting Table 1, we observe that $\bigcap_{k=1}^{\kappa} k\text{-RSTAB}(G)$ is tighter on instances which are either very sparse or very dense (as, on such instances, we manage to reach a multiplicative

[4] http://web.stanford.edu/~yyye/yyye/Gset

**Table 1 Comparison of Upper Bounds (UB) and multiplicative gaps on the small instances. For no instance the timelimit is incurred. The best values of Upper Bound (UB) and gap are highlighted in boldface. A checkmark in the last column indicates that the instance is solved to optimality.**

| | | QSTAB(G) | | $\bigcap_{k=1}^{\kappa} k$-RSTAB$(G)$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | | $\kappa=1$ | | $\kappa=2$ | | $\kappa=3$ | | $\kappa=4$ | | $\kappa=5$ | | |
| Den | $\alpha(G)$ | UB | gap | UB | gap | UB | gap | UB | gap | UB | gap | Is Integer |
| 10% | 22 | 23.00 | 1.05 | **22.00** | **1.00** | | | | | | | ✓ |
| 15% | 19 | 19.33 | 1.02 | **19.00** | **1.00** | | | | | | | ✓ |
| 20% | 17 | **17.00** | **1.00** | | | | | | | | | ✓ |
| 25% | 13 | 14.52 | 1.12 | 13.47 | 1.04 | 13.14 | 1.01 | **13.00** | **1.00** | | | ✓ |
| 30% | 12 | 12.69 | 1.06 | **12.00** | **1.00** | | | | | | | ✓ |
| 31% | 11 | 12.46 | 1.13 | 11.68 | 1.06 | 11.28 | 1.03 | **11.00** | **1.00** | | | ✓ |
| 32% | 10 | 12.20 | 1.22 | 11.17 | 1.12 | 10.74 | 1.07 | 10.49 | 1.05 | **10.23** | **1.02** | |
| 33% | 10 | 11.92 | 1.19 | 10.89 | 1.09 | 10.47 | 1.05 | 10.15 | 1.02 | **10.00** | **1.00** | ✓ |
| 34% | 10 | 11.63 | 1.16 | 10.65 | 1.07 | 10.21 | 1.02 | **10.00** | **1.00** | | | ✓ |
| 35% | 9 | 11.35 | 1.26 | 10.37 | 1.15 | 9.94 | 1.10 | 9.63 | 1.07 | **9.39** | **1.04** | |
| 40% | 8 | 10.59 | 1.32 | 9.48 | 1.19 | 8.99 | 1.12 | 8.66 | 1.08 | **8.40** | **1.05** | |
| 50% | 7 | 8.73 | 1.25 | 7.86 | 1.12 | 7.38 | 1.05 | 7.07 | 1.01 | **7.00** | **1.00** | |
| 60% | 6 | 7.15 | 1.19 | 6.31 | 1.05 | **6.00** | **1.00** | | | | | ✓ |
| 70% | 5 | 5.65 | 1.13 | 5.02 | 1.00 | **5.00** | **1.00** | | | | | ✓ |
| 80% | 4 | 4.45 | 1.11 | **4.00** | **1.00** | | | | | | | ✓ |
| 90% | 3 | 3.14 | 1.05 | **3.00** | **1.00** | | | | | | | ✓ |
| **Mean:** | | | **1.141** | | **1.056** | | **1.028** | | **1.014** | | **1.007** | |

optimality gap of 1 with values of $\kappa$ smaller than 5). In Sewell (1998), the authors already observed that stable set problems on random graphs tend to be easy for very sparse and very dense problems, and harder at an intermediate level of density. In the context of rank inequalities, this behaviour can be expected for the dense graphs in the dataset as, due to their small $\alpha(G)$ (no larger than 3 for a density of 90%), $\bigcap_{k=1}^{\kappa} k$-RSTAB$(G)$ with $\kappa=5$ contains the RI $\sum_{j\in V} x_j \leq \alpha(G)$ (which, on its own, suffices to achieve a multiplicative gap of 1). This is not the case for the sparse graphs in the dataset, though, for which, due to their stability number $\alpha(G)$ being often very large (up to 22 for a density of 10%), $\bigcap_{k=1}^{\kappa} k$-RSTAB$(G)$ with $\kappa=5$ does not contain the RI $\sum_{j\in V} x_j \leq \alpha(G)$. In spite of this, the experiments suggest that, on these sparse instances, $\kappa=5$, which is fairly small w.r.t. $\alpha(G)$, suffices for $\bigcap_{k=1}^{\kappa} k$-RSTAB$(G)$ to achieve a multiplicative gap very close to 1.

Table 2 summarizes the results obtained on the whole *small* random dataset. We report the average multiplicative gaps (remember that a multiplicative gap equal to 1.00 indicates that the MSSP has been solved to optimality). The gray row reports the average over the graphs of size $n=50,75,100$ with fixed density; the other rows (in white) report the average for each value of the graph size $n$ (10 instances per value of $n$). The very last row reports the overall average over 270 instances for each polyhedral upper bound. From the table, we observe that the $\vartheta(G)$ number achieves strong upper bounds on all the instances, while the bounds corresponding to CSTAB$(G)$ and WSTAB$(G)$ are interesting only for very sparse graphs—notice that, since WSTAB$(G) \subseteq CSTAB(G)$, the bound

**Table 2 Multiplicative gaps with respect to $\alpha(G)$, defined as $gap := \frac{UB}{\alpha(G)}$, achieved by different LP relaxations (lower values are better). Each gray row gives the average over all the instances of the same density $d$. Each row for each value of $n$ reports the average over 10 instances of the same size and density. The very last row report the overall averages over 270 instances for the same LP relaxation.**

