Les Cahiers du GERAD ISSN: 0711–2440

Integer programming models for the partial directed weighted improper coloring problem

A. Hertz, R. Montagné,
F. Gagnon

G–2016–112

November 2016

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Integer programming models for the partial directed weighted improper coloring problem

Alain Hertz [a]
Romain Montagné [b]
François Gagnon [c]

[a] *GERAD & Department of Mathematics and Industrial Engineering, Polytechnique Montréal (Québec) Canada, H3C 3A7*

[b] *Department of Electrical Engineering École de Technologie Supérieure, Montréal (Québec) Canada, H3C 1K3*

alain.hertz@gerad.ca
francois.gagnon@etsmtl.ca

November 2016

Les Cahiers du GERAD
G–2016–112

Abstract: Given a complete directed graph G with weights on the vertices and on the arcs, a θ-improper k-coloring of G is an assignment of at most k different colors to the vertices of G such that the weight of every vertex v is greater, by a given factor $1/\theta$, than the sum of the weights on the arcs (u, v) entering v, with the tail u of the same color as v. For a given real number θ and an integer k, the Partial Directed Weigthed Improper Coloring Problem ((θ, k)-PDWICP) is to determine the order of the largest induced subgraph G' of G such that G' admits a θ-improper k-coloring. The (θ, k)-PDWICP appears as a natural model when solving channel assignment problems that aim to maximize the number of simultaneously communicating mobiles in a wireless network. In this paper, we compare integer programming approaches to solve this $\mathcal{NP}$-hard problem to optimality. A polynomial-time computable upper bound inspired by the Lovász $\vartheta(G)$ function, and valid inequalities based on the knapsack and set-packing problems are embedded in a branch-and-bound procedure. Comparisons with a branch-and-price scheme are reported.

Keywords: Weighted improper vertex coloring, integer programming, valid inequalities, branch-and-cut, branch-and-price

Résumé : Étant donné un graphe G complet, orienté, avec des poids sur les sommets et les arcs, une k-coloration θ-impropre de G est une affectation d'au plus k couleurs aux sommets de G de telle sorte que le poids de chaque sommet v soit au moins aussi grand, par un facteur $\frac{1}{\theta}$, que la somme des poids sur les arcs (u, v) qui entrent en v, avec la couleur de u égale à celle de v. Pour un nombre réel θ et un entier k, le problème de la coloration partielle d'un graphe orienté pondéré ((θ, k)-PDWICP) est de déterminer la taille du plus grand sous-graphe induit G' de G tel que G' admet une k-coloration θ- impropre. Le (θ, k)-PDWICP est un modèle naturel lorsqu'on résout des problèmes d'affectation de fréquences, et que l'objectif est de maximiser le nombre d'utilisateurs qui communiquent simultanément dans un réseau sans fil. Dans cet article, nous comparons des approches de programmation linéaire en nombres entiers pour résoudre ce problème NP-dur. Une borne supérieure, calculable en temps polynomial, et inspirée de la fonction de Lovász $\vartheta(G)$, ainsi que des inégalités valides basées sur le problème du sac à dos et celui du "set packing", sont combinées dans une procédure de type branch-and- bound. Des comparaisons sont faites avec une approche branch-and-price.

Acknowledgments: The second author would like to thank Ted Ralphs for his help when using the `DipPy` software.

1 Introduction

Given a graph G and an integer k, the standard *k-coloring problem* (k-CP) is to determine an assignment of at most k different colors to the vertices of G, such that no two adjacent vertices share the same color. In 1966, Lovász [10] introduced a variation of the k-CP by allowing each vertex to have at most θ neighbors sharing the same color, giving rise to the *θ-improper k-coloring problem* ((θ, k)-ICP). The standard k-coloring problem is the particular case where $\theta = 0$.

In this paper, we consider an extension of the (θ, k)-ICP to weighted directed graphs. More precisely, consider a directed graph $G(V, A, W, \omega)$ with vertex set V and arc set A, where W and ω are positive weight functions on the vertices and arcs of G, and let $\theta \in \mathbb{R}^+$ be a positive threshold. The *directed weighted θ-improper k-coloring problem* ((θ, k)-DWICP) is to assign a color $c(v) \in \{1, \ldots, k\}$ to all vertices v of G such that the weight $W(v)$ of every vertex v is greater, by a factor $\frac{1}{\theta}$, than the sum of the weights $\omega(u, v)$ on the arcs (u, v) entering v, with the tail u of the same color as v:

$$\sum_{(u,v)\in A \mid c(u)=c(v)} \omega(u, v) \leq \theta \cdot W(v) \quad \forall v \in V. \tag{1}$$

The optimization problem we focus on in this paper is called the *partial directed weighted θ-improper k-coloring problem* ((θ, k)-PDWICP) [7]: given a weighted directed graph $G(V, A, W, \omega)$, a real number $\theta \geq 0$, and an integer $k < |V|$, the (θ, k)-PDWICP is to color a maximum number of vertices with the k available colors, while respecting constraints (1). This maximum number is denoted $\alpha_k^\theta(G)$.

Definition 1 *A subset $I \subseteq V$ of vertices in a weighted directed graph $G(V, A, W, \omega)$ is θ-independent if all vertices in I can receive the same color without violating constraints (1), i.e.,*

$$\sum_{u\in I, u\neq v} \omega(u, v) \leq \theta \cdot W(v) \quad \forall v \in I.$$

Consider, for example, the graph in Figure 1, where the numbers on the vertices and arcs correspond to their weight. If $\theta = \frac{1}{2}$, $\{a, b\}$ is θ-independent, as $\omega(a, b) = 2 \leq \frac{5}{2} = \theta \cdot W(b)$, and $\omega(b, a) = 1 \leq \frac{4}{2} = \theta \cdot W(a)$. If only one color is available (i.e., $k = 1$), then $\alpha_k^\theta(G) = 2$. Indeed, if we assign the same color to vertices b and c, the sum of the weights on the arcs entering b is $3 > \theta \cdot W(b) = \frac{5}{2}$.

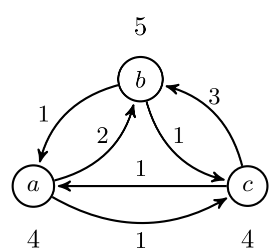

Figure 1: $\{a, b\}$ **is** $\frac{1}{2}$**-independent and** $\alpha_1^{\frac{1}{2}}(G) = 2$.

The undirected versions of the abovementioned problems are obtained by replacing the arc set A of G with an edge set E, and by imposing that the weight $W(v)$ of every vertex v is greater, by a factor $\frac{1}{\theta}$, than the sum of the weights on the edges $\{u, v\}$ incident to v with $c(u) = c(v)$. If $\theta \in \mathbb{N}$, and if all weights $W(v)$ and $\omega(u, v)$ equal 1, the undirected version of the (θ, k)-DWICP is equivalent to the (θ, k)-ICP. Also, the stability number of an undirected graph (i.e., the order of the largest independent set), commonly denoted by $\alpha(G)$, is equal to $\alpha_1^0(G)$, with all weights $W(v)$ and $\omega(u, v)$ equal to 1.

The (θ, k)-PDWICP is a generalization of the *maximum k-colorable induced subgraph problem* (k-MCISP) introduced in [13]. Indeed, given any undirected graph G, one can replace every edge $\{u, v\}$ by two arcs (u, v) and (v, u), and by setting θ equal to 0, and $W(v)$ and $\omega(u, v)$ equal to 1 for all vertices v and all arcs (u, v)

in G, one gets an instance of the (θ, k)-PDWICP which is equivalent to the original k-MCISP. Since the k-MCISP is $\mathcal{NP}$-hard, this reduction proves that the (θ, k)-PDWICP is also $\mathcal{NP}$-hard.

The undirected version of the (θ, k)-DWICP with unit weights on the vertices was studied by Araujo et al. [2]. Motivated by a practical channel assignment problem, the authors tackled the optimization problem of minimizing k so that G admits a θ-improper k-coloring. Upper bounds were proposed, as well as integer programming algorithms. For similar motivations, Archetti et al. [3] developed a branch-and-price scheme for the same problem in the directed case. Keeping in mind that channel assignments require very fast computing times, Hertz et al. [7] developed multiple heuristics for the (θ, k)-PDWICP, based on classical coloring algorithms such as RLF [9] and DSATUR [5], as well as Voronoï diagram considerations.

The aim of this paper is to analyse integer programming models for the (θ, k)-PDWICP. In Section 2, three different integer programming formulations are proposed and compared theoretically. A polynomial-time computable upper bound is developed in Section 3, while valid inequalities for a branch-and-cut algorithm are proposed in Section 4. Ingredients for a branch-and-price algorithm are given in Section 5, and numerical experiments are reported and analyzed in Section 6.

Unless explicitly stated otherwise, the rest of the paper only considers directed graphs. But the proposed algorithms can be applied to undirected graphs by replacing every edge $e = \{u, v\}$ by two arcs (u, v) and (v, u), and by setting $\omega(u, v)$ and $\omega(v, u)$ equal to the weight of edge e. Observe also that, without loss of generality, we can assume G is a complete weighted directed graph, as non-arcs can be replaced by arcs of weight 0.

2 Problem formulations

2.1 $0-1$ integer programming formulations

Formulation I. Let x_{vi} be a binary variable that equals 1 if and only if vertex v takes color i. One can compute $\alpha_k^\theta(G)$ with the following 0–1 integer program:

$$\max \sum_{v\in V}\sum_{i=1}^{k} x_{vi} \tag{2}$$

$$\text{s.t.} \sum_{i=1}^{k} x_{vi} \le 1 \qquad \forall v \in V \tag{3}$$

$$\sum_{u\neq v} \omega(u,v)\cdot(x_{ui}+x_{vi}-1) \le \theta\cdot W(v)\cdot x_{vi} \qquad \forall v\in V, \forall i \in \{1,\cdots,k\} \tag{4}$$

$$x_{vi}\in\{0,1\} \qquad \forall v\in V, \forall i \in \{1,\cdots,k\} \tag{5}$$

The objective function (2) aims at maximizing the number of colored vertices. Constraints (3) ensure that at most one color is assigned to every vertex, while Equations (4) ensure that the coloring is θ-improper.

