Université Sorbonne Paris Nord
Laboratoire d'Informatique de Paris Nord - LIPN
Équipe: Algorithmes et Optimisation Combinatoire - AOC

Co-k-plexes and k-defective coloring: polytopes and algorithms

Thèse de doctorat
présentée par

Alexandre Dupont-Bouillard

à L'Ecole Doctorale Galilée

pour obtenir le grade de
Docteur en Informatique

Quentin Louveaux	Université de Liège	Rapporteur
Marc Pfetsch	Die Technische Universität Darmstadt	Rapporteur
Aurélie Lagoutte	Université Grenoble Alpes	Examinatrice
Christophe Tollu	Université Sorbonne Paris Nord	Examinateur
Annegret Wagler	Université Clermont Auvergne	Examinatrice
Pierre Fouilhoux	Université Sorbonne Paris Nord	Directeur de thèse
Roland Grappe	Université Paris Dauphine	Directeur de thèse
Mathieu Lacroix	Université Sorbonne Paris Nord	Encadrant de thèse

Résumé

Cette thèse propose plusieurs résultats sur les co-k-plexes, qui sont des ensembles de sommets induisant un graphe avec un degré maximum $k-1$.

La première partie de la thèse explore plusieurs caractérisations d'une nouvelle sous-classe de graphes parfaits, appelés graphes contraction parfaits. Ce sont l'ensemble des graphes parfaits restant parfaits après contraction de n'importe quel ensemble d'arêtes. Cette classe de graphes est caractérisée de quatre manières différentes. Parmi ces résultats, un graphe est contraction parfait si et seulement s'il est parfait et que la contraction de n'importe quelle arête préserve sa perfection. Cela permet une caractérisation des graphes contraction parfait en termes de sous-graphes induits interdits, ainsi qu'un algorithme polynomial pour leur reconnaissance. Le utter graphe $u(G)$ d'un graphe G est un graphe dont les stables sont en bijection avec les co-2-plexes de G. Il est démontré que $u(G)$ est parfait si et seulement si G est contraction parfait. Puisque le problème de stable de poids maximum est solvable en temps polynomial sur les graphes parfaits, il en va de même pour le problème de co-2-plex de poids max sur les graphes contraction parfait. Une formulation étendue pour le polytope co-2-plex des graphes contraction parfait est obtenue en considérant le polytope de l'ensemble des stable de son utter graphe. Notez que cette formulation devient compacte pour les graphes chordaux. De plus, avec des contraintes d'intégrité, cette formulation devient une formulation ILP valide pour n'importe quel graphe G, conduisant à une comparaison entre cette formulation ILP étendue et la formulation classique de la littérature. D'un point de vue théorique, cette nouvelle formulation a une relaxation linéaire de meilleure qualité, et d'un point de vue expérimental, plusieurs implémentations de cette nouvelle formulation sont proposées et se révèlent compétitives.

La deuxième partie de la thèse concerne le problème k-defective coloring qui consiste à recouvrir les sommets d'un graphe avec un nombre minimum de co-k-plexes. Ce problème est formulé comme une formulation set covering contenant une variable par co-k-plex, et résolu en utilisant une méthode de branch and price. Un algorithme de branch and price pour le k-defective coloring proposé par Furini et al. est d'abord décrit et mis en œuvre en utilisant le framework SCIP, et plusieurs améliorations sont proposées pour les cas $k = 1, 2$. Pour $k = 1$, des inégalité de rang un de Chvátal-Gomory sont d'abord ajoutées de manière dynamique lors de la résolution de la relaxation linéaire à chaque nœud de l'arbre de branchement. L'ajout de contraintes dans un cadre de génération de colonnes est généralement réalisé lorsque le critère de convergence de la génération de colonnes est atteint, ce qui donne la priorité à la génération de nouvelles colonnes plutôt qu'à de nouvelles lignes. D'autre part, la génération de colonnes est connue pour souffrir du tailing off effect qui consiste en une séquence d'ajout de variables non améliorantes. Pour résoudre ce problème, plusieurs alternances entre les phases de cutting et de pricing sont étudiées et en particulier deux stratégies paramétrées sont conçues pour se débarrasser du tailing off effect. La génération de colonnes peut être stabilisée en utilisant des inégalités optimales duales. Cette technique consiste à ajouter au RMP un ensemble de colonnes supplémentaires correspondant à des inégalités ne coupant aucun point optimal du problème dual. Cette méthode est expérimentalement efficace pour stabiliser la génération de colonnes à chaque nœud de l'arbre de branchement. Différentes façons de combiner les inégalités de renforcement et les inégalités optimales duales sont étudiées. Pour $k = 2$, le problème de pricing se réduit à calculer un problème de co-2-plex pondéré maximum. Par conséquent, une variante de la méthode de branch and price dans laquelle le pricing est résolu en utilisant la formulation ILP de la première partie de la thèse est évaluée expérimentalement. De nouvelles inégalités duales sont proposées pour la stabilisation du cas k=2. Contrairement à ce qui existe dans la littérature, les inégalités duales proposées peuvent couper chaque solution optimale du dual. Cependant, chaque solution primale entière avec des valeurs positives dans les colonnes supplémentaires peut être transformée en une solution entière du problème d'origine avec la même valeur optimale. Une amélioration expérimentale significative est observée pour le problème de 2-defective coloring.

Extended abstract

This thesis proposes several results about co-k-plexes, which are vertex sets inducing a graph with maximum degree $k-1$ and the k-defective coloring which consists in covering the vertices of a graph with a minimum number of co-k-plexes. The thesis is split into two parts, each starting with a dedicated state-of-the-art chapter.

The first part investigates several characterizations of a new subclass of perfect graphs that we call *contraction-perfect graphs*, which is the set of perfect graphs remaining perfect after the contraction of any edge set.

Contractions in perfect graphs We characterize this graph class in four different manners. Among those results, we prove that a graph is contraction-perfect if and only if it is perfect and the contraction of any single edge preserves its perfection. This yields a characterization of contraction-perfect graphs in terms of forbidden induced subgraphs, and a polynomial algorithm to recognize them. On such graphs, solving the maximum weighted co-2-plex problem is doable in polynomial time. This is done by considering what we call the utter graph $u(G)$ of a graph G, whose stable sets are in bijection with the co-2-plexes of G. We show that $u(G)$ is perfect if and only if G is contraction-perfect and as the maximum stable set problem is solvable in polynomial time on perfect graphs [1] we obtain the polynomiality result of the maximum weighted co-2-plex problem on contraction-perfect graphs.

The maximum weighted co-2-plex problem We investigate an extended formulation for the co-2-plex polytope of contraction-perfect graphs obtained by considering the stable set polytope of its utter graph. Note that this formulation becomes compact for chordal graphs. Moreover, with additional integrity constraints, this formulation becomes a valid ILP formulation for any graph G, leading to a comparison between this extended ILP formulation with a formulation of the literature. From a theoretical point of view, our formulation has a better relaxation value than the formulation of the literature, by considering a subset of inequalities obtained when projecting our formulation on the natural variable set. From an experimental point of view, we propose different implementation variants of our formulation and conclude that they are competitive with the one from the literature.

The second part of the thesis is about column generation: this method permits to solve exponential variable set-sized ILPs, by considering a sequence of subsets of variables and recursively solving the associated restricted master problem (RMP) until a convergence criterion is met. We first describe a general column generation method from the literature solving the k-defective coloring proposed by Furini et al. [2]. We then implement this method within the SCIP framework to let us add new techniques. This method is based on a *set covering formulation*, that is an Integer Linear Programming Formulation obtained by considering a $0/1$ matrix M and that has the following form where $\mathbb{1}$ is a column vector of ones:

$$\begin{cases} \min \mathbb{1}^\top x & & \\ \quad Mx & \geq & \mathbb{1} \\ \quad x & \leq & \mathbb{1} \\ \quad x & \geq & 0 \end{cases}$$

We investigate how the results on set covering polytopes behave within this framework for the k-defective coloring for $k \in \{1, 2\}$. Note that, for $k = 1$, this problem is the well-known graph coloring problem.

The graph coloring problem and alternation strategies The latter meets an algorithm proposed by Mehrotra and Trick for the coloring problem [3] in the case $k = 1$. The study of covering polytopes leads us to consider Chvátal-Gomory constraints to improve the case $k = 1$. Adding constraints within a column generation framework is often done in the literature when the column generation convergence criterion is met, which set a priority to generate new columns instead of new rows. On the other hand, column generation is known to suffer from the *tailing off effect* that consists of a sequence of non-improving subsets of variables. To tackle this issue, we tried to alternate between cutting and pricing and in particular proposed two parameterized strategies hoping that cutting earlier would help to get rid of the tailing-off effect. The second improvement of this framework has been proposed in the literature [4] under the name of *dual optimal inequalities*: this technique consists in adding to the RMP a set of columns that are inequalities cutting no optimal points of the dual RMP. This method has been experimentally successful in stabilizing the column generation for the case $k = 1$, we hence tried to combine primal/dual inequalities and alternation strategies in several manners.

Defective colorings and integer linear programming Finally, we investigate whether the results of the first part of the thesis give improvements on this framework for the case $k = 2$. Note that, for a given k, the general framework needs an efficient algorithm to compute an exact maximum weighted co-k-plex: we then use our ILP formulation for solving the co-2-plex problems. For $k = 2$, Chvátal-Gomory constraints do not improve the relaxation value of the associated formulation, but, we propose to use dedicated valid inequalities, called triangle constraints. We also propose a set of dual inequalities for any k and showed that it implied significant improvements for the case $k = 2$. We then applied a technique similar to our work on the case $k = 1$ combining primal and dual inequalities, for the case $k = 2$. Note that the dual inequalities we propose are of a new type that has not been investigated in the literature: they may cut every dual optimal point but, when considered as variables of the primal, they never add integer solutions that cannot be mapped into an integer solution of the original problem with the same optimum value. Surprisingly, these variables interact well with SCIP's primal heuristics. We then obtain a significant experimental improvement for the 2-defective coloring problem.

Finally, we recap the results of the thesis and give perspectives for each of the chapters.

Remerciements

Avant toute chose, je remercie chaleureusement tous les membres du jury pour avoir accepté de lire mon travail ainsi que pour leurs conseils et la pertinence de leurs questions. Tout a commencé lors de ma rencontre avec Pierre Fouilhoux, il était alors un de mes enseignants de master. A cette époque, je m'étais fait une raison quant à l'absence d'algèbre dans mon cursus suite à l'abandon de mes études de mathématiques au profit de l'informatique. Le cours de Pierre sur les approches polyhédrales fût alors une révélation pour moi: cet élégant équilibre entre algorithmique, théorie des graphes et algèbre linéaire confirma mon attrait pour la recherche. Il me fallait alors trouver un stage en laboratoire, Pierre s'est imposé comme l'encadrant idéal non seulement thématiquement mais aussi pour son évidente bienveillance. Pierre accepta de me prendre comme stagiaire et de là commençait notre aventure. Alors que j'étais sûr de vouloir continuer en thèse avec Pierre, il m'incita a candidater à d'autres sujets, avec d'autres encadrants afin d'être sûr de commencer une thèse à la rentrée. C'est alors qu'il me fît rencontrer Roland Grappe et Mathieu Lacroix les deux autres encadrants de ma thèse. Finalement Pierre était recruté comme professeur au LIPN et me proposait de l'y suivre afin d'y collaborer avec Roland, Mathieu et lui. Sans hésiter, je fît confiance à Pierre et c'est un choix sur lequel je ne reviendrai pour rien au monde. Ce trio d'encadrants fût incroyablement riche: riche scientifiquement, personne n'en doute mais il facile d'imaginer que trois chercheurs aient du mal à trouver un équilibre dans leur façon d'encadrer un même étudiant. Ils ont parfaitement réussi cette tâche, chacun amenant sa propre expertise scientifique, mention spéciale pour Pierre qui m'a beaucoup soutenu pour l'enseignement et l'administratif. Merci à tous les trois pour avoir pris à coeur mon encadrement et m'avoir fait découvrir le monde de la recherche.

J'aimerai aussi remercier toutes les personnes qui ont travaillé en B103 et avec qui nous avons partagé tant de choses ... Cette première année de thèse pendant le confinement aurait pu être bien solitaire mais heureusement, mon compagnons des premières heures Théophile était là. Nos échanges scientifiques et pas que rythmèrent nos journées jusqu'à la fin du confinement et au retour de nos collègues, Ghazi, Francesco D., Francesco P. qui devinrent rapidement des amis. Débuta alors ma première collaboration scientifique avec une personne extérieure à mon encadrement: Francesco P. Ce fût un plaisir de travailler avec toi. Pendant la deuxième année, arrivèrent Dasha, Victor, Mathieu, Yue, Thibault puis plus tard Alexandre, Alexis et Jules, autant de personnes que d'horizons scientifiques et que de nouveaux amis. Cette richesse humaine et scientifique n'est en réalité pas limitée aux membre de la B103, le LIPN aura été un environnement idéal pour découvrir la recherche. Les évènements scientifiques internes au laboratoire m'auront permis de m'émerveiller autant de fois que j'y aurais participé. La proximité entre les membres m'aura donné l'impression d'une grande famille. Je remercie amicalement tous les membres du laboratoire.

Je souhaite aussi remercier mes parents pour m'avoir supporté, hébergé et facilité la tâche pendant ces 4 années. Ils auront été des soutiens indéfectibles! Merci Sylvie et Jean-Jacques d'avoir été là, mon attrait pour les sciences me vient en bonne partie de vous. Jordi, Celio, aussi solides qu'on peut l'être sur tous les plans, vous êtes vraiment trop mes potes. Merci Dasha !

Contents

Notations, definitions and mathematical tools

Graph definitions

A graph is a mathematical structure composed of a set of vertices, some linked by edges and some not. We denote such a structure by $G = (V, E)$ meaning that the graph G has vertex set V and edge set E. All the graphs in this thesis are simple and connected meaning that every edge is unique (E is a set) and a path exists between all pair of vertices. Given a graph $G = (V, E)$, we denote its *complement* by $\overline{G} = (V, \overline{E})$, where $\overline{E} = \{uv \in E : uv \notin E,\ u \neq v\}$. We denote by $V(G)$ (resp. $E(G)$) the vertex (resp. edge) set of G. Two vertices u and v are *adjacent* if $uv \in E(G)$. A vertex is *universal* if it is adjacent to all the other vertices. Given a subset of vertices $W \subseteq V$, let $E(W)$ denote the set of edges of G having both endpoints in W and $\delta(W)$ the set of all edges having exactly one endpoint in W. When W is a singleton $\{w\}$, we will simply write $\delta(w)$. We say that the edges in $\delta(w)$ are *incident* to w, and two edges sharing an endpoint are also said *incident*. A *matching* is a set of pairwise nonincident edges. For $F \subseteq E$, let $V(F)$ denote the set of vertices incident to any edge of F. Given $W \subseteq V$, the graph $G[W] = (W, E(W))$ is the *subgraph induced by W in G*. When H is an induced subgraph of G, we say that G *contains* H. Given a vertex $u \in V$, we denote by $N_u = \{w \in V : uw \in E\}$ its *neighborhood* in G, and by $\overline{N}_u = N_u \cup \{u\}$ its *closed neighborhood*. Two vertices u and v are *false twins* if $N_u = N_v$ and *true twins* if $\overline{N}_u = \overline{N}_v$.

A *clique* (resp. *stable set*) is a set of pairwise adjacent (resp. nonadjacent) vertices. A *coloring* consists in associating a color with every vertex such that no two adjacent vertices have the same color. To cover any clique K, it is necessary to use at least $|K|$ colors, which implies that for any graph G, $\omega(G) \leq \chi(G)$. We denote by $\omega(G)$ (resp. $\alpha(G)$) the size of the largest clique (resp. stable set) of G and the *chromatic number* $\chi(G)$ is the minimum number of colors of a coloring. A *path* (resp. *hole*) is a graph induced by a set of vertices $\{v_1, \ldots, v_p\}$ whose edge set is $\{v_i v_{i+1} :\ i = 1, \ldots, p-1\}$ (resp. $\{v_i v_{i+1} :\ i = 1, \ldots, p-1\} \cup \{v_1 v_p\}$ with $p \geq 4$). Note that this definition usually corresponds to induced paths. A subset of vertices induces a path (resp. hole) if its elements can be ordered into a sequence inducing a path (resp. hole). An *antipath* (resp. *antihole*) of G is a path (resp. hole) of $\overline{G}$. The *size* of a graph H is $|V(H)|$, and its *parity* is the parity of $|V(H)|$.

The *contraction* of an edge uv in G consists of deleting u and v, and adding a new vertex w and the edges wz for all $z \in N_u \cup N_v$. This new graph is denoted by G/uv. For $F \subseteq E$, we denote by G/F the graph obtained from G by contracting all the edges in F. The *image* of a vertex v of G in G/F is the vertex of G/F to which v is contracted, and the *image* of a set of edges L is the set of the images of the vertices of $V(L)$ in G/F. An edge uv and a vertex w are *adjacent by contraction* if w is adjacent to x_{uv} in G/uv, where x_{uv} is the image of uv in G/uv. In other words, at least one of uw and vw is in E. Two nonadjacent edges uv and xy are *adjacent by contraction* if contracting both edges results in two adjacent vertices, that is, $\delta(\{u,v\}) \cap \delta(\{x,y\}) \neq \emptyset$. A *minor* of a graph is a subgraph obtained by deleting vertices, edges, or contracting edges.

Given a total order on a finite set of elements, an *interval* is a subset of consecutive elements following that order. *The interval graph $G_{\mathcal{I}}$* of a given finite set of intervals $\mathcal{I}$ is the graph having one vertex per interval in $\mathcal{I}$ and an edge between two vertices if their corresponding intervals have a nonempty intersection. A set of intervals is *nested* if every couple of intervals either has an empty intersection or one is contained in the other. A *minor* of a graph is obtained by deleting vertices, edges, and/or contracting edges.

A *split graph* is a graph whose vertex set can be partitioned into a stable set and a clique. A graph is split if and only if it does not contain C_4 or its complement as an induced subgraph. A *trivially perfect graph* is an interval graph built from a nested interval set. Equivalently, a *trivially perfect graph* is a graph with no hole or path of size 4 as an induced subgraph. For $k \geq 3$, a k-hole-free graph is a graph whose holes are all of size at most k as an induced subgraph. The 3-hole-free graphs are referred to as *chordal graphs*.

A graph G is called *perfect* if $\omega(H) = \chi(H)$ for every induced subgraph H of G. Perfect graphs are recognizable in polynomial time [5]. The strong perfect graph theorem characterizes perfect graphs in terms of forbidden induced subgraphs and will be described in the first chapter of Part I.

A *series-parallel* graph is a graph that is obtained from a single edge by applying recursively one of the two following operations: add a parallel edge to an existing one, replace an edge uv by a new vertex w and two edges uw and vw. The graphs are characterized as those having no clique of size 4 as a minor.

The notion of k-plexes and co-k-plexes will be present in all the thesis so we give it particular attention.

Definition 0.0.1. *A* k-plex *is a set W of vertices inducing a graph where every vertex has a degree at least $|W| - k$.*

Cliques are 1-plexes. This notion was introduced in 1978 by Seidman and Foster [6] to seek communities in a graph with more freedom than when looking for cliques.

Definition 0.0.2. *A* co-k-plex *is the complement of a k-plex. Accordingly, a* co-2-plex *is a set of vertices inducing a subgraph of maximum degree at most one.*

Analogously, stable sets are co-1-plex. The underlying optimization problem is to find a maximum weight k-plex in a given weighted graph. For any fixed k and hence for $k = 1$, this problem is NP-hard [7]. Hence, by complementing the graph, so is the problem of finding a maximum co-2-plex.

Polyhedral definitions

A *half-space* H is a set of points of a d-dimensional space respecting a single linear constraint, *i.e.* $H = \{x \in \mathbb{R}^d | a^T x \leq b\}$ for $a \in \mathbb{R}^d$ and $b \in \mathbb{R}$. A polyhedron $\mathcal{P}$ is the intersection of a finite set of half-spaces $\{H_1, \dots, H_k\}$ where $H_i = \{x \in \mathbb{R}^d | a_i x \leq b_i\}$, *i.e.* if A is the matrix whose lines are a_i for $i \in \{1, \dots, k\}$, and b the vector of b_i for all $i \in \{1, \dots, k\}$, then $\mathcal{P} = \{x \in \mathbb{R}^d | Ax \leq b\}$. The system $Ax \leq b$ is called the *H-representation* of $\mathcal{P}$. A bounded polyhedron is a *polytope*. The *dimension* of a polytope is the dimension of the smallest affine space which contains it.

Let $\mathcal{P} = \{x \in \mathbb{R}^q | Ax \leq b\}$ be a d-dimensional polyhedron, if $d = q$, we say that $\mathcal{P}$ is *full-dimensional*. An inequality $\alpha x \leq \alpha_0$ is called a *facet defining inequality* if $\mathcal{P} \cap \{x \in \mathbb{R}^q \mid \alpha x = \alpha_0\}$ is a $(d-1)$-dimensional polyhedron. Given $x^* \in \mathcal{P}$, let $\overline{A}x \leq \overline{b}$ be the maximal set of linear constraints of the H-representation of $\mathcal{P}$ such that $\overline{A}x^* = \overline{b}$, if $\overline{A}$ has rank d, then x^* is a *vertex* of $\mathcal{P}$. A polytope $\mathcal{P}$ can equivalently be described as the convex hull of a finite set of points, we call such a description a *V-representation* of $\mathcal{P}$.

A cone $\mathcal{C}$ is a set of points satisfying the property that $x \in \mathcal{C}$ implies that $\lambda x \in \mathcal{C}\ \forall \lambda \in \mathbb{R}^+$. A polyhedron whose H-description is $Ax \leq 0$ is called a *polyhedral cone*. In this thesis, all cones will be polyhedral.

Given a set of points $x_1, \dots, x_k \in \mathbb{R}^n$, a *linear combination* is a weighted sum of such points: $\sum_{i=1}^k \lambda_i x_i$ where $\lambda_i \in \mathbb{R}$. A *conic* combination is a linear combination where each λ_i is nonnegative. A *convex* combination is a linear combination where each λ_i is nonnegative and $\sum_{i=1}^k \lambda_i = 1$.

Integer linear programming

An *integer linear program* (ILP) is defined by a set of variables x_i $i \in \{1, \dots, q\} = I$, linear constraints on variables x, a linear function to maximize/minimize, and integrality constraints on variables x:

$$\left\{ \begin{array}{l} \max\ wx \\ Ax\ \leq\ b \\ x_i \in \mathbb{Z}^+ \quad \forall i \in I \end{array} \right.$$

Optimizing a linear function over an H-polyhedron is known to be polynomial in the size of A using the ellipsoid method [8], but optimizing over such a discrete set is much more complicated since the optimal given by the ellipsoid method may not be an integer. It is, actually, NP-hard since knowing if a polyhedron contains an integer point is already NP-complete as 3-CNF-SAT can be reduced to it [9]. We recall that 3-CNF-SAT is the problem of finding an assignment of binary variables verifying a binary formula in conjunctive natural form with at most 3 literals by clauses.

Since a polyhedron $\mathcal{P}$ is a convex set, the optimum of a linear function over $\mathcal{P}$ is attained on one of its vertices. This implies that when all vertices of $\mathcal{P}$ have integer I components, solving any MILP on $\mathcal{P}$ is polynomial with the ellipsoid method.

A linear system $Ax \leq b$ is *Totally Dual Integral* (TDI) if for all $c \in \mathbb{Z}^n$ such that $\begin{cases} \max cx \\ Ax \leq b \end{cases}$ is bounded, the *dual* of such program $\begin{cases} \min by \\ A^T y = c \\ y \geq 0 \end{cases}$ has an integer optimal solution. It is an important concept in MILP since such a system satisfying $b \in \mathbb{Z}^m$ induces an integer polyhedron [10] that is a polyhedron having integer valuated vertices.

A *totally unimodular matrix* is a matrix whose determinant and subdeterminants are all 1,-1 or 0. It is well known that totally unimodular matrices yield TDI systems [11].

When solving an ILP, we often try to find *valid inequalities* that are inequalities cutting no integer points to the constraint set $Ax \leq b$ to try cutting its fractional extreme points. Any conic combination of the constraints $Ax \leq b$ is always *dominated*: it is verified by all points satisfying $Ax \leq b$, but when it contains fractional coefficients it can be strengthened as follows: let $ax \leq \alpha$ be such a combination, $\lfloor a \rfloor x \leq \alpha$ is also a valid inequality, knowing that x must be integer, $\lfloor a \rfloor x \leq \lfloor \alpha \rfloor$ also gives a valid inequality. This constraint is then called a *Chvátal-Gomory constraint*. The (finite) set of all non-dominated such constraints is called the *elementary Chvátal closure*. Chvátal [12] proved that by applying iteratively a finite number of times this procedure, one always obtains the convex-hull of the integer set of points.

Given an ILP formulation $\mathcal{F}$, we call the polytope obtained by replacing integrality constraints by bounds on variables, the *polytope subjacent to $\mathcal{F}$*.

Branch and Bound

A *Branch and Bound* is an exact algorithm solving an optimization problem, it is an enumeration algorithm based on a branching tree whose leaves correspond to solutions of the problem. To avoid enumerating all the solutions, at each node of the tree, local and upper bounds on the optimal solution should be computed. Let us consider a maximization problem: when the local upper bound of some node n_1 is lower or equal to the lower bound of n_2, all the solutions associated with leaves having n_1 as a parent will be dominated by a solution associated with a leaf having n_2 as a parent, the subtree starting at node n_1 can hence be safely pruned.

For such an algorithm to be efficient, it is important to compute good-quality upper-lower bounds. When solving a MILP formulation, the linear relaxation objective value given by any exact linear optimization algorithm (meaning that we ignore the integrality constraints) gives a valid upper bound, while every feasible solution yields a valid lower bound. For this reason, one may be interested in having a linear relaxation objective value close to the objective value of the optimal point satisfying integrality constraints. To do so, additional reinforcement constraints may be added to the MILP to cut the fractional optimal points and hence improve the local upper bounds.

Branch and Cut

A *Branch and Cut algorithm* is used when the MILP formulation we want to solve has an exponential number of constraints or when we want to add reinforcement cuts such as Chvátal-Gomory constraints for example, it is a branch and bound approach to which we add separation rounds. In that case, a subset of constraints is considered until an optimal fractional point x^* for that subset is found. After that, the *separation* round is launched: it consists of finding an inequality (among the ones unconsidered) that is violated by x^*. If no inequality cutting x^* is found, and if x^* does not satisfy the integrality constraints, then it is necessary to *branch*: choose a fractional component x_u of x^* that should be integer, create a subproblem solving the same formulation with the additional constraint $x_u \leq \lfloor x_u^* \rfloor$ and another subproblem with the additional constraint $x_u \geq \lceil x_u^* \rceil$. This can be iterated until the best solution is found.

Sometimes, a polynomial set of constraints/variables and integrality constraints may be sufficient to describe the solution set. But such a formulation, called *compact*, may have a bad relaxation value: the fractional optimum is far from the integer optimum. Considering extended space formulation (adding variables) can sometimes help to have strong formulations by making it easier to formulate strong constraints. It also happens that considering exponential-sized variable sets yields the best exact approach for some problems. The second part of this thesis will highlight a family of problems for which it is the case.