| | | | | | | QSTAB(G) | | $\bigcap_{k=1}^{\kappa} k$-RSTAB$(G)$ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $d$ | $n$ | $\vartheta(G)$ | CSTAB(G) | WSTAB(G) | $\{0,\frac{1}{2}\}$-cuts | $\kappa=1$ | $\kappa=2$ | $\kappa=3$ | $\kappa=4$ | $\kappa=5$ |
| 0.1 | | 1.05 | 1.09 | 1.08 | 1.10 | 1.12 | 1.06 | 1.04 | 1.04 | **1.03** |
| | 50 | 1.02 | 1.01 | 1.01 | 1.00 | 1.03 | 1.02 | 1.01 | 1.00 | 1.00 |
| | 75 | 1.05 | 1.08 | 1.07 | 1.09 | 1.13 | 1.06 | 1.03 | 1.02 | 1.02 |
| | 100 | 1.09 | 1.17 | 1.16 | 1.19 | 1.21 | 1.12 | 1.09 | 1.08 | 1.08 |
| 0.2 | | **1.08** | 1.41 | 1.20 | 1.25 | 1.22 | 1.12 | 1.09 | 1.09 | **1.08** |
| | 50 | 1.03 | 1.15 | 1.08 | 1.09 | 1.11 | 1.03 | 1.02 | 1.01 | 1.01 |
| | 75 | 1.08 | 1.42 | 1.20 | 1.26 | 1.22 | 1.12 | 1.09 | 1.07 | 1.05 |
| | 100 | 1.12 | 1.67 | 1.32 | 1.40 | 1.32 | 1.22 | 1.18 | 1.18 | 1.18 |
| 0.3 | | **1.10** | 1.88 | 1.43 | 1.42 | 1.29 | 1.18 | 1.14 | 1.13 | 1.12 |
| | 50 | 1.06 | 1.49 | 1.18 | 1.21 | 1.18 | 1.09 | 1.05 | 1.03 | 1.02 |
| | 75 | 1.08 | 1.87 | 1.40 | 1.40 | 1.28 | 1.16 | 1.12 | 1.08 | 1.07 |
| | 100 | 1.15 | 2.27 | 1.70 | 1.64 | 1.41 | 1.29 | 1.26 | 1.26 | 1.26 |
| 0.4 | | **1.11** | 2.40 | 1.80 | 1.56 | 1.33 | 1.21 | 1.16 | 1.15 | 1.14 |
| | 50 | 1.05 | 1.80 | 1.35 | 1.26 | 1.17 | 1.08 | 1.04 | 1.02 | 1.01 |
| | 75 | 1.10 | 2.41 | 1.81 | 1.55 | 1.32 | 1.20 | 1.14 | 1.10 | 1.09 |
| | 100 | 1.18 | 2.98 | 2.24 | 1.85 | 1.49 | 1.35 | 1.33 | 1.33 | 1.33 |
| 0.5 | | **1.08** | 2.89 | 2.17 | 1.59 | 1.30 | 1.17 | 1.12 | 1.11 | 1.11 |
| | 50 | 1.04 | 2.20 | 1.65 | 1.29 | 1.17 | 1.07 | 1.03 | 1.01 | 1.01 |
| | 75 | 1.07 | 2.92 | 2.19 | 1.62 | 1.30 | 1.16 | 1.09 | 1.03 | 1.03 |
| | 100 | 1.12 | 3.56 | 2.67 | 1.87 | 1.44 | 1.29 | 1.29 | 1.29 | 1.29 |
| 0.6 | | **1.07** | 3.62 | 2.71 | 1.69 | 1.30 | 1.15 | 1.10 | 1.09 | 1.09 |
| | 50 | 1.02 | 2.74 | 2.05 | 1.38 | 1.16 | 1.04 | 1.00 | 1.00 | 1.00 |
| | 75 | 1.07 | 3.59 | 2.69 | 1.69 | 1.28 | 1.13 | 1.06 | 1.01 | 1.00 |
| | 100 | 1.13 | 4.52 | 3.39 | 2.00 | 1.45 | 1.28 | 1.27 | 1.27 | 1.27 |
| 0.7 | | **1.05** | 4.35 | 3.26 | 1.68 | 1.24 | 1.10 | 1.07 | 1.07 | 1.06 |
| | 50 | 1.02 | 3.17 | 2.38 | 1.32 | 1.11 | 1.02 | 1.00 | 1.00 | 1.00 |
| | 75 | 1.04 | 4.33 | 3.25 | 1.67 | 1.24 | 1.07 | 1.02 | 1.01 | 1.00 |
| | 100 | 1.10 | 5.56 | 4.17 | 2.06 | 1.38 | 1.19 | 1.19 | 1.19 | 1.19 |
| 0.8 | | 1.02 | 5.40 | 4.05 | 1.67 | 1.17 | 1.03 | 1.02 | **1.01** | **1.01** |
| | 50 | 1.02 | 4.17 | 3.13 | 1.37 | 1.12 | 1.01 | 1.00 | 1.00 | 1.00 |
| | 75 | 1.03 | 5.38 | 4.03 | 1.68 | 1.14 | 1.03 | 1.01 | 1.00 | 1.00 |
| | 100 | 1.02 | 6.67 | 5.00 | 1.97 | 1.23 | 1.04 | 1.04 | 1.02 | 1.02 |
| 0.9 | | 1.03 | 6.98 | 5.24 | 1.63 | 1.08 | 1.01 | **1.00** | **1.00** | **1.00** |
| | 50 | 1.03 | 5.42 | 4.06 | 1.26 | 1.10 | 1.00 | 1.00 | 1.00 | 1.00 |
| | 75 | 1.04 | 7.08 | 5.31 | 1.65 | 1.08 | 1.03 | 1.00 | 1.00 | 1.00 |
| | 100 | 1.01 | 8.44 | 6.33 | 1.96 | 1.06 | 1.01 | 1.00 | 1.00 | 1.00 |
| **Mean:** | | **1.07** | **3.34** | **2.55** | **1.51** | **1.23** | **1.12** | **1.08** | **1.07** | **1.07** |

obtained with the former is always at least as tight as the bound obtained with the latter. The last five columns report the upper bound obtained with $k$-rank inequalities. Recall that $k$-RIs with $k=1$ coincide with clique inequalities and, therefore, the bound obtained with $\kappa=1$ coincides with that of QSTAB$(G)$.

**Table 3 Comparison of Upper Bounds (UB) and multiplicative gaps on the sparse dataset. The best Upper Bound and gaps are highlighted in boldface. Results obtained at the time limit are highlighted in italics.**