Formulation II. Formulation I can be slightly strengthened as follows. Let y_{uv}^i be another set of binary variables equal to 1 if both vertices u and v have color $i \in \{1, ..., k\}$. The following $0-1$ integer program is an alternative to compute $\alpha_k^\theta(G)$:

$$\max \sum_{v\in V}\sum_{i=1}^{k} x_{vi} \tag{6}$$

$$\text{s.t.} \sum_{i=1}^{k} x_{vi} \le 1 \qquad \forall v \in V \tag{7}$$

$$\sum_{u\neq v} \omega(u,v)\cdot y_{uv}^i \le \theta\cdot W(v)\cdot x_{vi} \qquad \forall v\in V, \forall i \in \{1,\cdots,k\} \tag{8}$$

$$x_{ui}+x_{vi}-1 \le y_{uv}^i \qquad \forall u\neq v, \forall i \in \{1,\cdots,k\} \tag{9}$$

$$x_{vi},\ y^i_{uv} \in \{0,1\} \qquad \forall u \neq v, \forall i \in \{1,\cdots,k\} \tag{10}$$

Constraints (9) enforce y^i_{uv} to equal 1 if both u and v have color i. The other equations have the same role as in the first model.

Formulation III. The third formulation is a Dantzig-Wolfe based decomposition which gives rise to a *set packing* problem. More precisely, let Ω be the set of all θ-independent subsets of G, and let λ_p be a binary variable equal to 1 if and only if subset $p \in \Omega$ is selected as a color class. The formulation can be expressed as follows :

$$\max \sum_{v\in V}\sum_{p\in\Omega|v\in p} \lambda_p \tag{11}$$

$$\text{s.t.} \quad \sum_{p\in\Omega|v\in p} \lambda_p \leq 1 \qquad \forall v \in V \tag{12}$$

$$\sum_{p\in\Omega} \lambda_p \leq k \tag{13}$$

$$\lambda_p \in \{0,1\} \qquad \forall p \in \Omega \tag{14}$$

The objective function (11) aims at maximizing the number of vertices in the selected θ-independent subsets. Constraints (12) ensure every vertex is colored at most once, while constraints (13) limit the number of θ-independent subsets (and thus the number of colors) to k.

2.2 Comparison of the continuous relaxations

For a complete directed weighted graph $G(V,A,W,\omega)$, a real number $\theta \geq 0$, and an integer $k < |V|$, let $z^I_R(G,\theta,k)$, $z^{II}_R(G,\theta,k)$ and $z^{III}_R(G,\theta,k)$ denote the optimal value of the linear relaxation of formulations I, II and III, respectively (i.e., the constraints that each variable must be 0 or 1 are replaced by weaker constraints imposing that each variable must belong to the interval $[0,1]$).

Proposition 1 *$z^I_R(G,\theta,k) \geq z^{II}_R(G,\theta,k) \geq z^{III}_R(G,\theta,k)$, and both inequalities are possibly strict.*

Proof. Consider any feasible solution $\{x_{vi}, y^i_{uv}\}$ to the linear relaxation of formulation II. Given any vertex $v \in V$ and any color $i \in \{1,\ldots,k\}$, we can multiply both sides of constraints (9) by $\omega(u,v)$ and sum up these constraints for all $u \neq v$ to get

$$\sum_{u\neq v} \omega(u,v)\cdot(x_{ui}+x_{vi}-1) \leq \sum_{u\neq v}\omega(u,v)\cdot y^i_{uv}.$$

Constraints (8) then imply constraints (4), which proves that $z^I_R(G,\theta,k) \geq z^{II}_R(G,\theta,k)$, since $\{x_{vi}\}$ is a feasible solution to the linear relaxation of formulation I.

Consider now any feasible solution $\{\lambda_p\}$ to the linear relaxation of formulation III. If $\lambda_p = 0$ for all $p \in \Omega$, we have $z^{III}_R(G,\theta,k) = 0 \leq z^{II}_R(G,\theta,k)$. So assume $\lambda_p > 0$ for at least one p in Ω. Let $(p_1,\ldots,p_{|\Omega|})$ be the ordered set of elements in Ω such that $\lambda_{p_r} \geq \lambda_{p_s}$ for $r < s$. Also, for every $j = 1,\ldots,|\Omega|$ let

$$\sigma(j) = \left\lceil \sum_{r=1}^{j} \lambda_{p_r} \right\rceil.$$

It follows from constraint (13) that $\sigma(j) \in \{1,\ldots,k\}$ for all j. For every vertex v we can therefore define its colors $c(v)$ as follows:

$$c(v) = \min_{v\in p_j} \sigma(j).$$

We can now give values to the variables of the linear relaxation of formulation II as follows:

- set $x_{vi} = \sum_{p\in\Omega|v\in p} \lambda_p$ if $i = c(v)$, and $x_{vi} = 0$ otherwise;
- set $y^i_{uv} = 1$ if $x_{ui} \cdot x_{vi} > 0$, and $y^i_{uv} = 0$ otherwise.

These values define a feasible solution to the linear relaxation of formulation II. Indeed,

- for all $v \in V$ we have $\sum_{i=1}^{k} x_{vi} = x_{vc(v)} = \sum_{p\in\Omega|v\in p} \lambda_p \leq 1$, the last inequality being valid by constraint (12). Hence, constraints (7) are satisfied;
- if $i \neq c(v)$ then $x_{vi} = 0$ and $y^i_{uv} = 0$ for all $u \neq v$, which implies that constraints (8) are satisfied. So assume $i = c(v)$ and let j be such that $c(v) = \sigma(j)$. We then have

$$\sum_{u\neq v} \omega(u, v) \cdot y^i_{uv} = \sum_{u\neq v|x_{ui}>0} \omega(u, v) \leq \sum_{u\neq v|u\in p_j} \omega(u, v) \leq \theta \cdot W(v)$$

  the last inequality being valid since p_j is a θ-independent set. Hence, constraints (8) are satisfied;
- variables y^i_{uv} are defined in such a way that constraints (9) are satisfied.

This constructed feasible solution to the linear relaxation of formulation II has the same value as the considered feasible solution to the linear relaxation of formulation III. Indeed,

$$\sum_{v\in V} \sum_{i=1}^{k} x_{vi} = \sum_{v\in V} x_{vc(v)} = \sum_{v\in V} \sum_{p\in\Omega|v\in p} \lambda_p.$$

We therefore have $z^{II}_R(G, \theta, k) \geq z^{III}_R(G, \theta, k)$.

To show that we can have $z^{III}_R(G, \theta, k) < z^{II}_R(G, \theta, k) < z^{I}_R(G, \theta, k)$, it is sufficient to consider the graph on Figure 2. It is an easy exercise to check that $z^I_R(G, 1, 1) = 1.8$. Such a value is reached by setting $x_{a1} = 1$ and $x_{b1} = x_{c1} = 0.4$. No values for the y^i_{uv} variables can fit with these x_{vi} values to get a feasible solution to the linear relaxation of formulation II. Indeed, we should then have $y^1_{ab} \geq 0.4$ and $y^1_{cb} \geq 0$, and constraint (8) for vertex b would impose $0.8 \leq \omega(a, b) \cdot y^1_{ab} + \omega(c, b) \cdot y^1_{cb} \leq \theta \cdot W(b) \cdot x_{b1} = 0.4$, a contradiction. The optimal solution to the linear relaxation of formulation II has value $z^{II}_R(G, 1, 1) = 1.75$. It is reached by setting $x_{a1} = 0.75$, $x_{b1} = x_{c1} = 0.5$, $y^1_{ab} = y^1_{ba} = y^1_{ac} = y^1_{ca} = 0.25$, and $y^1_{bc} = y^1_{cb} = 0$. Such a value cannot be reached with the linear relaxation of formulation III. Indeed, there are only three 1-independent sets, namely $p_1 = \{a\}$, $p_2 = \{b\}$ and $p_3 = \{c\}$, and constraint (13) imposes $\lambda_{\{a\}} + \lambda_{\{b\}} + \lambda_{\{c\}} \leq 1$. Since the objective function, is also equal to $\lambda_{\{a\}} + \lambda_{\{b\}} + \lambda_{\{c\}}$, we conclude that $z^{III}_R(G, 1, 1) \leq 1$, this value been reached by setting $\lambda_{\{a\}} = 1$ and $\lambda_{\{b\}} = \lambda_{\{c\}} = 0$. In summary, we have $z^{III}_R(G, 1, 1) = 1 < z^{II}_R(G, 1, 1) = 1.75 < z^{I}_R(G, 1, 1) = 1.8$. □

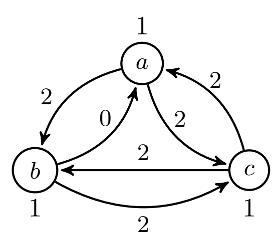

Figure 2: A graph G with $z^{III}_R(G, 1, 1) < z^{II}_R(G, 1, 1) < z^{I}_R(G, 1, 1)$.

These inequalities hint that it is very likely that formulation I will give the worse integrality gap, while formulation III will give the best one. Note however that formulation III has an exponential number of variables and can thus hardly be used as such. We will see how to bypass this difficulty is Section 5.

We close this section by proving that the strict inequality $z^I_R(G, \theta, k) > z^{II}_R(G, \theta, k)$ is possible only with $k = 1$.