Column generation

When the variable space has an exponential size in the size of the input of a problem, it is classic to use *column generation*. A solution for such a MILP often has a lot of zero components. This remark leads to considering only a subset of variables and iteratively adding new variables until reaching the optimum. The linear program associated with such a subset of variables is called *Restricted Master Problem*. After each resolution of an RMP, a *pricing subproblem* will be solved, consisting of finding an improving variable, which will be added to the RMP. If no improving variable exists, then the optimal of the current RMP is optimal for the original problem. This method works analogously to the simplex method, which adds an improving variable to the current variable base if it exists until no such variable exists; the only difference is that for the simplex method, finding such a variable corresponds to iterating over the variable array. With column generation, this "variable array" has exponential size, which implies solving an optimization problem, and hence, it is necessary to have a dedicated method to find an improving variable that will try not to enumerate all the possible variables. This technique will be described by an example in the first chapter of Part II.

Fourier-Motzkin elimination procedure

The Fourier-Motzkin elimination procedure is an algorithm that, given the H-representation of a polytope $\mathcal{P} = \{(x, y) \in \mathbb{R}^{p \times q} | Ax + By \leq u\}$, permits to compute the H-representation of $proj_x(\mathcal{P})$, where $proj_x(\mathcal{P})$ is the orthogonal projection of $\mathcal{P}$ onto the subspace associated with x. The method consists of recursively removing one variable at a time.

Projecting a single variable

Let us consider that $\mathcal{P} = \{(x, y) \in \mathbb{R}^p \times \mathbb{R} | Ax + yb \leq u\}$ where b is a column vector. Let us partition the indices of A's rows into three sets, I^+, I^-, I^0 where I^+ contains the indices i where $b_i > 0$, I^- where $b_i < 0$ and I^0 where $b_i = 0$. Let A^+, A^-, and A^0 be the submatrices of A whose rows are respectively indexed by I^+, I^-, I^0 and u^+, u^-, u^0 their rights hand sides. Let us now consider the matrix $A^\times$ whose rows are $|b_j|A_i^+ + b_i A_j^-$, $\forall i \in I^+, j \in I^-$. By construction, $b_i|b_j|y + b_i b_j y = 0$, hence, $A^\times$ is obtained from (A, b) by combining the inequalities two by two and such that the y variable disappears. Let us also consider the vector $u^\times$ whose components are equal to $|b_j|u_i + b_i u_j$, $\forall i \in I^+$, $j \in I^-$. The Fourier-Motzkin elimination procedure tells us that:

$$proj_x(\mathcal{P}) = \left\{ x \in \mathbb{R}^p \middle| \begin{array}{ccc} A^0 x & \leq & u^0 \\ A^\times x & \leq & u^\times \end{array} \right\}. \tag{1}$$

This method removes $|I^+| + |I^-|$ rows but adds $|I^+| \times |I^-|$ new rows. It linearly combines every row in I^+ with every row in I^- such that the y variable vanishes. Another way to understand it comes with the following rewriting:

$$proj_x(\mathcal{P}) = \left\{ x \in \mathbb{R}^p \middle| \exists y \in \mathbb{R} \begin{array}{ccc} yb^+ & \leq & u^+ - A^+ x \\ yb^- & \geq & A^- x - u^- \end{array} \right\}. \tag{2}$$

Considering two inequalities of the upside system $yb_i^+ \leq u_i^+ - A_i^+ x$ and $yb_j^- \geq A_j^- x - u_j^-$, respectively combining them with coefficient b_j^-, b_i^+ yields a valid inequality for $proj_x(\mathcal{P})$ whose only nonzeros coefficient are associated with x variables:

$$b_i^+ A_j^- x - b_i^+ u_j^- \leq b_i^+ y b_j^- = b_j^- y b_i^+ \leq b_j^- u_i^+ - b_j^- A_i^+ x.$$

Projecting several variables

Similarly, when considering a multidimensional y, $proj_x(\mathcal{P})$ will be obtained by taking any linear positive combinations of rows such that the y variables disappear, *i.e.*:

$$proj_x(\mathcal{P}) = \{x \in \mathbb{R}^p | cAx \leq cb, \forall c \in \mathbb{Q}_+ \text{ such that } cB = 0\}. \tag{3}$$

This is more complicated to consider since c can have many non-zero coefficients (when y is one dimensional, it is sufficient to consider c with two non-zero coefficients). This means that the combinations we consider are no longer combinations of only two rows. The set of vectors c form the following *projection cone* where m is the number of rows of B:

$$\mathcal{C}_B = \{c \in \mathbb{Q}_+^m | cB = 0\}$$

To every coefficient in $\mathcal{C}_B$ corresponds a valid inequality for $proj_x(\mathcal{P})$, but the non-dominated inequalities are associated with extreme rays of $\mathcal{C}$. Each element in $\mathcal{C}$ is a conic combination of its generators. This implies that each inequality associated with an interior point of $\mathcal{C}_B$ is dominated by the inequalities associated with its generators. Note that the number of facets defining inequalities of $proj_x(\mathcal{P})$ can be exponential in the dimension of y in the worst case. Sometimes, it still results in a polynomial number of constraints. Consider the trivial case of projecting a hypercube of dimension n ($2n$ variables) on a subset of m variables, it results in a hypercube of dimension m ($2m$ constraints when the worst case is 2^{n-m}). We will show an example of a projection cone in Section 3.4.4.

Balas's Theorem

Balas's Theorem is used to obtain an extended formulation of the convex hull of a finite union of polyhedrons. We give the theorem in the case of the union of two polytopes.

Theorem 0.0.3 ([13])**.** *Let $P_1 = \{x^1 \in \mathbb{R}^d | A^1x^1 \leq b^1\}$ and $P_2 = \{x^2 \in \mathbb{R}^d | A^2x^2 \leq b^2\}$ be two non-empty polytopes:*

$$conv(P_1 \cup P_2) = proj_x\{x \in \mathbb{R}^d | x = x^1 + x^2,\ A^1x^1 \leq \lambda b^1,\ A^2x^2 \leq (1-\lambda)b^2,\ 0 \leq \lambda \leq 1\}$$

In practice, if P_1 and P_2 are not defined with the same number of variables, 0 components can be added to the solutions of the polytope defined with the lowest variables set's size.

This theorem can be generalized to the union of k polytopes $P_1, \dots, P_k$ by introducing k variables, $\lambda_1, \dots, \lambda_k$ such that $\sum_{i=1}^k \lambda_i = 1$ and k points $x^1, \dots, x^k$ such that $x^i \in \lambda_i P_i$ for all $i \in \{1, \dots, k\}$ and then $x = \sum_{i=1}^k x_i$ belongs to $conv(\bigcup_{i=1}^k P_i)$.

The graph coloring problem

This problem is NP-hard [9]. The coloring problem has historically been among the most important in graph theory because of two beautiful theorems that remained conjectures for a long time: the 4 color theorem and the strong perfect graph theorem. I will present the latter in Chapter 1.

The four-color theorem

The 4 color theorem states that every *planar graph*, a graph one can draw on a plane without crossing edges, can be colored with at most 4 colors. This result was conjectured by Francis Guthrie in 1852. Several famous proofs turned out to be false. In 1976, Kenneth Appel and Wolfgang Haken gave a computer-assisted proof [14]. It was one of the first times that this type of proof was used and it raised the question of whether a short proof exists for each provable statement. Since then, people have been looking for a non-computer-assisted proof with no success so far.

A few words on parameterized complexity

Tackling the graph coloring problem is a particularly hard challenge from a computational point of view. It has been studied under the spectrum of parameterized complexity. Parameterized complexity is a field of research that aims to find the combinatorial barrier to solve some instances by bounding a parameter of the instance of a problem. We say that a parameterized problem is FPT if there exists an algorithm solving this parameterized problem in time $f(k)|x|^{O(1)}$ where f is a computable function, $|x|$ is the input size of the instance and k the size of the parameter. A parameter commonly used is the size of the solution, for example, if we consider the set of graphs for which the optimal vertex cover has size at most k, there exists a simple algorithm finding such a cover in time $2^k|x|^{O(1)}$ (as every edge is covered by one of its extremities, it is sufficient to make the arbitrary disjunction on k vertices). This proves that the problem of vertex cover parameterized by the size of the cover is Fixed-Parameter Tractable. XP is the set of parameterized problems solvable in time $|x|^{f(k)}$ for some computable f, it is known to strictly contain FPT. Note that knowing if a graph is 3-colorable is already NP-hard. In particular, if $P \neq NP$, there exists no algorithm deciding whether a graph is k colorable in time $|x|^{f(k)}$. This problem is hence para-NP-hard which stands for parameterized NP-hard, meaning that it is NP-hard even when the parameter (number of colors in this case) is fixed. This shows that the coloring problem is particularly hard.

Part I

Contraction perfect graphs and co-2-plexes

Chapter 1

Literature review on co-k-plexes formulations and polytopes

Given a network, it is classic to seek "communities." When it is a social network, these communities refer to sets of individuals that share characteristics *i.e.* there exists a relation between the individuals of a same community. This problem also finds applications in other fields, such as biological network analysis, see [15] where two prominent examples are given: protein interaction network where each node is a protein, and two proteins are adjacent if they are known to interact; and gene co-expression network where each node is a gene and two genes are adjacent if their expression is sufficiently correlated (the correlation of co-expression is higher than a given threshold). In graph theory, cliques are the first objects one would think about to represent communities, but the fact that each pair of vertices must be adjacent is often too restrictive. For example, given a sports club, the set of its members can be called a community because **most** of its members know each other. This remark leads to many "relaxed" clique definitions to accept a few "missing edges" between the vertices associated with individuals of a community, the k-plexes are one of these relaxed cliques. In this chapter, we recall the literature on perfect graphs and the integer linear programming (ILP) models used to compute a maximum weighted co-k-plex. As stable sets are co-1-plexes, we start with results for the maximum weighted stable set problem and discuss how they extend to the maximum weighted co-k-plex problem. On the other hand, we recall the polyhedral results associated with the subjacent polytope of these ILP formulations.

Finding a maximum weight k-plex is a topic that has been especially lively for the past few years. For this problem, there are algorithms based on local search [16, 17, 18, 19, 20], heuristics [21], branch and bound algorithms, some of which incorporate machine learning ingredients [7, 22, 23, 24, 25, 26, 27], and quadratic models [28]. The problem of enumerating the k-plexes also received some attention [29, 30, 31]. We focus in this literature review on results concerning integer programming formulations, and polyhedral theory. We present the results on perfect graphs with a focus on the clique polytope but the results on the stable sets polytope are obtained by complementing the graph. This choice has been made because of its links with the coloring problem, these will be detailed in Section 1.2.

Let us recall the complexity of finding a maximum weighted co-k-plex problem. Balasundaram et al. [7] showed that finding a maximum k-plex is NP-hard by reducing the existence of a clique of size t to the existence of a k-plex of size kt in another polynomial-sized graph. We translate their results to the case of co-k-plex by complementing the graph.

Theorem 1.0.1 ([7]). *For any $k \in \mathbb{Z}_+ \setminus \{0\}$, the maximum weighted co-k-plex problem is NP-hard.*

1.1 Integer programming for the maximal w-weighted co-k-plex problem

In this section, we present several existing ILP formulations and their reinforcement inequalities for the maximum weighted co-k-plex problem. Variables x are always associated with vertices of a graph. First, we recall a basic ILP formulation solving the maximum weighted stable set problem.

Proposition 1.1.1. *The following formulation is valid for solving the maximum w-weighted stable set problem of a graph $G(V, E)$:*

$$\max w^\top x \tag{1.1}$$
$$x_u + x_v \leq 1 \qquad \forall uv \in E, \tag{1.2}$$
$$x_u \in \{0,1\} \qquad \forall u \in V. \tag{1.3}$$

Inequalities (1.2) ensure that at most one extremity of each edge is chosen. However, they can also be interpreted as: for each vertex v and each subset of size 1 of its neighborhood at most one vertex is taken. Analogously, one can formulate the problem of finding a maximum co-k-plex using constraints with vertices v and subsets of its neighborhood of size k.

Proposition 1.1.2. *The following formulation is valid for solving the maximum w-weighted k-plex problem of a graph $G(V, E)$:*

$$\max w^\top x \tag{1.4}$$
$$x_u + x(U) \leq k \qquad \forall u \in V,\ U \subseteq N_u \text{ with } |U| = k, \tag{1.5}$$
$$x_u \in \{0,1\} \qquad \forall u \in V. \tag{1.6}$$

This formulation is not satisfying as its number of inequalities is $O(|V|^k)$ and it is unclear whether these constraints are strong (we will see in Chapter 3 later that a very few of them define facets). Balasundaram et al. [7] proposed a formulation with $|V|$ linear inequalities for finding a k-plex. By complementing the graph, we obtain the following.

Theorem 1.1.3 ([7]). *The following formulation $\mathcal{N}_k$ is valid for finding a maximum w-weighted co-k-plex of a graph $G = (V, E)$:*

$$\max w^\top x \tag{1.7}$$
$$x(N_u) + (|N_u| - k + 1)x_u \leq |N_u| \qquad \forall u \in V \tag{1.8}$$
$$x_u \in \{0, 1\} \qquad \forall u \in V. \tag{1.9}$$

Constraints (1.8) correspond to the fact that in any co-k-plex S, if v belongs to S, at most $k-1$ of its neighbors belong to S. Adding such a linear set of constraints is sufficient to formulate the problem.

The stable set formulation (1.2) (1.3) can be reinforced using the following clique inequalities.

$$x(K) \leq 1 \quad \forall \text{ clique } K \text{ of } G. \tag{1.10}$$

Analogously, cliques yield valid inequalities for $\mathcal{N}_k$.

Theorem 1.1.4 ([7]). *The following inequalities are valid for the co-k-plex polytope:*

$$x(K) \leq k \qquad \forall \textit{ clique } K \textit{ of } G. \tag{1.11}$$

As cliques are complements of stable sets and yield strong inequalities for formulation (1.2) (1.3), k-plexes that are complements of co-k-plexes, yield strong inequalities for $\mathcal{N}_k$ as stated in the following theorem.

Theorem 1.1.5 ([7]). *The following inequalities are valid for the co-k-plex polytope:*

$$x(K) \leq 2k - 2 + (k \mod 2) \qquad \forall \textit{ k-plex of } K \textit{ of } G \textit{ with } |K| \geq 3. \tag{1.12}$$

Note that for the case $k = 2$, Constraints (1.12) are sufficient to yield a valid formulation.

Theorem 1.1.6 ([32]). *The following formulation is valid for the maximum weighted co-2-plex problem of a graph $G = (V, E)$:*

$$\max w^\top x \tag{1.13}$$
$$x(K) \leq 2 \qquad \forall \textit{ 2-plex } K \textit{ of } G \tag{1.14}$$
$$x_u \in \{0, 1\} \qquad \forall u \in V \tag{1.15}$$

One could naively think that Theorem 1.1.6 can be generalized to give a formulation for the co-k-plex with constraints of the type $x(K) \leq k$ where K is a k-plex(, but these constraints are not valid in general: the hole of size 4 has 4 vertices and is both a 3-plex and a co-3-plex. Note that Theorem 1.1.6 coincides with Theorem 1.1.3 for paths as each vertex and its neighborhood form a 2-plex of size at most 3.

Finally, another family of valid inequality associated with holes has been given.

Theorem 1.1.7 ([7])**.** *Let H be an antihole of size at least $k+3$ of G, the following inequality is valid for the co-k-plex polytope:*

$$x(H) \leq k+1 \tag{1.16}$$

On the other hand, Stetsyuk [28] et, al. gave an integer quadratic programming formulation for the maximum weighted co-k-plex problem.

Theorem 1.1.8 ([28])**.** *The following quadratic formulation $\mathcal{Q}_k$ is valid for the maximum w-weighted co-k-plex problem:*

$$\max w^\top x \tag{1.17}$$

$$x_v \times x(N_v) \leq k-1 \qquad \forall v \in V \tag{1.18}$$

$$x_v \in \{0,1\} \qquad \forall v \in V \tag{1.19}$$

1.2 Polyhedral results

In this section, we recall the literature on the co-1-plex, co-2-plex, and co-k-plex polytopes. The study of the co-1-plex polytope goes through the links between the packing polytope and perfect graphs and ends with the strong perfect graph theorem.

For any graph $G = (V, E)$, let $\mathcal{P}_G^k$ be the co-k-plex polytope, that is the convex hull of the incidence vectors of the co-k-plexes of G:

$$\mathcal{P}_G^k = conv(\{\chi^S \mid S \text{ is a co-}k\text{-plex of } G\}).$$

Clique polytope and strong perfect graph theorem

First, note that if a graph contains a clique of size k, then its smallest coloring needs at least k colors. The "Strong perfect Graph conjecture" enunciated by Claude Berge states that the graphs for which the chromatic number is equal to the size of its largest clique and for which this property is true for every induced subgraph are precisely the graphs containing no odd hole nor odd antihole as an induced subgraph. This conjecture is essential, in particular for its polyhedral aspects. Let M be a 0/1 matrix, M is *perfect* if the polyhedron

$\mathcal{P}_M = \{x \in \mathbb{R}^n \mid Mx \leq 1, x \geq 0\}$ has only integer vertices. A *packing polytope* is a polytope that can be described as $conv\{x \in \{0,1\}^n \mid Mx \leq 1, x \geq 0\}$ for some 0/1 matrix M. In 1972, Lovász [33] proved that the matrix M is perfect if and only if $\{\max w^\top x \mid x \in \mathcal{P}_M\}$ has an integral optimum solution x for all $w \in \{0,1\}^{|V|}$ and from this result we start to understand the beautiful hidden link between graph theory and polyhedral theory under this conjecture.

Theorem 1.2.1 ([33])**.** *For a 0/1 matrix M of size $n \times m$ with no column of 0's, the following statements are equivalent:*

(i) M is perfect

(ii) $\max\{w^\top x \mid x \in \mathcal{P}_M\}$ *has an integral optimal solution x for all $w \in \{0,1\}^n$*

(iii) the linear system $Mx \leq 1,\ x \geq 0$ is TDI.

Given a graph G, its stable set incidence matrix is a 0/1 matrix whose columns are associated with vertices of G and the rows with inclusion-wise maximal stable sets of G. The coefficient associated $v \in V$ and $S \subseteq V$ is 1 if $v \in S$ and 0 otherwise. Let us highlight the link between Theorem 1.2.1 and perfect graphs by showing that when M is the stable set incidence matrix of a perfect graph, $\max\{w^\top x \mid x \in \mathcal{P}_M\}$ has an integral optimal solution x for all $w \in \{0,1\}^n$.

Finding a maximum w-weighted clique of a graph G with weights 0/1 corresponds to finding the maximum cardinality clique problem in the subgraph induced by the vertices of cost 1, say G'. It is equivalent to solving:

$$\max\{w^\top x \mid x \in \mathcal{P}_M \cap \mathbb{Z}^n_+\}$$

where M is the stable set incidence matrix of G. Every integer point of $\mathcal{P}_M$ corresponds to a clique of G whose vertices of weight 1 form a clique of G'. Also, by linear programming duality, the dual of $\{\max w^\top x | x \in \mathcal{P}_M\}$ is:

$$(\mathcal{D}) \left\{ \min \sum_{S \in \mathcal{S}} y_S \;\middle|\; \sum_{S \in \mathcal{S}_v} y_S \geq w \ \forall v \in V,\ y_S \geq 0 \ \forall S \in \mathcal{S} \right\}$$

where $\mathcal{S}$ is the set of maximal stable sets of G and $\mathcal{S}_v$ is the set of stable sets of G containing v. $\mathcal{D}$ is a linear program whose variables are associated with inclusion-wise maximal stable set and each constraint to a vertex of G, requiring that the sum of the variables associated with stable sets containing v is greater or equal to 1 when $w_v = 1$, *i.e.* when v is a vertex of G'. An integer solution of this dual corresponds to a covering of the vertices of G' by stable sets, it hence corresponds to a coloring of G'. Hence, if G is perfect, the maximal clique and the minimal coloring of G' coincide, which implies that the primal's optimal integer solution coincides with the dual's optimal integer solution which implies that $\mathcal{P}_M$ has an optimal integer vertex for w.

By Theorem 1.2.1, it follows that if G is a perfect graph, then its stable set incidence matrix M is perfect. This gives one side of Chvátal's Theorem: a graph is perfect if and only if its clique polytope is described by clique and trivial inequalities 1975 [34]. The other side of the proof uses the following proposition by Edmonds, the replication lemma: "Adding twins preserves perfection" and the characterization of imperfect graphs from Lovasz.

Proposition 1.2.2 ([35])**.** *Let $\mathcal{I}$ be the set of integer solutions of a system $\mathcal{T} = \{Ax \leq b,\ x \geq 0\}$, $\mathcal{T}$ yields an integer polytope if and only if for all integer vector c*

$$\max\{c^\top I \mid I \in \mathcal{I}\} = \min\{b^\top y \mid y^\top A \geq c,\ y \geq 0\}$$

.

Theorem 1.2.3 ([36])**.** *An imperfect graph contains an induced subgraph $G = (V, E)$ such that $\omega(G) \times \alpha(G) < |V|$.*

Note that Theorem 1.2.3 also proves that G is perfect if and only if its complement is. Finally, another alternative proof of Chvátal' is given by Cornuéjols [37]. This proof is the strongest I know in the sense that it additionally proves that G is perfect if and only if its complement also is perfect. Moreover, this proof does not make use of Theorem 1.2.1 nor Proposition 1.2.2 but extensively uses the replication lemma. It shows that every fractional point of $\mathcal{P}_M$ is a convex combination of integer points.

Theorem 1.2.4 ([34])**.** *Let $G = (V, E)$ be a graph and M be its incidence stable set matrix. Then G is perfect if and only if $\mathcal{P}_M$ is integer.*

This beautiful theorem characterizes the packing polyhedra defined by trivial inequalities and facets with right-hand side 1. In fact, for any 0/1 matrix M, if it is not the incidence matrix of the stable sets of some graph, $\mathcal{P}_M$ is not integer by using the following characterization. We denote by J the matrix of all ones and I the identity matrix.

Theorem 1.2.5 ([37])**.** *Let M be a 0/1 matrix with no column of 0. The following statements are equivalent:*

(i) M is the incidence matrix of the stable sets of a graph.

(ii) If $\begin{pmatrix} 0 & 1 & 1 & 1 & \dots & 1 \\ 1 & 0 & 1 & 1 & \dots & 1 \\ 1 & 1 & 0 & 1 & \dots & 1 \end{pmatrix}$ is a submatrix of M with columns $S_1, \dots, S_p$, then M contains a row v such that $M_v^{S_i} = 1\ \forall i \in \{1, \dots, p\}$.

(iii) If $J - I$ is a $p \times p$ submatrix of M where $p \geq 3$, then M contains a row v satisfying $M_v^j = 1$ for all columns j of $J - I$.

Theorem 1.2.5 has been told to G. Cornuéjols by M. Conforti in private communications.

Theorem 1.2.1 also states that $\mathcal{P}_G$ is integer if and only if the linear system $Mx \leq 1,\ x \geq 0$ is TDI, which gives a surprising result: integrality of the polytope and TDI-ness of its natural H-representation coincide.

Later on, in 1984, Martin Grötschel, László Lovász, and Alexander Schrijver [1] proved using the ellipsoid method that the maximum clique and minimal coloring problem are solvable in polynomial time on perfect graphs.

And finally, the Strong Perfect Graph Theorem was proven by Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas in 2006 [38], which proves Berge's conjecture.

Theorem 1.2.6 ([38])**.** *A graph is perfect if and only if it does not contain any odd hole or odd antihole as an induced subgraph.*

I suggest "Packing and Covering" by Gérard Cornuéjols [37] for a survey on this topic. It has been such an inspiring book.

Co-k-plex polytope

In this section, we present the known facial results on the co-k-plex polytope.

Theorem 1.2.7 ([7])**.** *For $k \in \mathbb{Z}^+ \setminus \{0\}$, and the co-$k$-plex polytope $\mathcal{P}_G^k$ of G, the following hold:*

- *$\mathcal{P}_G^k$ is full dimensional*
- *$x_u \geq 0$ defines a facet of $\mathcal{P}_G^k$*
- *$x_u \leq 1$ defines a facet of $\mathcal{P}_G^k$ if $k \geq 2$*

Theorem 1.2.8 ([7])**.** *For a subset $K \subseteq V$ with $|K| \geq 3$, the inequality $x(K) \leq 2k - 2 + (k \mod 2)$ defines a facet of $\mathcal{P}_G^k$ if and only if K is a maximal k-plex of G.*

Co-2-plex polytope

Benjamin McClosky and Illya V. Hicks investigated when the following polytope is integer:

$$\mathcal{Q}_G = \left\{ x \in \mathbb{R}^V \mid x(K) \leq 2 \ \forall K \text{ 2-plex of } G,\ 0 \leq x \leq 1 \right\}$$

Theorem 1.2.9 ([32])**.** *$\mathcal{Q}_G$ is integer if and only if G is a:*

- *path*
- *co-2-plex*
- *2-plex*
- *chordless cycle with size equal to $0 \mod 3$.*

Note that the proof of Theorem 1.2.9 proves that the constraints matrix is totally unimodular for paths, co-2-plexes, and 2-plexes. This remark yields a variant of perfect graphs: paths, co-2-plexes, and 2-plexes are the graphs for which the smallest covering by 2-plexes coincide with the largest co-2-plex and for which this property is true in each induced subgraph.

An edge $e \in E$ of $G = (V, E)$ is k-critical if the largest co-k-plex of $G \setminus e$ is strictly bigger than the largest co-k-plex of G. Let E_k^* be the set of k-critical edges, and ξ_G^k be the size of the largest co-k-plex of G. This definition looks like the definition of critical graphs recalled in Chapter 5 for the proofs of rank facets for the covering polytope, and seems to be standard in polyhedral studies to prove the facial structure of rank inequalities.

Theorem 1.2.10 ([32])**.** *If (V, E_k^*) is connected, then the inequality $x(V) \leq \xi_G^k$ defines a facet of $\mathcal{P}_G^k$.*

Benjamin McClosky and Illya V. Hicks [32] use Theorem 1.2.10 to give facets when G is a wheel or a web or a hole whose size differs from $0 \mod 3$. This article shows that describing the co-2-plex polytope in the natural space variable is pretty hard. By analogy with the stable set polytope, when formulating it with inequalities associated with complements of stable sets that are cliques, we describe the stable set polytope of perfect graphs. When we consider a similar approach for the co-2-plex polytope, that is, formulating it with 2-plex inequalities, we obtain a much smaller graph class. For this reason, it seems relevant to investigate extended formulations. This is done in Chapter 3.

Chapter 2

Contractions in perfect graphs

Contractions in perfect graphs already have been studied under nonedge contractions. The *contraction of nonedges* corresponds to merging the extremities of nonedges of a graph. This topic yielded a subclass of perfect graphs called perfectly contractile. Two nonadjacent vertices in a graph form an *even pair* if every induced path between them has an even number of edges. Fonlupt and Uhry [39] proved that contracting an even pair in a perfect graph preserves perfection, and Meyniel proved what is called the Even Pair Lemma [40]: no minimally imperfect graph contains an even pair. Bienstock [41] proved that the following decision problems are CoNP-complete in the general case: deciding whether a given pair of vertices of a graph forms an even pair, and deciding whether a given graph contains an even pair. Nevertheless, in perfect graphs, both problems become solvable in polynomial time [42].

A graph is *perfectly contractile* if, for every induced subgraph, a sequence of even pair contractions yields a clique. In [43], a subclass of perfectly contractile graphs is given, leading to a polynomial combinatorial algorithm for the coloring problem for that class.