| | | | | | | | | | QSTAB(G) | | $\bigcap_{k=1}^{\kappa} k$-RSTAB$(G)$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | | | $\vartheta(G)$ | | CSTAB(G) | | $\{0,\frac{1}{2}\}$-cuts | | $\kappa=1$ | | $\kappa=2$ | | $\kappa=3$ | | $\kappa=4$ | |
| Name | Den | $\alpha$ | $\vartheta(G)$ | gap | UB | gap | UB | gap | UB | gap | UB | gap | UB | gap | UB | gap |
| g150.4 | 4 | 59 | 61.80 | 1.05 | 61.34 | 1.04 | 62.32 | 1.06 | 67.00 | 1.14 | 62.08 | 1.05 | 60.80 | 1.03 | ***60.21*** | ***1.02*** |
| g150.5 | 5 | 55 | 58.73 | 1.07 | 58.52 | 1.06 | 61.06 | 1.11 | 64.00 | 1.16 | 58.56 | 1.06 | 57.69 | 1.05 | ***57.27*** | ***1.04*** |
| g170.3 | 3 | 71 | 73.34 | 1.03 | 72.05 | 1.01 | 73.32 | 1.03 | 78.50 | 1.11 | 73.53 | 1.04 | 72.16 | 1.02 | ***71.60*** | ***1.01*** |
| g200.3 | 3 | 83 | 86.52 | 1.04 | 85.01 | 1.02 | 87.38 | 1.05 | 94.50 | 1.14 | 86.61 | 1.04 | 84.97 | 1.02 | ***84.65*** | ***1.02*** |
| g200.2 | 2 | 96 | 97.17 | 1.01 | **96.00** | **1.00** | **96.00** | **1.00** | 100.00 | 1.04 | 97.00 | 1.01 | **96.00** | **1.00** | | |
| g300.2 | 2 | 122 | 129.47 | 1.06 | **127.60** | **1.05** | 131.64 | 1.08 | 141.00 | 1.16 | 130.43 | 1.07 | *127.81* | *1.05* | | |
| g350.2 | 2 | 133 | **143.43** | **1.08** | 143.91 | 1.08 | 149.47 | 1.12 | 161.00 | 1.21 | 146.11 | 1.10 | *144.02* | *1.08* | | |
| g400.1 | 1 | 191 | 194.79 | 1.02 | **191.60** | **1.00** | 192.53 | 1.01 | 201.50 | 1.05 | 195.50 | 1.02 | 193.50 | 1.01 | *193.00* | *1.01* |
| | | **Mean:** | | **1.045** | | **1.034** | | **1.057** | | **1.125** | | **1.049** | | **1.033** | | **1.029** |

The computational results in Table 2 clearly show that the contribution of RI with a small RHS in terms of gap is very significant. Their bound is clearly superior to the bound obtained with any of the polyhedral relaxations we consider, and substantially tighter than the bound obtained with QSTAB$(G)$ (or, equivalently, with RIs with $k=1$). The most substantial differences are observed when comparing $\bigcap_{k=1}^{\kappa} k$-RSTAB$(G)$ with CSTAB$(G)$ and WSTAB$(G)$ as the graph size and density increases. We notice that the bounds obtained with $\bigcap_{k=1}^{\kappa} k$-RSTAB$(G)$ are very close to $\vartheta(G)$ in a number of cases, and strictly better in roughly half of them. As already observed in Table 1, we observe a sort of *phase transition* phenomenon for densities between 0.3 and 0.5, where the bounds for, in particular, graphs with $n=100$ nodes, are larger than for other densities and graph sizes. We remark that the multiplicative gaps we report for $\bigcap_{k=1}^{\kappa} k$-RSTAB$(G)$ are exact only for the graphs of size $n=50$ (see Table 5 for the number of timeouts). Indeed, for $n=75$ and $\kappa=4,5$ and for $n=100$ and $\kappa=2$, the time limit is hit for many instances and, hence, the bounds we report constitute an estimation on the bound of the corresponding closure $\bigcap_{k=1}^{\kappa} k$-RSTAB$(G)$ (which is likely to be tighter).

Table 3 shows the results obtained on the *sparse* dataset. The results obtained when the time limit is reached are reported in *italics*. The table shows that, due to the size of these graphs and their very small density, the running time required to separate R-$k$-RIs can be substantial (see Subsection 5.4 for a more thorough discussion on computing times and efficiency), causing the cutting plane algorithm to reach the time limit already for $\kappa=3$ on `g300.2` and `g350.2` and for $\kappa=4$ for all the other graphs. In spite of this, by optimizing over $\bigcap_{k=1}^{\kappa} k$-RSTAB$(G)$ with $\kappa\leq 4$ we obtain bounds tighter than $\vartheta(G)$ on 7 instances out of 8, and we close the optimality gap on `g200.2`. We observe that CSTAB$(G)$ is very tight on these very sparse instances, and that it always outperforms QSTAB$(G)$ in terms of gap (since these graphs are very sparse, they only rarely contain a wheel and, thus, the bounds obtained with WSTAB$(G)$ and CSTAB$(G)$ coincide). Moreover, for

**Table 4 Comparison of Upper Bound (UB) and multiplicative gaps on the DIMACS instances. The best UB and gap is highlighted in boldface. Results obtained at the time limit are highlighted in italics.**