Proposition 2 *$z_R^I(G, \theta, k) = z_R^{II}(G, \theta, k) = n$ for every graph G with n vertices, every $\theta \in \mathbb{R}^+$, and every integer $k > 1$.*

Proof. It is sufficient to observe that by setting $x_{vi} = \frac{1}{k}$ and $y_{uv}^i = 0$ for all $u \neq v$ and all $i \in \{1, \cdots, k\}$, one gets feasible solutions to the linear relaxations of formulations I and II. Since these feasible solutions have value $\sum_{v \in V} \sum_{i=1}^k x_{vi} = n$, constraints (3) in formulation I and (7) in formulation II prove that they are optimal. □

3 A polynomial-time computable upper bound

The Lovász *theta function* is a polynomial-time computable parameter that provides an upper bound $\vartheta(G)$ on the order of the largest clique in G [11]. As a corollary, if $\bar{G}$ denotes the complement of G, $\vartheta(\bar{G})$ is an upper bound on the stability number of an undirected graph G (i.e., the order of the largest independent set). Since the (θ, k)-PDWICP is a generalization of the largest independent set problem, we show how to adapt these concepts to derive an upper bound on the order of the largest θ-improper k-colorable subgraph. This bound will prove to be tighter than the optimal value of the continuous relaxations of formulations I and II.

Definition 2 *Let $G = (V, E)$ be an undirected graph, and let $\mathcal{A}(G)$ be the set of symmetric real matrices, such that $a_{ij} = 1$ if $i = j$ or if $\{i, j\} \in E$. The other coefficients are arbitrary, as long as the matrix remains symmetric. The generalized Lovász $\vartheta_k(G)$ function introduced in [13] is defined as follows:*

$$\vartheta_k(G) = \min_{A \in \mathcal{A}(G)} \left\{ \sum_{i=1}^k \lambda_i(A) \right\},$$

where $\lambda_i(A)$ is the i^{th} largest eigenvalue of A.

Narasimhan [13] has shown how to compute this parameter in polynomial time (with an ellipsoid method), and how it can be related to the order of the largest (proper) k-colorable induced subgraph of G. More precisely, if $\alpha_k(G)$ denotes the order of the largest k-colorable induced subgraph of G, we have the following.

Proposition 3 [13] $\alpha_k(G) \leq \vartheta_k(\bar{G})$ for all (undirected and unweighted) graphs G and all positive integers k.

In order to adapt this bound to complete weighted directed graphs, we introduce the following definition.

Definition 3 *For every complete weighted directed graph $G(V, A, W, \omega)$ and any real number $\theta \geq 0$, we define the following auxiliary graph $H(G, \theta)$:*

i. the vertex set of $H(G, \theta)$ is V, as in G;

ii. two vertices u and v of $H(G, \theta)$ are linked by an edge if and only if $\omega(u, v) > \theta \cdot W(v)$ or (non-exclusive) $\omega(v, u) > \theta \cdot W(u)$.

We can now derive an upper bound for the optimal value to the (θ, k)-PDWICP, based on the Lovász generalized $\vartheta_k(G)$ function.

Proposition 4 *Given any complete directed weighted graph $G(V, A, W, \omega)$, any real number $\theta \geq 0$, and any integer $k < |V|$, we have*

$$\alpha_k^\theta(G) \leq \left\lfloor \vartheta_k(\bar{H}(G, \theta)) \right\rfloor.$$

Proof. By definition, every θ-improper k-coloring of a subgraph G' of G corresponds to a (proper) k-coloring of the subgraph of $H(G, \theta)$ induced by the vertices in G'. Hence, $\alpha_k^\theta(G) \leq \alpha_k(H(G, \theta)) \leq \vartheta_k(\bar{H}(G, \theta))$. □

Since the construction of the auxiliary graph $H(G, \theta)$ can clearly be done in polynomial time, and since parameter $\vartheta_k(\bar{H}(G, \theta))$ is also polynomial-time computable, the proposed upper bound can be computed in polynomial time as well. An example is illustrated in Figure 3, with $k = \theta = 2$. It can be shown [13] that $\vartheta_2(C_5) = 2\sqrt{5}$, which implies $\alpha_2^2(G) \leq \lfloor 2\sqrt{5} \rfloor = 4$. This bound is reached by coloring vertices a and b of $H(G,\theta)$ with one color, and vertices c and d with the other. In this example, the bound is tight and the proper 2-coloring of $H(G,\theta)$ is a 2-improper 2-coloring of G. This is not always the case, and the gap between $\vartheta_k(\bar{H}(G,\theta))$ and $\alpha_k^\theta(G)$ can be arbitrarily large. If, for example, $\theta = W(v) = 1$ for every vertex v, and $\omega(u,v) > \frac{1}{2}$ for every arc (u,v), it is clear that $\alpha_k^1(G) \leq 2k$ (color classes are necessarily pairs of vertices, or singletons). In this case, $\bar{H}(G,\theta)$ is a complete graph or order n, which implies that $\vartheta_k(\bar{H}(G,\theta)) = n$, and $\frac{\vartheta_k(\bar{H}(G,\theta))}{\alpha_k^\theta(G)} \geq \frac{n}{2k}$.

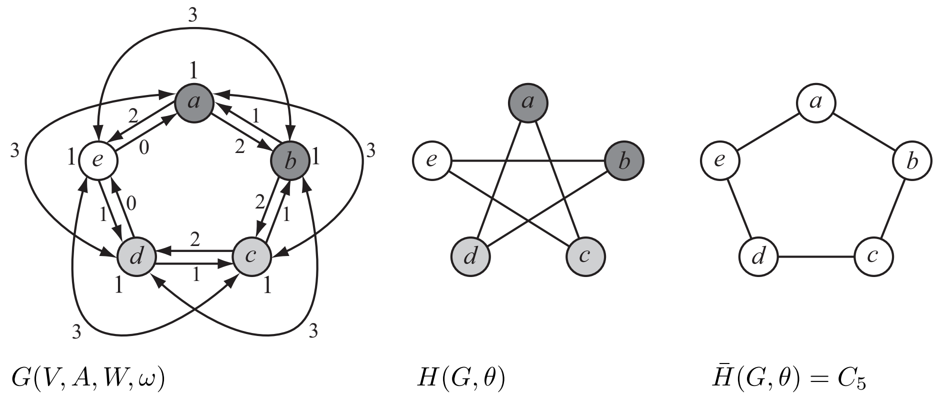

Figure 3: A graph G with $\alpha_2^2(G) \leq \lfloor \vartheta_2(H(\bar{G},2)) \rfloor = \lfloor 2\sqrt{5} \rfloor = 4$.

4 Valid inequalities for a branch-and-cut algorithm

Branch and cut involves running a branch-and-bound algorithm and using cutting planes to tighten the continuous relaxations. In this section, we first propose valid inequalities that break symmetries and avoid infeasible solutions, and we then exhibit a set of constraints that possibly strengthen the IP formulations of Section 2. The violation of a valid inequality, when detected, corresponds to a cut that can be used in a branch-and-cut algorithm to prevent exploring useless branches.

Symmetry breaking constraints. Since the solution space possibly contains multiple equivalent solutions that have identical objective function values, and differ only in the selected colors, one way to cope with such a symmetry is by adding the following ordering constraints:

$$\sum_{u \in V} x_{ui} \geq x_{v(i+1)} \qquad \forall v \in V,\ \forall i \in \{1,\ldots,k-1\} \tag{15}$$

$$\sum_{u \in V} x_{ui} \geq \sum_{u \in V} x_{u(i+1)} \qquad \forall i \in \{1,\ldots,k-1\} \tag{16}$$

Equations (15) ensure that color $i+1$ is not used until color i is, and Equations (16) order the color classes by non-increasing size.

Incompatibility constraints. Given a complete weighted directed graph $G(V,A,W,\omega)$, consider the auxiliary graph $H(G,\theta)$ of Definition 3, and let E be its edge set. It follows from the construction of $H(G,\theta)$ that the endpoints u and v of an edge $\{u,v\} \in E$ cannot have the same color, which can be imposed as follows:

$$y_{uv}^i = 0 \quad \forall \{u,v\} \in E,\ \forall i \in \{1,\ldots,k\}. \tag{17}$$

We now focus on formulation II, but all following results can easily be adapted to formulation I.

Knapsack constraints. As already mentioned, Constraints (8) ensure that the obtained coloring is θ-improper. Such a constraint for a specific vertex v and a given color $i \in \{1,\ldots,k\}$ reads as follows:

$$\sum_{u \neq v} \omega(u,v) \cdot y_{uv}^i \leq \theta \cdot W(v) \cdot x_{vi}.$$

If x_{vi} equals 0, then y^i_{uv} equals 0 for all vertices $u \neq v$; but if x_{vi} equals 1, the above inequation is equivalent to a binary knapsack constraint where the vertices $u \neq v$ are items of weight $\omega(u, v)$ that need to fit into a knapsack of capacity $\theta \cdot W(v)$. The following properties, which can be found, for example, in [8], were proven in a knapsack context, and can easily be adapted to our problem.

Consider a $0-1$ knapsack problem with n items. Let x_i be a binary variable that equals 1 if and only if item i is chosen. Also, let v_i be the weight of item i, and let b be the capacity of the knapsack. A subset S of items is called a *cover* if it satisfies $\sum_{i \in S} v_i > b$. Given any cover S, the following *cover inequality* is clearly valid:

$$\sum_{i \in S} x_i \leq |S| - 1.$$

Also, a subset P of items is called a *pack* if it satisfies $\sum_{i \in P} v_i < b$. For a pack P, let $\mu = b - \sum_{i \in P} v_i$ denote the residual capacity. It can be proved [8] that the following *weight inequality* is valid:

$$\sum_{i \notin P} \max\{0, v_i - \mu\} \cdot x_i \leq \sum_{i \in P} v_i \cdot (1 - x_i).$$

In order to adapt such inequalities to the (θ, k)-PDWICP, we consider the following definitions:

Definition 4 *Consider a complete weighted directed graph $G(V, A, W, \omega)$, a real number $\theta \geq 0$, and a vertex $v \in V$.*

- *A (θ, v)-cover is a subset S of vertices $u \neq v$ such that:*

$$\sum_{u \in S} \omega(u, v) > \theta \cdot W(v).$$

- *A (θ, v)-pack is a subset P of vertices $u \neq v$ such that:*

$$\sum_{u \in P} \omega(u, v) < \theta \cdot W(v).$$

The residual capacity of a (θ, v)-pack P is defined as $\theta \cdot W(v) - \sum_{u \in P} \omega(u, v)$.