In this chapter, instead of contracting nonedges, we investigate contracting edges in perfect graphs. Contracting any pair of vertices, that is, identifying both vertices, does not always preserve perfection. In Section 2.1, we first provide a forbidden induced subgraph characterization for the graphs remaining perfect after the contraction of any single edge. We show that these graphs are exactly contraction perfect graphs in Section 2.1.2. This yields the forbidden induced subgraph characterization of contraction perfect graphs in Section 2.1.3. Consequently, deciding if a graph is contraction perfect can be done in polynomial time.

In Section 2.2, we introduce the utter graph of a graph. This yields another characterization of contraction perfect graphs in Section 2.2.1 as those whose utter graphs are perfect. As a byproduct, a maximum weight co-2-plex can be found in polynomial time in contraction perfect graphs. In Section 2.2.2, we strengthen the link with the utter graph for a few subclasses of contraction perfect graphs.

This chapter is overall independent of the state of the art given previously in Chapter 1 but is the first mandatory step for us to design new algorithms to solve the maximum weighted co-k-plex problem. Still, this chapter makes use of the strong perfect graph theorem characterizing perfect graphs in terms of forbidden induced subgraphs.

2.1 Contraction perfect graphs

This section investigates how contracting edges in a perfect graph impacts its perfection. It relies on the strong perfect graph theorem [38], which states that perfect graphs are the graphs that contain neither odd holes nor odd antiholes. We start by characterizing when the contraction of an edge destroys the perfection of a graph. We define contraction perfect graphs as the perfect graphs remaining perfect under the contraction of any edge set. In the second and third subsections, we provide two characterizations of contraction perfect graphs: the first one compares contractions of single edges with contractions of sets of edges, and the second one is in terms of forbidden induced subgraphs.

We start by proving a parity result about interval graphs that will be used in Section 2.1.1 but might be of independent interest. A set of intervals is an *odd intersection interval set* if the non-empty intersection of any subset of intervals is an interval of odd cardinality.

Lemma 2.1.1. *Given an odd intersection interval set $\mathcal{I}$, the union of the intervals associated with every connected component of $G_{\mathcal{I}}$ has odd cardinality.*

Proof. It is enough to prove the result when $G_{\mathcal{I}}$ is connected. We proceed by induction on $|\mathcal{I}|$, the case $|\mathcal{I}| = 1$ being immediate. Let I and I' be two intersecting intervals of $\mathcal{I}$. Hence, $I \cup I'$ is an interval. Let $\mathcal{I}'$ be obtained from $\mathcal{I}$ by replacing I and I' by $I \cup I'$. Since $\mathcal{I}$ is an odd intersection interval set, $|(I \cup I') \cap K| = |I \cap K| + |I' \cap K| - |I \cap I' \cap K|$ is odd or zero for each intersection K of any subset of intervals of $\mathcal{I}$. Hence, $\mathcal{I}'$ is also an odd intersection interval set. Moreover, $G_{\mathcal{I}'}$ is connected as it is obtained from $G_{\mathcal{I}}$ by contracting the edge whose extremities are the vertices associated with I and I'. Since $|\mathcal{I}'| < |\mathcal{I}|$, the induction hypothesis implies that the union of intervals of $\mathcal{I}'$ has odd cardinality. By definition, the latter set is equal to the union of the intervals of $\mathcal{I}$, which ends the proof. ☐

2.1.1 When does the contraction of an edge destroy perfection?

By the strong perfect graph theorem, if the contraction of an edge e in a perfect graph G destroys its perfection, this implies that G/e contains an odd hole or an odd antihole. Moreover, since G is perfect, the image of e must be a vertex of such a forbidden induced subgraph. Lemmas 2.1.2 and 2.1.4 characterize when contracting an edge yields an odd hole or an odd antihole, respectively. Both results will be used in the proof of Theorem 2.1.5 which characterizes when the contraction of an edge destroys the perfection of a graph.

Lemma 2.1.2. *If G/F contains a hole H for some edge set F, then G contains a hole of size at least $|V(H)|$.*

Proof. We proceed by induction on $|F|$. The result holds if $|F| = 0$. Otherwise, some vertex v of H is obtained by contracting the edges of a connected subgraph (V', F') of G with $\emptyset \neq F' \subseteq F$. Let a and b be the neighbors of V' in H, and let P be an ab-path in $G[V' \cup \{a, b\}]$. Then, $V(H) \setminus \{v\} \cup V(P)$ induces a hole of $G/(F \setminus F')$ of size at least $|V(H)|$, and induction concludes. □

As we shall see in Lemma 2.1.4, when the contraction of an edge destroys the perfection of G by creating an odd antihole, this implies that G contains the following specific structure.

Definition 2.1.3. *An edge e and an even antipath P induced by $(w_1, \ldots, w_p)$ with $p \geq 6$ form an* expanded antihole *if one extremity of e is adjacent to all vertices of P except w_1, w_{p-1}, w_p, and the other extremity is adjacent to all vertices of P except w_1, w_2, w_p.*

Figure 2.1 represents an expanded antihole where $(w_1, \ldots, w_p)$ induces an even antipath. An edge e is *involved* in an expanded antihole if there exists an even antipath forming with e an expanded antihole. Note that, by definition of an expanded antihole, the extremities of e are not vertices of P, and contracting uv yields an odd antihole. Moreover, if $e = uv$, either $(u, w_2, \ldots, w_{p-1}, v)$ or $(v, w_2, \ldots, w_{p-1}, u)$ induces an even antipath that forms an expanded antihole with edge $w_1 w_p$.

Note that if one relaxes the adjacency requirement in the definition an expanded antihole to: "each vertex $w_2, \ldots, w_{p-1}$ is adjacent to one extremity of e", contracting e still yields an odd antihole of G/e. However, in this case, G is not necessarily perfect. The following lemma asserts that the only case where G is perfect and G/e contains an odd antihole of size at least 7 is when G contains an expanded antihole.

Lemma 2.1.4. *Let G be a perfect graph, H a subgraph of G, and uv an edge of $E(H)$. H/uv is an odd antihole of G/uv of size at least 7 if and only if H is an expanded antihole formed by uv and the antipath induced by $V(H) \setminus \{u, v\}$.*

Proof. ($\Leftarrow$) By definition, if H is an expanded antihole involving uv, then H/uv is an odd antihole of G/uv.
($\Rightarrow$) Let us denote by w_{uv} the vertex obtained by contracting uv. Since H/uv is an antihole, $V(H) \setminus \{u, v\}$ can be ordered into the sequence $(w_1, \ldots, w_p)$ with $p \geq 6$ inducing an even antipath in G. The set of vertices of a maximal antipath

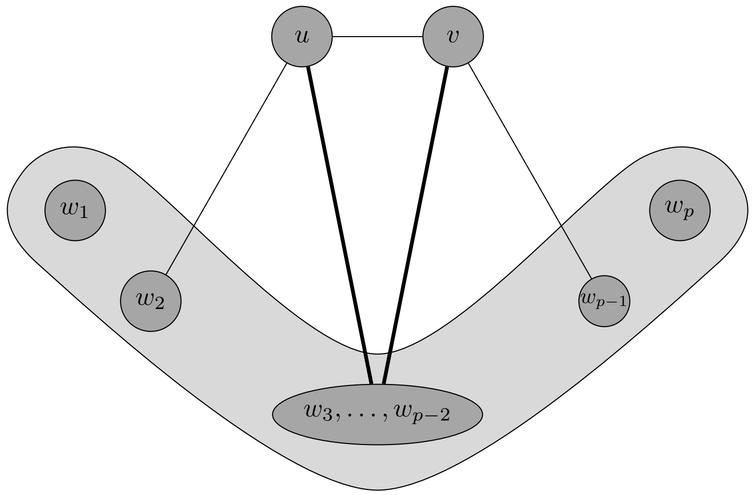


Figure 2.1: An expanded antihole.

of $H[N_u \setminus v]$ (resp. $H[N_v \setminus u]$) forms an interval of $(w_2, \ldots, w_{p-1})$. Let $\mathcal{P}^u$ (resp. $\mathcal{P}^v$) be the set of these intervals, and $\mathcal{I} = \mathcal{P}^u \cup \mathcal{P}^v$. Since G is perfect, H contains no antihole and then, the interval $(w_2, \ldots, w_{p-1})$ does not belong to $\mathcal{I}$.

Suppose that interval graph $G_{\mathcal{I}}$ associated to $\mathcal{I}$ is not connected. Then, there exists $i \in \{2, \ldots, p-2\}$ such that no interval of $\mathcal{I}$ contains both w_i and w_{i+1}. By definition of $\mathcal{I}$, $\{u, v, w_i, w_{i+1}\}$ induces a path of H with extremities w_i and w_{i+1}. Since $p \geq 6$, there exists $z \in \{w_1, w_p\}$ adjacent to both w_i and w_{i+1}. As z is adjacent to neither u nor v, $\{z, w_i, u, v, w_{i+1}\}$ induces an odd hole of G, a contradiction to its perfection. Hence, $G_{\mathcal{I}}$ is connected.

Note that every interval $I = \{w_i, \ldots, w_j\}$ of $\mathcal{I}$ has odd cardinality, as otherwise $\{w_{i-1}, w_i, \ldots, w_j,\ w_{j+1}\}$ would induce, together with u if $I \in \mathcal{P}^u$ or v if $I \in \mathcal{P}^v$, an odd antihole of G, contradicting its perfection. By Lemma 2.1.1, since p is even and $G_{\mathcal{I}}$ is connected, $\mathcal{I}$ is not an odd intersection interval set. Moreover, the intersection of more than two intervals of $\mathcal{I}$ is empty by definition of $\mathcal{P}^u$ and $\mathcal{P}^v$. Hence, there exists $I \in \mathcal{P}^u$ and $I' \in \mathcal{P}^v$ whose intersection is nonempty and has even cardinality. Let $I = \{w_i, \ldots, w_k\}$ and $I' = \{w_j, \ldots, w_\ell\}$ with $2 \leq i \leq j \leq k \leq \ell \leq p-1$, Figure 2.2 gives an illustration. Since $|I|$ and $|I'|$ are odd and $|I \cap I'|$ is even, $2 \leq i < j < k < \ell \leq p-1$. Suppose that $j > 3$. Then, w_1 is adjacent to $w_{j-1}, \ldots, w_{k+1}$ so $\{w_1, v, w_{j-1}, \ldots, w_{k+1}, u\}$ induces an odd antihole of G, a contradiction. Hence, $j = 3$. Similarly, one can prove that $k = p-2$. Hence, u is adjacent to $w_2, \ldots, w_{p-2}$ and v is adjacent to $w_3, \ldots, w_{p-1}$. Therefore, $(w_1, \ldots, w_p)$ and the edge uv form an expanded antihole. $\square$

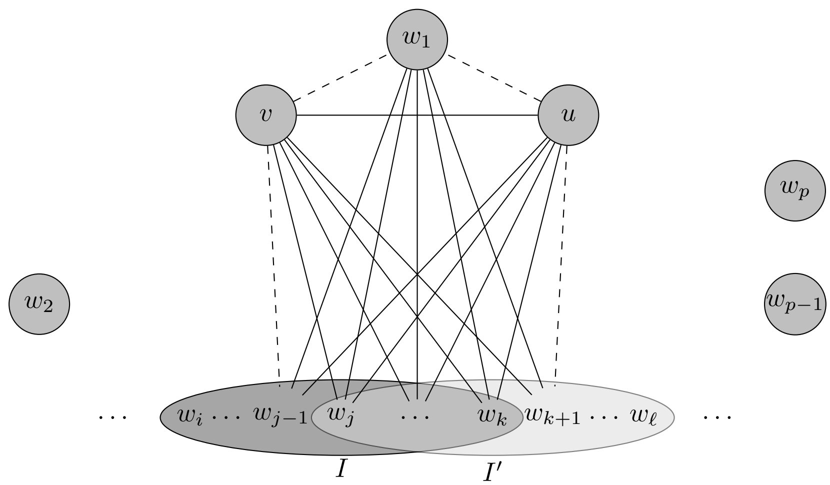


Figure 2.2: Illustration of the proof of Lemma 2.1.4 (the doted lines represent nonedges and dots the remaining vertices of the antipath).

Theorem 2.1.5. *Let $G = (V, E)$ be a perfect graph and $uv \in E$. The graph G/uv is not perfect if and only if G contains an even hole of size at least 6 containing uv or an expanded antihole involving uv.*

Proof. ($\Leftarrow$) In both cases, contracting uv destroys perfection.
($\Rightarrow$) Suppose that G/uv is not perfect. If G/uv contains an odd hole H, then G contains a hole H' of size at least $|V(H)|$ by Lemma 2.1.2. Since G is perfect, $|V(H')|$ is even and at least 6, and contains uv. Otherwise, G/uv contains an odd antihole hence, by Lemma 2.1.4, G contains an expanded antihole involving uv. $\square$

2.1.2 Contracting an edge or a set of edges?

We prove that contraction perfect graphs are characterized by the contraction of single edges.

Theorem 2.1.6. *A perfect graph is contraction perfect if and only if it remains perfect by the contraction of any single edge.*

Proof. To prove the nontrivial direction, let $G = (V, E)$ be a perfect graph and $F \subseteq E$ be such that G/F is not perfect. We will prove that G/e is not perfect for some $e \in E$. Without loss of generality, suppose that F has minimum size, that is, G/F' is perfect for all $F' \subsetneq E$ such that $|F'| < |F|$. Since G/F is not perfect, it contains an odd hole or an odd antihole. Suppose that G/F contains an odd hole C. By Lemma 2.1.2, G contains a hole D with $|V(D)| \geq |V(C)|$. Since G is perfect, $|V(D)|$ is even and at least 6. Then, G/e contains the odd hole D/e for any $e \in E(D)$ and thus is not perfect.

From now on, we may assume that G and G/F' for all $F' \subseteq E$ contain no hole of size at least 5. Hence, G/F contains odd antiholes and let us denote by $(w_0, \ldots, w_p)$ one of them where $p \geq 6$.

We will first prove that F is a matching. To do this, let us suppose that F contains two adjacent edges uv and vw whose image in G/F is w_0. Let $G' = G/(F \setminus \{uv, vw\})$. Since $G'/\{uv, vw\}$ contains an odd antihole, and since G'/uv is perfect by minimality of F and has no hole of size at least 5, then, by Theorem 2.1.5, G'/uv contains an expanded antihole formed by the even antipath $P_0 = (w_1, \ldots w_p)$, and the edge $x_{uv}w$ where x_{uv} is the image of uv in G'.

Without loss of generality suppose that x_{uv} is adjacent to $\{w_2, \ldots, w_{p-2}\}$ and w is adjacent to $\{w_3, \ldots, w_{p-1}\}$. Note that v is not adjacent to w_2 in G, as otherwise G'/vw contains the odd antihole induced by $\{x_{vw}, w_1, \ldots, w_p\}$ —where x_{vw} is the image vw in G'/vw— which contradicts the minimality assumption on F. Now, G' contains uw as otherwise $\{u, v, w, w_{p-1}, w_2\}$ induces an odd hole of G', contradicting its perfection. But now, G'/uw contains the odd antihole induced by $\{x_{uw}, w_1, \ldots, w_p\}$ where x_{uw} is the vertex obtained by contracting edge uw. This contradicts the minimality assumption on F, hence F is a matching.

We now prove that F only contains one edge. Let uv and $u'v'$ be two edges of F. By the minimality assumption, their images are vertices of $\{w_0, \ldots, w_p\}$, and up to node relabelling, let us suppose that the image of uv is w_0 and the one of $u'v'$ is w_i with $i \in \{1, \ldots, p/2\}$. Set $G' = G/(F \setminus \{uv, u'v'\})$. By Lemma 2.1.4, uv forms with $P_0 = \{w_1, \ldots, w_p\}$ an expanded antihole of $G'/u'v'$. Similarly, $u'v'$ forms with $P_i = \{w_{i+1}, \ldots, w_p, w_0, \ldots, w_{i-1}\}$ an expanded antihole of G'/uv. By definition of expanded antiholes, we suppose without loss of generality that u (resp. u') is adjacent to w_2 (resp. w_{i-2}) but not to w_{p-1} (resp. w_{i+2}). Hence, $H_u = \{u, w_1, \ldots, w_{p-1}\}$ induces an even antihole of $G'/u'v'$, similarly $\{w_0, \ldots, w_{i-1}, u', w_{i+2}, \ldots, w_p,\}$ induces an even antihole of G'/uv. Figure 2.3 gives an illustration of $H_u \cup u' \setminus \{w_i, w_{i+1}\}$, which induces an antihole in G' of size $p-1$, and hence is odd. This contradicts the minimality assumption on F. Therefore F cannot contain two edges, so G does not remain perfect by the contraction of the edge of F. $\square$

2.1.3 Characterization by forbidden induced subgraphs

By definition, a perfect graph remains perfect by any vertex set deletion. Thus, a forbidden induced subgraph characterization may be hoped for, and this tremendous result is the strong perfect graph theorem. Since perfect graphs do not remain perfect by any edge set contraction (a hole of size 6—which is perfect— does not remain perfect by the contraction of a single edge), there is no characterization of perfect graphs by forbidden induced minors.

By definition of contraction perfect graphs, a characterization in terms of forbidden induced minors can be directly deduced from the strong perfect graph theorem. Indeed, if a graph is not contraction perfect, then there exists an edge

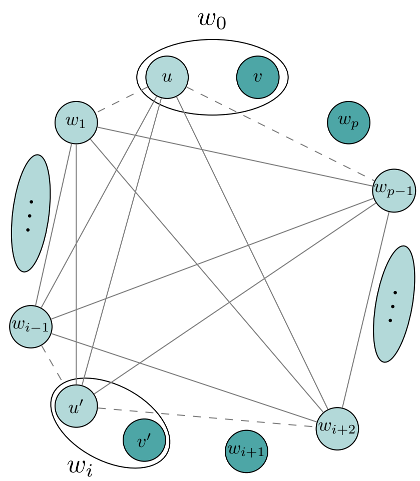


Figure 2.3: Illustration for the proof of Theorem 2.1.6:
the odd antihole $H_u \cup u' \setminus \{w_i, w_{i+1}\}$ of G' is given in light green

set contraction and a vertex set deletion that gives an odd hole or an odd antihole. Therefore, odd hole and odd antiholes are the induced minors to be forbidden in order to ensure contraction perfection. This list of forbidden induced minors is inclusionwise minimal because of the strong perfect graph theorem.

A list of forbidden induced subgraphs that characterizes contraction perfect graphs would be the set of graphs that yield an odd hole or an odd antihole by an edge set contraction. Using Theorem 2.1.6, which limits the set of such candidates, this section gives the minimal forbidden induced subgraph characterization of contraction perfect graphs.

Corollary 2.1.7. *A graph is contraction perfect if and only if it contains no hole of size at least 5, no odd antihole, and no expanded antihole as an induced subgraph.*

Proof. ($\Rightarrow$) By Theorem 2.1.5 and the strong perfect graph theorem, a graph containing one of these structures is not contraction perfect.
($\Leftarrow$) If the graph is not perfect, then it contains an odd hole or an odd antihole. If the graph is perfect but not contraction perfect, then there exists an edge whose contraction destroys the perfection of the graph by Theorem 2.1.6. By Theorem 2.1.5, this implies that the graph contains an even hole of size at least 6 or an expanded antihole. □

The next lemma gives an example of a non trivial contraction perfect graph class and will be used as a technical lemma. Remark that H is an induced subgraph of G if and only if $\overline{H}$ is an induced subgraph of $\overline{G}$.

Lemma 2.1.8. *Antipaths are contraction perfect.*

Proof. The complement $\overline{P}$ of an antipath P is a path. Since the path $\overline{P}$ is perfect, P is perfect by the weak perfect graph theorem [44]. An antipath only contains holes of size at most 4. An antipath contains no antihole since $\overline{P}$ contains no hole. Moreover, an antipath contains no expanded antihole. Indeed, any contraction of an edge uv in P yields a vertex whose neighborhood has size at least $|V(P)| - 2$. Hence the degree of such a vertex is too large for the latter to belong to an antihole. Hence, by Lemma 2.1.4, P does not contain any expanded antihole involving uv. The result holds by Corollary 2.1.7. □

Note that that the list of forbidden induced subgraphs of Corollary 2.1.7 is inclusionwise minimal. Indeed, if it were not the case, then one forbidden induced subgraph of Corollary 2.1.7 would contain another one as a proper induced subgraph, which is not the case as shown in the next observation.

A graph is *minimally non contraction perfect* if it is not contraction perfect and each of its proper induced subgraphs is contraction perfect.

Observation 1. *Expanded antiholes, odd antiholes and holes of size at least 5 are minimally non contraction perfect.*

Proof. Every proper induced subgraph of a hole is a set of disjoint paths and hence is contraction perfect. Every proper induced subgraph of an odd antihole is a set of fully connected antipaths, the complement of such a graph is a set of disjoint paths that is contraction perfect.

We will show that expanded antiholes are minimally non contraction perfect by considering the complement of an expanded antihole. Let us consider G an expanded antihole. Note that, $\overline{G}$ contains no antihole of size more than 6 since it contains precisely 6 vertices whose degree is greater or equal to 3, that are $W = \{u, v, w_1, w_2, w_{p-1}, w_p\}$. Indeed, since an antihole of size at least 6 is a regular graph whose vertices have degree at least 3, these are the only candidates that may belong to such an antihole. Since $\overline{G}[W]$ is not regular, it is not an antihole of size 6, this proves that G does not contain any holes of size at least 6. Obviously, $\overline{G}$ does not contain any complement of an expanded antihole as proper induced subgraphs. Now, note that the holes of $\overline{G}$ are even, which means that G only contains even antiholes and hence, no hole of size 5. By Corollary 2.1.7, expanded antiholes are contraction perfect. □

Corollary 2.1.9. *Recognizing contraction perfect graphs can be done in polynomial time.*

Proof. By Theorem 2.1.6, deciding whether a graph $G = (V, E)$ is contraction perfect amounts to check whether G is perfect and if, for each contraction of a single edge, the resulting graph is perfect. Each of these $|E| + 1$ perfection tests can be done in polynomial time [5]. □

2.2 Utter graph and co-2-plexes

2.2.1 Perfection of the utter graph

Let S be a co-2-plex. The vertex edge representation of a S is a pair (W, F) where W are the isolated vertices of $G[S]$ and F are its isolated edges. Then, by definition of co-2-plexes, F is a matching and contracting F gives $|F|$ isolated vertices nonadjacent to W. In other words, W and the image of F in G/F form a stable set of G/F of size $|W| + |F|$.

We use this remark to define the utter graph of a graph G in which the stable sets are in bijection with the co-2-plexes of G. The *utter graph* $u(G)$ of a graph $G = (V, E)$ has vertex set $V \cup E$ and two vertices in $u(G)$ are adjacent if and only if their corresponding elements in G are either adjacent, incident, or adjacent by contraction in G (defined in the introduction). Figure 2.4 gives an illustration of this definition where the vertices 12 and 23 of $u(G)$ respectively correspond to the edges 12 and 23 of G. For each edge uv of G, G/uv is the subgraph of $u(G)$ induced by $V \setminus \{u, v\} \cup uv$ where uv denotes the vertex of $u(G)$ associated with edges uv. More generally, for a matching $F \subseteq E$, G/F is the subgraph of $u(G)$ induced by $V \setminus V(F)$ and the vertices of $u(G)$ associated with edges in F.

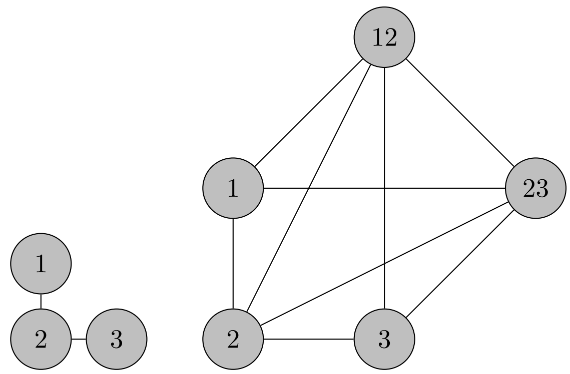
The utter graph is a new type of graph that is inspired by the total graph. The *total graph* of $G = (V, E)$ also has vertex set $V \cup E$. Still, its edge set is quite different: two elements are adjacent if they are incident edges of G, adjacents vertices of G, a vertex of G, and an edge of G incident to it. Let us give an example, let $G = (\{u, v, w\}, \{uv, vw\})$, in its utter graph, u and vw are adjacent, which is not the case in its total graph. These two graphs, even if they look alike, permit us to tackle different problems; the total graph permits to tackle a problem called total coloring, where we try to color every element of G, that is, its edge set and vertex set such that two edges have different colors if they are incident, two vertices have different colors if they are adjacent and a vertex and its incident edges have different colors and such with a minimum number of colors. It corresponds to a classic graph coloring of the total graph.

Lemma 2.2.1. *There is a bijection between the co-2-plexes of G and the stable sets of $u(G)$.*

Proof. By definition of utter graphs, a vertex subset W and an edge set F is a vertex edge representation (W, F) if and only if $W \cup T$ is a stable set of $u(G)$. □

Note that by definition of contraction perfect graphs, adding a true twin to any vertex preserves contraction perfection. Moreover, the forbidden induced subgraph characterization also proves that adding false twins preserves contraction perfection. Indeed, none of these forbidden induced subgraphs contain false twins. This gives the following replication lemma for contraction perfect graphs.

Lemma 2.2.2. *Adding true or false twins preserves contraction perfection.*



(a) G (b) $u(G)$

Figure 2.4: A graph G and its utter graph $u(G)$.

We say that a graph class is *replicable* if adding true twins to any graph of that class yields another graph of that class.

Theorem 2.2.3. *Let $\mathcal{C}$ be an induced minor closed and replicable graph class, then $G \in \mathcal{C}$ if and only if $u(G) \in \mathcal{C}$.*

Proof. ($\Leftarrow$) Since G is an induced subgraph of $u(G)$, the result follows from the assumptions on $\mathcal{C}$.
($\Rightarrow$) Let $\widetilde{G}$ be obtained from G by adding, for every edge $uv \in E$, a twin u' (resp. v') to u (resp. v). By construction, $\widetilde{G}$ contains the edge $u'v'$. Let F be the set of such edges for every edge $uv \in E$. Note that, $\widetilde{G}/F = u(G)$ and that $\widetilde{G}/F$ is obtained by adding true twins and contracting edges. Since $\mathcal{C}$ is replicable and induced minor closed, $u(G)$ belongs to $\mathcal{C}$. $\square$

Observation 2. *The conditions given in Theorem 2.2.3 are not necessary.*

Proof. Consider the sequence $G_{i+1} = u(G_i)$, for $i \in \mathbb{Z}^+$, where $G_0 = C_4$. Let $\mathcal{C}$ be the set of all G_i completed by induction with the graphs whose utter graph is in $\mathcal{C}$. Then, for any graph G and its utter graph $u(G)$, we have that $G \in \mathcal{C}$ if and only if $u(G) \in \mathcal{C}$. Note that each G_i contains a C_4 and by construction, if G is in $\mathcal{C}$, then there exists $i \in \mathbb{Z}^+$ such that, $G = G_i$ or there exists $k \in \mathbb{Z}^+$ such that $u^k(G) = G_i$. Since G is a clique if and only if $u(G)$ is a clique, no element of $\mathcal{C}$ is a clique, in particular $K_2 = (\{v_1, v_2\}, \{v_1v_2\})$ is not in $\mathcal{C}$ but is an induced subgraph of every element in $\mathcal{C}$. Hence $\mathcal{C}$ is not induced minor closed. $\square$

Since the class of contraction perfect graphs is replicable by Lemma 2.2.2 and induced minor closed by definition, the following equivalence *i)* $\Leftrightarrow$ *ii)* can be deduced from Theorem 2.2.3. We show that contraction perfect graphs are precisely those for which the utter graph is perfect. This means that every perfect utter graph is also contraction perfect.