| | | | | | | | | | | | | QSTAB(G) | | $\bigcap_{k=1}^{\kappa} k$-RSTAB$(G)$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | | | | $\vartheta(G)$ | | CSTAB(G) | | WSTAB(G) | | $\{0,\frac{1}{2}\}$-cuts | | $\kappa=1$ | | $\kappa=2$ | | $\kappa=3$ | |
| Name | $n$ | Den | $\alpha$ | $\vartheta(G)$ | gap | UB | gap | UB | gap | UB | gap | UB | gap | UB | gap | UB | gap |
| brock200_1 | 200 | 25% | 21 | 27.46 | **1.31** | 66.67 | 3.17 | 50.00 | 2.38 | 47.00 | 2.24 | 38.02 | 1.81 | 34.75 | 1.65 | 34.63 | *1.65* |
| brock200_2 | 200 | 50% | 12 | 14.23 | **1.19** | 66.67 | 5.56 | 50.00 | 4.17 | 31.44 | 2.62 | 21.13 | *1.76* | 21.13 | *1.76* | 21.13 | *1.76* |
| brock200_3 | 200 | 39% | 15 | 18.82 | **1.25** | 66.67 | 4.44 | 50.00 | 3.33 | 37.45 | 2.50 | 27.23 | 1.82 | 24.97 | *1.66* | 24.97 | *1.66* |
| brock200_4 | 200 | 34% | 17 | 21.29 | **1.25** | 66.67 | 3.92 | 50.00 | 2.94 | 41.15 | 2.42 | 30.63 | 1.80 | 27.99 | *1.65* | 27.99 | *1.65* |
| brock400_1 | 400 | 25% | 27 | 39.70 | **1.47** | 133.33 | 4.94 | 100.00 | 3.70 | 88.28 | 3.27 | 63.90 | 2.37 | 61.52 | *2.28* | 61.52 | *2.28* |
| brock400_2 | 400 | 25% | 29 | 39.56 | **1.36** | 133.33 | 4.60 | 100.00 | 3.45 | 91.20 | 3.14 | 64.27 | 2.22 | 61.59 | *2.12* | 61.59 | *2.12* |
| brock400_3 | 400 | 25% | 31 | 39.48 | **1.27** | 133.33 | 4.30 | 100.00 | 3.23 | 88.05 | 2.84 | 64.12 | 2.07 | 61.68 | *1.99* | 61.68 | *1.99* |
| brock400_4 | 400 | 25% | 33 | 39.70 | **1.20** | 133.33 | 4.04 | 100.00 | 3.03 | 90.38 | 2.74 | 64.17 | 1.94 | 61.40 | *1.86* | 61.40 | *1.86* |
| C125.9 | 125 | 10% | 34 | 37.81 | **1.11** | 43.00 | 1.26 | 41.53 | 1.22 | 43.55 | 1.28 | 43.06 | 1.27 | 39.75 | 1.17 | 38.71 | *1.14* |
| C250.9 | 250 | 10% | 44 | 56.24 | **1.28** | 83.33 | 1.89 | 70.22 | 1.60 | 76.00 | 1.73 | 71.38 | 1.62 | 65.77 | 1.49 | 65.10 | *1.48* |
| C500.9 | 500 | 10% | 57 | 84.19 | **1.48** | 166.67 | 2.92 | 125.00 | 2.19 | 140.10 | 2.46 | 122.95 | 2.16 | 114.22 | 2.00 | 114.17 | *2.00* |
| hamming6-4 | 64 | 65% | 4 | 5.33 | 1.33 | 21.33 | 5.33 | 16.00 | 4.00 | 6.48 | 1.62 | 5.33 | 1.33 | 4.00 | **1.00** | 4.00 | 1.00 |
| keller4 | 171 | 35% | 11 | 14.01 | 1.27 | 57.00 | 5.18 | 42.75 | 3.89 | 18.92 | 1.72 | 14.83 | 1.35 | 13.79 | **1.25** | 13.79 | *1.25* |
| MANN_09 | 45 | 93% | 16 | 17.47 | 1.09 | 18.00 | 1.13 | 18.00 | 1.13 | 17.25 | **1.08** | 18.00 | *1.13* | 18.00 | *1.13* | 18.00 | *1.13* |
| p_hat300-2 | 300 | 51% | 25 | 26.96 | **1.08** | 100.00 | 4.00 | 75.00 | 3.00 | 50.01 | 2.00 | 33.58 | 1.34 | 31.46 | *1.26* | 31.46 | *1.26* |
| p_hat300-3 | 300 | 26% | 36 | 41.17 | **1.14** | 100.00 | 2.78 | 75.00 | 2.08 | 68.42 | 1.90 | 54.31 | 1.51 | 50.53 | *1.40* | 50.53 | *1.40* |
| sanr200_0.7 | 200 | 30% | 18 | 23.84 | **1.32** | 66.67 | 3.70 | 50.00 | 2.78 | 45.26 | 2.51 | 33.34 | 1.85 | 30.81 | *1.71* | 30.77 | *1.71* |
| sanr200_0.9 | 200 | 10% | 45 | 49.27 | **1.09** | 66.67 | 1.48 | 58.98 | 1.31 | 63.79 | 1.42 | 59.82 | 1.33 | 55.10 | 1.22 | 54.27 | *1.21* |
| sanr400_0.7 | 400 | 30% | 22 | 34.27 | **1.56** | 133.33 | 6.06 | 100.00 | 4.55 | 79.29 | 3.60 | 56.90 | 2.59 | 56.90 | *2.59* | 56.90 | *2.59* |
| **Mean:** | | | | | **1.26** | | **3.35** | | **2.63** | | **2.16** | | **1.71** | | **1.591** | | **1.586** |

graphs with density smaller or equal to 2, it achieves the best upper bounds (except for `gr350.2`). The bound obtained with $\bigcap_{k=1}^{\kappa} k$-RSTAB$(G)$ improves on the bound obtained with CSTAB$(G)$ for graphs with higher density. This is expected, as only for graphs with a not too small density we can expect to find rank inequalities which, while having a small right-hand side (by construction), feature a dense left-hand side and can improve over classical inequalities where $G[U]$ has a special structure.

Table 4 shows the results on the *DIMACS* instances. With only two exceptions (`hamming6-4` and `keller4`), on which we obtain bounds better than $\vartheta(G)$ already for $\kappa=2$, the bound obtained by optimizing over $\bigcap_{k=1}^{\kappa} k$-RSTAB$(G)$ in the time limit is no tighter than $\vartheta(G)$, but it is always significantly better than the upper bounds obtained with any other LP relaxation. We remark that, while the bounds corresponding to QSTAB$(G)$, CSTAB$(G)$, and WSTAB$(G)$ are exact, those corresponding to $\bigcap_{k=1}^{\kappa} k$-RSTAB$(G)$ with $\kappa \geq 2$ are not as, when computing them, the time limit is reached already for $\kappa=2$ for almost all the instances. This prevents us from drawing any meaningful conclusions on the tightness of $\bigcap_{k=1}^{\kappa} k$-RSTAB$(G)$ with a small $\kappa$ on this dataset, besides showing that the separation problem of R-$k$-RIs (the MW/BSSP) is challenging to solve to optimality on larger graphs.

### 5.4. Efficiency of the separation algorithms

We now compare the efficiency of the branch-and-cut and branch-and-bound algorithms presented in Section 3.

Table 5 reports the average computing time (in seconds) over 10 instances of the same size and density for the two separation algorithms on the *small* datasets. For each value of $k$, the table reports the time spent separating R-$k$-RIs with that value of $k$ only and the number of instances that reach the timeout (columns `To`). The time limit is reached when the total time spent separating R-$k$-RIs with $k$ up to $\kappa$ is exceeded. The average computing time is computed by only taking into account the instances for which the time limit is not reached.

Table 5 shows that, on the *small* dataset, the branch-and-cut algorithm is faster for $k = 4,5$ than the branch-and-bound algorithm, which is one order of magnitude faster for $k = 1,2$. This is well expected as, by relying on the complete enumeration of all stable sets of size $k-1$, the branch-and-bound algorithm is more efficient when $k$ is small. The table also shows that for $k = 1,2$ the branch-and-bound algorithm is two-to-three orders of magnitude faster than the branch-and-cut algorithm. The latter is still faster for $k = 3$, but the speedup it achieves is substantially smaller than the one that is observed for smaller values of $k$. This suggests that, if we were to increase $k$ above 3, we would most likely witness a situation similar to the one we observed on the *small* dataset, where the branch-and-bound algorithm becomes inefficient for larger values of $k$. This further shows that the separation problem of R-$k$-RIs (the MW/BSSP) is very challenging to solve to optimality as the size of the graph and its density increase.