Given a (θ, v)-cover S and a color $i \in \{1, \dots, k\}$, the following inequality is clearly valid for formulation II, since it is a straightforward adaptation of the cover inequality of the knapsack problem:

$$\sum_{u \in S} y^i_{uv} \leq |S| - 1. \tag{18}$$

Essentially, this inequality imposes that not all vertices of the (θ, v)-cover S can have the same color as v. It is violated by a given fractional solution y^i_{uv} if $\sum_{u \in S}(1 - y^i_{uv}) < 1$. As suggested in [8], one can try to generate such cuts heuristically, by inserting vertices into S in non-decreasing value $\frac{1-y^i_{uv}}{\omega(u,v)}$.

Given a (θ, v)-pack P with residual capacity μ, and a color $i \in \{1, \dots, k\}$, the following inequality is valid for formulation II, since it is a straightforward adaptation of the weight inequality of the knapsack problem:

$$\sum_{u \notin P} \max\{0, \omega(u, v) - \mu\} \cdot y^i_{uv} \leq \sum_{u \in P} \omega(u, v) \cdot (1 - y^i_{uv}) \tag{19}$$

Since our goal is to reduce the overall computing time, we use the simple greedy approach described in [14] to generate (θ, v)-packs for which the above inequalities are possibly violated. Roughly speaking, a (θ, v)-pack is built sequentially, giving priority to vertices u that fit in the pack (i.e., $\omega(u, v)$ is smaller than the residual capacity) and for which the associated y^i_{uv} variable is closest to 1. More details can be found in [14].

Packing constraints. As already mentioned, the auxiliary graph $H(G,\theta)$ of Definition 3, associated with a complete weighted directed graph $G(V,A,W,\omega)$, is such that the endpoints u and v of an edge $\{u,v\}$ in $H(G,\theta)$ cannot have the same color. Hence, for every clique K in $H(G,\theta)$ and every color $i \in \{1,\ldots,k\}$, we have the following constraint:

$$\sum_{v\in K} x_{vi} \leq 1. \tag{20}$$

Finding a clique that violates such a constraint is an $\mathcal{NP}$-hard problem. We therefore try to generate cuts by sequentially selecting vertices with highest value x_{vi}, and which are linked to previously selected vertices, until, eventually, $\sum_{v\in K} x_{vi} > 1$.

Also, every cycle C in $H(G,\theta)$ contains at most $\lfloor\frac{|C|}{2}\rfloor$ vertices with the same color. Hence, for every odd cycle C in H and every color $i \in \{1,\ldots,k\}$, we have the following constraint:

$$\sum_{v\in C} x_{vi} \leq \frac{|C|-1}{2}. \tag{21}$$

As shown in [15] finding odd-cycles that violate the above constraint can be done as follows. For each color i, consider the bipartite graph $B=(V_1\cup V_2, E)$, where $V_1=V_2=V$, and a vertex $u\in V_1$ is linked to a vertex $v\in V_2$ with an edge of weight $1-x_{ui}-x_{vi}$ if and only if u and v are adjacent in $H(G,\theta)$. A minimum weight odd cycle in $H(G,\theta)$ containing a given vertex v corresponds to a shortest path in B linking the two copies of v. If such a shortest path has length strictly smaller than 1, then Equation (21) is violated for the corresponding odd cycle. Note that the weights on the edges of B are not necessarily positive, unless incompatibility constraints (17) are added to the model. Adding them significantly simplifies the process since shortest paths can then be determined with Dijkstra's algorithm.

5 A branch-and-price algorithm

In this section, we briefly describe how to solve formulation III using a branch-and-price algorithm. As already mentioned, this formulation contains many variables. We therefore consider a restriction of the problem, by dynamically generating a small subset of variables, considering only those that are likely to improve the objective function. The continuous relaxation of formulation III is the *Master Problem*, while its restriction that only considers a subset of variables (also called columns) is the *Restricted Master Problem* (RMP). To check whether an optimal solution to the RMP is also optimal for the Master Problem, a subproblem, called *pricing problem*, is solved and eventually produces variables that can be added to the RMP to improve the objective function. This involves finding a column that has a positive reduced cost (for a maximization problem). Each time a column is found with strictly positive reduced cost, it is added to the RMP and the relaxation is reoptimized. If no columns can enter the basis and the solution to the relaxation is not integer, then branching occurs. More details about this famous technique can be found, for example, in [4].

Branch-and-price was used to solve problems in a variety of application areas, including graph coloring. For example, it was used in [12] for classical coloring, and in [3] to minimize the number k of colors so that a given complete directed graph G admits a θ-improper k-coloring.

Let π_v and γ be the dual variables for constraints (12) and (13) of the continuous relaxation of formulation III. Let x_v be a binary variable that equals 1 if and only if vertex v is selected. The pricing problem reads as follows:

$$\max \sum_{v\in V}(1-\pi_v)\cdot x_v - \gamma \tag{22}$$

$$\text{s.t.} \sum_{u\neq v}\omega(u,v)\cdot y_{uv} \leq \theta\cdot W(v)\cdot x_v \qquad \forall v\in V \tag{23}$$

$$x_u + x_v \leq y_{uv} + 1 \qquad \forall u\neq v \tag{24}$$

$$x_v \in \{0,1\} \qquad \forall v \in V \tag{25}$$

The objective function (22) aims at maximizing a column's reduced cost. The other constraints are identical to the ones of formulation II, except for the fact that only one color is considered.

Solving the pricing problem to optimality at each iteration can be time consuming and unnecessary. We use a greedy algorithm. If it fails to determine a column with positive reduced cost, we then solve the pricing problem to optimality. The above pricing problem can be viewed as a maximum weighted θ-improper independent set problem. The greedy algorithm we use generates a θ-independent set by considering the vertices in non-increasing value $1-\pi_v$, and by sequentially adding them to the θ-independent set, when possible. If the total weight of the obtained set is strictly larger than γ, than a positive reduced cost column has been found.

6 Computational experiments

The instances used to test the proposed algorithms are characterized by the number $n = |V|$ of vertices, the number k of available colors, parameter θ, and the weight functions W and ω. We have run tests with $n \in \{10, 15, 20, 25, 30, 35\}$, $k \in \{1, \ldots, 9\}$, and $\theta \in \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$ (in wireless communications, $\frac{1}{\theta}$ represents the minimum Signal-to-Interference-Ratio (SIR); current technology typically imposes $\frac{1}{\theta} < 2$). All weights $W(v)$ and $\omega(u,v)$ were generated according to a uniform distribution in $[1, n-1]$, so that $\sum_{u \neq v} \omega(u,v)/W(v) \geq 1$ for every vertex v.

Numerical experiments were done using `DipPy` [14], a Python-based modeling language that provides a simple interface to `DIP` [6], an open source framework part of the COIN-OR project [1] for implementing decomposition-based branch-and-bound algorithms for mixed integer linear programs. All tests were run on a MAC OS X machine, with an Intel Core i5, 1.80 GHz.

6.1 The Lovász upper bound versus continuous relaxations

In this section, we compare the value of the optimal solution (z^*) with the Lovász upper bound ($\overline{z}_L$), and with the optimal values of the continuous relaxations of formulations I, II and III (z_R^I, z_R^{II}, z_R^{III}). Although the Lovász upper bound can be computed in polynomial time, it requires the use of the ellipsoid method which can be time consuming. To avoid this, we compute the k largest eigen values of the adjacency matrix with 1's on the diagonal (i.e., the restriction of $\vartheta_k(\bar{H}(G,\theta))$ where $a_{ij} = 0$ if and only if $i \neq j$ and $\{i,j\} \notin E$). Tables 1, 2, 3, and 4 report these values for $\theta = 1, \frac{3}{4}, \frac{1}{2}$, and $\frac{1}{4}$. Every result is an average over 10 tests. We observe that z_R^{II} is better than z_R^I in very few cases and, as proved in Proposition 2, this only occurs with $k = 1$. Clearly, z_R^{III} and $\overline{z}_L$ are the best bounds.

It is easier to analyze these results with the performance profiles drawn in Figure 4. For every $\theta \in \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$ and each value $\overline{z}_L$, z_R^I, z_R^{II}, and z_R^{III}, we indicate the percentage of instances for which the bound lies within a given gap with respect to z^*. For example, we observe that with $\theta = 1$, 46% of the instances have $z_R^{III} = z^*$, while this is the case for 81% of the instances with $\theta = 0.75$, and 87% with $\theta = 0.5$ and 0.25. The worst gap between z_R^{III} and z^* is 6%. For comparison, z_R^I and z_R^{II} are never equal to z^* and only 19% of the instances with $\theta = 1$ have a gap smaller than 20%. This is in accordance with the theoretical results of Section 2.2 in which it was proved that formulation III is tighter than formulations I and II. As shown in Proposition 2, formulations I and II have similar profiles since z_R^{II} can be strictly smaller than z_R^I only for $k = 1$.