Corollary 2.2.4. *The following statement are equivalent:*

i) G is contraction perfect

ii) $u(G)$ is contraction perfect

iii) $u(G)$ is perfect

Proof. Let $G = (V, E)$ be a graph.
($\Leftarrow$) By definition of utter graphs, G and G/e for $e \in E$ are induced subgraphs of $u(G)$. Hence, all those graphs are perfect since $u(G)$ is. By Theorem 2.1.6, this implies that G is contraction perfect.
($\Rightarrow$) If G is contraction perfect, then so is $u(G)$ by Corollary 2.2.4. By definition of contraction perfect graphs, this implies that $u(G)$ is perfect. $\square$

Given a cost function c on the vertices of G, let $\widetilde{c}_u = c_u$, for all $u \in V$, and $\widetilde{c}_{uv} = c_u + c_v$, for all $uv \in E$. Then finding a maximum c-weighted co-2-plex in G is equivalent to finding a maximum $\widetilde{c}$-weighted stable set in $u(G)$, and this can be done in polynomial time [1]. This gives the following corollary by Theorem 2.2.4.

Corollary 2.2.5. *Finding a maximum weight co-2-plex in a contraction perfect graph is solvable in polynomial time.*

Therefore when $u(G)$ is perfect, computing a minimum covering by stable sets in $u(G)$ can be done in polynomial time. Note that a minimum cover of a graph by stable sets is exactly a graph coloring. A further question is whether a graph coloring of $u(G)$ helps us to find a minimum covering of G by co-2-plexes. However, coloring the utter graph yields several coverings of G. Two opposite examples can be obtained as follows: a covering using only the stable sets of $u(G)[V]$, which are stable sets of G; and a covering using the stables of $u(G)[E]$ which are sets of edges of G pairwise nonadjacent by contraction.

In Figure 2.4, a covering of G by co-2-plexes involves two co-2-plexes, whereas a coloring of $u(G)$ is composed of four stables sets. Hence the two problems are not equivalent. An open question is whether a minimum co-2-plex covering of G can be deduced from a given coloring of $u(G)$. Indeed, in Figure 2.4, both examples given in the previous paragraphs yield minimal coverings of G $\{\{1,3\},\{2\}\}$ and $\{\{12\},\{23\}\}$.

2.2.2 Subclasses of perfect graphs

This section shows how Theorem 2.2.3 applies to other graph classes.

Trivially perfect graphs are interval graphs, split and interval graphs are chordal. All these graph classes are contraction perfect, which is not the case for k-hole-free graphs with $k \geq 4$.

A graph class $\mathcal{C}$ is said *stable by utter graphs* if: $G \in \mathcal{C}$ if and only if $u(G) \in \mathcal{C}$.

Corollary 2.2.6. *The split (resp. trivially perfect, interval, chordal, k-hole-free) graphs are stable by utter graphs. However, perfect graphs are not stable by utter graphs.*

Proof. One can show that all these graph classes are induced minor closed and replicable. We only detail the case of interval graphs.

Every induced subgraph $G_{\mathcal{I}}[W]$ of an interval graph $G_{\mathcal{I}}$ is the interval graph built from the interval set containing intervals of $\mathcal{I}$ associated with W. Contracting an edge uv in $G_{\mathcal{I}}$ corresponds to replacing the intervals associated with u and v in $\mathcal{I}$ by the union of both intervals. Adding a true twin to a vertex v in an interval graph $G_{\mathcal{I}}$ corresponds to duplicating the interval associated with v in $\mathcal{I}$. Then, from Theorem 2.2.3, interval graphs are stable by utter graphs.

Note that, for the k-hole-free graphs, the induced minor closeness follows from their definition and Lemma 2.1.2. A counter example for a perfect graph to be not stable by utter graphs is C_6 which is perfect, but whose utter graph contains many C_5. $\square$

Since the complements of perfect graphs are perfect, a natural question is to investigate the complement of contraction perfect graphs. However, $\overline{C_6}$ is an antihole of size 6, hence contraction perfect, but C_6 is not.

Observation 3. *The complement of a contraction perfect graph is not necessarily contraction perfect.*

Conclusion

In this chapter, we approach edge contraction in perfect graphs by characterizing the contraction perfect graphs in several manners: by contracting either a single edge or an edge set, by forbidden induced subgraphs and by the utter graph perfection. As a byproduct, the recognition of contraction perfect graphs is polynomial. Moreover, using utter graphs, the maximum weight co-2-plex problem becomes solvable in polynomial time in contraction perfect graphs. Finally, we focus on particular subclasses of perfect graphs.

For $k = 2$, it remains an intriguing question whether a coloring of the utter graph can be used to get a minimal covering of the starting graph with co-2-plexes. When $k > 2$, another interesting question is whether some extension of utter graphs could capture co-k-plexes as familiar combinatorial objects, like stable sets in utter graphs do for co-2-plexes. From a polyhedral point of view, the equivalence between co-2-plexes of a graph and stable sets of its utter graph will give extended formulations for the co-2-plexe polytope from the stable set polytope.

Contraction perfect graphs being a new subclass of perfect graphs, one may also be interested in combinatorial algorithms to find a maximum clique/stable set or a minimum coloring on such graphs. Since, unlike for perfect graphs, the complement of a contraction perfect graph is not necessarily contraction perfect, finding a maximum clique or a stable set may lead to distinct studies.

Acknowledgments

We thank Daria Pchelina for her contribution to Lemma 2.1.1 and recommend her thesis: "Density of Dics and Sphere Packings" which helps to understand the idea behind the proof of Kepler's conjecture with a set of beautiful illustrations going from her work on the packing of discs with three different sizes in two dimensions to the packing of spheres with two different sizes in dimension 3. Since she considers packings in unbounded spaces, with an unbounded number of elements (discs/spheres), the results we present in Chapter 1 on packing polytope and perfect graphs do not seem to be linked to her work.

Chapter 3

The maximum weighted co-2-plex problem

In this chapter, we first present a new ILP formulation for the maximum weighted co-2-plex problem. This formulation can be seen as an improvement of a linearization of the quadratic model $\mathcal{Q}_2$. We show that this formulation provides a tighter linear relaxation than the formulation $\mathcal{N}_2$ of Balasundaram [7] that we will denote by $\mathcal{N}$ as we focus on the case $k = 2$.

We present a second formulation using the bijection of Lemma 2.2.1 between the co-2-plexes of a graph and the stable sets of its utter graph. The inequalities of this second formulation yield valid inequalities for the first formulation. Adding these inequalities gives a formulation with a tighter linear relaxation than the one of Balasundaram strengthened by 2-plex constraints (1.12).

The second part of this chapter is dedicated to the study of the co-2-plex polytope. As a direct consequence of the previous chapter, we get an extended formulation for the co-2-plex polytope of contraction perfect graphs. This extended formulation becomes compact for chordal graphs. By projecting it onto the natural variable's space we obtain a complete description of the co-2-plex polytope of trees. Finally, we present our tries at projecting this formulation onto the natural variable's space for split graphs.

In the last part of this chapter, we compare different implementations of our formulations and compare them with $\mathcal{N}$ reinforced by 2-plex inequalities.

3.1 Integer linear programming models

In this section, we exhibit new extended space ILP formulations for the maximum co-2-plex problem and compare them from a theoretical point of view.

3.1.1 Linearizing the quadratic model

A way to obtain an integer linear programming model for the max co-2-plex problem is to apply the classic Reformulation-Linearization Technique (RLT) [45] on the quadratic model $\mathcal{Q}_2$ introduced in Chapter 1 page 23. Since a quadratic term $x_u x_v$ appears in this formulation for each edge uv, the linearization consists of adding binary y variables associated with the edges of the graph and ensuring that $y_{uv} = x_u x_v$ with the following RLT inequalities [45]:

$$x_u + x_v - y_{uv} \leq 1 \qquad \forall uv \in E \tag{3.1}$$

$$y_{uv} \leq x_u \qquad \forall u \in V, \forall uv \in \delta_u \tag{3.2}$$

Replacing the quadratic terms of inequalities (1.18) by variables y yields the following inequalities:

$$y(\delta_v) \leq 1 \quad \forall v \in V \tag{3.3}$$

The model maximizing $w^\top x$ with x, y binary variables satisfying inequalities (3.1)-(3.2) is a valid ILP model for the maximum weighted co-2-plex. However, it can be strengthened as follows.

Proposition 3.1.1. *The following* edge formulation*, denoted by $\mathcal{E}$ is valid for the maximum w-weighted co-k-plex problem:*

$$\max w^\top x \tag{3.4}$$

$$x_u + x_v - y_{uv} \leq 1 \qquad \forall uv \in E \tag{3.5}$$

$$y(\delta_v) \leq x_v \qquad \forall v \in V \tag{3.6}$$

$$y_e \in \{0,1\} \qquad \forall e \in E \tag{3.7}$$

$$x_v \in \{0,1\} \qquad \forall v \in V \tag{3.8}$$

Proof. Every binary point satisfying Inequalities (3.2) and (3.3) also satisfies constraints (3.6). Moreover, Inequalities (3.2) and (3.3) are redundant with respect to Inequalities (3.6) and the trivial inequalities: the Inequality (3.2) associated with $u \in V$ and $uv \in \delta_u$ is the sum of the Inequality (3.6) associated with u and $y_e \geq 0$ for all $e \in \delta_v \setminus uv$, and the Inequality (3.3) associated with $v \in V$ is the sum of (3.6) associated with v and $x_v \leq 1$. $\square$

As a consequence, we get that the linear relaxation of $\mathcal{E}$ is tighter than the linear relaxation of the standard linear reformulation of $\mathcal{Q}_2$. We now show that $\mathcal{E}$ provides a tighter linear relaxation than $\mathcal{N}$. For this, we show that the projection onto the x variables of the polytope subjacent to $\mathcal{E}$ is included in the polytope subjacent to $\mathcal{N}$ and that there exist graphs for which this inclusion is strict. The polytope subjacent to $\mathcal{E}$ is defined as:

$$\mathcal{P}_{\mathcal{E}} = \{(x, y) \in \mathbb{R}^V \times \mathbb{R}^E_+ \mid (x, y) \text{ satisfies (3.5) and (3.6)}\}.$$

Indeed, trivial inequalities $0 \leq x_v \leq 1 \ \forall v \in V$ and $y_e \leq 1 \ \forall e \in E$ are redundant for $\mathcal{E}$ as they are obtained as sums of its constraints. Note that no trivial constraints are needed of x variables.

Theorem 3.1.2. *The projection of $\mathcal{P}_{\mathcal{E}}$ onto the space associated with x variables is equal to the following:*

$$\left\{\begin{aligned} x(W) + (|W|-1)x_w &\leq |W| & \forall w \in V,\ W \subseteq N_w, \\ -x_v &\leq 0 & \forall v \in V \\ x_v &\leq 1 & \forall v \in V \end{aligned}\right\} \qquad \begin{aligned}(3.9)\\(3.10)\\(3.11)\end{aligned}$$

Proof. This projection is made as a direct application of Fourier-Motzkin procedure. The only inequalities of $\mathcal{P}_{\mathcal{E}}$ having a negative coefficient for variables y are nonnegativity inequalities and inequalities (3.5). They have exactly one negative coefficient -1 on y variables. For this reason, a nondominated inequality of $proj_x\mathcal{P}_{\mathcal{E}}$ involves exactly one inequality (3.6) associated with a vertex w, a set of trivial inequalities associated with $F_1 \subseteq \delta_w$, a set of inequalities (3.5) associated with F_2 such that (F_1, F_2) partition δ_w. By setting $F_2 = \delta_w \cap \delta(W)$ and hence, $F_1 = \delta_w \setminus F_2$ we prove the result. □

Recall that the polytope subjacent to $\mathcal{N}$ denoted by $\mathcal{P}_{\mathcal{N}}$ is defined by the following:

$$\left\{\begin{aligned} x(N_u) + (|N_u|-1)x_u &\leq |N_u| & \forall u \in V \\ x_u &\leq 1 & \forall u \in V \\ -x_u &\leq 0 & \forall u \in V \end{aligned}\right\} \qquad (3.12)$$

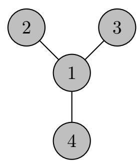
Note that the Inequality (3.12) associated with $v \in V$ is nothing but the Inequality (3.9) associated with v and N_w. Hence, $proj_x(\mathcal{P}_{\mathcal{E}}) \subseteq \mathcal{P}_{\mathcal{N}}$ which implies that $\mathcal{E}$ provides a linear relaxation as tight as the one of $\mathcal{N}$. We prove that it is actually tighter.

Corollary 3.1.3. *Formulation $\mathcal{E}$ provides a tighter linear relaxation than Formulation $\mathcal{N}$.*

Proof. Consider the star with node 1 as center and nodes 2, 3, and 4 as leaves, see Figure 3.1. The point $\bar{x}_1 = \frac{1}{2}$, $\bar{x}_2 = \bar{x}_3 = 1$ and $\bar{x_4} = 0$ is an extreme point of $\mathcal{P}_{\mathcal{N}}$ since its satisfies with equality three trivial inequalities and the inequality (3.12) associated with node 1. However, $\bar{x}$ does not belong to $\mathcal{P}_{\mathcal{E}}$ since it violates the constraint (3.9) associated with node 1 and neighbors 2 and 3. □

3.1.2 Formulations based on the utter graph

In Chapter 2, we established a bijection between the co-2-plexes of a graph and the stable sets of its utter graph. Indeed, a co-2-plex S of $G = (V, E)$ is a vertex subset inducing graph $G[S]$ of maximum degree one. Hence, the vertex edge representation of S is a pair (W, F) where W contains the isolated vertices of $G[S]$ and F contains its isolated edges. By construction, the vertex set of



$$\begin{cases} x_1 + x_2 + x_3 & \leq & 2 \\ x_1 + x_2 + x_4 & \leq & 2 \\ x_1 + x_3 + x_4 & \leq & 2 \\ 2x_1 + x_2 + x_3 + x_4 & \leq & 3 \\ x_i & \geq & 0 \quad \forall i \in \{1, 2, 3, 4\} \end{cases}$$

Figure 3.1: A star and the linear description of its co-2-plex polytope

$u(G)$ is equal to $V \cup E$. Lemma 2.2.1 asserts that (W, F) is the vertex edge representation of a co-2-plex if and only if $W \cup F$ is a stable set in $u(G)$.

This bijection allows us to model the co-2-plex problem as a stable set problem on $u(G)$ as follows. Let z_v $\forall v \in V$ and y_e $\forall e \in E$ be binary variables indicating which nodes of $u(G)$ belong to the stable set. Since a stable set is a vertex subset intersecting at most once each clique, we get the following stable set formulation $\mathcal{S}$.

$$\max w^\top z + \sum_{uv \in E} (w_u + w_v) y_{uv} \tag{3.13}$$

$$z(W) + y(F) \leq 1 \qquad \forall \text{ maximal clique } W \cup F \text{ of } u(G) \tag{3.14}$$

$$z_u \in \{0, 1\} \qquad \forall u \in V \tag{3.15}$$

$$y_e \in \{0, 1\} \qquad \forall e \in E \tag{3.16}$$

Note that the solutions of $\mathcal{S}$ are the incidence vectors of the vertex edge representations of co-2-plexes of G. As in the vertex edge representation (W, F) of a co-2-plex S, $uv \in F$ implies that both u and v belong to S, the objective coefficient of y_{uv} has to be $w_u + w_v$.

In the stable set formulation $\mathcal{S}$, a vertex $v \in V$ belongs to the co-2-plex corresponding to a solution if $z_v = 1$ or $y_e = 1$ for some $e \in \delta_v$. One can apply a linear transformation to have a binary variable associated with each vertex $v \in V$ that equals one if and only if v belongs to the co-2-plex. This leads to the following utter clique formulation $\mathcal{UK}$, where an *utter clique* of G is a couple

(W, F) with $W \subseteq V$ and $F \subseteq E$ such that the vertices associated with W, F form a clique in $u(G)$. An utter clique (W, F) is *maximal* if $W \cup F$ is a maximal clique of $u(G)$. There is hence a one-to-one correspondence between maximal utter cliques of G and maximal cliques of $u(G)$.

Theorem 3.1.4. *The following* utter clique formulation $\mathcal{UK}$ *is valid for the maximum co-2-plex problem:*

$$\max w^\top x$$

$$x(W) + y(F \cap E(V \setminus W)) - y(E(W)) \leq 1 \quad \forall \textit{ maximal utter clique } (W, F) \textit{ of } G \tag{3.17}$$

$$y(\delta_v) \leq x_v \quad \forall v \in V \tag{3.18}$$

$$x_v \in \{0, 1\} \quad \forall v \in V \tag{3.19}$$

$$y_e \in \{0, 1\} \quad \forall e \in E \tag{3.20}$$

Proof. First, we add x variables associated with vertices of G to the stable set formulation of its utter graph and the following constraints:

$$x_v = z_v + y(\delta_v) \quad \forall v \in V \tag{3.21}$$

This gives the following formulation.

$$\max w^\top z + \sum_{uv \in E} (w_u + w_v) y_{uv}$$

$$z(W) + y(F) \leq 1 \quad \forall \text{ maximal utter clique } (W, F) \text{ of } G \tag{3.22}$$

$$-z_u \leq 0 \quad \forall u \in V \tag{3.23}$$

$$-y_{uw} \leq 0 \quad \forall uw \in E$$

$$z_u \in \{0, 1\} \quad \forall u \in V$$

$$y_e \in \{0, 1\} \quad \forall e \in E$$

$$x_u - z_u - y(\delta_u) = 0 \quad \forall u \in V$$

The theorem is then proven by projecting the latter formulation onto its x and y variable space. This is done by replacing every occurrence of z_u in the above formulation by $x_u - y(\delta_u)$:

- Constraints (3.17) arise from Constraints (3.22): the combination yields $x(W) + y(F) - \sum_{u \in W} y(\delta(u)) \leq 1$. As if $v \in V$ belongs to V, then by maximality of the clique, every vertex of $u(G)$ associated with an edge $uv \in E$ also belongs to F. Hence, every variable associated with an edge of G adjacent to only one vertex of W in $u(G)$ will vanish. The edges of G adjacent to two vertices of W in $u(G)$ will have a -1 coefficient. The coefficients of the other variables remain the same.
- Constraints (3.18) will arise from the trivial Constraints (3.23).

- And finally, $w^\top x = w^\top z + \sum_{uv \in E}(w_u + w_v) y_{uv}$

$\square$

Note that nonmaximal utter cliques of G yield valid but redundant inequalities for $\mathcal{UK}$.

Proposition 3.1.5. *Formulation $\mathcal{UK}$ is tighter than the linear relaxation of $\mathcal{E}$.*

Proof. Note that the Constraint (3.2) associated with edge $uv \in E$ is nothing but the utter clique inequality (3.17) associated with the utter clique $(\{u, v\}, \delta_u \cup \delta_v)$. Hence, $\mathcal{P}_{\mathcal{UK}} \subseteq \mathcal{P}_{\mathcal{E}}$. Consider a hole of size 4 (u, v, z, w). Then, the point $x^*_u = x^*_v = x^*_z = x^*_w = y^*_{uv} = \frac{1}{2}$ belongs to $\mathcal{P}_{\mathcal{E}}$ but violates the utter clique inequality (3.17) associated with utter clique $(\{z, w\}, \{uv\})$. $\square$

By Corollary 3.1.3 and Proposition 3.1.5, Formulation $\mathcal{UK}$ provides a tighter linear relaxation than $\mathcal{N}$. We prove in Proposition 3.1.6 that this still holds even if we strenghten Formulation ($\mathcal{N}$) with the following 2-plex inequalities introduced in [32]:

$$x(K) \leq 2 \quad \text{for all 2-plexes } K \text{ of } G \tag{3.24}$$

Let $\mathcal{T}$ denote the polytope corresponding to the linear relaxation of $\mathcal{N}$ strenghtened by the 2-plex inequalties, that is,

$$\mathcal{T} = \mathcal{P}_{\mathcal{N}} \cap \{x \in \mathbb{R}^{|V|} \mid x \text{ satisfies (3.24)}\}.$$

Proposition 3.1.6. *Formulation $\mathcal{UK}$ provides a tighter linear relaxation than $\mathcal{N}$ strenghtened by* (3.24)

Proof. By Corollary 3.1.3 and Proposition 3.1.5, to prove that $proj(\mathcal{P}_{\mathcal{UK}})$ is included in $\mathcal{T}$, it suffices to show that the 2-plex inequalities are redundant with respect to (3.17), (3.18) and the trivial inequalities. Let K be a 2-plex of G. By definition of 2-plexes, K induces a complete graph in which a matching has been removed. Taking one extremity of each edge of this matching yields a node set K_1. Then, K_1 is a clique of G, as well as $K_2 = K \setminus K_1$. Moreover, for all $v \in K_1$ and $e \in E(K_2)$, v and e are adjacent by contraction so $(K_1, E(K_2))$ is an utter clique of G, and so is $(K_2, E(K_1))$. Thus, the inequality associated with K is the sum of the utter clique inequalities associated with $(K_1, E(K_2))$ and $(K_2, E(K_1))$ and is redundant.

It remains to exhibit a graph for which the inclusion is strict. Consider the star with node 1 as center and nodes 2, 3, 4 and 5 as leaves. The point $\bar{x}_i = \frac{2}{3}$ for $i = 1, \ldots, 4$ and $\bar{x}_5 = 0$ is valid for $\mathcal{T}$. Indeed, each maximal 2-plex contains node 1 and two nodes among 2, 3, 4 and 5 so all 2-plex inequalities are satisfied. Moreover, it satisfies inequalities of $\mathcal{P}_{\mathcal{N}}$. However, it violates the inequality (3.12) associated with node 1 and neighbors 2, 3 and 4. As this inequality is valid for $proj(\mathcal{P}_{\mathcal{E}})$ and $proj(\mathcal{P}_{\mathcal{E}}) \subseteq proj(\mathcal{P}_{\mathcal{UK}})$ by Proposition 3.1.5, it is valid for $proj(\mathcal{P}_{\mathcal{UK}})$ which ends the proof. $\square$

3.2 Extended space polyhedral characterizations

In this section, we present an extended formulation describing the co-2-plex polytope of any contraction perfect graph G using the utter graph $u(G)$ defined in Chapter 2. As a corollary, we will obtain a compact extended formulation for the co-2-plex polytope of chordal graphs.

The following Theorem can be seen as a Corollary of Theorem 2.2.4 and the weak perfect graph theorem.

Corollary 3.2.1. *A graph G is contraction perfect if and only if $\mathcal{P}_{\mathcal{UK}}$ is integer.*

Proof. By Theorem 2.2.4, G is contraction perfect if and only if $u(G)$ is perfect and, by the weak perfect graphs theorem if and only if the stable set polytope of $u(G)$ is described by maximal clique inequalities if and only if the polytope subjacent to $\mathcal{UK}$ is integer. Note that $x_v = z_v + y(\delta_v)$ preserves integrality in the sense that (x, y) is integer if and only (z, y) is also which implies the equivalence. □

Any orthogonal projection of an integer polyhedron is an integer polyhedron, which yields the following.

Corollary 3.2.2. *If G is contraction perfect, then the co-2-plex polytope of G is exactly $proj_x(\mathcal{P}_{\mathcal{UK}})$.*

We do not know if a polynomial bounds the number of maximal cliques of a contraction perfect graph, hence, the formulation given in Corollary 3.2.1 may not be compact.

In the following, we exhibit a compact extended formulation of the co-2-plex polytope for chordal graphs. Consider the following set of inequalities:

$$x(K) - y(E(K)) \leq 1 \quad \forall \text{ maximal clique } K \text{ of } G \tag{3.25}$$

and let $\mathcal{P}_{\mathcal{K}} = \{(x, y) \in \mathbb{R}^V \times \mathbb{R}^E_+ \mid (x, y) \text{ satisfies (3.18) and (3.25)}\}$.

Theorem 3.2.3. *A graph G is chordal if and only if $\mathcal{P}_{\mathcal{K}}$ is integer.*

Proof. ($\Rightarrow$) Suppose that $G = (V, E)$ is chordal and let us show that for any maximal utter clique (W, F) of G, $F' = F \cap E(V \setminus W)$ is empty. By contradiction, suppose that there exists an edge $uv \in F'$. Then, u and v do not belong to W. Since $W \cup F$ is a maximal clique of $u(G)$, we know that there exists at least one vertex $w_u \in W \cup F$ (resp. w_v) that is non-adjacent to u (resp. v) in $u(G)$. We have $w_u \neq w_v$ as otherwise, by contradiction, uv would be non-adjacent to w_u in $u(G)$, a contradiction with the fact that $W \cup F$ is a clique. We consider three cases: *i)* both w_u and w_v belong to W, *ii)* only one belongs to W, *iii)* and finally both belong to F.

i) Since $uv \in F$, we know that uv is adjacent to w_u and w_v (in $u(G)$), in which case u is adjacent to w_v and v is adjacent to w_u which means that $G[\{u, v, w_u, w_v\}]$ is a hole of size 4, a contradiction.

ii) We suppose that w_u is associated to the edge $zt \in E$. Now, since uv is adjacent to zt and u is not adjacent to zt, v is adjacent to z or t, and w_v is adjacent to t or z since w_v is adjacent to w_u. By taking a shortest path P from w_v to v in $G[\{w_v, t, z, v\}]$, we obtain that $G[P \cup u]$ is a hole of size 4 or 5, a contradiction.

iii) Now we suppose that w_u (resp. w_v) is associated to an edge $tz \in E$ (resp. $rs \in E$). Analogously, by taking a shortest path P from u to v in $G[u, v, t, z, r, s] \setminus uv$, we obtain that $G[u, v, t, z, r, s]$ contains a hole of size 4, 5 or 6, a contradiction.

Hence F' is empty. Since by definition, W is a clique of G, the utter clique inequality (3.17) associated with (W, F) is nothing but the clique inequality (3.25) associated with W. Note that the trivial inequalities $0 \leq x_v \leq 1$ and $y_e \leq 1$ are redundant with respect to inequalities (3.18), (3.25) and $y_e \geq 0$ for all $e \in E$. This implies that $\mathcal{P}_{\mathcal{K}} = \mathcal{P}_{\mathcal{UK}}$. Since $G = (V, E)$ is chordal, it is contraction perfect, and hence by Corollary 3.2.1, $\mathcal{P}_{\mathcal{UK}}$ is an integer polytope, and so is $\mathcal{P}_{\mathcal{K}}$.

($\Leftarrow$) By contradiction, suppose that G is not chordal. Since $(K, E(K))$ is an utter clique when K is a clique, and since the trivial inequalities except $y_e \geq 0$ for all $e \in E$ are redundant for $\mathcal{P}_{\mathcal{K}}$, it follows that $\mathcal{P}_{\mathcal{UK}} \subseteq \mathcal{P}_{\mathcal{K}}$. Both polytopes have the same set of integer points by Propositions 3.1.1 and 3.1.4 since the inequality (1.2) associated with edge uv is the inequality (3.25) associated with clique $\{u, v\}$ if $\{u, v\}$ is maximal, or dominated by a inequality (3.25) associated with a maximal clique containing both u and v. If G is not contraction perfect, by Theorem 2.2.4, $\mathcal{P}_{\mathcal{UK}}$ is not integer, so it contains a fractional point $(\bar{x}, \bar{y})$ that is not a convex combination of its integer points. Since $(\bar{x}, \bar{y})$ belongs to $\mathcal{P}_{\mathcal{K}}$, $\mathcal{P}_{\mathcal{K}}$ is not integer.