Table 6 reports, for the *small* and *sparse* datasets and for increasing densities, the number of R-$k$-RIs generated during the execution of the cutting plane algorithm and the average cardinality of $U$. As the table shows, on the *sparse* dataset and for each value of $k$, $|U|$ is almost constant and it grows very slowly with the density of the graph, whereas, on the *small* dataset (which contains denser graphs), it grows much faster with the graph density, at a speed which gets larger for increasing values of $k$. The table also shows that, while the number of inequalities generated for each instance is almost constant for the different values of $k$ on the *small* dataset, it increases quite rapidly with $k$ on the *sparse* dataset. This suggests that, on the *sparse* dataset, not only a single instance of the separation problem is, in general, harder to solve, but also that more separation problems have to be solved, further explaining why, on this dataset, the time limit is reached more often.

Figure 4 reports, as a function of the number of R-$k$-RIs that we generated (first subplot) and of the running time (second subplot), the evolution of the upper bound and the value of $k$ during a single execution of our cutting plane algorithm on an instance taken from the *small* dataset (featuring 50 vertices and an edge density of 33%). The branch-and-cut algorithm is employed as the separation routine. The figure shows that, while the number of violated inequalities is almost independent of $k$

**Table 5 Comparison of branch-and-bound (BnB) and branch-and-cut (BnC) separation w.r.t. average runtime (in seconds) and number of timeouts (To) when separating R-$k$-RIs with different values of $k$. Each row reports the average over 10 instances. Total timeouts: 106 for BnC and 138 for BnB.**

| | | QSTAB(G) | | | | $\bigcap_{k=1}^{\kappa} k$-RSTAB($G$) | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | | $\kappa=1$ | | | | $\kappa=2$ | | | | $\kappa=3$ | | | | $\kappa=4$ | | | | $\kappa=5$ | | | |
| $n$ | Den | BnC | To | BnB | To | BnC | To | BnB | To | BnC | To | BnB | To | BnC | To | BnB | To | BnC | To | BnB | To |
| 50 | 10% | 0.9 | 0 | 0 | 0 | 1.7 | 0 | 0 | 0 | 3.2 | 0 | 0.2 | 0 | 4.3 | 0 | 1.2 | 0 | 8.5 | 0 | 11.6 | 0 |
| 50 | 20% | 3.7 | 0 | 0 | 0 | 13.5 | 0 | 0.1 | 0 | 23.9 | 0 | 1.6 | 0 | 43.9 | 0 | 14.2 | 0 | 97.8 | 0 | 210.8 | 0 |
| 50 | 30% | 5.1 | 0 | 0 | 0 | 19.7 | 0 | 0.2 | 0 | 56.7 | 0 | 17.9 | 0 | 112.9 | 0 | 197.6 | 0 | 182.4 | 0 | 645.4 | 1 |
| 50 | 40% | 6.4 | 0 | 0.1 | 0 | 28.6 | 0 | 0.4 | 0 | 66.8 | 0 | 13.9 | 0 | 126.4 | 0 | 184.4 | 0 | 160.5 | 0 | 258.1 | 0 |
| 50 | 50% | 8.2 | 0 | 0.1 | 0 | 30.7 | 0 | 0.5 | 0 | 55.1 | 0 | 9.2 | 0 | 128.7 | 0 | 85.7 | 0 | 141.1 | 0 | 96.2 | 0 |
| 50 | 60% | 11.3 | 0 | 0.1 | 0 | 39 | 0 | 1.2 | 0 | 56.1 | 0 | 7.8 | 0 | 75.8 | 0 | 24.1 | 0 | | | | |
| 50 | 70% | 10.8 | 0 | 0.1 | 0 | 26.1 | 0 | 0.5 | 0 | 46 | 0 | 2.1 | 0 | 61.3 | 0 | 8.2 | 0 | | | | |
| 50 | 80% | 12.4 | 0 | 0.1 | 0 | 21.1 | 0 | 0.3 | 0 | 27.6 | 0 | 0.6 | 0 | | | | | | | | |
| 50 | 90% | 5.3 | 0 | 0.1 | 0 | 9.9 | 0 | 0.1 | 0 | 12 | 0 | 0.1 | 0 | | | | | | | | |
| 75 | 10% | 3.2 | 0 | 0 | 0 | 20.4 | 0 | 0.2 | 0 | 89.9 | 0 | 21.3 | 0 | 310.8 | 0 | 829.8 | 0 | 646.3 | 0 | 1703.6 | 7 |
| 75 | 20% | 10.7 | 0 | 0.1 | 0 | 70.3 | 0 | 0.6 | 0 | 468.2 | 0 | 278.9 | 0 | 1655.6 | 0 | 6437.5 | 9 | 4578.8 | 6 | 0 | 1 |
| 75 | 30% | 14.3 | 0 | 0.1 | 0 | 178.5 | 0 | 1.9 | 0 | 876.5 | 0 | 1638.4 | 0 | 3994 | 0 | 0 | 10 | 3257.4 | 9 | | |
| 75 | 40% | 19.5 | 0 | 0.2 | 0 | 317.4 | 0 | 6.2 | 0 | 1498.8 | 0 | 4993.4 | 5 | 3837.4 | 7 | 0 | 5 | 0 | 3 | | |
| 75 | 50% | 27.1 | 0 | 0.3 | 0 | 436.3 | 0 | 25.1 | 0 | 1663 | 0 | 0 | 10 | 2102.5 | 5 | | | | | | |
| 75 | 60% | 35.7 | 0 | 0.4 | 0 | 463.5 | 0 | 119.2 | 0 | 1185 | 0 | 80.4 | 9 | 1483.1 | 0 | | | 1479.1 | 1 | | |
| 75 | 70% | 45.2 | 0 | 0.5 | 0 | 358.7 | 0 | 417.3 | 0 | 415.9 | 0 | 458.9 | 2 | 739.3 | 0 | | | 739.6 | 0 | | |
| 75 | 80% | 50.9 | 0 | 0.9 | 0 | 82.7 | 0 | 25.6 | 0 | 217.3 | 0 | 445.1 | 0 | 204.8 | 0 | 260.4 | 0 | | | | |
| 75 | 90% | 24.6 | 0 | 0.3 | 0 | 49.5 | 0 | 0.8 | 0 | 67.4 | 0 | 1.2 | 0 | | | | | | | | |
| 100 | 10% | 31.5 | 0 | 0.1 | 0 | 321.4 | 0 | 0.7 | 0 | 2035.1 | 0 | 298.7 | 0 | 0 | 10 | 0 | 10 | | | | |
| 100 | 20% | 72.4 | 0 | 0.2 | 0 | 1558.1 | 0 | 3.4 | 0 | 0 | 10 | 5077.1 | 0 | | | 0 | 10 | | | | |
| 100 | 30% | 98.6 | 0 | 0.4 | 0 | 2532.9 | 0 | 16.9 | 0 | 0 | 10 | 0 | 10 | | | | | | | | |
| 100 | 40% | 145.6 | 0 | 0.5 | 0 | 4956.5 | 0 | 97.8 | 0 | 0 | 10 | 0 | 10 | | | | | | | | |
| 100 | 50% | 270.7 | 0 | 0.7 | 0 | 0 | 10 | 787.5 | 0 | | | 0 | 10 | | | | | | | | |
| 100 | 60% | 387.7 | 0 | 1.3 | 0 | 3965.8 | 8 | 5856.3 | 6 | 0 | 2 | 0 | 4 | | | | | | | | |
| 100 | 70% | 405.4 | 0 | 2.4 | 0 | 0 | 10 | 0 | 10 | | | | | | | | | | | | |
| 100 | 80% | 628.8 | 0 | 6.7 | 0 | 3271 | 5 | 0 | 9 | 3538.5 | 0 | | | | | | | | | | |
| 100 | 90% | 309.3 | 0 | 14.3 | 0 | 449.2 | 0 | 44.1 | 0 | 995.2 | 0 | 111.9 | 0 | | | | | | | | |
| **Total Timeouts:** | | **0** | | **0** | | **33** | | **25** | | **32** | | **60** | | **22** | | **44** | | **19** | | **9** | |