We observe that the Lovász upper bound curves are below the z_R^{III} curves and above the z_R^I and z_R^{II} curves. While $\overline{z}_L$ is slightly better than z_R^I and z_R^{II} with $\theta = 1$, it improves when θ decreases. For $\theta = 0.25$, the worst gap between $\overline{z}_L$ and z^* is 15%, while the gap between z_R^{II} and z^* is smaller than 15% for only 2% of the instances. Since the Lovász upper bound appears as tighter than the optimal value of the continuous relaxations of formulation I and II, it is likely that using this bound during a branch-and-bound procedure will reduce the number of vertices, and consequently the total computing time.

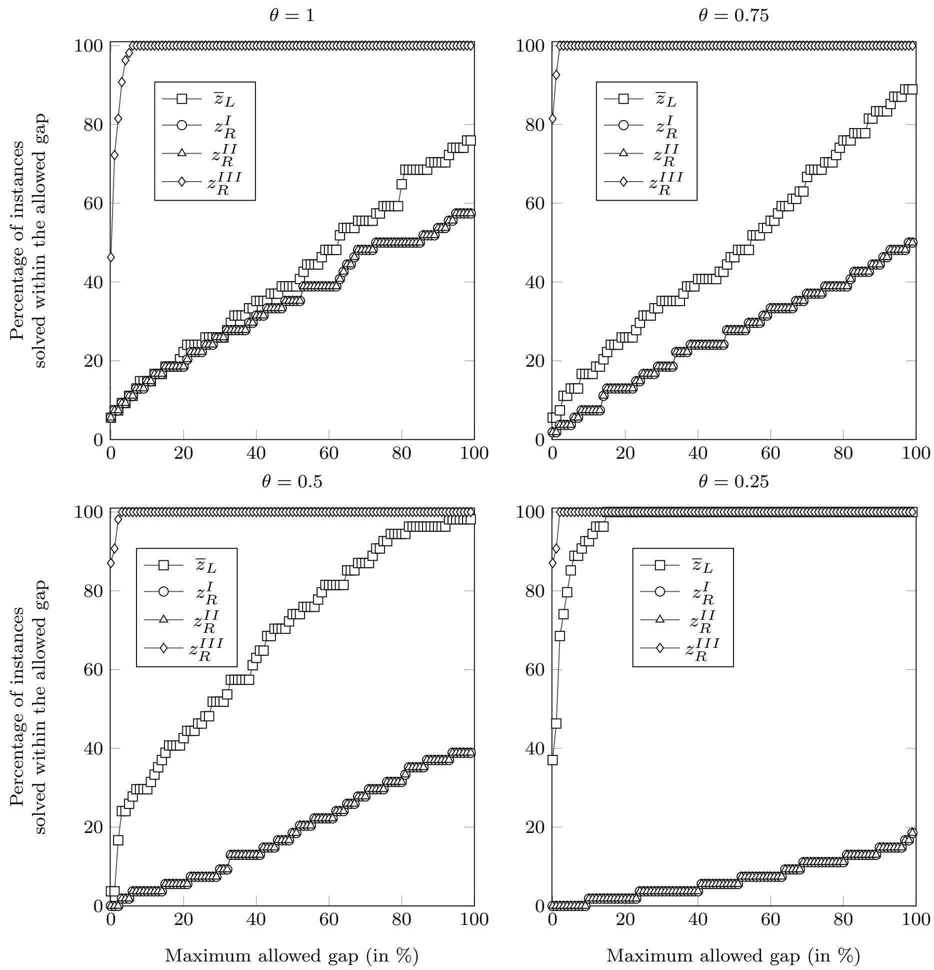

Figure 4: Performance profiles of the results contained in Tables 1–4.

6.2 Comparison of branch-and-cut and branch-and-bound strategies

In Tables 5 and 6, we compare the computing times and the number of branching nodes in the branch-and-cut and branch-and-price algorithms for the following strategies:

DEF	This is the default strategy for formulation II, that uses the Dip solver which has its own library of cuts.
L	The Lovász upper bound is added to the default strategy.
L-SI	Symmetry breaking and incompatibility constraints (15), (16), and (17) are added to strategy L.
L-SI-C	Cover constraints (18) are added to L-SI.
L-SI-W	Weight constraints (19) are added to L-SI.
L-SI-K	Clique constraints (20) are added to L-SI.
L-SI-O	Odd cycle constraints (21) are added to L-SI.
ALL	Constraints (18), (19), (20), and (21) are added to L-SI.
BP	Branch-and-price with an exact solution for the pricing problem.
BP-H	Branch-and-price with a heuristic algorithm for the pricing problem.

There is a trade-off between reducing the computing time and adding cuts to an integer programming formulation. Indeed, generating more cuts tends to increase the time taken to process each node of a branch-and-cut procedure, but also tightens the formulation and thus prevents exploring useless branches. Constraints (18)–(21) are valid for every color in $\{1, \ldots, k\}$, but in order to reduce the computing time, we have decided to choose a color at random each time we get a new optimal solution to a linear relaxation, and to try to produce a violated constraint for that chosen color.

Table 5 reports results obtained for graphs with $n \in \{10, 15, 20\}$ vertices, $k \in \{1, \ldots, 9\}$ colors, and $\theta = 1$. For every algorithm, we report the total number of branching nodes, as well as the total computing time (in seconds). All results are averages over 10 graphs. A limit of 900 seconds was given to each algorithm, and the symbol "-" is used to indicate that, for at least one of the 10 instances, no proven optimal solution could be generated during the allotted time. Table 6 gives the same results for $\theta = 0.25$.

We observe in Table 5 that the Lovász upper bound reduces the number of branching nodes in most cases, but most significantly, adding it with symmetry breaking and incompatibility constraints turns out to drastically decrease the number of nodes, and the computing times. For example, with $(n, k) = (10, 6)$, the average number of nodes with the default strategy is equal to 24153.9, while only 545.2 nodes are necessary with strategy L-SI, and this allows to decrease the computing time from 133.4 seconds to 3 seconds. We also observe that adding cuts generally decreases the number of branching nodes and computing times even more. For example, considering again the graphs with $(n, k) = (10, 6)$, the average number of nodes is reduced to 13.9 with L-SI-K, and the computing time is then reduced to 1 second. This is however not always the case, since adding cuts sometimes increases the computing time and/or the number of branching nodes. For example, for $(n, k) = (15, 2)$, L-SI-K needs 7.9 seconds to visit 117.5 nodes, while strategy L visits 1134.7 nodes in 5.7 seconds. Also, for $(n, k) = (15, 4)$, L-SI-K needs 3.6 seconds to visit 67.7 nodes, while ALL (which has more cuts than L-SI-K) visits 897 nodes in 33.6 seconds. However, the computing times, when using cuts, always remain competitive in comparison with the default strategy.

We also observe that solving the problem with a branch-and-price algorithm, via formulation III, is an efficient strategy, as computing times are very small when compared to the branch-and-cut strategies. This is mainly due to the fact that the problem is often solved at the root node of the branching scheme. Solving the subproblem heuristically slightly improves computing times, but does not appear as a key element.

As already noticed in the previous section, the Lovász upper bound gets closer to the optimal solution when θ decreases. This can be observed in Table 6 where, for example, the 90747.2 nodes visited by the default strategy with $(n, k) = (10, 5)$ are reduced to 7.6 by only introducing the Lovász upper bound.

Note that the computing times are increasing with k when k has a small value, while the problem is much easier to solve with large values of k, since the upper bound n is then often the optimal value. This is illustrated in Figure 5, where we indicate the computing times of DEF, L, L-SI-K, and BP-H, for graph with 10 vertices and $\theta = 1$.

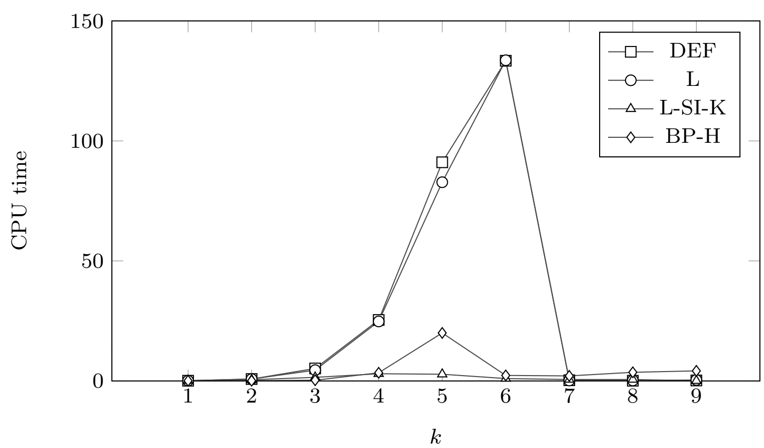

Figure 5: Computing times for $(n, \theta) = (10, 1)$**.**

The winning strategies are clearly L-SI-K and BP-H, which are the only algorithms able to find optimal solutions for all (n, k) with $n = 20$, $k \in \{1, \ldots, 9\}$ and $\theta = 0.25$.

7 Conclusion

We have developed and compared three integer programming formulations for the (θ, k)-PDWICP problem. Valid inequalities have been proposed to cut solutions obtained by solving continuous relaxations of the original formulations. We have also shown how to adapt the Lovász $\vartheta(G)$ function to obtain an upper bound on the optimal solution value. Numerical tests have shown that these elements, embedded in a branch-and-bound procedure, are important features that delay the exponential growth of the computing time. We have also analyzed a branch-and-price scheme that turns out to be competitive with the best branch-and-cut strategy. Numerical experiments have however shown that the computing time increases exponentially with the number of vertices and colors, making all these algorithms impractical for real-life problems. Indeed, while the (θ, k)-PDWICP can be used to model cellular channel allocation problems in wireless communication networks, such problems typically have hundreds or even thousands of vertices, dozens of colors, and the computing times must be nearly instantaneous. It is clear that heuristics are then more appropriate. Such fast algorithms have been developed, for example in [7], and the quality of the produced solutions can now be validated, with the proposed exact methods, on medium scale instances with few dozens of vertices.