Suppose now that G is contraction perfect but not chordal. Hence, it contains a hole of size 4, say (u, v, w, z). Let K_1, K_2 and K_3 be three maximal cliques satisfying $K_1 \cap \{u, v, w, z\} = \{u, z\}$, $K_2 \cap \{u, v\} = \{v, w\}$ and $K_3 \cap \{u, v, w, z\} = \{v, w\}$. Let (x^*, y^*) be such that $x^*_u = x^*_v = x^*_z = x^*_w = y^*_{zw} = \frac{1}{2}$ and all other components are 0. (x^*, y^*) satisfies the clique inequalities associated with K_1, K_2 and K_3, with equality. The inequality (3.18) associated with each vertex but u and v, and the trivial inequality $y_e \geq 0$ associated with $e \in E \setminus \{vw\}$. These inequalities are linearly independent so (x^*, y^*) is a fractional extreme point of $\mathcal{P}_{\mathcal{K}}$. □

Corollary 3.2.4. *If G is chordal, then $\mathcal{P}_{\mathcal{K}}$ is a compact extended formulation of $\mathcal{P}_2(G)$.*

Proof. If G is chordal then, by Theorem 3.2.3, $\mathcal{P}_{\mathcal{K}}$ is integer and, as the number of cliques of a chordal graph is linear in the number of vertices [46], it has compact size. □

3.3 On the co-2-plex polytope of trees

The edge formulation without integrality constraints is an extended space polyhedral description of the co-2-plex polytope of trees.

Corollary 3.3.1. *A graph $G = (V, E)$ is a tree, if and only if $\mathcal{P}_\mathcal{E}$ is integer.*

Proof. A graph is a tree if and only if it is chordal and its maximal cliques are edges. By Theorem 3.2.3 it is chordal and its maximal cliques are edges if and only if $\mathcal{P}_\mathcal{K}$ is integer and its maximal cliques are edges. Noticing that edges are cliques of size 2 ends the proof. □

As a corollary of Theorem 3.1.2 and Theorem 3.3.1, we obtain the following theorem.

Theorem 3.3.2. *The co-2-plex polytope of a tree $G = (V, E)$ is equal to the following:*

$$\left\{\begin{array}{lr} x(W) + (|W| - 1)x_w \leq |W| & \forall w \in V,\ W \subseteq N_w, |W| \geq 2 \\ -x_v \leq 0 & \forall v \in V \\ x_v \leq 1 & \forall v \in V \end{array}\right\} \quad \begin{array}{r}(3.26)\\(3.27)\\(3.28)\end{array}$$

The system is minimal as shown in the next theorem. We denote by χ^W the incidence vector of the vertex set $W \subseteq V$.

Theorem 3.3.3. *If $G = (V, E)$ is a tree, then the following inequalities define facets of its co-2-plex polytope:*

(i) $x(W) + (|W| - 1)x_w \leq |W|\ \forall w \in V,\ W \subseteq N_w$

(ii) $-x_v \leq 0\ \forall v \in V$

(iii) $x_v \leq 1\ \forall v \in V$

Proof. The two first statements are proven by exhibiting $|V|$ affinely independent points on the face.

Statement *iii*): $\chi^{u,v}\ \forall u \in V \setminus v$ and χ^v belong to the face $x_v = 1$

Statement *ii*): $\chi^u\ \forall u \in V \setminus v$ and $\chi^\emptyset$ belongs to the face induced by $x_v = 0$.

Statement *i*): Note that if $W = \{u\}$, the obtained inequality is $x_u \leq 1$ which defines a facet. We prove the remaining cases by maximality. Take $w \in V$, $W \subseteq N_w$ of size at least 2 and consider the inequality $x(W)+(|W|-1)x_w \leq |W|$ and the face F of $\mathcal{P}^2_G$ that it defines. Suppose that there exists a face F' of $\mathcal{P}^2_G$ containing F and described by $ax \leq \alpha$.

Claim 1 $\mathbf{a_u = 0\ \forall u \in V \setminus \overline{N}_w}$: as G is a tree, $\forall u \in V \setminus \overline{N}_w$ there exists a vertex v of W non adjacent to u, in this case $\chi^{v,w,u}$ and $\chi^{w,v}$ both belong to F, and hence belong to F' implying that $a_v + a_w + a_u = a_v + a_w$ and hence $a_u = 0$.

Claim 2 $\mathbf{a_u = 0\ \forall u \in N_w \setminus W}$: similarly, $\chi^{W \cup u}$ and χ^W both belong to F.

Claim 3 $\mathbf{a_u = a_v\ \forall u, v \in W}$: $\chi^{v,w}$ and $\chi^{u,w}$ both belong to F.

Claim 4 $\boldsymbol{\alpha} = \mathbf{a_u |W|\ \forall u \in V}$: this is obtained by Claim 3 and the fact that χ^W belongs to F.

Finally, as for $u \in W$, $\chi^{w,u} \in F$, we have $a_w = \alpha - a_u$ we obtain that $ax \leq \alpha$ is a multiple of $x(W) + (|W| - 1)x_w \leq |W|$ which implies that $F = F'$. □

3.4 On the co-2-plex polytope of split graphs

In this section, I will summarize our research process on split graphs. Unfortunately, we were not able to conclude and characterize the co-2-plex polytope of split graphs in the natural variable space. However, I believe that this section enlightens the broadness of both Balas's theorem and Fourier-Motzkin elimination procedure by giving nontrivial applications of these results.

3.4.1 The existence of a compact extended formulation for the co-2-plex polytope of split graphs

Bala's theorem is not only strong for its explicit extended formulation of the convex hull of two known polytopes, but, as a corollary, it gives a framework to prove the existence of a compact extended formulation of some polytopes. We show how it can be applied for such a purpose in the case of co-2-plex polytope of split graphs. After that, we discuss what are the requirements to apply Bala's theorem and deduce from it the existence of a compact formulation for such a polytope.

Theorem 3.4.1. *Let $G = (V, E)$ be a split graph whose vertex set is partitioned into a clique K and a stable set S, there exists a compact description of $\mathcal{P}_G^2$ with $O(|V||K|)$ constraints.*

Proof. We proceed by induction on the size of K. When $|K| = 0$, $G = (V, E)$ is a stable set, hence, the co-2-plex polytope of G is the hypercube of size $|V|$ for which the description in the natural variables set has $|V|$ variables and $2|V|$ constraints. Now, let us suppose that there exists a compact formulation $P_1 = \{x^1 \in \mathcal{R}^{|V|} | Ax^1 \leq b,\ x_u^1 = 0\}$ for the co-2-plex polytope of $G \setminus u$ with at most ℓ constraints for some $u \in K$.

Now, we aim for an explicit formulation of the polytope whose extreme points are co-2-plexes containing u incidence vectors. Any co-2-plex S_u of G containing u contains at most one vertex of N_u by definition of co-2-plexes. If a vertex $v \in S_u \setminus N_u$ belongs to a co-2-plex C, then no vertex of $K \cap N_v \setminus u = N_v \setminus u = N_v$ belongs to C by definition of split graphs and co-2-plexes. This implies that $S_u \setminus \overline{N}_u$ is a stable set of the graph G' obtained from G by removing u and adding an edge between all pairs of vertices in N_u. G' is a split graph since it is partitioned into $K \cup N_u \setminus u$ and $S_u \setminus N_u$. Hence, it is perfect, and its stable set polytope is described by clique inequalities by the weak perfect graph Theorem. Let $\mathcal{S}$ be the set of stable sets of G', the convex hull of the incidence vectors of $S' \cup \{u\} \forall S' \in \mathcal{S}$ is then described by the following where the variables x^2 are associated with vertices of G:

$$P_2 = \left\{ x^2 \in \mathbb{R}^{|V|} \;\middle|\; \begin{array}{rcll} x^2(\overline{N}_v) & \leq & 1 & \forall v \in S \setminus N_u \\ x^2(K \cup N_u \setminus u) & \leq & 1 & \\ -x^2(v) & \leq & 0 & \forall v \in V \setminus u \\ x_u^2 & = & 1 & \end{array} \right\}$$

Remember that x^1 has size at most l by the induction hypothesis, after adding sufficiently enough zero components to x^2 (so its size is l), we apply Balas's theorem to compute the convex hull of $P_1 \cup P_2$:

$$\mathcal{P}_G^2 = proj_x \left\{ \begin{array}{rcll} x^1 + x^2 & = & x & \\ x^2(\overline{N}_v) & \leq & \lambda & \forall v \in S \setminus N_u \\ x^2(K \cup N_u \setminus u) & \leq & \lambda & \\ -x_v^2 & \leq & 0 & \forall v \in V \setminus u \\ x_u^2 & = & \lambda & \\ Ax^1 & \leq & (1-\lambda)b & \\ x_u^1 & = & 0 & \\ \lambda & \in & [0,1] & \end{array} \right\}$$

Now, using relations $x - x^1 = x^2$, $x_u = 0 + x_u^2 = \lambda$, we get rid of variables x^2, x_u^1, λ and the zero components of x associated with null components of x^2:

$$\mathcal{P}_G^2 = proj_x \left\{ \begin{array}{rcll} x(\overline{N}_v) - x^1(\overline{N}_v) & \leq & x_u & \forall v \in S \setminus N_u \\ x(K \cup N_u \setminus u) - x^1(K \cup N_u \setminus u) & \leq & x_u & \\ x_v^1 - x_v & \leq & 0 & \forall v \in V \setminus u \\ Ax^1 & \leq & (1-x_u)b & \\ x_u & \in & [0,1] & \end{array} \right\}$$

The latter formulation has $|V| + \ell$ variables, and $|V| + |S \setminus N_u| + \ell + 2$ constraints. This ends our proof. $\square$

Note that the proof of Theorem 3.4.1 is generic and could be used for various polytopes whose extreme points are other combinatorial objects. The requirement is to exhibit a disjunction on this combinatorial object set for which at most one element of the disjunction 'has the same nature' (and for which the induction hypothesis will be used to characterize its polytope with an implicit extended formulation). In this example, the disjunction is: "A co-2-plex of G, is *either* a co-2-plex of G containing u, *or* a co-2-plex of $G \setminus u$". The first element of this disjunction is equivalent to a stable set in an 'easy' graph, for which we know an explicit extended formulation. The second one has the same nature: "a co-2-plex of G". Hence, we use an implicit extended formulation to describe its polyhedron, here $A^1 x \leq b$. If for more than one element of the disjunction, it is necessary to use the implicit formulation to describe their polytope, the resulting formulation is no more compact since at each iteration of the induction, we may give a formulation with $k * \ell$ variables/constraints for $k \geq 2$, which would end up in an exponential procedure in the size of the induction: it is unclear how to simplify the variables associated with the several implicit formulations which imply that each iteration may result in doubling the variable set.

Theorem 3.4.1 is generally not satisfying since it does not exhibit this formulation but can help find an extended formulation in particular by giving an upper bound on its size. Moreover, it permits confirmation that it exists, which is a good preliminary in a research process.

The formulation of Theorem 3.2.3 yields an explicit version of Theorem 3.4.1 for split graphs since the set of edges in a split graph can have size $O(\frac{K(K-1)}{2} + |S||K|) = O(|V||K|)$ and its number of cliques is $|S| + 1$.

3.4.2 Expressions obtained using Fourier-Motzkin

For any split graph $G = (V, E)$ partitioned into the maximal clique K, and the stable set S, its set of maximal cliques is $\{K\} \cup \{\overline{N}_u | u \in S\}$. Hence, by Corollary3.2.4, its co-2-plex polytope is equal to the projection onto the x variable space of the following polytope:

$$\left\{\begin{array}{rl} x(\overline{N}_u) - y(E(\overline{N}_u)) \leq 1 & \forall u \in S \\ x(K) - y(E(K)) \leq 1 & \\ y(\delta_u) - x_u \leq 0 & \forall u \in V \\ -y_e \leq 0 & \forall e \in E \end{array}\right\} \begin{array}{r} (3.29) \\ (3.30) \\ (3.31) \\ (3.32) \end{array}$$

The Fourier-Motzkin elimination procedure tells us that every facet defining inequality for $\mathcal{P}_2(G)$ on split graphs is obtained by a combination of inequalities such that y variables vanish. Let us consider such an integer combination defined by:

- $n_u \in \mathbb{Z}_+$ for $u \in S$ the number of clique inequalities of type 3.29 associated with $\overline{N}_u$
- $m_u \in \mathbb{Z}_+$ for $u \in V$ the number of inequalities of type 3.31 associated with u
- k the number of inequality associated with K
- p_E the number of trivial inequalities associated with e

The obtained inequality has the form:

$$\sum_{u \in S} n_u x(\overline{N}_u) + kx(K) - \sum_{v \in V} m_v x_v \leq \sum_{u \in S} n_u + k \tag{3.33}$$

Since the y variables vanish, we know that:

$$m_u + m_v = p_{uv} + n_u \ \forall u \in S,\ uv \in \delta_u \tag{3.34}$$

$$m_u + m_v = p_{uv} + k + \sum_{w \in S | u,v \in \overline{N}_w} n_w \quad \forall uv \in E(K) \tag{3.35}$$

Every combination can be seen as a cover of the edges of a multiset of cliques by a multiset of stars (edges in δ_u for some u). Note that if n, m, k, p verify Equations 3.34 and 3.35, for any $\lambda \in \mathbb{R}^*$, $\lambda n, \lambda m, \lambda k, \lambda p$ also verify these constraints. Hence, the set of such combinations forms a cone (a polyhedral cone since the constraints are linear) called the *projection cone*. We will look at such an object in section 3.4.4.

Lemma 3.4.2. *If $\alpha \leq a$ is facet defining for $\mathcal{P}_2(G)$ different than $-x_u \leq 0$, then $\forall u \in V$ $0 \leq \alpha_u \leq a$.*

Proof. First, since $S = \{u\}$ is a co-2-plex, $\alpha_u = \alpha\chi_u \leq a$. Let us now prove that $\alpha \geq 0$. Suppose that α_u is negative, and let S be a co-2-plex containing u such that $\alpha\chi_S = a$ if it exists, $S' = S \setminus u$ is also a co-2-plex and $\alpha\chi_{S'} > a$ which constradicts the validity of $\alpha \leq a$. If S does not exist, then by setting $\alpha' = \alpha + \chi_u$ we obtain a new valid inequality $\alpha' x \leq a$ for $\mathcal{P}_G$ dominating $\alpha x \leq a$ if the latter is not a trivial inequality. □

The proof of Lemma 3.4.2 is very generic and applicable to every type of polytopes whose extreme points are incidence vectors of subset closed combinatorial objects: any subset of a co-2-plex is a co-2-plex, any subset of a stable set is a stable set, every subset of a matching is a matching.

Let us reduce the space of the parameters of a combination giving a redundant inequality.

Theorem 3.4.3. *If for some $u \in S$, $m_u > 0$, the inequality obtained by Fourier-Motzkin is redundant as obtained by another combination satisfying $m_u = 0$ $\forall u \in S$.*

Proof. If $m_u > n_u$, according to Equation 3.33, u has a negative coefficient, in which case the inequality obtained is not facet defining by Lemma 3.4.2 or is a nonnegativity inequality.

If $n_u > m_u$, we set $m'_u = 0$, $n'_u = n_u - m_u$, $k' = k + m_u$, $m'_v = m_v + m_u \forall v \in K \setminus \overline{N}_u$ and finally, $\mathcal{P}'_E = \mathcal{P}_E + m_u$ $\forall e \in E(K \setminus N_u)$, all other components of n', m', p' are equals to the corresponding component of n, m, p.

Let us show that n', m', k', p' verify Equations (3.34) and (3.35). The relation remains unchanged for $vw \in \delta(S) \setminus \delta_u$. For $uv \in \delta_u$:

$$m_u + m'_u + m'_v = m_u + m_v = p_{uv} + n_u = p'_{uv} + n'_u + m_u$$

For $vw \in E(K) \cap E(N_u) = E(N_u)$:

$$\begin{aligned} m'_v + m'_w = m_v + m_w = p_{vw} + k + \sum_{s \in S | v,w \in \overline{N}_s} n_s &= p'_{vw} + (k' - m_u) \\ &+ \sum_{s \in S | v,w \in \overline{N}_s,\ s \neq u} n'_s + m_u + n'_u \end{aligned}$$

For $vw \in E(K \setminus N_u)$:

$$\begin{aligned} (m'_v - m_u) + (m'_w - m_u) = m_v + m_w &= p_{vw} + k + \sum_{s \in S | v,w \in \overline{N}_s} n_s \\ &= (p'_{vw} - m_u) + (k' - m_u) + \sum_{s \in S | v,w \in \overline{N}_s} n'_s \end{aligned}$$

For $v \in K \setminus N_u, w \in N_u$:

$$(m'_v - m_u) + m'_w = m_v + m_w = p_{vw} + k + \sum_{s \in S | v,w \in \overline{N}_s} n_s$$
$$= p_{vw} + (k' - m_u) + \sum_{s \in S | v,w \in \overline{N}_s} n'_s$$

Now that n', m', k', p' induces a valid inequality for $\mathcal{P}_2(G)$, let us compare it to the one obtained by n, m, k, p; both right hand side are equals:

$$\sum_{v \in S} n_v + k = \sum_{v \in S \setminus u} n'_v + k' - m_u + n'_u + m_u = \sum_{v \in S} n'_v + k'$$

Similarly, every vertex has the same coefficient in the inequality associated with n', m', k', p'. $\square$

3.4.3 Analyzing PORTA files under the light of Fourier-Motzkin

PORTA [47] is a software used to manipulate H-representation and V-description of polyhedrons and compute one from the other. It is also able to enumerate the integer points inside polytopes. The algorithms used by PORTA are based on Fourier-Motzkin procedure, hence, they are exponential. In small instances, it remains possible to use this software, and it often helps to deduce families of valid inequalities and facets for a given family of polytopes. We used PORTA accordingly to obtain the H-representation of a split graph co-2-plex polytope into their natural variable set. I recently started using the Julia library 'JuliaPolyhedra' for that same purpose [48].

Let us consider a nontrivial inequality $\alpha x \leq a$ defining a facet of $\mathcal{P}^2_G$. By Theorem 3.4.3, we know that there exists a combination n, m, k, p of constraints of $\mathcal{P}^2_G$ satisfying $m_u = 0 \ \forall u \in S$ that yields $\alpha x \leq a$.

In the following, we show how to deduce the value of the other coefficients in the combination that yields $\alpha x \leq a$.

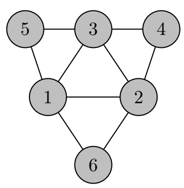
Since we restrict ourselves to coefficients satisfying $m_u = 0 \ \forall u \in S$, by Equation (3.33) $\alpha_u = n_u \ \forall u \in S$. This implies that $k = a - \sum_{u \in S} n_u$.

Finally, $m_v = \sum\limits_{u \in S,\ v \in N_u} n_u + k - \alpha_v \ \forall v \in K$ and $\mathcal{P}_{\mathcal{E}}$ can be deduced from relations (3.34) and (3.35).

Let us show an example with the sun see Figure 3.2 where a split graph and its co-2-plex polytope are given. The nonnegativity constraint $x_u \geq 0$ is obtained with $m_u = 1$, $p_{uv} = 1 \ \forall uv \in \delta_u$. The trivial inequality constraint $x_u \leq 1$ is obtained with:

- $n_u = 1$, $m_v = 1 \ \forall v \in N_u \ p_{vw} = 1 \ \forall vw \in E(N_u)$ if $u \in S$
- $k = 1$, $m_v = 1 \ \forall v \in K \setminus u$, $p_{vw} = 1 \ \forall vw \in E(K \setminus v)$ otherwise

The inequalities of right-hand size 2 are maximal 2-plex inequalities, and each 2-plex of G is a clique minus one edge since G is split. They are obtained when $\sum_{u \in S} n_u + k = 2$ and $m_v = 1$ for every vertex v in the intersection of the corresponding two cliques and p set accordingly. The same procedure applies to the other inequalities.



$$\left\{\begin{array}{rcll} x_i & \geq & 0 & \forall i \in \{1,\dots,6\} \\ x_6 & \leq & 1 & \\ x_5 & \leq & 1 & \\ x_4 & \leq & 1 & \\ x_3 & \leq & 1 & \\ x_2 & \leq & 1 & \\ x_1 & \leq & 1 & \\ x_3 + x_4 + x_5 & \leq & 2 & \\ x_2 + x_4 + x_6 & \leq & 2 & \\ x_1 + x_5 + x_6 & \leq & 2 & \\ x_1 + x_2 + x_3 + x_6 & \leq & 2 & \\ x_1 + x_2 + x_3 + x_5 & \leq & 2 & \\ x_1 + x_2 + x_3 + x_4 & \leq & 2 & \\ x_1 + x_2 + 2x_3 + x_4 + x_6 & \leq & 3 & \\ x_1 + 2x_2 + x_3 + x_5 + x_6 & \leq & 3 & \\ 2x_1 + x_2 + x_3 + x_4 + x_5 & \leq & 3 & \\ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 & \leq & 3 & \\ 2x_1 + 2x_2 + 2x_3 + x_4 + x_5 + x_6 & \leq & 4 & \end{array}\right.$$

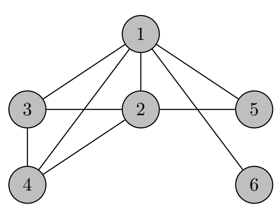
Figure 3.2: The sun and its co-2-plex polytope

To give a finite family (which is not the case in Equation 3.33 since these linear constraints correspond to a cone), we would like to understand more precisely the combinatorial structure behind the coefficients and know which combination of n, m, k, p gives a facet defining inequality (the above technique permit to do the converse that is, given a facet, deduce the associated combination).

With our procedure, we see that fixing n, α and a, determines the coefficients $m,\ k,\ p$. Now that we seek a combination inducing a facet defining inequality without knowing the priors α and a, we wonder if fixing n suffices to fix the other coefficients, this would permit us to focus on the structure of n. Figure 3.2 answers by the negative, for all the facets with right-hand side 3 or 4, there exists a combination verifying $n_4 = n_5 = n_6 = 1$. But, for any such combination, $k = 0$ when the right-hand side of the associated inequality equals 3, and $k = 1$ when the right-hand side equals 4, proving that k can be free even when n is fixed.

A second question is: when n and k are fixed, is it possible to deduce a unique m and p. Figure 3.2 also answers by the negative. For every facet with right-hand side 3, a valid combination exists satisfying $n_4 = n_5 = n_6 = 1$ and $k = 0$, but the coefficients m and p are still free, leading to 3 different facet defining inequalities.

Still, given n and k, the number of possible p and m satisfying Equations (3.34) and (3.35) and such that the obtained inequality only has positive coefficients is bounded. In Figure 3.2, every clique is taken at most once, implying that $\sum_{u \in S} n_u + k$ is bounded by $|S| + 1$. But in general, it is not true: there exist graphs for which some facet defining inequalities are obtained by taking several times the same clique, see Figure 3.3 where the facet defining inequality is obtained by setting $n_6 = 2$. This graph is partitioned into the clique $\{1, 2, 3, 4\}$ and the stable set $\{5, 6\}$. We managed to build graphs for which the right-hand side of a facet inequality becomes even greater than the number of vertices of the graph, and what is even more surprising is that they are obtained from the graph of Figure 3.3 by adding false twins to vertex 6.



$$3x_1 + 2x_2 + x_3 + x_4 + x_5 + 2x_6 \leq 5$$

Figure 3.3: A graph and a facet defining inequality

From the Fourier-Motzkin elimination procedure, we deduce a bound U on the right-hand side of a facet defining inequality for $\mathcal{P}_G$ on chordal graphs, which gives an upper bound on $\sum_{u \in S} n_u + k$ for the particular case of split graphs and hence, by enumerating the multisets of cliques of size at most U and enumerating the candidates m and p satisfying relations 3.34 and 3.35, we obtain a finite family of inequalities. This bound is proven by induction and is not satisfying since it does not give information on the combinatorial structure of the facets, but it is generic and applicable to any extended formulation; the only changing point of the proof would be the initialization of the induction.

Theorem 3.4.4. *When $G = (V, E)$ is a chordal graph, the right-hand side of a facet defining inequality for $\mathcal{P}_G$ is bounded by $2^{(2^{|E|-1}-1)}$.*

Proof. We prove such bound by induction. First, let us exhibit a sequence bounding the right-hand side of a facet defining inequality at each iteration of the Fourier-Motzkin elimination procedure, *i.e.* at each removal of a single variable associated with an edge of E of formulation $\mathcal{P}_{\mathcal{K}}$.

Let us consider $E = \{e_1, \ldots, e_{|E|}\}$ and $\mathcal{P}^0, \ldots, \mathcal{P}^\ell, \ldots, \mathcal{P}^{|E|}$ the sequence of formulations obtained by the Fourier-Motzkin elimination procedure to remove the y variables from $\mathcal{P}_{\mathcal{K}} = \mathcal{P}^0$, where $\mathcal{P}^{i+1}$ is obtained from $\mathcal{P}^i$ by projecting out the variable y_{i+1} and finally, $\mathcal{P}^{|E|} = \mathcal{P}_G$ the co-2-plex polytope of G on its natural variable set.

Let us denote by U_ℓ an upper bound on the right-hand side of a facet defining inequality of $\mathcal{P}^\ell$. First, by definition of $\mathcal{P}$ we can set $U_0 = 1$. Also, the projection of the very first variable, that is y_{e_1}, results in inequalities with a right-hand side 1 or 0: this happens because the set of constraints having a positive coefficient for the variable y_{e_1} in $\mathcal{P}^0$ have a coefficient one or zero for each y variable and a right-hand side equal to 0. Also, the set of constraints having a negative coefficient for the variable y_{e_1} has a coefficient -1 or 0 for y variables and a right-hand side equal to 0 or 1. By combining two of these inequalities such that the y_{e_1} disappear, one can only obtain an inequality with a right-hand side equal to 1, and where the absolute value of the coefficients associated with y is lower or equal to 1. Hence, $U_1 = 1$. Let β_ℓ be the largest absolute value of the coefficient of a y variable in formulation $\mathcal{P}^\ell$. The right-hand side of a facet defining inequality of $\mathcal{P}^{\ell+1}$ is bounded by $2U_\ell * \beta_\ell$ (it corresponds to the combination of two inequalities maximizing the right-hand side). Also, one can see that $\beta_\ell \leq U_\ell \ \forall \ell \in \{0, \ldots, |E|\}$ (the fact that it is true at rank 0 propagates all over the recursion). Hence, by setting $U_{\ell+1} = 2 * (U_\ell)^2$, we obtain that $U_{|E|}$ bounds the right-hand side of a facet defining inequality of $\mathcal{P}_G$ for chordal graphs.