for this instance, the time needed for their separation more than doubles whenever $k$ is increased by one. At the same time, though, it shows that the bound improvement that is obtained when $k$ is increased by one is quite substantial, especially immediately after the increase has taken place.

## 6. Concluding Remarks

We have proposed a methodology for optimizing over the closure of all rank inequalities with a right-hand side no larger than a small constant that relies on the separation of a relaxation of rank inequalities which we called relaxed $k$-rank inequalities. We have investigated the corresponding separation problem (which turns out to be an interesting bilevel programming problem), its computational complexity, and its polyhedral structure. For its solution, we have proposed a branch-and-cut method suitable for large values of $k$ (it relies on the separation of a family of facet-defining inequalities introduced in this paper) and a combinatorial branch-and-cut algorithm which turns out to be more efficient when $k$ is small. Overall, our computational experiments suggest that the closure of all rank inequalities with a right-hand side no larger than 5 can be, not in general but in a number of cases, tighter than Lovász's Theta function $\vartheta(G)$, and that it is often tighter than the bound obtained

**Table 6 Statistics of the generated cuts reporting the number of generated inequalities and the average cardinality of $U$ for the small and sparse dataset for different values of $k$. For the random instances, each row reports results averaged over 10 instances.**

| | | | QSTAB(G) | | $\bigcap_{k=1}^{\kappa} k$-RSTAB$(G)$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | | | $k=1$ | | $\kappa=2$ | | $\kappa=3$ | | $\kappa=4$ | | $\kappa=5$ | |
| Name | $n$ | Den | cuts | $\lvert U\rvert$ | cuts | $\lvert U\rvert$ | cuts | $\lvert U\rvert$ | cuts | $\lvert U\rvert$ | cuts | $\lvert U\rvert$ |
| r-50-0.1 | 50 | 10% | 11 | 3.0 | 5 | 5.1 | 6 | 7.4 | 3 | 9.5 | 6 | 12.6 |
| r-50-0.2 | 50 | 20% | 58 | 3.2 | 67 | 6.5 | 15 | 10.1 | 23 | 13.3 | 62 | 17.3 |
| r-50-0.3 | 50 | 30% | 75 | 3.7 | 80 | 8.1 | 89 | 12.6 | 57 | 17.3 | 38 | 21.7 |
| r-50-0.4 | 50 | 40% | 91 | 4.5 | 105 | 10.0 | 82 | 15.7 | 95 | 21.4 | 49 | 27.5 |
| r-50-0.5 | 50 | 50% | 112 | 5.3 | 101 | 12.2 | 46 | 19.1 | 157 | 27.2 | 21 | 35.3 |
| r-50-0.6 | 50 | 60% | 138 | 6.5 | 140 | 15.3 | 34 | 24.6 | 7 | 33.6 | | |
| r-50-0.7 | 50 | 70% | 152 | 7.9 | 37 | 13.9 | 29 | 27.3 | 2 | 38.0 | | |
| r-50-0.8 | 50 | 80% | 189 | 10.1 | 73 | 24.1 | 19 | 34.2 | | | | |
| r-50-0.9 | 50 | 90% | 115 | 11.7 | 100 | 28.5 | 7 | 36.1 | | | | |
| r-75-0.1 | 75 | 10% | 37 | 3.0 | 74 | 5.5 | 62 | 8.3 | 85 | 11.2 | 75 | 14.2 |
| r-75-0.2 | 75 | 20% | 111 | 3.4 | 130 | 7.4 | 147 | 11.6 | 171 | 15.8 | 201 | 19.8 |
| r-75-0.3 | 75 | 30% | 137 | 4.2 | 159 | 9.4 | 193 | 14.7 | 274 | 20.3 | 113 | 24.7 |
| r-75-0.4 | 75 | 40% | 167 | 5.2 | 186 | 11.7 | 255 | 18.5 | 352 | 25.2 | 1 | 34.0 |
| r-75-0.5 | 75 | 50% | 205 | 6.3 | 229 | 14.6 | 315 | 23.6 | 129 | 32.6 | | |
| r-75-0.6 | 75 | 60% | 245 | 7.7 | 272 | 18.4 | 218 | 30.3 | 99 | 42.8 | 4 | 54.5 |
| r-75-0.7 | 75 | 70% | 302 | 9.6 | 339 | 23.5 | 26 | 38.4 | 342 | 56.6 | 7 | 72.3 |
| r-75-0.8 | 75 | 80% | 365 | 12.7 | 39 | 31.4 | 285 | 52.7 | 24 | 66.6 | | |
| r-75-0.9 | 75 | 90% | 275 | 16.5 | 81 | 31.8 | 136 | 68.9 | | | | |
| r-100-0.1 | 100 | 10% | 86 | 3.0 | 114 | 5.8 | 147 | 8.9 | 150 | 11.9 | | |
| r-100-0.2 | 100 | 20% | 156 | 3.7 | 205 | 8.0 | 134 | 12.4 | | | | |
| r-100-0.3 | 100 | 30% | 198 | 4.6 | 268 | 10.1 | 157 | 16.4 | | | | |
| r-100-0.4 | 100 | 40% | 244 | 5.7 | 312 | 12.8 | 65 | 20.0 | | | | |
| r-100-0.5 | 100 | 50% | 294 | 6.9 | 339 | 16.0 | | | | | | |
| r-100-0.6 | 100 | 60% | 358 | 8.6 | 470 | 20.3 | 172 | 32.9 | | | | |
| r-100-0.7 | 100 | 70% | 445 | 11.0 | 483 | 27.6 | | | | | | |
| r-100-0.8 | 100 | 80% | 546 | 14.6 | 657 | 37.4 | 9 | 60.3 | | | | |
| r-100-0.9 | 100 | 90% | 611 | 20.8 | 6 | 51.2 | 163 | 94.0 | | | | |
| g400.1 | 400 | 1% | 6 | 3.3 | 32 | 5.1 | 64 | 7.0 | *6* | *9.2* | | |
| g200.2 | 200 | 2% | 6 | 3.3 | 23 | 5.1 | 23 | 7.1 | | | | |
| g300.2 | 300 | 2% | 25 | 3.1 | 143 | 5.0 | *130* | *7.1* | | | | |
| g350.2 | 350 | 2% | 37 | 3.1 | 316 | 5.0 | *201* | *7.2* | | | | |
| g170.3 | 170 | 3% | 22 | 3.1 | 86 | 5.1 | 107 | 7.1 | *56* | *9.5* | | |
| g200.3 | 200 | 3% | 17 | 3.1 | 133 | 5.0 | 155 | 7.1 | *22* | *9.5* | | |
| g150.4 | 150 | 4% | 27 | 3.1 | 102 | 5.1 | 135 | 7.4 | *37* | *9.7* | | |
| g150.5 | 150 | 5% | 43 | 3.0 | 152 | 5.1 | 181 | 7.6 | *32* | *10.2* | | |