Appendix

Table 1: The Lovász upper bound $\overline{z}_L$ versus the optimal values z_R^I, z_R^{II}, z_R^{III} of the continuous relaxations for instances with $\theta = 1$.

(n,k)	$(10,1)$	$(10,2)$	$(10,3)$	$(10,4)$	$(10,5)$	$(10,6)$	$(10,7)$	$(10,8)$	$(10,9)$
z^*	2.1	4.1	6.1	7.8	9.1	9.8	10.0	10.0	10.0
$\overline{z}_L$	3.8	6.3	8.1	9.3	9.8	10.0	10.0	10.0	10.0
z_R^I	5.3	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
z_R^{II}	5.3	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
z_R^{III}	2.1	4.1	6.1	7.9	9.2	9.8	10.0	10.0	10.0
(n,k)	$(15,1)$	$(15,2)$	$(15,3)$	$(15,4)$	$(15,5)$	$(15,6)$	$(15,7)$	$(15,8)$	$(15,9)$
z^*	2.5	4.7	6.7	8.7	10.5	12.3	13.5	14.4	14.9
$\overline{z}_L$	5.3	8.2	10.6	12.5	14.0	14.6	15.0	15.0	15.0
z_R^I	7.8	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0
z_R^{II}	7.7	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0
z_R^{III}	2.5	4.7	6.7	8.7	10.6	12.3	13.6	14.5	14.9
(n,k)	$(20,1)$	$(20,2)$	$(20,3)$	$(20,4)$	$(20,5)$	$(20,6)$	$(20,7)$	$(20,8)$	$(20,9)$
z^*	2.9	5.8	8.3	10.3	12.3	14.3	15.9	17.5	18.7
$\overline{z}_L$	8.3	11.9	14.9	17.3	19.4	19.9	20.0	20.0	20.0
z_R^I	10.3	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
z_R^{II}	10.3	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
z_R^{III}	2.9	5.8	8.4	10.4	12.4	14.4	16.1	17.7	18.8
(n,k)	$(25,1)$	$(25,2)$	$(25,3)$	$(25,4)$	$(25,5)$	$(25,6)$	$(25,7)$	$(25,8)$	$(25,9)$
z^*	3.1	6.0	8.6	10.9	13.0	15.0	17.0	19.0	20.8
$\overline{z}_L$	9.0	12.8	16.6	19.6	22.4	24.4	24.9	25.0	25.0
z_R^I	12.8	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0
z_R^{II}	12.7	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0
z_R^{III}	3.1	6.0	8.7	10.9	13.1	15.1	17.1	19.1	20.9
(n,k)	$(30,1)$	$(30,2)$	$(30,3)$	$(30,4)$	$(30,5)$	$(30,6)$	$(30,7)$	$(30,8)$	$(30,9)$
z^*	3.0	6.0	8.9	11.7	13.9	15.8	17.9	19.7	21.8
$\overline{z}_L$	11.1	15.7	19.6	23.1	26.1	28.3	29.3	29.9	30.0
z_R^I	15.3	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
z_R^{II}	15.2	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
z_R^{III}	3.0	6.0	8.9	11.8	14.2	16.4	18.4	20.4	22.3
(n,k)	$(35,1)$	$(35,2)$	$(35,3)$	$(35,4)$	$(35,5)$	$(35,6)$	$(35,7)$	$(35,8)$	$(35,9)$
z^*	3.0	6.0	9.0	12.0	14.5	16.9	18.9	21.3	22.9
$\overline{z}_L$	12.2	17.5	21.7	25.8	29.5	32.5	34.2	34.7	35.0
z_R^I	17.8	35.0	35.0	35.0	35.0	35.0	35.0	35.0	35.0
z_R^{II}	17.7	35.0	35.0	35.0	35.0	35.0	35.0	35.0	35.0
z_R^{III}	3.0	6.0	9.0	12.0	14.8	17.3	19.7	22.0	24.1

Table 2: The Lovász upper bound $\overline{z}_L$ versus the optimal values z_R^I, z_R^{II}, z_R^{III} of the continuous relaxations for instances with $\theta = 0.75$.

(n,k)	$(10,1)$	$(10,2)$	$(10,3)$	$(10,4)$	$(10,5)$	$(10,6)$	$(10,7)$	$(10,8)$	$(10,9)$
z^*	2.0	3.9	5.5	6.8	7.8	8.8	9.5	9.9	10.0
$\overline{z}_L$	2.5	4.5	5.9	7.0	8.0	8.9	9.5	9.9	10.0
z_R^I	5.2	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
z_R^{II}	5.2	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
z_R^{III}	2.0	3.9	5.5	6.8	7.8	8.8	9.5	9.9	10.0
(n,k)	$(15,1)$	$(15,2)$	$(15,3)$	$(15,4)$	$(15,5)$	$(15,6)$	$(15,7)$	$(15,8)$	$(15,9)$
z^*	2.2	4.2	6.2	8.2	9.8	11.2	12.2	13.2	14.0
$\overline{z}_L$	3.4	6.2	8.4	10.1	11.6	12.8	13.6	14.2	14.6
z_R^I	7.7	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0
z_R^{II}	7.7	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0
z_R^{III}	2.2	4.2	6.2	8.2	9.9	11.3	12.4	13.4	14.2
(n,k)	$(20,1)$	$(20,2)$	$(20,3)$	$(20,4)$	$(20,5)$	$(20,6)$	$(20,7)$	$(20,8)$	$(20,9)$
z^*	2.5	4.5	6.5	8.5	10.5	12.5	14.5	16.0	17.4
$\overline{z}_L$	4.9	8.4	10.9	13.6	15.6	17.4	18.8	19.6	19.8
z_R^I	10.2	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
z_R^{II}	10.2	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
z_R^{III}	2.5	4.5	6.5	8.5	10.5	12.5	14.5	16.1	17.4
(n,k)	$(25,1)$	$(25,2)$	$(25,3)$	$(25,4)$	$(25,5)$	$(25,6)$	$(25,7)$	$(25,8)$	$(25,9)$
z^*	2.6	5.0	7.0	9.0	11.0	13.0	15.0	16.9	18.7
$\overline{z}_L$	5.4	9.4	12.6	15.3	17.9	20.1	21.8	23.0	23.8
z_R^I	12.7	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0
z_R^{II}	12.7	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0
z_R^{III}	2.6	5.0	7.0	9.0	11.0	13.0	15.0	16.9	18.7
(n,k)	$(30,1)$	$(30,2)$	$(30,3)$	$(30,4)$	$(30,5)$	$(30,6)$	$(30,7)$	$(30,8)$	$(30,9)$
z^*	2.8	5.2	7.2	9.2	11.2	13.2	15.2	17.2	19.1
$\overline{z}_L$	7.0	11.1	14.7	17.8	20.9	23.6	25.7	27.7	28.8
z_R^I	15.2	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
z_R^{II}	15.2	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
z_R^{III}	2.8	5.2	7.2	9.2	11.2	13.2	15.2	17.2	19.1
(n,k)	$(35,1)$	$(35,2)$	$(35,3)$	$(35,4)$	$(35,5)$	$(35,6)$	$(35,7)$	$(35,8)$	$(35,9)$
z^*	3.0	5.8	8.3	10.6	12.7	14.7	16.7	18.7	20.7
$\overline{z}_L$	7.2	11.7	16.0	19.3	22.6	25.6	28.5	31.0	32.7
z_R^I	17.7	35.0	35.0	35.0	35.0	35.0	35.0	35.0	35.0
z_R^{II}	17.7	35.0	35.0	35.0	35.0	35.0	35.0	35.0	35.0
z_R^{III}	3.0	5.8	8.3	10.6	12.7	14.8	16.8	18.8	20.8

Table 3: The Lovász upper bound $\overline{z}_L$ versus the optimal values z_R^I, z_R^{II}, z_R^{III} of the continuous relaxations for instances with $\theta = 0.5$.