Now, we compute $U_{|E|}$: let $W_\ell = log_2(U_\ell)$, we have $W_{\ell+1} = log_2(U_{\ell+1}) = log_2(2 * (U_\ell)^2) = log_2(2) + 2log_2(U_\ell) = 1 + 2W_\ell$. And finally $Z_\ell = 1 + W_\ell$ hence, $Z_{\ell+1} = 1 + W_{\ell+1} = 1 + 1 + 2W_\ell = 2(1 + W_\ell) = 2Z_\ell$. Also, $W_1 = log_2(U_1) = 0$ hence, $Z_1 = 1$. This implies that $Z_\ell = 2^{\ell-1}$, $W_\ell = 2^{\ell-1} - 1$ and $U_\ell = 2^{2^{\ell-1}-1}$ which ends our proof. □

3.4.4 Projection cone

Another possibility to understand the structure of the combinations inducing facet-defining inequalities is to study the structure of the projection cone that is:

$$
\begin{array}{cc}
\begin{array}{cccc} E & V & k & S \end{array} & \\
\begin{pmatrix} ? & I_{|V|} & 0 & 0 \\ ? & ? & 1 & 0 \\ ? & ? & 0 & I_{|S|} \end{pmatrix} & \begin{array}{c} \chi_V \\ \chi_k \\ \chi_S \end{array}
\end{array}
$$

Figure 3.4: The matrix built in the proof of Theorem 3.4.5

$$
\mathcal{C}_G = \left\{ (p,n,m,k) \in \mathbb{R}^{|E|} \times \mathbb{R}^{|V|} \times \mathbb{R}^{|V|} \times \mathbb{R} \;\middle|\; \begin{array}{rcll} p_{uv} + n_u & = & m_u + m_v & \forall u \in S,\ uv \in \delta_u \\ p_{uv} + k + \sum\limits_{w \in S | u,v \in \overline{N}_w} n_w & = & m_u + m_v & \forall uv \in E(K) \\ m_u & \geq & 0 & u \in V \\ p_e & \geq & 0 & e \in E \\ n_u & \geq & 0 & u \in S \\ k & \geq & 0 & \end{array} \right\}
$$

Theorem 3.4.5. *$\mathcal{C}_G$ has dimension $|S| + |V| + 1$.*

Proof. First, since $\mathcal{C}_G$ is defined using $t = |S| + |V| + 1 + |E|$ variables, its dimension d is at most t. Let us consider the constraint submatrix associated with edges of G. Every $\mathcal{P}_E$ $e \in E$ belongs to exactly one such constraint, hence, this submatrix contains an identity of size $|E|$, which proves that its rank is exactly $|E|$. This implies that $t - |E| \geq d$. Let us build enough affinely independent incidence points of $\mathcal{C}_G$. For each $u \in V$ let us consider the point having $m_u = 1$ and $\mathcal{P}_E = 1$ $\forall e \in \delta_u$, let us denote the matrix induced by all such points by χ_V. We denote by χ_k the point $k = 1$, $m_u = 1$ $\forall u \in K$, $\mathcal{P}_E = 1 \forall e \in E(K)$. For each vertex $u \in S$, let us consider the point satisfying $n_u = 1$, $m_v = 1$ $\forall v \in N_u$ and $\mathcal{P}_E = 1$ $\forall e \in \bigcup\limits_{v \in N_u} \delta_v \setminus \delta_u$, let us denote by χ_S the submatrix induced by all such points. Now, one can check that $\begin{pmatrix} \chi_V \\ \chi_k \\ \chi_S \end{pmatrix}$ has rank $|S| + |V| + 1$, see Figure 3.4 where a diagonal block of size $|V| + |S| + 1$ is exhibited, which ends the proof (as 0 belongs to the cone). □

Every point of $\mathcal{C}_G$ induces an inequality weakly dominated by the inequalities induced by the generators of $\mathcal{C}_G$. Hence, it is sufficient to consider the set of generators of the projection cone. By Theorem 3.4.3, we can restrict the research to the generators satisfying $m_u = 0$ $\forall u \in S$, hence we give the dimension of the face induced by such constraints.

Theorem 3.4.6. *$\mathcal{C}_G \cap \{m_u = 0 | u \in S\}$ has dimension $|V| + 1$.*

Proof. Similarly to the proof of Theorem 3.4.5, we first give an upper bound on its dimension. The equality submatrix has a size $|E|+|S|$, and its rank is at least $|E|+|S|$ since $\mathcal{P}_E$ $e \in E$ appears precisely one time in each edge constraint, which gives a diagonal, and $m_u = 0 \forall u \in S$ gives an identity. Now, by restricting the set of points exhibited in the proof of Theorem 3.4.5 to the lines of χ_V associated with vertices of n_u $u \in K$, χ_k, and χ_S we obtain $|V \setminus S|+|S|+1$ linearly independent integer points. □

Let us write down the polytope obtained by removing the null variables m_u $u \in S$:

$$\mathcal{C}_G^{\cap} = \left\{ (p,n,m,k) \in \mathbb{R}^{|E|} \times \mathbb{R}^{|V|} \times \mathbb{R}^{|K|} \times \mathbb{R} \;\middle|\; \begin{array}{rcll} p_{uv} + n_u & = & m_v & \forall u \in S,\ uv \in \delta_u \\ p_{uv} + k + \sum\limits_{w \in S | u,v \in \overline{N}_w} n_w & = & m_u + m_v & \forall uv \in E(K) \\ m_u & \geq & 0 & u \in K \\ p_e & \geq & 0 & e \in E \\ n_u & \geq & 0 & u \in S \\ k & \geq & 0 & \end{array} \right\}$$

It is still a cone, so we are still interested in its generators. Its generators satisfy at least $|V|$ linearly independent constraints with equality among the nonnegativity ones. This means that by enumerating the subsets of variables of size $|V|$ among the variables $\mathcal{X} = \{m_u | u \in K\} \cup \{n_u | u \in S\} \cup \{\mathcal{P}_E | e \in E\} \cup \{k\}$, one can enumerate the generators of $\mathcal{C}_G^{\cap}$. Let $\mathcal{X}'$ be such a subset, any non-zero integer solution of $\mathcal{C}_G^{\cap} \cap \{x = 0 | x \in \mathcal{X}'\}$ induces a valid inequality. The set of obtained inequalities defines the co-2-plex polytope of G. Note that given a rational non-zero solution, one can easily obtain an integer non-zero solution by multiplying it with a sufficiently big coefficient. A satisfying projection would summarize such inequalities by enumerating some graph structure(s) and their associated coefficient.

3.5 Experimental results for the maximum cardinality co-2-plex problem

As shown in Section 3.1, the extended clique formulation $\mathcal{UK}$ seems pretty strong. Its subjacent polytope $\mathcal{P}_{\mathcal{UK}}$ is integer when the graph is contraction perfect, and its projection on the natural variable set is contained in the subjacent polytope of the formulation of Balasundaram [7], even when 2-plex inequalities are added to the latter. In this section, we investigate from an experimental point of view the performances of our formulation by considering different variants. Gao et.al show in [49] that given a lower bound b for the size of the largest co-k-plex of a graph $G=(V,E)$, if a vertex has a neighborhood of size greater or equal to $|V|-b+k$, it does not belong to the largest co-k-plex of G. After noticing that, they proposed a preprocessing recursively removing the vertices verifying this property from the graph. Note that a co-k-plex of G is a

k-plex of $\overline{G}$, and the diameter of a k-plex is k. This remark led Gao et.al [49] to decompose an instance G in $|V|$ smaller instances: instead of computing the maximal size co-k-plex problem on the graph obtained after the preprocessing, they compute for each vertex $v \in V$ the maximal co-k-plex on the graph G_v obtained by restricting G to the vertices at distance at most k from v in $\overline{G}$. After that, they preprocess each obtained instance using the same method, and finally, they launch their exact algorithm on this set of smaller instances.

This section uses this preprocessing to solve the maximum co-2-plex problem.

3.5.1 The different algorithms

The 3 formulations/5 algorithms

In this section, 'N' will refer to the branch-and-cut algorithm based on Formulation $\mathcal{N}$ that dynamically adds the 2-plex inequalities (3.24) using the following heuristic greedy separation algorithm: it consists of ordering the vertices by decreasing LP value and then adding the vertex of higher cost at each iteration before removing the vertices adjacent to this latter from the list of potential candidates to first build a maximal clique and then add vertices to make it an inclusion wise maximal 2-plex.

We compare this baseline algorithm with the stable set formulation $\mathcal{S}$ based on the utter graph. In the algorithm 'S', only the clique inequalities associated with the cliques $\{u, v\} \cup \delta_u \cup \delta_v$ for all edge $uv \in E$ are considered. The formulation is weaker but valid, that is, every integer point satisfying these inequalities is a stable set of $u(G)$. The second algorithm 'SK' is a branch-and-cut solving the stable set formulation $\mathcal{S}$: it starts by only considering edge constraints (for each edge of the utter graph, at most, one of both sides belongs to the solution) and the clique inequalities are dynamically added using a greedy separation. It orders the vertices by decreasing LP value before adding them to the current clique while it is still possible.

We also compare these algorithms with the results obtained by solving $\mathcal{E}$ that corresponds to Algorithm 'P'. The algorithm 'PK' is a branch-and-cut that solves $\mathcal{E}$ with a dynamic separation of utter clique inequalities (3.17). These inequalities are separated as follows: the optimal fractional point (x^*, y^*) is first mapped into the corresponding point (z^*, y^*) of $\mathcal{P}_S$, using the formula $x_u - y(\delta_u) = z_u$, the greedy clique separation procedure is run and if a violated clique inequality is found, it is mapped into the associated clique inequality (3.17) and added to PK.

Correspondences between the branching scheme of these formulations

First, we did not implement any branching rule, we let CPLEX decide how to branch. We discuss here the classic branching rule used to solve any compact MILP that is setting on one child node a variable to 1 and on the other to 0.

Branching on x variables of 'P' corresponds to branching on x variables of formulation 'N', branching on variable y_{uv} of 'P' corresponds in formulation N to adding $x_u = x_v = 1$ on one child node and $x_u + x_v \leq 1$ on the other.

Branching on z_u variable of formulation 'S' corresponds in formulation 'P' to the addition of constraints $x_u = 1$ and $y(\delta_u) = 0$ on one node, and $x_u = y(\delta_u)$ on the other node. Branching on y variables in 'P' or 'S' is equivalent.

To have a balanced branching tree, branching on x variables of formulation N and P seems to give the most balanced branching tree since its semantic is: "on one child node, u belongs to the solution. On the other, it does not" when branching on z variables of formulation 'S' has the following semantic: " on one child node u and none of its neighbors belong to the solution and on the other, u is in the solution only if one of its neighbors belongs to it."

3.5.2 Experimental results

Instances

We compare our algorithm on several sets of graphs:

- 74 instances from Color02 [50]
- 120 connected random graphs with 70, 80, 85, and 90 vertices and a probability 0.5,0.7, 0.9 for each edge (10 graphs for each combination of such parameters)
- The 8 smallest instances (in terms of vertices) from the 10th DIMACS challenge and their complement

Performance profile

"Performance profile" is a statistic that permits the comparison of different algorithms on sets of instances for which they do not always converge. We describe how it is calculated in this section. The idea came from [51]. First, let us denote by r^i_A the time needed for the algorithm A to solve the instance I. Let $\mathcal{A}$ be the set of algorithms we want to compare. We also consider $\overline{r}^I_A = \frac{r^I_A}{\min_{A' \in \mathcal{A}} r^I_{A'}}$ if A manages to solve I before the timeout and ∞ otherwise . Given a set of instances $\mathcal{I}$ and a parameter $t \in [1, \dots, K]$ for some K we consider $p(t) = \frac{|\{I \in \mathcal{I} \mid \overline{r}^I_A \leq t\}|}{|\mathcal{I}|}$ which represent the proportion of instances solved with a factor of t from the fastest algorithm. The performance profile is then the graphic obtained by computing $p(t)$ for some $t \in \mathbb{R}$.

By this definition, the highest curve on such a graphic corresponds to the 'best' algorithm.

Tables and Figures

First, note that some algorithms do not solve some instances. For this reason, we consider the performance profile on the whole sets of instances, see Figures 3.5, 3.6, 3.7, 3.8, the ratio between the CPU time consumed by our algorithms (S, SK, P, PK) divided by the CPU time consumed by N to solve the set of instances that each algorithm manages to solve in less than an hour,

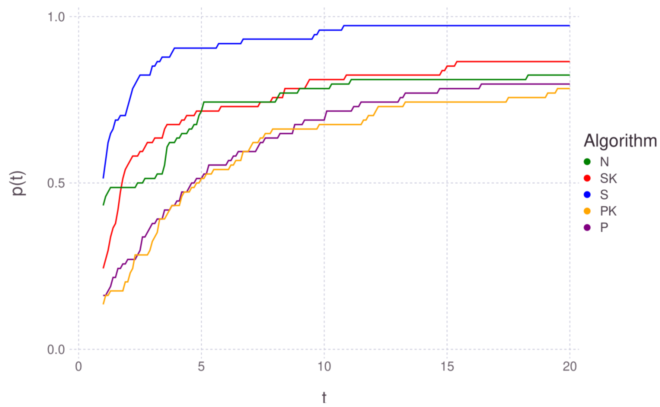


Figure 3.5: Performance profile on Color02

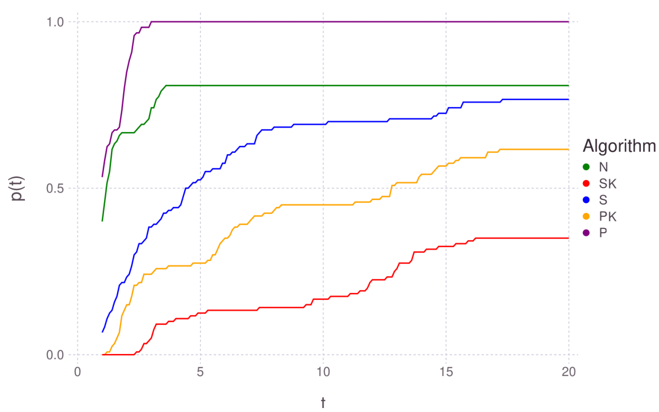


Figure 3.6: Performance profile on Random instances

we refer to this sets as 'restricted' and finally the number of instances solved, these latter statistics are given in Table 3.2.

In Table 3.1, we give the CPU time consumed by N and its number of branching nodes, one can see that DIMACS and comp-DIMACS both contain easy instances.

By looking at the performance profiles on Figures 3.5, 3.6, 3.7, 3.8, it is clear that S is better than SK and P seems to be overall better than PK. Unless for DIMACS, where N solves every instance faster than our algorithm, S and P are the best algorithms depending on the instance set.

In Table 3.2, the data that pops out is the number of branching nodes, which is much lower when considering an extended variable set. This corresponds to the theoretical comparison made in Section 3.1. On the sets, DIMACS and comp-DIMACS, PK generates much more inequalities than SK. P manages to solve two sets of instances completely when all the other algorithms solve at

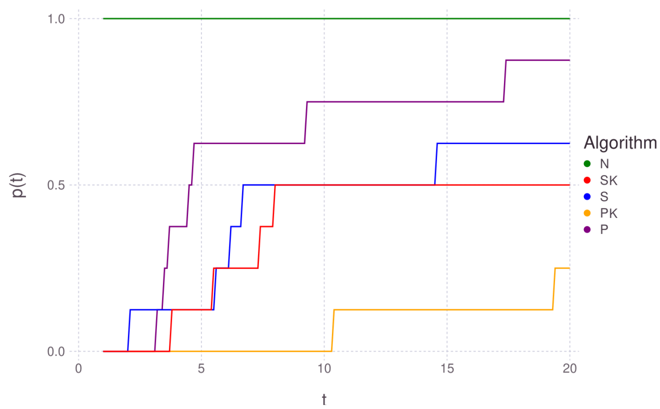


Figure 3.7: Performance profile on DIMACS

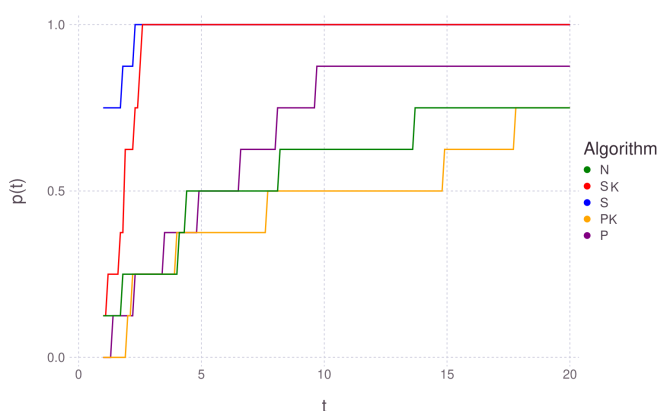


Figure 3.8: Performance profile on DIMACS-comp

Instances	CPU	# nodes
restrict Color02	8204	11 204 503
restrict Random	13 858	39 476 293
restrict DIMACS	33	18 897
restrict comp-DIMACS	127	239 401

Table 3.1: Flat results of N

most one set. P also performs quite well in terms of CPU consumed, in particular, when all the other extended formulations perform quite badly on DIMACS, P manages to solve it reasonably slower than N. Still, the algorithm that solves Color02 the fastest is S. This seems to show that the heuristic procedure computing a maximum clique does not improve the algorithms.

		Restricted instances			All instances
		ratio		flat	
Algorithm	Instances	CPU	# nodes	# cuts	% solved
N	Color02	1.0	1.0	701186	71.62
	Random	1.0	1.0	1926611	80
	DIMACS	1.0	1.0	25803	100
	comp-DIMACS	1.0	1.0	6347	75
SK	Color02	0.99	$< 10^{-2}$	79830	81.08
	Random	4.32	0.03	877485	35
	DIMACS	55.01	$< 10^{-2}$	35	87.5
	comp-DIMACS	0.4	$< 10^{-2}$	98	100
S	Color02	0.39	$< 10^{-2}$	-	85.13
	Random	3.33	0.03	-	76.6
	DIMACS	24.31	2.6	-	87.5
	comp-DIMACS	0.17	$< 10^{-2}$	-	100
PK	Color02	1.32	0.01	62764	77.02
	Random	3.76	0.06	1372674	61.6
	DIMACS	51.63	0.79	24540	87.5
	comp-DIMACS	0.87	0.01	2695	87.5
P	Color02	0.7	0.2	-	74.32
	Random	0.71	0.07	-	100
	DIMACS	5.7	0.01	-	87.5
	comp-DIMACS	0.44	$< 10^{-2}$	-	100

Table 3.2: Experimental results of the 5 algorithms solving the maximum co-2-plex problem

Conclusion

In this chapter, we proposed new extended formulations for the maximum weighted co-2-plex, and studied their associated polyhedra. From a theoretical point of view, we proved that our formulations provide tighter linear relaxations than the existing formulations. We also study when the polytopes are integer. As a consequence, we provided an extended formulation for the co-2-plex polytope for contraction perfect graphs which is compact if the graph is chordal. We also give a description of the co-2-plex polytope on the natural variable space when the graph is a tree. From a computational point of view, we compared several implementations of our formulations and showed that our algorithms perform quite better than the state of the art on such a problem.

Part II

k-Defective coloring: cutting primal and dual spaces in column generation

Chapter 4

Literature review on defective colorings and covering polytopes

In this second part, we will study how to solve the k-defective problem through integer linear programming. This first chapter is dedicated to the state-of-the-art on defective colorings. In the first part of this thesis, we motivated the study of 2-plexes (co-2-plexes by complementing the graph) from the fact that cliques can be too restrictive in defining a strongly connected subgraph, as it can correspond to communities in social networks or genetic similarities.

4.1 Defective colorings

Defective colorings were introduced in 1986 [52] to study variants of problems related to the chromatic number of graphs. From a computational point of view, they are known to be challenging as the k-defective coloring is NP-hard even on split graphs for $k \geq 2$ as proven among other complexity results in [53].

4.1.1 Defective coloring number of a graph class

A variant of defective colorings is the *the defective coloring number* of a graph class $\mathcal{F}$, denoted by $\chi_\Delta(\mathcal{F})$, as the infimum $k \in \mathbb{Z}^+$ such that there exists an integer $d \in \mathbb{Z}_+$ for which every graph in $\mathcal{F}$ is d-colorable with co-k-plexes. This number may not exist, in particular, $\omega(\mathcal{F}) = sup\{\omega(G)|G \in \mathcal{F}\}$ must be bounded. Finding $\chi_\Delta(\mathcal{F})$ has been introduced to study a variant of the following conjecture by Hadwiger:

Conjecture 4.1.1 ([54])**.** *For every $t \in \mathbb{Z}^+$, if $\mathcal{F}$ is the set of graphs with no clique of size $t+1$ as minor, then $\chi(\mathcal{F}) \leq t$.*

The case $t = 1$ corresponds to coloring isolated vertices, which is trivially 1-colorable. The case $t = 2$ corresponds to coloring trees: since they are bipartite, they are also 2-colorable. The case $t = 3$ corresponds to coloring series-parallel graphs for which a simple induction proves the result. The case $t = 4$ implies the 4-color theorem. At first sight, it seems counter-intuitive with the fact that planar graphs are graphs with no cliques of size 5 minors or biclique of size 3 minors. Indeed, the biclique of size 3 is 2-colorable which suggests that having no minor clique of size 5 could be sufficient for a graph to be 4-colorable. The case $t = 5$ has been proven in [55].

The following variant of the Hadwiger conjecture has been proven in 2015:

Theorem 4.1.2 ([56])**.** *For every $t \in \mathbb{Z}^+$, if $\mathcal{F}$ is the set of graphs with no clique of size $t+1$ as minor, then $\chi_\Delta(\mathcal{F}) \leq t$.*

4.1.2 An exact approach for the k-defective coloring

Furini et al. [2] propose a generic column generation framework to solve the partitioning/covering of a graph with relaxed cliques. In particular, they propose a framework to solve the k-defective coloring by generalizing the formulation of Mehrotra and Trick [3] for the vertex coloring problem (corresponding to the case $k = 1$).

The primal and pricing formulations of Furini et al.

Let $\mathcal{S}_G^k$ be the set of inclusion-wise maximal co-k-plexes of $G = (V, E)$, $\mathcal{S}_v^k \subseteq \mathcal{S}_G^k$ be the set of maximal inclusion wise co-k-plexes of G containing v, and $x_S,\ S \in \mathcal{S}_G^k$, be binary variables associated with the inclusion-wise maximal co-k-plexes of G. The extended formulation of Furini et al. in [2] for the k-defective problem is the following (P_k):

$$\min \sum_{S \in \mathcal{S}_G^k} x_S \tag{4.1}$$

$$\sum_{S \in \mathcal{S}_v^k \mid v \in S} x_S \geq 1 \qquad \forall\ v \in V \tag{4.2}$$

$$x_S \geq 0 \qquad \forall\ S \in \mathcal{S}_G^k \tag{4.3}$$

$$x_S \in \mathbb{Z} \qquad \forall\ S \in \mathcal{S}_G^k \tag{4.4}$$

Given $\mathcal{S}^*$ a subset of $\mathcal{S}_G^k$ covering the vertices of G, the restricted master problem $RMP_{\mathcal{S}^*}$ is the formulation (P_k) restricted to the subset of variables associated with $\mathcal{S}^*$. Let $\lambda_v,\ v \in V$, the optimal dual cost associated with the covering inequalities of $RMP_{\mathcal{S}^*}$. The *pricing subproblem* of the column generation process is then to find a co-k-plex S whose associated primal variable has a negative *reduced cost* $c_S(\lambda) = 1 - \sum_{v \in S} \lambda_v$, that is, to find a co-$k$-plex

S maximizing $\sum_{v \in S} \lambda_v$. If $c_S(\lambda)$ is nonnegative for all $S \in \mathcal{S}$, $RMP_{\mathcal{S}^*}$ gives an optimal solution of the linear relaxation (P), otherwise we iterate by solving $RMP_{\mathcal{S}^* \cup S}$ for some S satisfying $c_S(\lambda) < 0$.

The exact pricing subproblems are solved by (SP) the following ILP formulation [7]:

$$\begin{aligned} \max \sum_{v \in V} \lambda_v \pi_v & & (4.5)\\ (|N_u| + 1 - k)\pi_u + \pi(N_u) \leq |N_u| & \qquad \forall\, u \in V \\ \pi_v \in \{0,1\} & \qquad \forall\, v \in V \end{aligned}$$

Branching rules

Furini et al. [2] compare several branching rules and conclude that the best is obtained by analogy with the Ryan and Foster rule proposed by Mehrotra et al. [3] that is: branching on pairs of vertices (u, v) such that in one child node, u and v always belong to the same co-k-plex. In the other child node, they never belong to the same co-k-plex.

Let $\mathcal{S}^*$ (resp. x^*) be the subset of co-k-plexes (resp. the primal solution) obtained at the end of the column generation process. Let (u, v), a couple of vertices minimizing the following quantity:

$$\left|0.5 - \sum_{S \in \mathcal{S}^*:\ u \in S, v \in S} x^*_S\right|$$

This quantity is a way to evaluate if solution x^* contains some information on whether u and v belong to the same co-k-plex in an optimal solution. Indeed, if it is close to 0, it lets us think that no information can be deduced from x^*; and if it is close to 0.5, it lets us think that x^* give the information whether u and v belong to the same co-k-plex in an optimal covering. Now, the pricing procedure has to be modified. The exact procedure is based on an MILP (SP) with the additional inequalities $x_u = x_v$ in one branching node, and $x_u + x_v \leq 1$ in the other.

In the next section, we describe how the case $k = 1$ naturally yields a *robust branching rule* i.e., adding these branching decisions to the master program preserves the structure of the pricing subproblems. When $k \geq 2$, these branching decisions are nonrobust. Indeed, the pricing subproblem is no longer a maximum co-k-plex problem. For example, suppose that at the current node, u, v always belong to the same co-k-plex and v, w never belong to the same co-k-plex, the pricing subproblem is then to find the maximum co-k-plex either containing u, v but not w, or containing w but not u, v or containing none of these three vertices. This latter problem probably shares a little bit of structure with the maximum co-k-plex problem on G, but the more one added branching decisions, the more you break the similarities between these problems disappear. In general, a robust branching decision manages to keep the similarities by considering the same problem definition but on a slightly modified instance instead of keeping the instance and changing the definition of the problem slightly.

4.2 Integer linear programming formulation for the coloring problem

Extensive work has been done to solve the coloring problem on any graph. One of the best methods has been proposed by Anuj Mehrotra and Michael A. Trick [3] and corresponds to finding the best integer point of formulation $\mathcal{D}$ presented in Chapter 1. It also corresponds to formulation P_1 by noticing that co-1-plexes are stable sets. This section considers that $\mathcal{S} = \mathcal{S}^1$.

4.2.1 Covering formulation for the coloring problem

Note that the graph coloring problem corresponds to the 1-defective coloring problem, as co-1-plexes are stable sets. In this case $\mathcal{S}_G$ is the set of inclusion wise maximal stable sets. The resulting stable set formulation (P_1) is based on the observation that any color class in a coloring is a stable set and is as follows:

$$\min \sum_{S \in \mathcal{S}_G} x_S \tag{4.6}$$

$$\sum_{S \in \mathcal{S}_v} x_S \geq 1 \qquad \forall\, v \in V \tag{4.7}$$

$$x_S \in \{0,1\} \qquad \forall\, S \in \mathcal{S}_G \tag{4.8}$$

Recall that with this objective function, covering is equivalent to partitioning as subsets of stable sets are stable: any vertex covered more than one time could be assigned arbitrarily to one of the corresponding colors. The dual of the linear relaxation of P is an LP whose integer points correspond to cliques of the graph. This remark is particularly interesting because of the work of Jan Mycielski [57], who gives an operation on a graph that does not change the size of its largest clique but increases by 1 its coloring number.