with classical inequalities that are valid for the stable set problem, including odd-cycle and wheel inequalities, as well as clique inequalities (which coincide with rank inequalities with a right-hand side equal to 1).

## Acknowledgments

This research was partially supported by the Italian Ministry of Education, University and Research (MIUR): Dipartimenti di Eccellenza Program (2018–2022) - Dept. of Mathematics “F. Casorati”, University of Pavia. The authors would like to thank two anonymous referees whose comments helped improve the quality of the paper.

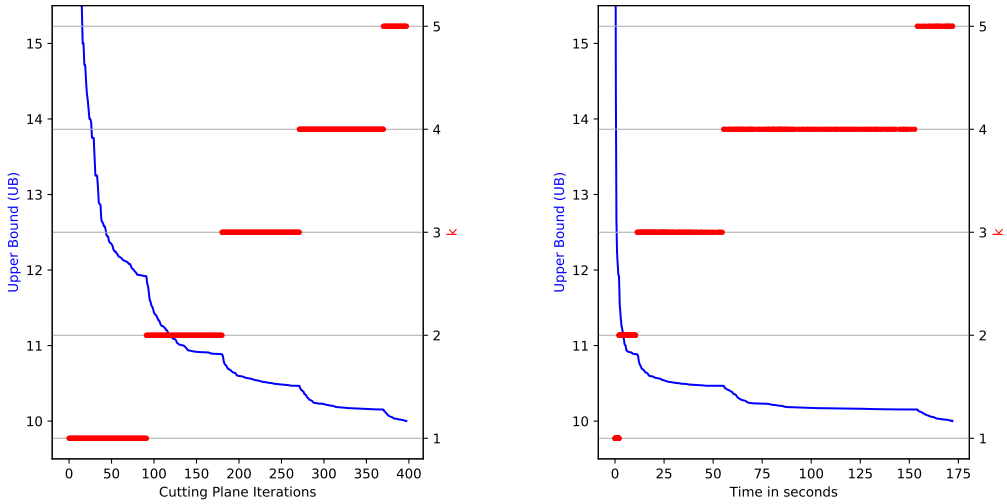


**Figure 4 Bound over time.**

*Note.* Upper bound as a function of the number of iterations (left) and computing time (right) when separating R-$k$-RIs with the branch-and-cut algorithm for increasing values of $k$ on an instance of the *small* dataset featuring 50 vertices and an edge density of 33%.

## References

Amaldi E, Coniglio S, Gualandi S (2010) Improving cutting plane generation with 0-1 inequalities by bi-criteria separation. *International Symposium on Experimental Algorithms*, 266–275 (Springer).

Amaldi E, Coniglio S, Gualandi S (2014) Coordinated cutting plane generation via multi-objective separation. *Mathematical Programming* 143(1-2):87–110.

Balasundaram B, Butenko S, Hicks I (2011) Clique relaxations in social network analysis: The maximum k-plex problem. *Operations Research* 59(1):133–142.

Benson SJ, Ye Y, Zhang X (2000) Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization* 10(2):443–461.

Bolusani S, Coniglio S, Ralphs TK, Tahernejad S (2020) A unified framework for multistage and multilevel mixed integer linear optimization. Dempe S, Zemkoho A, eds., *Bilevel Optimization: Advances and Next Challenges*, chapter 18, 513–560 (Springer).

Bourjolly JM, Laporte G, Mercure H (1997) A combinatorial column generation algorithm for the maximum stable set problem. *Operations Research Letters* 20(1):21–29.

Butenko S, Wilhelm W (2006) Clique-detection models in computational biochemistry and genomics. *European Journal of Operational Research* 173(1):1–17.

Campêlo M, Cornuéjols G (2009) Stable sets, corner polyhedra and the chvátal closure. *Operations Research Letters* 37(6):375–378.

Cheng E, Cunningham W (1997) Wheel inequalities for stable set polytopes. *Mathematical programming* 77(2):389–421.

Cheng E, de Vries S (2002) Antiweb-wheel inequalities and their separation problems over the stable set polytopes. *Mathematical Programming* 92(1):153–175.

Chvátal V (1970) The smallest triangle-free 4-chromatic 4-regular graph. *Journal of Combinatorial Theory* 9(1):93–94.

Chvátal V (1975) On certain polytopes associated with graphs. *Journal of Combinatorial Theory, Series B* 18(2):138–154.

Coniglio S, Furini F, San Segundo P (2021) A new combinatorial branch-and-bound algorithm for the knapsack problem with conflicts. *European Journal of Operational Research* 289(2):435–455.

Coniglio S, Gualandi S (2017) On the separation of topology-free rank inequalities for the max stable set problem. *16th International Symposium on Experimental Algorithms (SEA 2017)* (Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik).

Coniglio S, Tieves M (2015) On the generation of cutting planes which maximize the bound improvement. *International Symposium on Experimental Algorithms*, 97–109 (Springer).