(n,k)	$(10,1)$	$(10,2)$	$(10,3)$	$(10,4)$	$(10,5)$	$(10,6)$	$(10,7)$	$(10,8)$	$(10,9)$
z^*	1.8	3.5	4.7	5.7	6.7	7.7	8.7	9.5	9.8
$\overline{z}_L$	1.9	3.6	4.8	5.8	6.8	7.8	8.8	9.5	9.8
z_R^I	5.1	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
z_R^{II}	5.1	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
z_R^{III}	1.8	3.5	4.8	5.8	6.8	7.8	8.8	9.5	9.8
(n,k)	$(15,1)$	$(15,2)$	$(15,3)$	$(15,4)$	$(15,5)$	$(15,6)$	$(15,7)$	$(15,8)$	$(15,9)$
z^*	2.0	4.0	5.9	7.2	8.3	9.3	10.3	11.3	12.3
$\overline{z}_L$	2.3	4.5	6.3	7.5	8.5	9.5	10.5	11.5	12.5
z_R^I	7.6	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0
z_R^{II}	7.6	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0
z_R^{III}	2.0	4.0	5.9	7.2	8.3	9.3	10.3	11.3	12.3
(n,k)	$(20,1)$	$(20,2)$	$(20,3)$	$(20,4)$	$(20,5)$	$(20,6)$	$(20,7)$	$(20,8)$	$(20,9)$
z^*	2.2	4.2	6.2	8.1	10.0	11.7	12.9	14.1	15.1
$\overline{z}_L$	3.1	6.0	8.2	10.3	12.0	13.5	14.7	15.7	16.7
z_R^I	10.1	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
z_R^{II}	10.1	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
z_R^{III}	2.2	4.2	6.2	8.1	10.0	11.7	12.9	14.1	15.1
(n,k)	$(25,1)$	$(25,2)$	$(25,3)$	$(25,4)$	$(25,5)$	$(25,6)$	$(25,7)$	$(25,8)$	$(25,9)$
z^*	2.0	4.0	6.0	8.0	10.0	12.0	13.8	15.2	16.5
$\overline{z}_L$	3.5	6.6	9.4	11.9	14.0	15.8	17.3	18.7	19.8
z_R^I	12.6	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0
z_R^{II}	12.6	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0
z_R^{III}	2.0	4.0	6.0	8.0	10.0	12.0	13.8	15.3	16.6
(n,k)	$(30,1)$	$(30,2)$	$(30,3)$	$(30,4)$	$(30,5)$	$(30,6)$	$(30,7)$	$(30,8)$	$(30,9)$
z^*	2.2	4.2	6.2	8.2	10.2	12.2	14.2	16.1	17.9
$\overline{z}_L$	3.8	7.4	10.4	13.0	15.6	17.6	19.7	21.3	22.8
z_R^I	15.1	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
z_R^{II}	15.1	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
z_R^{III}	2.2	4.2	6.2	8.2	10.2	12.2	14.2	16.1	17.9
(n,k)	$(35,1)$	$(35,2)$	$(35,3)$	$(35,4)$	$(35,5)$	$(35,6)$	$(35,7)$	$(35,8)$	$(35,9)$
z^*	2.1	4.1	6.1	8.1	10.1	12.1	14.1	16.1	18.1
$\overline{z}_L$	4.4	7.9	11.1	13.9	16.6	19.1	21.1	23.0	25.1
z_R^I	17.6	35.0	35.0	35.0	35.0	35.0	35.0	35.0	35.0
z_R^{II}	17.6	35.0	35.0	35.0	35.0	35.0	35.0	35.0	35.0
z_R^{III}	2.1	4.1	6.1	8.1	10.1	12.1	14.1	16.1	18.1

Table 4: The Lovász upper bound $\overline{z}_L$ versus the optimal values z_R^I, z_R^{II}, z_R^{III} of the continuous relaxations for instances with $\theta = 0.25$.

(n,k)	$(10,1)$	$(10,2)$	$(10,3)$	$(10,4)$	$(10,5)$	$(10,6)$	$(10,7)$	$(10,8)$	$(10,9)$
z^*	1.1	2.1	3.1	4.1	5.1	6.1	7.1	8.1	9.1
$\overline{z}_L$	1.1	2.1	3.1	4.1	5.1	6.1	7.1	8.1	9.1
z_R^I	5.1	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
z_R^{II}	5.1	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
z_R^{III}	1.1	2.1	3.1	4.1	5.1	6.1	7.1	8.1	9.1
(n,k)	$(15,1)$	$(15,2)$	$(15,3)$	$(15,4)$	$(15,5)$	$(15,6)$	$(15,7)$	$(15,8)$	$(15,9)$
z^*	1.8	2.9	3.9	4.9	5.9	6.9	7.9	8.9	9.9
$\overline{z}_L$	1.8	2.9	3.9	4.9	5.9	6.9	7.9	8.9	9.9
z_R^I	7.6	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0
z_R^{II}	7.6	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0
z_R^{III}	1.8	2.9	3.9	4.9	5.9	6.9	7.9	8.9	9.9
(n,k)	$(20,1)$	$(20,2)$	$(20,3)$	$(20,4)$	$(20,5)$	$(20,6)$	$(20,7)$	$(20,8)$	$(20,9)$
z^*	2.0	3.9	5.1	6.1	7.1	8.1	9.1	10.1	11.1
$\overline{z}_L$	2.1	4.0	5.2	6.2	7.2	8.2	9.2	10.2	11.2
z_R^I	10.1	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
z_R^{II}	10.1	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
z_R^{III}	2.0	3.9	5.2	6.2	7.2	8.2	9.2	10.2	11.2
(n,k)	$(25,1)$	$(25,2)$	$(25,3)$	$(25,4)$	$(25,5)$	$(25,6)$	$(25,7)$	$(25,8)$	$(25,9)$
z^*	2.0	3.9	5.6	7.0	8.2	9.4	10.5	11.5	12.5
$\overline{z}_L$	2.0	4.0	5.7	7.1	8.3	9.5	10.6	11.6	12.6
z_R^I	12.6	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0
z_R^{II}	12.6	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0
z_R^{III}	2.0	3.9	5.6	7.0	8.2	9.4	10.5	11.5	12.5
(n,k)	$(30,1)$	$(30,2)$	$(30,3)$	$(30,4)$	$(30,5)$	$(30,6)$	$(30,7)$	$(30,8)$	$(30,9)$
z^*	2.0	4.0	5.9	7.4	8.7	9.8	10.8	11.8	12.8
$\overline{z}_L$	2.0	4.3	6.2	7.7	9.0	10.0	11.0	12.0	13.0
z_R^I	15.1	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
z_R^{II}	15.1	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
z_R^{III}	2.0	4.0	5.9	7.4	8.7	9.8	10.8	11.8	12.8
(n,k)	$(35,1)$	$(35,2)$	$(35,3)$	$(35,4)$	$(35,5)$	$(35,6)$	$(35,7)$	$(35,8)$	$(35,9)$
z^*	2.0	4.0	6.0	7.9	9.6	11.2	12.4	13.5	14.6
$\overline{z}_L$	2.3	4.6	6.7	8.7	10.4	11.7	12.9	14.0	15.1
z_R^I	17.6	35.0	35.0	35.0	35.0	35.0	35.0	35.0	35.0
z_R^{II}	17.6	35.0	35.0	35.0	35.0	35.0	35.0	35.0	35.0
z_R^{III}	2.0	4.0	6.0	7.9	9.6	11.2	12.4	13.5	14.6

Table 5: Numerical experiments with $\theta = 1$.

(n, k)	$(10, 1)$		$(10, 2)$		$(10, 3)$		$(10, 4)$		$(10, 5)$		$(10, 6)$		$(10, 7)$		$(10, 8)$		$(10, 9)$	
Strategy	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time
DEF	33.3	0.1	323.6	0.8	1855.3	5.2	7824.4	25.4	20541.2	91.1	24153.9	133.4	40.2	0.3	23.2	0.1	24.1	0.2
L	26.9	0.1	376.0	0.8	1740.0	4.5	7624.3	24.8	20341.7	82.8	23710.9	133.6	27.1	0.2	23.7	0.2	31.6	0.4
L-SI	16.3	0.1	65.8	0.1	216.0	0.7	537.9	2.2	984.4	4.5	545.2	3.0	24.0	0.5	11.3	0.3	2.2	0.1
L-SI-C	15.4	0.1	60.3	0.2	221.9	1.1	482.3	3.0	1023.0	7.0	343.1	3.0	16.4	0.6	11.3	0.5	2.2	0.1
L-SI-K	10.4	0.1	47.7	0.5	139.0	1.5	233.8	3.0	176.3	2.8	13.9	1.0	9.6	0.6	7.0	0.6	2.1	0.1
L-SI-O	16.7	0.1	91.3	0.6	234.8	1.9	481.5	4.6	812.2	8.2	421.0	5.3	15.3	0.7	10.3	0.6	2.2	0.1
L-SI-W	16.8	0.1	63.3	0.3	231.5	1.8	501.5	3.9	976.4	8.1	409.4	3.8	26.8	0.9	11.1	0.5	2.2	0.1
ALL	16.7	0.1	89.4	0.6	211.2	1.7	519.4	5.0	894.2	9.2	402.7	5.0	16.7	0.7	12.6	0.6	1.9	0.1
BP	1.0	0.1	1.0	0.5	1.0	1.3	14.8	15.3	78.4	72.3	3.2	11.8	3.5	14.3	4.2	15.9	3.8	19.2
BP-H	1.0	0.1	1.0	0.1	1.0	0.3	6.7	3.4	44.5	20.0	1.6	2.3	2.1	2.1	3.4	3.6	4.6	4.2

(n, k)	$(15, 1)$		$(15, 2)$		$(15, 3)$		$(15, 4)$		$(15, 5)$		$(15, 6)$		$(15, 7)$		$(15, 8)$		$(15, 9)$	
Strategy	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time
DEF	90.1	0.4	1663.6	9.4	17626.2	121.0	-	-	-	-	-	-	-	-	-	-	-	-
L	68.2	0.3	1134.7	5.7	13134.1	73.8	-	-	-	-	-	-	-	-	-	-	-	-
L-SI	36.1	0.1	264.6	1.3	1848.7	12.0	1108.0	8.7	7522.0	84.7	-	-	-	-	-	-	-	-
L-SI-C	33.4	0.2	256.3	2.1	1710.3	17.7	1196.0	19.4	6945.0	112.5	-	-	-	-	-	-	-	-
L-SI-K	17.1	1.4	117.5	7.9	1443.9	67.7	9.0	3.6	19137.5	824.3	15008.0	778.8	-	-	-	-	-	-
L-SI-O	37.5	0.7	252.0	5.0	1823.8	34.3	702.0	21.5	9811.0	239.6	14.0	8.3	-	-	-	-	-	-
L-SI-W	35.8	0.5	236.0	3.3	1922.3	30.1	1021.0	19.9	6794.0	126.5	-	-	-	-	-	-	-	-
ALL	37.5	0.8	260.9	5.3	2083.7	38.2	897.0	33.6	13237.0	312.9	-	-	-	-	-	-	-	-
BP	1.0	0.1	1.0	0.9	1.0	1.9	1.0	3.6	1.0	6.7	1.0	9.1	1.0	12.1	1.0	15.0	1.0	17.8
BP-H	1.0	0.1	1.0	0.1	1.0	0.4	1.0	1.8	1.0	2.0	1.0	2.8	1.1	3.2	1.0	3.5	1.0	5.7

(n, k)	$(20, 1)$		$(20, 2)$		$(20, 3)$		$(20, 4)$		$(20, 5)$		$(20, 6)$		$(20, 7)$		$(20, 8)$		$(20, 9)$	
Strategy	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time
DEF	95.5	1.0	3087.5	44.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-
L	97.8	0.9	2657.3	43.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
L-SI	61.2	0.4	578.7	6.1	6549.3	67.5	-	-	-	-	-	-	-	-	-	-	-	-
L-SI-C	68.0	0.8	601.1	9.8	6075.6	103.4	-	-	-	-	-	-	-	-	-	-	-	-
L-SI-K	50.4	14.1	386.4	86.5	3780.5	609.3	-	-	-	-	-	-	-	-	-	-	-	-
L-SI-O	69.9	3.5	726.1	29.4	7253.9	254.9	-	-	-	-	-	-	-	-	-	-	-	-
L-SI-W	67.3	1.9	608.9	18.2	6876.9	174.7	-	-	-	-	-	-	-	-	-	-	-	-
ALL	69.9	3.5	551.0	25.2	8330.0	301.1	-	-	-	-	-	-	-	-	-	-	-	-
BP	1.0	4.1	1.0	10.6	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BP-H	1.0	1.7	1.0	5.2	3.4	54.3	11.4	233.0	14.7	354.3	-	-	-	-	-	-	-	-

Table 6: Numerical experiments with $\theta = 0.25$.