As (P_1) has exponentially many variables in the size of G, it is solved using column generation: the idea is to solve the linear relaxation of (P) to get a lower bound for the optimal solution. It is done by considering a subset of maximal stable sets $\mathcal{S}^*$ that is non-optimal for the coloring problem and then, by solving the following linear program called *Restricted Master Problem* and denoted by $(RMP_{\mathcal{S}^*})$:

$$\min \sum_{S \in \mathcal{S}^*} x_S \tag{4.9}$$

$$\sum_{S \in \mathcal{S}^* \cap \mathcal{S}_v} x_S \geq 1 \qquad \forall\, v \in V \tag{4.10}$$

$$x_S \geq 0 \qquad \forall\, S \in \mathcal{S}^* \tag{4.11}$$

Then, given the dual values λ_v $v \in V$ associated with the covering constraints, the reduced cost of a variable associated with the stable set S is as follows:

$$c_S(\lambda) = 1 - \sum_{v \in S} \lambda_v.$$

Finding a minimum reduced cost variable corresponds to finding a maximum weighted stable set S with weight λ_v, we call it the *Pricing Subproblem.*

After that, if a negative reduced cost column associated with the stable set S is found, S is added to $\mathcal{S}^*$, and we iterate. When no negative reduced cost variables exist for a given $\mathcal{S}^*$, the optimal solution x^* of $RMP_{\mathcal{S}^*}$ gives the linear relaxation value of (P) but does not necessarily give its optimal solution since x^* can be fractional. To deal with this problem, the Ryan and Foster's rule leads to computing for each couple of vertices $\{u, v\}$ the quantity:

$$\left| 0.5 - \sum_{S \in \mathcal{S}^* \cap \mathcal{S}_u \cap \mathcal{S}_v} x^*_S \right|$$

This quantity is a way to evaluate if solution x^* contains some information on whether u and v belong to the same co-k-plex in an optimal solution. Indeed, being close to 0, let us think that no information can be deduced from x^*; and being close to 0.5, let us think that x^* gives the information whether u and v belong to the same co-k-plex in the optimal covering. This quantity always equals 0.5 for adjacent vertices since no stable set contains both.

Given the pair $\{u, v\}$ minimizing this quantity, we create two branching nodes, the first where u and v will always belong to the same stable set and another where they never belong to the same co-k-plex. If such a rule is applied to any couple of vertices, we can deduce the associated integer solution, which gives a complete branching rule.

In practice, the left-hand branching node corresponds to solving the coloring problem on the graph obtained by merging u and v while the right-hand branching node corresponds to solving the coloring problem on the graph obtained by adding an edge between u and v. Hence, the leaves of this branching tree correspond to coloring cliques, for which it is trivial to compute an optimal coloring even without linear programming. Note that, in this case, the constraint matrix is the identity and hence yields an integral polyhedron. This type of branching, where the structure of the problem is similar in the branching nodes (every branching node corresponds to a coloring problem), is called *robust.* Robust branchings are generally preferable since they imply that the pricing subproblems in each branching node have the same structure. In our example, the pricing subproblem of each branching node is a maximum weighted stable set, this permits using all the knowledge on computing maximum stable sets at each branching node.

Extensive work has been done on this ILP formulation. In 2010, Stefano Gualandi and Frederico Malucelli [58] proposed a constraint satisfaction scheme for the pricing subproblems and solved 22 instances still open from the reference benchmark [59]. Enrico Malaguti, Michele Monaci, and Paolo Toth [60], using heuristics for the pricing subproblems, solved 5 additional instances in 2012. The formulation of Mehrotra et al. suffers from the lack of precision of some linear programming solvers. In 2011, Stephan Held, William Cook, and Edward C. Sewell proposed a new technique assuring the validity of the lower bound given by the formulation independent of the solvers's precision. They gave a method to parallelize the resolution [61]. In 2014, David Morrison, Jason Sauppe, Edward Sewell, and Sheldon Jacobson [62] presented a new branching rule, making them able to solve 7 instances faster than Malaguti, Monaci, and Toth [60]. Pierre Hansen, Martin Labbé, and David Schindl [63] tried to reinforce this formulation by adding cuts associated with odd holes and antiholes of the graph. These inequalities indicate that at least 3 stable sets are needed to cover an odd hole. For an antihole of length $2k+1$, at least $k+1$ stable sets are needed. These inequalities are *robust* [64], that is, they do not change the pricing subproblems structure; it is still a maximum weighted stable set in a slightly modified graph. Unfortunately, experiments of [63] showed that, in general, they do not siginificantly improve the relaxation value.

4.3 Mycielski's graphs: a worst case scenario

Jan Mycielski [57] gave an operation on a graph $G = (V, E)$ (connected and with at least two vertices), resulting in a new graph $\widetilde{G}$ for which $\omega(G) = \omega(\widetilde{G})$ but $\chi(G) + 1 = \chi(\widetilde{G})$. Let us consider $V = \{v_1, \dots, v_{|V|}\}$. This operation consists of adding for each v_i $i \in \{1, \dots, |V|\}$ a vertex w_i adjacent to every vertex of N_i, and adding a vertex u adjacent to all w_{v_i}.

Theorem 4.3.1 ([57]). *For any graph G, $\chi(G) = \chi(\widetilde{G}) + 1$ and $\omega(G) = \omega(\widetilde{G})$.*

Here is a sketch of the proof: any clique of $\widetilde{G}$ contains at most one vertex from $W = \{w_1, \dots, w_{|V|}\}$, and any clique K containing w_i has the same size of the clique of G, $K \cup v_i \setminus w_i$. Moreover, any clique containing u has size at most 2. The chromatic number of $\widetilde{G}$ is at least $\chi(G)$ since it contains G. Let us show that it is strictly greater by contradiction. Suppose that there exists a coloring $\mathcal{C} = \{S_1, \dots, S_{\chi(G)}\}$ of $\widetilde{G}$. First, $G_i = \widetilde{G}[V \cup w_i \setminus \{v_i, u\} \setminus W]$ is isomorphic to G, hence the restriction of $\mathcal{C}$ to the vertices of G_i has size $\chi(G)$ and is optimal. If $W \cap S_k = \emptyset$ for some $k \in \{1, \dots, \chi(G)\}$, by coloring each vertex v_i of S_k with the same color used (in $\mathcal{C}$) for w_i, we obtain a $\chi(G) - 1$ coloring of G, a contradiction. This means that for all $\chi(G)$ coloring of $\widetilde{G}$, $\chi(G)$

different colors appear in W, which implies having an extra color for vertex u, which contradicts the fact that it is $\chi(G)$-colorable.

This permits the construction of instances for which the integrality gap of (P_1) is as large as wanted [65].

Theorem 4.3.2 ([65])**.** *The optimal fractional coloring of* $\widetilde{G}$ *is equal to* $\chi_f(G)+\frac{1}{\chi_f(G)}$.

Here is a sketch of the proof: from any stable set S of G, $S \cup u$, $\{w_i|v_i \in S\} \cup S$ and W all form stable sets of $\widetilde{G}$. Given a fractional coloring x^* of G of value $\chi_f(G)$, by setting $x'_{S\cup u} = \frac{x^*_S}{\chi_f(G)}$, $x'_{S\cup\{w_i|v_i\in S\}} = \frac{(\chi_f(G)-1)x^*_S}{\chi_f(G)}$ and finally $x'_W = \frac{1}{\chi_f(G)}$ we obtain a fractional coloring of $\widetilde{G}$ of size $\chi_f(G)+\frac{1}{\chi_f(G)}$. The hard part of the proof in [65] shows that when x^* is an optimal fractional coloring of G, x' is optimal for $\widetilde{G}$.

Let us consider a sequence of graphs obtained by recursively applying Mycielski's operation. Their fractional coloring numbers follow the sequence $U_{n+1} = U_n + \frac{1}{U_n}$ while their coloring numbers follow $V_{n+1} = V_n + 1$. Both sequences strictly increase and diverge, but it can be shown that the difference between both also grows to infinity, which proves the arbitrarily large gap.

4.4 Strengthening the linear relaxation of (P_k)

Valid inequalities may be added to strengthen the linear relaxation of P_k. We first present the elementary Chvátal closure, which gives valid inequalities for any integer polytope given valid inequalities.

The *coloring polytope*, that is, the integral polytope subjacent to formulation P_1, is a particular case of covering polytope. These polytopes have been extensively studied, and the hope is that the theoretical analysis of such polytopes extends to experimental improvements on (P_1). Hence, we present classes of facet-defining inequalities for covering polytopes and then describe valid inequalities for the coloring polytope.

4.4.1 Separating Chvátal-Gomory inequalities

For an ILP defined with polynomially many variables, black box solvers often use heuristics to separate the elementary Chvátal closure since it is a generic way to reinforce a formulation, and it often helps to reduce significantly the integrality gap [66].

Separating the elementary Chvátal closure is doable using the MILP proposed in [66]. It is a heuristic separation procedure based on a MILP with a maximum number of 1000 branching nodes. It is generic, hence usable to separate heuristically the Chvátal elementary closure for any ILP. We describe the particular case of their formulation corresponding to our setup, that is, improving the linear relaxation of P and, hence, considering improving the linear relaxation of an RMP.

It takes as input a solution x^* of our RMP and computes a Chvátal-Gomory (CG) constraint such that its violation by x^* is maximized. The separating MILP will then be given by:

$$\max \alpha_0 - \sum_{S \in \mathcal{S}^*} x^*_S \alpha_S \tag{4.12}$$

$$f_S = \alpha_S - u(S) \forall S \in \mathcal{S}^* \tag{4.13}$$

$$f_0 = \alpha_0 - \mathbb{1}u \tag{4.14}$$

$$f_S \geq 0 \forall S \in \mathcal{S}^* \tag{4.15}$$

$$f_S \leq 1 - \delta \forall S \in \mathcal{S}^* \tag{4.16}$$

$$u_v \geq 0 \forall v \in V \tag{4.17}$$

$$u_v \leq 1 - \delta \forall v \in V \tag{4.18}$$

$$\alpha_S \in \mathbb{Z} \forall\ S \in \mathcal{S}^* \tag{4.19}$$

$$\alpha_0 \in \mathbb{Z} \tag{4.20}$$

.

In this MILP, the variables u are associated with the covering constraints of P and correspond to the linear combination associated with a CG constraint, the variable α_S (resp. α_0) represents the integer coefficient of the stable S (resp. right-hand side) in the CG constraint. The variable f_S (resp. f_0) corresponds to the gap between the coefficient α_S (resp. α_0) and the linear combination $u(S)$ (resp. $\mathbb{1}u$), that is $\lceil u(S) \rceil - u(S)$ (resp. $\lceil \mathbb{1}u \rceil - \mathbb{1}u$) and finally, δ is some small value to avoid having u coefficients greater or equal to 1. The objective function is to maximize the constraint violation associated with u. Note that some constraints are redundant, but the authors [66] explain that it helps stabilize the numerical behavior. In [66], it is proposed to add a little penalty in the objective value on variables u, weighted by a small value ϵ, to add numerical stability.

4.4.2 Set covering polytope

A covering polytope can be described as follows. Let M be a 0/1 matrix, a *covering polytope* $\mathcal{P}_M$ whose V-description has the following structure:

$$\mathcal{P}_M = conv\{x \in \{0,1\}^m \mid Mx \geq 1\}$$

. Rows of M can be seen as elements to cover, and columns as subsets of these elements. In the coloring problem, these elements are vertices in V and these subsets are stable sets of $\mathcal{S}$, we hence define $M_V^{\mathcal{S}}$ as the 0/1 matrix whose rows are vertices of V and columns are stable sets of $\mathcal{S}$.

Unlike packing polytopes, no beautiful combinatorial characterization is known for covering polyhedrons for which $\{x \in [0,1]^m \mid Mx \geq 1\}$ would be integer.

However, extensive work on this topic has given families of facets and valid inequalities. Our first idea was to use this knowledge to reinforce the stable set formulation P.

In this section, we describe some of these results and the limits to their compatibility within a column generation scheme.

A first attempt to link P and the facial structure of covering polyhedrons was initiated in [63]: they work on the *coloring polytope*, that is, the convex hull of P's solutions. Even if the study of covering polytopes goes beyond the study of the coloring polytope, we will use similar notations that we used in the previous section to create a smooth link between both studies.

Assuming that $M_V^{\mathcal{S}}$ has no zero row or column, Antonio Sassano [67] gives many classes of facets for the set covering polytope, characterizations of trivial inequalities defining facets, and a lifting procedure building facets from facets.

Theorem 4.4.1. *[67] $\mathcal{P}_M$ is full dimensional if and only if every row of M contains at least two 1s.*

Theorem 4.4.2. *[67] If $\mathcal{P}_M$ is full dimensional, then the following hold:*

- $x_S \geq 0$ *defines a facet of* $\mathcal{P}_{M_V^{\mathcal{S}}}$ *if and only if* $|\mathcal{S}_v \setminus \{S\}| \geq 2\ \forall v \in S$
- $x_S \leq 1$ *defines a facet of* $\mathcal{P}_M$
- *All facet defining inequalities* $\alpha^\top x \geq \alpha_0$ *for* $\mathcal{P}_M$ *have* $\alpha \geq 0$ *and* $\alpha_0 \geq 0$
- *The inequality* $x(\mathcal{S}_v) \geq 1$ *defines a facet if and only if* (i) *there exists no* $u \in V$ *such that* $\mathcal{S}_u \subsetneq \mathcal{S}_v$, *and* (ii) *for each* $S_1 \in \mathcal{S} \setminus \mathcal{S}_v$, *there exists* $S_v \in \mathcal{S}_v$ *such that* $\{S_v\} \cup \mathcal{S} \setminus \mathcal{S}_v \setminus \{S_1\}$ *covers* V
- *The only minimal valid inequalities for* $\mathcal{P}_M$ *with right-hand side equal to 1 belong to the system* $Mx \geq 1$

As a corollary of Theorem 4.4.2, we can obtain results for the coloring polytope, and this has been done by Hansen et.al., we describe these results in the next section.

The covering garden is full of roses

The families of facets given by Sassano [67], are *rank* facets, *i.e.* they have the form $\alpha^\top x \geq \beta$ where α only has $\{0,1\}$ coefficient. Their characterization goes through a bipartite graph having as vertex set $V \cup \mathcal{S}$ and where v is adjacent to S if and only if $v \in S$. Let us denote this graph by $G(V, \mathcal{S})$.

Definition 4.4.3. *[67] A bipartite graph $G(V,\mathcal{S})$ is a* complete q-rose of order p *if:*

(i) $\mathcal{S} = \{S_1, \ldots, S_p\}$

(ii) $V = \{v_1, \ldots, v_{\binom{p}{q}}\}$

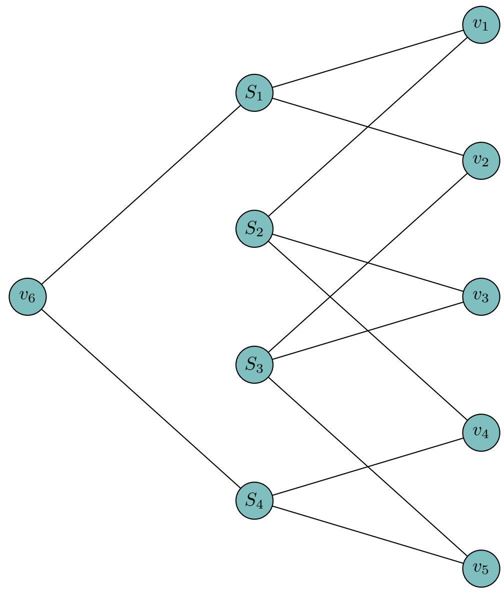


Figure 4.1: A 2-rose of order 4

(iii) each v_i *is contained in exactly* q *elements of* $\mathcal{S}$

(iv) $\forall v_i, v_j \in V,\ i \neq j \Rightarrow \mathcal{S}_{v_i} \neq \mathcal{S}_{v_j}$

Moreover, we say that $\widetilde{\mathcal{S}} \subseteq \mathcal{S}$ *is a* q-clique *(of order* $|\widetilde{\mathcal{S}}|$*) if there exists* $\widetilde{V} \subseteq V$*, such that* $G(\widetilde{V}, \widetilde{\mathcal{S}})$ *is a complete* q*-rose or order* $|\widetilde{V}|$*.*

Figure 4.1 gives an example of 2-rose of order 4. Note that the matrix associated with a complete q-rose of order p is never the transpose of a stable set incidence matrix, see Theorem 1.2.5.

Definition 4.4.4. *[67] For an integer* $t \leq p$ *, let us denote by* $\mathcal{T}^i$ *the set* $\{S_i, \ldots, S_{i+t-1}\} \subseteq \mathcal{S}$ *(indices taken modulo* p*). A bipartite graph* $G(V, \mathcal{S})$ *is a* (q,t)*-rose of order* p *if:*

(i) $\mathcal{S} = \{S_1, \ldots, S_p\}$

(ii) V *is the union of subsets* $V^1, \ldots, V^p$ *where* $\bigcup_{v \in V^i} \mathcal{S}_v = \mathcal{T}^i\ \forall i \in \{1, \ldots, p\}$

(iii) each subgraph $G(V^i, \mathcal{T}^i)$ *is a complete* q*-rose of order* $t \geq q \geq 2$

On Figure 4.2 we show a $(2,3)$-rose of order 7: $\mathcal{S} = \{S_1, \ldots, S_7\}$, $V = \{v_1, \ldots, v_14\}$, $\mathcal{T}^1 = \{S_1, S_2, S_3\}$, $\mathcal{T}^2 = \{S_2, S_3, S_4\} \ldots, \mathcal{T}^p = \{S_1, S_2, S_7\}$, $V^1 = \{v_1, v_2, v_3\}$, $V^2 = \{v_3, v_4, v_5\}, \ldots, V^p = \{v_{12}, v_{13}, v_{14}\}$. Note that the intersection between these families of bipartite graphs is not empty: the graph of Figure 4.1 is also a (2,3)-rose of order 4.

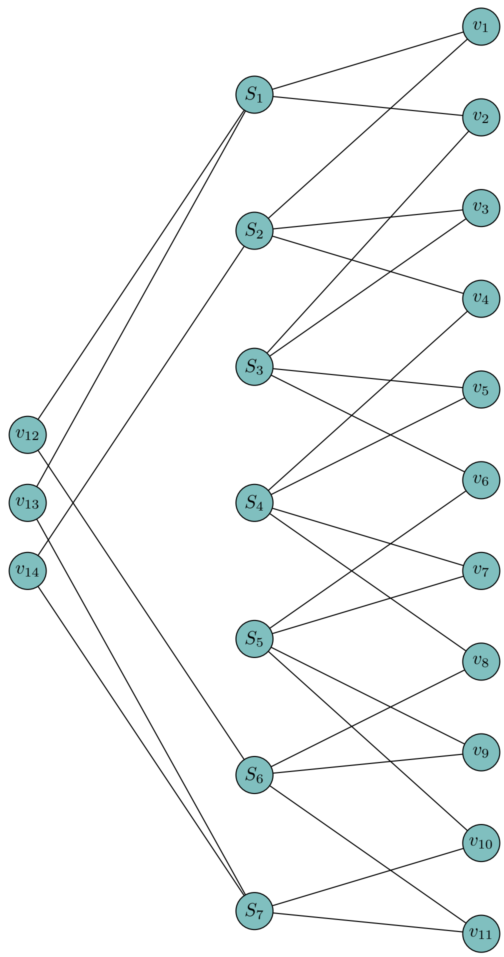
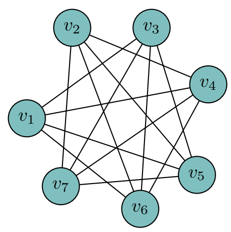


Figure 4.2: A (2,3)-rose of order 7



$$
\begin{array}{c c}
\begin{array}{ccccccc} \{1,7\} & \{1,2\} & \{2,3\} & \{3,4\} & \{4,5\} & \{5,6\} & \{6,7\} \end{array} & \\
\begin{pmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix} & \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{array}
\end{array}
$$

Figure 4.3: An odd antihole and the transpose of its stable set incidence matrix

Most proofs for rank facets of the covering polytope use a Lemma characterizing such facets in terms of connectivity of the so-called critical graph (given by a very technical definition). Still, the validity of such inequalities is easier to understand than it seems by looking at the right-hand side.

Let us introduce some facet defining inequalities for the covering polytope $\mathcal{P}_{M_V^{\mathcal{S}}}$.

By definition of a complete q-rose of order p $G(V,\mathcal{S})$, the covering number of $G(V,\mathcal{S})$ is exactly $p-q+1$: for every set $\widetilde{\mathcal{S}} \subset \mathcal{S}$ of size $|\widetilde{\mathcal{S}}| < p-q+1$, there exists an element of $v \in V$ contained in the q elements of $\mathcal{S}$ that do not belong to $\widetilde{|\mathcal{S}|}$. The associated valid inequality for $\mathcal{P}_{M_V^{\mathcal{S}}}$ is $\sum_{S\in\mathcal{S}} x_S \geq p-q+1$.

Now, given a (q,t)-rose of order p $G(V,\mathcal{S})$, and the associated $\mathcal{T}^i$, V^i $\forall i \in \{1,\ldots,p\}$, we obtain the validity of the inequalities defined in the following Theorem 4.4.5 by a Chvátal sum with coefficient $\frac{1}{t}$ of the inequalities associated with a complete q-rose of order p $G(\mathcal{T}^i, V^i)$ for all $i \in \{1,\ldots,p\}$.

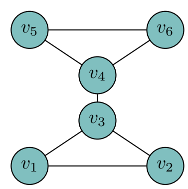
Theorem 4.4.5. *[67] Given a (q,t)-rose of order p $G(V,\mathcal{S})$, the inequality $\sum_{S\in\mathcal{S}} x_S \geq \lceil \frac{(t-q+1)p}{t} \rceil$ defines a facet of $\mathcal{P}_{M_V^{\mathcal{S}}}$ if and only if $p(q-1) \mod t \neq 0$.*

Corollary 4.4.6. *[67] If $G(V,\mathcal{S})$ is a (q,q)-rose of order p, $p \neq q$, then the inequality: $\sum_{S\in\mathcal{S}} x_S \geq \lceil \frac{p}{q} \rceil$ defines a facet of $\mathcal{P}_{M_V^{\mathcal{S}}}$.*

Application to the coloring polytope of antiholes

It is well known that to color any antihole of odd size k, $\frac{k+1}{2}$ colors are needed. If G is an odd antihole, then $M_V^{\mathcal{S}}$ is a matrix having two 1 per line and columns and has an odd size. This matrix is the incidence matrix of a $(2,2)$-rose of order k. Hence, Corollary 4.4.6 applies and gives a facet with right-hand side $\frac{k+1}{2}$, see Figure 4.3 gives an illustration with an odd antihole of size 7.

However, these families of facets depend on the whole structure of the constraint matrix. Indeed, in packing problems, a local constraint is always valid for the whole problem but this is not the case with covering problems. An example based on the coloring problem is given in Figure 4.4: when looking at the red submatrix, we deduce a local constraints of the form $x_{1,4} + x_{1,5} +$



$$\begin{array}{c c}
\begin{array}{cccccccc} \{1,4\} & \{1,5\} & \{2,4\} & \{2,5\} & \{3,5\} & \{1,6\} & \{2,6\} & \{3,6\} \end{array} & \\
\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} & \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array}
\end{array}$$

Figure 4.4: Nonvalidity of local constraints

$x_{2,4} + x_{2,5} + x_{3,5} \geq 3$. This inequality is not valid for the whole problem as $\{\{1,4\},\{2,5\},\{3,6\}\}$ is a covering but does not satisfy this constraint

Unfortunately many covering polytopes do not have such a well-structured constraint matrix as in Theorem 4.4.5 and Corollary 4.4.6, so these results do not apply directly even if a strict submatrix has such a structure.

To tackle this problem, lifting theorems should lead to valid inequalities for the whole problem. Given $\alpha^\top x \leq a_0$ a facet defining inequality for $\mathcal{P}_{V_0}^{S^*}$, we captures the smallest coefficient β such that $\alpha^\top x + \beta^\top x_S v \geq a_0$ is valid for $\mathcal{P}_{V_0}^{S^* \cup \{S\}}$ in the next definition.

Definition 4.4.7. *[67] Given $G(V_0, \mathcal{S}^*)$ a subgraph of $G(V, \mathcal{S})$ and a facet defining inequality $a^\top x \geq a_0$ for $\mathcal{P}_{M_{V_0}^{\mathcal{S}^*}}$. For $S \in \mathcal{S} \backslash \mathcal{S}^*$, we denote by $\alpha_S(G(V_0, \mathcal{S}^*)) = min_{\mathcal{C} \subseteq \mathcal{S}^*}\{a\chi^{\mathcal{C}} | \mathcal{C}$ covers $V_0 \setminus S\}$, where $\chi^{\mathcal{R}}$ is the incidence vector of $\mathcal{R} \subseteq \mathcal{S}$.*

Theorem 4.4.8. *[67] Given $G(V_0, \mathcal{S}^*)$ a subgraph of $G(V, \mathcal{S})$ and a facet defining inequality $a^\top x \geq a_0$ for $\mathcal{P}_{M_{V_0}^{\mathcal{S}^*}}$.*

For each $S \in \mathcal{S} \setminus \mathcal{S}^*$, the inequality

$$a^\top x + (a_0 - \alpha_S(G(V_0, \mathcal{S}^*)))x_S \geq a_0$$

defines a facet of the polytope $\mathcal{P}_{M_{V_0}^{\mathcal{S}^* \cup \{S\}}}$.

Note that if $a^\top x \geq a_0$ is valid but not facet defining, $a^\top x + (a_0 - \alpha_S(G(V_0, \mathcal{S}^*)))x_S \geq a_0$ will still be a valid inequality for $\mathcal{P}_{M_{V_0}^{\mathcal{S}^* \cup \{S\}}}$.

Note also that by repeatedly applying this procedure, one can obtain a facet defining inequality for $\mathcal{P}_{M_{V_0}^{\mathcal{S}}}$. Hence, this latter inequality is valid for $\mathcal{P}_{M_V^{\mathcal{S}}}$. Unfortunately, as we show in Chapter 5, this lifting procedure is not compatible within a column generation scheme.

Rank 2 and 3 inequalities

Antonio Sassano and Gérard Cornuéjols [68] characterize the facets with right-hand side 2 by giving an exponential procedure to enumerate them. It consists of enumerating the subsets W of V and summing the associated inequalities before dividing by $|W| - \epsilon$ and up-rounding every coefficient of the obtained inequality. This shows that they all belong to the elementary Chvátal closure.

Theorem 4.4.9. *Every facet of $\mathcal{P}_{M_V^S}$ with right-hand side $\{0, 1, 2\}$ belongs to the elementary Chvátal closure.*

They also give two characterizations for an inequality of the type $\alpha^\top x \geq 2$ to be minimal and a characterization of the ones defining facets. They show that for each $k \geq 3$, a covering polytope exists for which a facet defining inequality with right-hand side k does not belong to its elementary Chvátal closure. In [69], the same authors discuss how to lift such inequalities. *Lifting* an inequality often refers to a procedure that, given a facet of a polyhedron, deduces a new facet defining inequality of that same polyhedron or of another one whose constraint defining matrix is close. Finally, M. Sánchez-Garcia M.I. Sobroón and B. Vitoriano [70] characterize all facets with coefficient $\{0, 1, 2, 3\}$. According to the increasing difficulty of [71] [68], [69],[70], we understand that no one tried to characterize all the facets with coefficients greater than 3.

To evaluate the ability of the rank 2 inequalities to improve the relaxation value, we enumerated such constraints and added them to P for a few little instances of coloring problems. Unfortunately, this did not improve the relaxation value.

4.4.3 Polyhedral study of the coloring polytope

Hansen, Labbé et Schindl [63] proposed a polyhedral study of the coloring polytope that is the polytope subjacent to formulation P_1 using the generics results for the covering polytope that I recalled in Section 4.4.