Corrêa R, Donne DD, Koch I, Marenco J (2018) General cut-generating procedures for the stable set polytope. *Discrete Applied Mathematics* 245:28–41.

Dukanovic I, Rendl F (2007) Semidefinite programming relaxations for graph coloring and maximal clique problems. *Mathematical Programming* 109(2-3):345–365.

Furini F, Ljubic I, Martin S, San Segundo P (2019) The maximum clique interdiction game. *European Journal of Operational Research* 277(1):112–127.

Giandomenico M, Letchford A, Rossi F, Smriglio S (2015) Ellipsoidal relaxations of the stable set problem: theory and algorithms. *SIAM Journal on Optimization* 25(3):1944–1963.

Giandomenico M, Rossi F, Smriglio S (2013) Strong lift and project cutting planes for the stable set problem. *Mathematical Programming* 141(1-2):165–192.

Gouveia L, Martins P (2015) Solving the maximum edge-weight clique problem in sparse graphs with compact formulations. *EURO J. Computational Optimization* 3(1):1–30.

Grötschel M, Lovász L, Schrijver A (1981) The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica* 1(2):169–197.

Grötschel M, Lovász L, Schrijver A (1988) *Geometric algorithms and combinatorial optimization*, volume 2 of *Algorithms and Combinatorics* (Springer-Verlag).

Gruber G, Rendl F (2003) Computational experience with stable set relaxations. *SIAM Journal on Optimization* 13(4):1014–1028.

Hastad J (1999) Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica* 182:105–142.

Held S, Cook W, Sewell E (2012) Maximum-weight stable sets and safe lower bounds for graph coloring. *Mathematical Programming Computation* 4(4):363–381.

Hoffman KL, Padberg M (1993) Solving airline crew scheduling problems by branch-and-cut. *Management science* 39(6):657–682.

Hosseinian S, Fontes D, Butenko S (2018) A nonconvex quadratic optimization approach to the maximum edge weight clique problem. *J. Global Optimization* 72(2):219–240.

Hosseinian S, Fontes D, Butenko S, Buongiorno MN, Fornari M, Curtarolo S (2017) The maximum edge weight clique problem: formulations and solution approaches. *Optimization Methods and Applications*, 217–237 (Springer).

Johnson D, Trick M (1996) *Cliques, coloring, and satisfiability: second DIMACS implementation challenge, October 11-13, 1993*, volume 26 (American Mathematical Soc.).

Karp R (1972) Reducibility among combinatorial problems. Miller R, Thatcher J, eds., *Proceedings of a Symposium on the Complexity of Computer Computations*, The IBM Research Symposia Series (Plenum Press).

Letchford A, Rossi F, Smriglio S (2020) The stable set problem: Clique and nodal inequalities revisited. *Computers & Operations Research* 123:105024.

Lodi A, Ralphs T, Woeginger G (2014) Bilevel programming and the separation problem. *Mathematical Programming* 146(1-2):437–458.

Lovász L (1979) On the shannon capacity of a graph. *IEEE Transactions on Information theory* 25(1):1–7.

Mannino C, Sassano A (1996) Edge projection and the maximum cardinality stable set problem. *DIMACS series in discrete mathematics and theoretical computer science* 26:205–219.

Marzi F, Rossi F, Smriglio S (2019) Computational study of separation algorithms for clique inequalities. *Soft Computing* 23(9):3013–3027.

Maslov E, Batsyn M, Pardalos P (2014) Speeding up branch and bound algorithms for solving the maximum clique problem. *Journal of Global Optimization* 59(1):1–21.

Nemhauser G, Sigismondi G (1992) A strong cutting plane/branch-and-bound algorithm for node packing. *Journal of the Operational Research Society* 43(5):443–457.

Östergård P (2002) A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics* 120(1):197–207.

Padberg M (1973) On the facial structure of set packing polyhedra. *Mathematical programming* 5(1):199–215.

Pardalos P, Xue J (1994) The maximum clique problem. *Journal of Global Optimization* 4(3):301–328.

Prosser P (2012) Exact algorithms for maximum clique: A computational study. *Algorithms* 5(4):545–587.

Rebennack S, Oswald M, Theis D, Seitz H, Reinelt G, Pardalos P (2011) A branch and cut solver for the maximum stable set problem. *Journal of Combinatorial Optimization* 21(4):434–457.

Rebennack S, Reinelt G, Pardalos P (2012) A tutorial on branch and cut algorithms for the maximum stable set problem. *International Transactions in Operational Research* 19(1-2):161–199.

Rossi F, Smriglio S (2001) A branch and cut algorithm for the maximum cardinality stable set problem. *Operations Research Letters* 28(2):63–74.

Rutenburg V (1994) Propositional truth maintenance systems: Classification and complexity analysis. *Annals of Mathematics and Artificial Intelligence* 10(3):207–231.

San Segundo P, Artieda J (2015) A novel clique formulation for the visual feature matching problem. *Applied Intelligence* 43(2):325–342.

San Segundo P, Coniglio S, Furini F, Ljubić I (2019) A new branch-and-bound algorithm for the maximum edge-weighted clique problem. *European Journal of Operational Research* 278(1):76–90.

San Segundo P, Lopez A, Pardalos P (2016) A new exact maximum clique algorithm for large and massive sparse graphs. *Computers & Operations Research* 66:81–94.

San Segundo P, Rodríguez-Losada D, Jiménez A (2011) An exact bit-parallel algorithm for the maximum clique problem. *Computers & Operations Research* 38(2):571–581.

Sewell EC (1998) A branch and bound algorithm for the stability number of a sparse graph. *INFORMS Journal on Computing* 10(4):438–447.

Shimizu S, Yamaguchi K, Masuda S (2018) A branch-and-bound based exact algorithm for the maximum edge-weight clique problem. *Computational Science/Intelligence & Applied Informatics, CSII2018*, 27–47.

Tomita E, Kameda T (2007) An efficient branch-and-bound algorithm for finding a maximum clique with computational experiments. *Journal of Global optimization* 37(1):95–111.

Trotter LJ (1975) A class of facet producing graphs for vertex packing polyhedra. *Discrete Mathematics* 12(4):373–388.

Wu Q, Hao JK (2016) A clique-based exact method for optimal winner determination in combinatorial auctions. *Information Sciences* 334:103–121.

Wu W, Hao JK (2015) A review on algorithms for maximum clique problems. *European Journal of Operational Research* 242(3):693–709.

Zuckerman D (2006) Linear degree extractors and the inapproximability of max clique and chromatic number. *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, 681–690 (ACM).