(n,k)	$(10,1)$		$(10,2)$		$(10,3)$		$(10,4)$		$(10,5)$		$(10,6)$		$(10,7)$		$(10,8)$		$(10,9)$	
Strategy	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time
DEF	21.9	0.1	228.3	0.4	2209.5	5.2	16242.0	56.2	90747.2	429.4	-	-	-	-	-	-	-	-
L	2.2	0.1	1.4	0.1	3.7	0.1	3.9	0.1	7.6	0.1	32.5	0.2	29.3	0.2	37.3	0.3	31.7	0.2
L-SI	1.5	0.1	1.7	0.1	1.5	0.1	1.4	0.1	2.1	0.0	2.0	0.1	1.5	0.1	1.7	0.0	1.3	0.1
L-SI-C	1.5	0.1	1.7	0.1	1.5	0.0	1.4	0.1	2.1	0.1	2.0	0.1	1.5	0.1	1.7	0.1	1.3	0.1
L-SI-K	1.0	0.1	1.0	0.1	1.1	0.1	1.1	0.1	1.3	0.1	1.7	0.1	1.1	0.1	1.1	0.1	1.3	0.1
L-SI-O	1.5	0.1	1.7	0.1	1.5	0.1	1.3	0.1	2.0	0.1	2.0	0.1	1.5	0.1	1.9	0.1	1.3	0.1
L-SI-W	1.5	0.1	1.7	0.1	1.5	0.1	1.4	0.1	2.1	0.1	2.0	0.1	1.5	0.1	1.7	0.1	1.3	0.1
ALL	1.5	0.0	1.7	0.0	1.3	0.1	1.4	0.1	2.1	0.1	1.1	0.1	1.5	0.1	1.7	0.1	1.3	0.1
BP	1.0	0.1	1.0	0.0	1.0	0.15	1.0	0.2	1.0	0.9	1.0	1.1	1.0	1.2	1.0	1.8	1.0	2.0
BP-H	1.0	0.1	1.0	0.1	1.0	0.1	1.0	0.1	1.0	0.3	1.0	0.3	1.0	0.7	1.0	0.7	1.0	0.8

(n,k)	$(15,1)$		$(15,2)$		$(15,3)$		$(15,4)$		$(15,5)$		$(15,6)$		$(15,7)$		$(15,8)$		$(15,9)$	
Strategy	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time
DEF	29.6	0.1	400.3	2.1	6344.4	28.3	70088.9	373.1	-	-	-	-	-	-	-	-	-	-
L	9.3	0.1	26.1	0.2	21.6	0.2	23.0	0.3	37.1	0.6	43.4	0.7	57.1	0.9	68.3	1.7	74.2	2.1
L-SI	6.2	0.1	8.3	0.1	10.4	0.1	37.2	0.5	40.1	0.7	19.8	0.4	17.4	0.5	28.1	1.1	13.7	1.1
L-SI-C	6.2	0.1	8.3	0.1	10.4	0.2	37.2	0.7	40.1	1.0	19.8	0.6	17.4	0.7	28.1	1.5	13.7	1.4
L-SI-K	1.0	0.2	1.9	1.1	3.9	2.5	2.3	1.8	4.6	3.2	3.4	2.2	6.8	4.6	3.4	2.8	8.3	5.6
L-SI-O	6.2	0.1	8.0	0.3	10.4	0.4	38.4	1.3	48.2	2.0	20.5	1.0	15.8	1.0	25.2	1.7	13.7	1.7
L-SI-W	6.2	0.1	8.0	0.1	10.4	0.2	32.5	0.6	40.1	1.0	21.0	0.6	12.8	0.6	15.5	0.9	19.4	1.9
ALL	6.2	0.1	9.0	0.3	9.4	0.4	24.6	0.9	35.3	1.5	18.3	0.9	16.5	1.0	25.0	1.8	11.4	1.4
BP	1.0	0.2	1.0	0.8	1.0	1.6	1.0	4.1	1.0	7.2	1.0	8.6	1.0	11.6	1.0	14.1	1.0	16.8
BP-H	1.0	0.1	1.0	0.4	1.0	0.9	1.0	1.6	1.0	2.0	1.0	2.6	1.0	2.7	1.0	3.3	1.0	5.4

(n,k)	$(20,1)$		$(20,2)$		$(20,3)$		$(20,4)$		$(20,5)$		$(20,6)$		$(20,7)$		$(20,8)$		$(20,9)$	
Strategy	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time
DEF	32.4	0.2	476.5	6.9	-	-	-	-	-	-	-	-	-	-	-	-	-	-
L	19.0	0.1	126.6	0.8	1435.9	9.7	-	-	-	-	-	-	-	-	-	-	-	-
L-SI	7.9	0.1	45.6	0.4	160.2	1.7	415.7	6.3	1620.7	43.5	-	-	-	-	-	-	-	-
L-SI-C	7.9	0.1	45.6	0.6	160.2	2.8	415.7	9.6	1620.7	59.8	-	-	-	-	-	-	-	-
L-SI-K	1.3	3.1	2.7	9.1	8.4	18.5	10.7	27.0	9.1	30.6	14.3	42.4	18.6	58.8	69.6	124.7	53.3	101.3
L-SI-O	7.7	0.4	39.4	1.8	175.9	8.2	400.5	19.8	1850.5	107.4	-	-	-	-	-	-	-	-
L-SI-W	7.9	0.1	45.6	0.6	160.2	3.0	452.3	12.0	1553.7	48.7	-	-	-	-	-	-	-	-
ALL	7.7	0.4	52.6	2.2	139.7	6.9	499.5	22.4	1115.1	67.0	1528.4	84.4	-	-	-	-	-	-
BP	1.0	0.9	1.0	2.7	2.6	17.4	10.3	55	4.2	50.5	-	-	-	-	-	-	-	-
BP-H	1.0	0.4	1.0	0.9	3.2	6.7	4.3	12.3	5.2	17.9	8.4	33.7	6.1	32.0	7.5	40.2	16.7	96.2

References

[1] Computational infrastructure for operations research. [Online]. Available: http://www.coin-or.org

[2] Julio Araujo, Jean-Claude Bermond, Frédéric Giroire, Frédéric Havet, Dorian Mazauric, and Remigiusz Modrzejewski. Weighted improper colouring. In *Proceedings of the 22nd international conference on Combinatorial Algorithms*, IWOCA'11, pages 1–18, Berlin, Heidelberg, 2011. Springer-Verlag. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-25011-8_1

[3] Claudia Archetti, Nicola Bianchessi, Alain Hertz, Adrien Colombet, and François Gagnon. Directed weighted improper coloring for cellular channel allocation. *Discrete Applied Mathematics*, (182):46–60, 2015.

[4] Cynthia Barnhart, Ellis L Johnson, George L Nemhauser, Martin WP Savelsbergh, and Pamela H Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

[5] Daniel Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979.

[6] Matthew V Galati and Ted K Ralphs. Dip: A framework for decomposition in integer programming. Technical report, COR@L Laboratory, Lehigh University, 2015.

[7] Alain Hertz, Romain Montagné, and François Gagnon. Constructive algorithms for the partial directed weighted improper coloring problem. *Journal of Graph Algorithms and Applications*, 20(2):159–188, 2016.

[8] Konstantinos Kaparis and Adam N Letchford. Separation algorithms for 0-1 knapsack polytopes. *Mathematical programming*, 124(1–2):69–91, 2010.

[9] Frank Thomson Leighton. A graph coloring algorithm for large scheduling problems. *Journal of research of the national bureau of standards*, 84(6):489–506, 1979.

[10] László Lovász. On decomposition of graphs. *Stud. Sci. Math. Hung*, 1:237–238, 1966.

[11] László Lovász. On the shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25(1):1–7, 1979.

[12] Anuj Mehrotra and Michael A Trick. A column generation approach for graph coloring. *Informs Journal on Computing*, 8(4):344–354, 1996.

[13] Giri Narasimhan et al. *The maximum k-colorable subgraph problem.* PhD thesis, University of Wisconsin, Madison, 1989.

[14] Michael O'Sullivan, Qi-Shan Lim, Cameron Walker, Iain Dunning, and Stuart Mitchell. Dippy: A simplified interface for advanced mixed-integer programming. Technical report, University of Auckland, School of Engineering, 2012.

[15] Steffen Rebennack. Stable set problem: Branch & cut algorithms. In *Encyclopedia of Optimization*, pages 3676–3688. Springer, 2009.