Proposition 4.4.10 ([63])**.** *$\mathcal{P}_{M_V^S}$ is full dimensional if and only if $\forall v \in V$, $V \setminus N(v)$ is not a stable set.*

If this condition is not verified, one can simplify the instance. Indeed, if there exists v such that $V \setminus N(v)$ is a stable set, the optimal coloring of G is the optimal coloring $G[V \setminus \overline{N}(v)]$ to which we add $v \cup V \setminus N(v)$. This transformation is doable in polynomial time and can be applied until we consider a subgraph whose coloring polytope is fully dimensional.

A vertex u is said to be dominated if there exists $v \in V$ such that $N(u) \subseteq N(v)$ and $u \neq v$.

Proposition 4.4.11 ([63])**.** *The only facet defining inequalities with a right-hand side equal to 1 are the cover inequalities associated with non-dominated vertices.*

Both Propositions 4.4.11 and 4.4.10 are respectively deduced from the Theorems 4.4.1 and 4.4.2. Moreover, they interpret the results of Cornuéjols et.al. [71], regarding the coloring polytope. They also give necessary conditions for an elementary Chvátal rank inequality to be non-dominated for the coloring polytope.

They did computational experiments adding to P_1 constraints associated with odd holes of size 5 to improve the linear relaxation, but they said it to be inefficient. The idea of such constraints is that every odd hole needs at least 3 stable sets to be covered. It can be extended with odd antiholes, stating that an odd antihole of size $2k+1$ needs at least $k+1$ stable sets to be covered.

4.5 Dual optimal inequalities for column generation

Column generation suffers from slow convergence, and to bypass this problem, different methods, called stabilization methods, have been derived [72]. One of them consists of introducing *Dual Optimal Inequalities* (DOI), that is, adding extra columns to the linear relaxation that do not perturbate its optimum value and for which an optimal solution with positive values for variables associated with such extra columns can be transformed into an optimal solution of the original relaxed problem, that is, with a null coefficient in these extra columns. The name comes from the fact that each extra column corresponds to a constraint that cuts the dual solution space but none of its optimal solutions. One could think that adding this columns to the rimal only makes the RMP harder to solve, but in fact it stabilizes the columns generation by cutting the space of dual values, which often results in a lower number of columns.

Gschwind et al. [4] proposed a set of DOI for different column generation formulations for different problems based on the following structural property of the constraint matrix.

Definition 4.5.1 ([4])**.** *Given, an $n \times m$ matrix M and two size m vectors I, J, matrix M has the (I, J)-exchange property if, for any of its column H, $H \geq I$ implies that $H - I + J$ is a column of M.*

Theorem 4.5.2 ([4])**.** *Given a linear system*

$$\min\{\mathbb{1}^\top x \mid Mx \geq b,\ x \geq 0\}$$

where M has positive integer coefficients and $b \geq 0$, if M has the (I, J)-exchange property and I is a canonical column vector, then $J^\top y \leq I^\top y$ does not cut off any optimal point of the dual

$$\max\{b^\top y \mid M^\top y \leq \mathbb{1}\}.$$

Theorem 4.5.2 permits to deduce that if M has the (I, J)-exchange property for some canonical unit vector I, then:

$$\min\{\mathbb{1}^\top x \mid Mx \geq b,\ y \geq 0\} = \min\{\mathbb{1}^\top x \mid Mx + (J^\top - I^\top)z \geq b,\ z \geq 0\}.$$

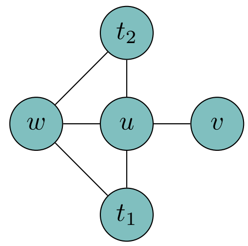


Figure 4.5: A graph for which $M_V^{\mathcal{S}}$ has an exchange property

where z is a new variable. Moreover, as the set of variables of the left-hand program is included in the set of variables of the right-hand program, the dual space of the right-hand one is included in the dual space of the other.

Gschwind et al. [4] use Theorem 4.5.2 to propose DOIs for formulation (P) solving the coloring problem on a graph $G = (V, E)$, whose constraint matrix is $M_V^{\mathcal{S}}$. Given $w \in V$ and a stable set T such that $T \subseteq N_t$ and $N(T) \subseteq \overline{N}_w$, then $M_V^{\mathcal{S}}$ has the (χ^w, χ^T)-exchange property. Then the DOI variable z has coefficients (χ^w, χ^T) as follows.

$$\begin{pmatrix} -1 \\ 1 \\ ... \\ 1 \\ 0 \\ ... \\ 0 \end{pmatrix} \begin{array}{l} \Longrightarrow\ w \\ \\ \Longrightarrow\ T \\ \\ \\ \\ \\ \end{array}$$

This matrix property translates in terms of graph structure: given a stable set S containing w, as $N(T) \in \overline{N}_w$, there is no edge between $S \setminus w$ and T, as T is a stable set, $(S \cup T) \setminus w$ is also a stable set of G. An example is given in Figure 4.5 where $T = \{t_1, t_2\}$. A stable set containing w is given by $\{w, v\}$, and $\{v,_1, t_2\}$ is another stable set.

This implies that adding a variable z with coefficients $\chi^T - \chi^w$ to P does not change its optimal value. However, given a solution x, z of P with the additional column $\chi^W - \chi^t$, where $z \neq 0$, x may not be a solution of P as vertices of W may not be sufficiently covered. But, since z has a negative coefficient in the constraints associated with t, t will be excessively covered. For that reason, by taking a stable set S containing t such that $x_S \neq 0$, reducing x_S and increasing $x_{S \cup W \setminus t}$ will permit to obtain a solution of (P_1) Note that this can be done only if $z \leq x_S$ otherwise, it will be necessary to iterate over several stable sets S.

As the number of such structures can be exponential in the size of a graph, Gschwind et al. [4] first propose to add a set of such DOI as a preprocessing, this has been experimentally successful. However, Gschwind et al. [4] also investigated adding DOI dynamically during the column generation process.

The DOIs based on the exchange property have appeared previously in several contexts linked to cutting stock problems but are not the only type of DOI that have been proposed in the literature. Another type of dual inequalities that have been proposed in [4] is called *deep dual optimal inequalities* (DDOI), they correspond to inequalities that do not cut off every dual optimal point.

Dual optimal inequalities DOI were introduced in 2005 by José M. Valério de Carvalho [73] for the cutting stock problem. This problem is to find the minimal number of patterns needed to satisfy a set of demands $d_i,\ i \in I$ in items of sizes $s_i \leq W$. It can be formulated as an ILP with exponentially many variables associated with every possible cutting pattern. Its constraints are associated with each item and having a right-hand side equal to the number of such items needed. Remark that given an item of size s_i we can cut a new item having size $s_j \leq s_i$, and this can be generalized in the following way: cutting an item of size s_i into a set of items J is possible when $\sum_{j \in J} s_j \leq s_i$. Hence, José M. Valério de Carvalho proposed to add new variables to the primal with zero objective coefficients, coefficient 1 in constraints associated with items in J, -1 is the constraint associated with i, and 0 in the other constraints. There exists exponentially many such DOI variables. Hence, he made experimental computations with $|J| = 1, 2$ only and added only one variable with coefficient -1 in line i for each item i. It resulted in less primal iterations and computation time to compute the linear relaxation. Later, Hatem Ben Amor et.al. [74] showed that aggregating constraints from the binary cutting stock (where each item has a unit demand and several items can have the same size) problem works equivalently as adding equality constraints in the dual formulation from a computational point of view. This corresponds to the use of deep dual optimal inequalities, *i.e.*, inequalities that do not cut every optimal solution of the dual. Later, Cláudio Alves and J.M Valério de Carvalho [75] proposed a stabilized branch and cut and price algorithm for the multiple length cutting stock problem, which is a generalization of the previous problem where the patterns have different sizes and for each size there is a maximum number of patterns available. They were the first to try branching with DOI and pointed out that some DOI conflicted with the branching decisions. Hence, they removed DOI from the master problem when such conflicting decisions were made. They also added DOI during the branching, considering the branching decisions as constraints of the primal and, hence, as variables of the dual. This also has been experimentally successful.

In another article on the cutting stock problem, François Clautiaux, Cláudio Alves, and J.M Valério de Carvalho [76] proposed to add *safe* lower and upper bounds on the dual variables: these bounds do not cut off any optimal solution. They also introduce a particular type of dual cut associated with a given dual solution y^*. This type of cut would separate every dual point that is a linear combination of y^* and some other solution. This type of inequality may sometimes cut every optimal dual solution, but in that case, they prove that y^* is already an optimal solution. The idea of their framework is to add a set of such cuts before solving the linear relaxation and then remove subsets of these cuts to check if it improves the linear relaxation. In other words, they check whether these cuts have cut every optimal point. Note that they never solve the linear relaxation without additional cuts. Nonetheless, they prove that this framework computes a right linear relaxation. They propose removing patterns associated with constraints whose dual value is zero. They give a lemma characterizing a set of such items. Given a valid lower bound for the number of patterns needed, if this lower bound is greater than some number (polynomially computable) depending on the instance, then every 'small' item can be removed from the instance. The last idea tried in this article is "Original trust region." The idea is to use a known dual solution close to the optimal one. Then add dual box constraints containing this solution, solve the LP, and check if removing the box improves the relaxation, if this is not the case, then it is a dual optimal solution. They show that these ideas stabilize the column generation on 10 sets of instances for which they compare different combinations of dual cuts/bounds/trust region combinations.

Chapter 5

The graph coloring problem and alternation strategies

In this chapter, we discuss whether it is possible to use the general results on the set covering polytope for developing a Branch and Price and Cut (B&C&P) for the graph coloring problem. Finally, we present a new B&C&P based on different alternation strategies between cutting and pricing. Finally, we propose a method using both dual inequalities and Chvátal-Gomory constraints.

5.1 Two remarks about graph coloring polytopes

First, we show that the inequalities used in [63] to reinforce (P) belong to the elementary Chvátal closure of $\{x \mid M_{\mathcal{S}}^{V} x \geq 1,\ x \geq 0\}$. We recall that $M_{V}^{\mathcal{S}}$ is the 0/1 matrix whose rows are vertices of G and columns are stable sets of G.

Lemma 5.1.1. *The following inequality families belong to the elementary Chvátal closure of* $\{x \mid M_{\mathcal{S}}^{V} x \geq 1,\ x \geq 0\}$*:*

(i) $\sum_{S \in \mathcal{S} \mid S \cap C \neq \emptyset} x_S \geq 3$ *for each odd hole* C *of* G*,*

(ii) $\sum_{S \in \mathcal{S} \mid S \cap C \neq \emptyset} x_S \geq \frac{|C|+1}{2}$ *for each odd antihole* C *of* G*.*

Proof. Let us sum the inequalities of $M_{V}^{\mathcal{S}} x \geq 1$ associated with vertices of C. If C is a hole, then a stable set contains at most $\frac{|C|-1}{2}$ vertices of C, and hence, by dividing the obtained inequality by this quantity and rounding up every coefficient gives inequality *(i)* associated with C. If C is an antihole, then every stable set contains at most two vertices of C, in which case, dividing the obtained inequality by 2 and rounding up gives the inequality *ii)* associated with C. $\square$

Lemma 5.1.1, and Theorem 4.4.9 let us think that separating the elementary Chvátal closure may be promising for the coloring problem as this procedure seems natural to obtain facet-defining inequalities for covering polytopes.

Lifting procedure of Theorem 4.4.8 and column generation A first question arising from the lifting operation of Theorem 4.4.8 is to see whether it can be used in a column generation scheme.

The first issue with such a procedure is that, generally, the lifting coefficient α_S is not linear in the elements of S. This changes the structure of the pricing subproblems. An idea to address this issue would be to compute heuristically a quantity a'_S higher or equal to a_S. This would still lead to a valid inequality, but weaker if $a'_S > a_S$. This upper bound could be linear on the elements of S, which would not change the structure of the pricing subproblems.

The second issue seems harder to address. Since the procedure given in Theorem 4.4.8 to obtain a valid inequality for $\mathcal{P}_{M^V_{\mathcal{S}}}$ is iterative, it is order dependent. We give a simple example of this limit with the graph in Figure 5.1. Here, a valid covering of the graph by stable sets is given by $\mathcal{S}^* = \{\{1,4\},\{2,5\},\{3,6\}\}$. The inequality $\sum_{S\in\mathcal{S}^*} x_S \geq 3$ is valid for $\mathcal{P}_{M^{1,2,3,4,5,6}_{\mathcal{S}^*}}$. This inequality is redundant since it is the half sum of the inequalities with right-hand side one restricted to the subset of columns in $\mathcal{S}^*$. Using Theorem 4.4.8, for $S' = \{1,2,3\}$ we obtain the valid inequality $\sum_{S\in\mathcal{S}^*} x_S \geq 3$ for $\mathcal{P}_{M^V_{\mathcal{S}^*\cup\{S'\}}}$. But now, if we apply it again with $S'' = \{4,5,6\}$ we obtain the inequality $\sum_{S\in\mathcal{S}^*} x_S + 3x_{S''} \geq 3$. Now, if we do it in the reverse order, we obtain $\sum_{S\in\mathcal{S}^*} x_S + 3x_{S'} \geq 3$.

Until now, this seems compatible with column generation since such an algorithm directly gives an order of columns. It turns out that this idea does not work well from a theoretical point of view, it hardens the convergence criteria of the column generation. Given a subset of columns $\mathcal{S}^*$ and a valid inequality $ax \geq a_0$ for $\mathcal{P}_{M^V_{\mathcal{S}^*}}$. Given an order of the columns in $\mathcal{S}\backslash\mathcal{S}^*$, say, $(S_1,\ldots,S_{|\mathcal{S}\backslash\mathcal{S}^*|})$, the reduced cost of a column S_i depends on a_{S_i}, which can be computed using Theorem 4.4.8 only by knowing the coefficients $\{a_{S_k} | k \leq i\}$. This seems to be possible only by enumerating such columns recursively and hence generating all the columns.

If one ignores such an issue and prices a column S, supposing that a_S (or a'_S) should be the coefficient obtained as if S was the first column in the given order. What can happen is that no column has a negative reduced cost at this iteration. Still, after adding S to the RMP (even if it has a positive reduced cost), another column, say S', now has a negative reduced cost since it is added as the second element of that order.

In conclusion, using such a lifting operation appears to contradict the principle of column generation, that is to generate the fewer possible columns.

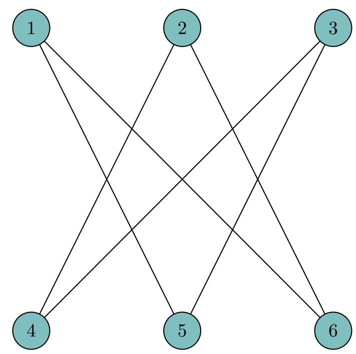


Figure 5.1: A limit in the use of Theorem 4.4.8 in a column generation scheme

5.2 B&C&P with Chvátal-Gomory constraints

In [68], all facet defining inequalities with right-hand side 0, 1, and 2 for the set covering polyhedron are given. The only facet defining inequalities with right-hand side 0 are $x_S \geq 0$, the ones of right-hand size 1 are: $\sum_{S|v \in S} x_S \geq 1$, and the ones with right-hand side 2 are proven to belong to the elementary Chvátal closure. Recall that the odd hole and antihole inequalities used in [63] also have rank 1. Hence, we decided to separate the elementary Chvátal closure.

5.2.1 Pricing with Chvátal-Gomory constraints

This section describes how to use Chvátal-Gomory (CG) constraints to reinforce (P_1). Note that a Chvátal-Gomory constraint is obtained by taking a fractional linear combination of constraints, which implies having a vector of fractional coefficients. In the particular case of (P_1), constraints are associated with vertices, hence we represent these coefficients by a vector $c \in \mathbb{R}^V$. Now, the obtained reinforced formulation $(\widetilde{P})$ is the following, where C is a set of CG multipliers c:

$$\min \sum_{S \in \mathcal{S}} x_S \tag{5.1}$$

$$\sum_{S \in \mathcal{S} | v \in S} x_S \geq 1 \qquad \forall\ v \in V \tag{5.2}$$

$$\sum_{S \in \mathcal{S}} x_S \lceil c(S) \rceil \geq \lceil \mathbb{1}^\top c \rceil \qquad \forall c \in C \tag{5.3}$$

$$x_S \geq 0 \qquad \forall\ S \in \mathcal{S} \tag{5.4}$$

.

We then must deal with additional dual variables associated with CG constraints in $(\widetilde{P})$. Let us denote by μ_c the dual cost of a CG constraint associated with $c \in C$. The reduced cost of a stable set S becomes:

$$r_S = 1 - \sum_{v \in S} \lambda_v - \sum_{c \in C} \mu_c \lceil c(S) \rceil.$$

Note that this quantity is no longer linear in the vertices of S (it is linear when no CG constraints are added to the primal) and modifies the structure of the pricing subproblems. To tackle this problem, in [77] it is proposed but not tried, to use a modified ILP. We propose a similar exact ILP (SP) solving the pricing subproblems with an additional variable z_c and one additional constraint for each CG constraint of the primal:

$$\begin{aligned}
\max \sum_{v \in V} \lambda_v w_v + \sum_{c \in C} \mu_c z_c & & \\
w_u + w_v \leq 1 & & \forall\ uv \in E, \\
z_c - \sum_{v \in V} c_v w_v \leq 1 - \epsilon & & \forall c \in C, \\
w_v \in \{0, 1\} & & \forall\ v \in V. \\
z_c \in \mathbb{N} & & \forall\ c \in C.
\end{aligned}$$

An important aspect of practically using (SP) as an exact algorithm is to compute an ϵ value such that SP is valid. For a given CG constraint associated with c, the tightest ϵ_c (the one that would give the strongest valid constraints for the formulation SP) is given by the following formula:

$$\epsilon_c = \begin{cases} \min\limits_{\substack{S \in \mathcal{S} \\ c(S) - \lfloor c(S) \rfloor > 0}} c(S) - \lfloor c(S) \rfloor & \text{if it exists,} \\ 0 & \text{otherwise.} \end{cases}$$

This quantity seems hard to compute. Hence, we propose every c_u to 10^{-4} which implies that $\epsilon = 10^{-4}$ gives a valid ILP pricing. This change may imply changes in the separated inequalities.

To avoid calling an exact pricer at each iteration, we decided also to use two heuristics. We first approximate c_S by a function that is linear on the vertices:

$$\widetilde{r}_S = 1 - \sum_{v \in S} \lambda_v - \sum_{c \in C} \mu_c \sum_{v \in S} c_v$$

Now, the weight of a vertex u becomes: $f_u = \lambda_u + \sum_{c \in C} \mu_c c_u$. This weight permits us to use the following three algorithms.

Greedy Algorithm: it consists of sorting vertices u by decreasing $f(u)$, starting from an empty set S in which we recursively add the first vertex not in $N(S) \cup S$.

Root Node Pricer: it corresponds to the resolution of (SP) without z variables and with f as objective, only solving the root branching node done with a black box ILP solver.

Exact Pricer: Finally, to ensure convergence, we solve (SP) until a negative

reduced cost column is found using a black box ILP solver.
Note that another possibility would be to let the solver solve (SP) to optimality. Still, a preliminary computational experiment showed that returning the first negative reduced cost column gives an overall faster method, even if the number of columns is slightly greater. Note that for the baseline algorithm, for which no cuts are generated, having a root node pricer and an exact pricer returning the first negative reduced cost column is redundant; no root node pricer will be launched for this algorithm.

5.2.2 Alternation strategies

Column generation is known to suffer from convergence issues: in particular, a phenomenon called the *tailing-off effect* consists of a sequence of iterations of subproblems, improving poorly the optimal values of the associated sequence of RMPs. Most of the time, with BCP, the strategy is to add cuts after primal convergence (when no reduced cost columns exist) because adding constraints to the RMPs adds dual variables and hence hardens the pricing subproblems. This is especially the case when such cuts are *non-robust*, that is, they change the structure of the pricing subproblems. On the other hand, it should be possible to minimize the tailing-off effect by cutting before primal convergence, *i.e.*, launching a cutting round in the case when adding a column does not improve the objective value of the RMP 'much' with the hope that it perturbates the column generation process by cutting the sequence of non-improving solutions. To investigate these techniques, we first need reinforcement inequalities, and then, we propose different alternation strategies and investigate their performances.

To investigate whether cutting before primal convergence can limit the tailing-off effect, we propose to compare four parametrized alternation strategies between pricing and cutting:

- Baseline: no cutting round is launched
- Price&Cut: a cutting round is launched any time the exact pricer does not find an improving column
- k-*alternation*: if the improvement of the objective value is less than 10^{-5} after k iterations of the restricted master problem, start a cutting round
- (α, k)-*alternation*: each kth iteration, launch a cutting round. If a violated constraint is found, set k to $k \times \alpha$ $(0 \leq \alpha \leq 1)$, otherwise set k to $\frac{k}{\alpha}$

Note that the baseline algorithm is the particular case when the alternation strategy never decides to cut. See Figure 5.2 for a diagram describing our BCP scheme. At each RMP resolution, that is solving the linear relaxation of P on a subset of variables, the chosen strategy decides whether to cut or to price. If it decides to cut, we launch the MILP separating the CG constraints (SEPA1) [66]. If it decides to price, three pricers are successively launched only when the previous one fails to find a negative reduced cost column always in the same order: Greedy $\rightarrow$ Root node $\rightarrow$ Exact.

The 'convergence' node aims to ensure that both the separator and the pricing procedures would not find any cut/column, this ensures that a valid lower bound is computed (*i.e.*, the relaxation value is computed): after a failing pricing round, meaning that no negative reduced cost column exists, the separator has to be launched to try to find a new cut, if this round fails, then convergence is achieved, otherwise, the cut is added and the algorithm goes back to solving the new RMP. If a separator fails to find a column, then a pricing round must be launched to try to find a negative reduced cost column if this round fails, then convergence is achieved. This latter does not appear in the figure for clarity's sake.

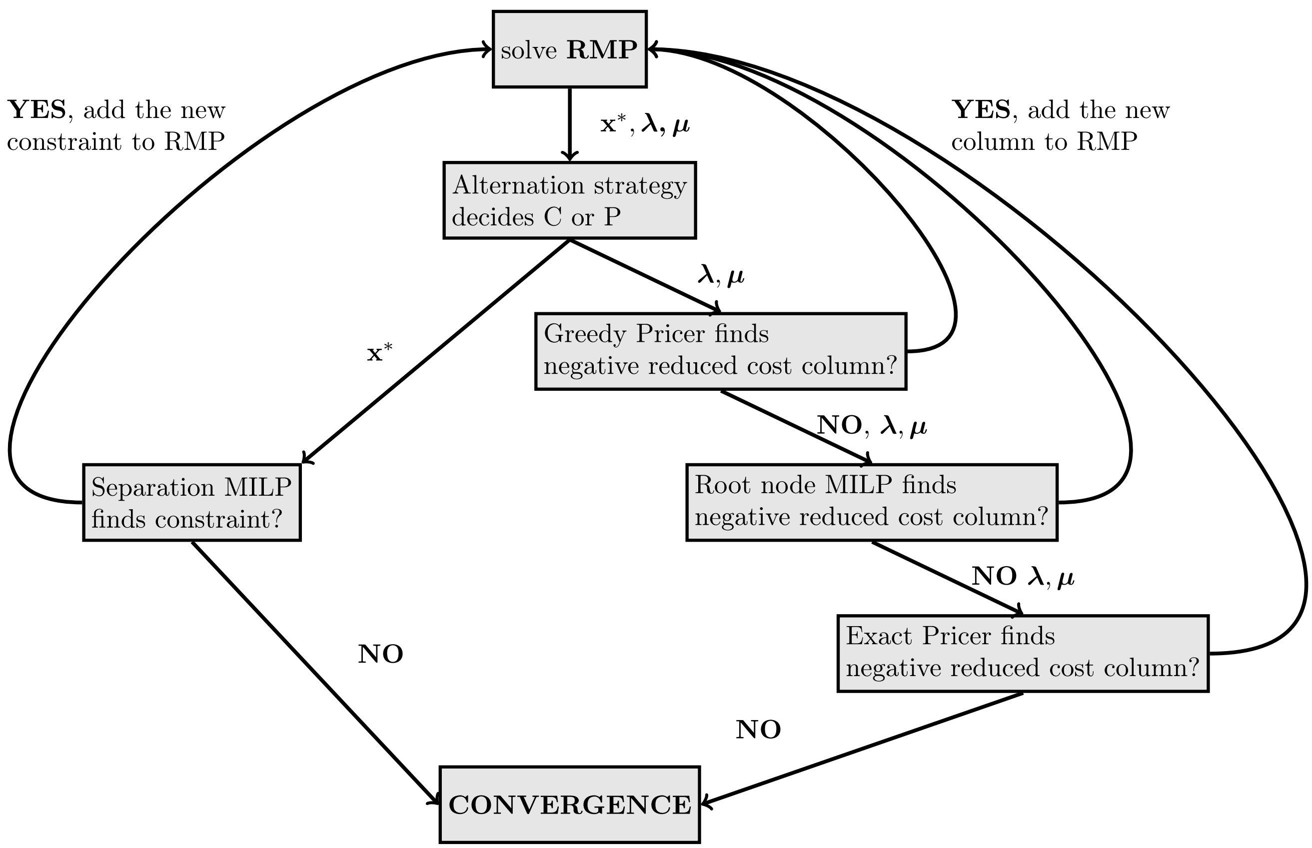


Figure 5.2: BCP scheme

5.2.3 Experimental results

Technical details

We ran the experiments on a computer with an intel core i5, 3.10 GHz processor, and 16 GB RAM. We used SCIP 7.0.3 [78] for the column generation framework, CPLEX [79] for the LPs resolutions, and the pricing heuristics using MILP and the separation procedure.

Instance description

We compare the different formulations we propose with the baseline algoithm on three sets of instances:

- Color02 instances [50] solved in less than three hours by the Baseline, for which χ_f is computed in less than an hour by the Baseline. This gives a set of 40 instances
- Instances of the second and tenth DIMACS challenge solved in less than three hours by the baseline and for which χ_f is computed in less than an hour by the Baseline. This gives a set of 9 instances.
- The complements of the instances of the second and tenth DIMACS challenge solved in less than three hours by the Baseline, for which χ_f is computed in less than an hour by the Baseline. It is a set of 7 instances.

Note that Color02 contains the three first Mycielski instances. These instances are obtained by recursively applying the construction of Mycielski 4.3.1 to the hole of size 5. In practice, such adversarial instances are hard to solve with P^G, and the 6th one (myciel6) already takes many hours with the implementation we will describe in the next section.

Tables

The meaning of the column's titles is as follows:

- multicolumn 'root node': statistics for the computation of the root branching node
- multicolumn 'Complete Branching': statistics for whole branching tree computation time
- 'CPU': ratio of time needed for the corresponding algorithm to solve this set of instances and the time needed by the Baseline to solve this set of instances
- '# col': ratio of columns generated by the corresponding algorithm to solve this set of instances and the number of columns generated by the Baseline to solve this set of instances
- '% gap': corresponds to average gap closure on the instances for which $\chi_f \neq \chi$ calculated by the formula $\frac{\chi-\chi_{alg}}{\chi-\chi_f}$ where χ_{alg} is the relaxation of the corresponding algorithm
- '# nodes': ratio of branching nodes generated by the corresponding algorithm to solve this set of instances and the number of branching nodes generated by the Baseline to solve this set of instances

- % faster is the proportion of instances solved faster by the corresponding algorithm than the Baseline
- '# cuts': number of generated cuts

Since the separation procedure we use for separating the elementary Chvátal closure is heuristic, our different alternation strategies may induce different relaxation values. 9 instances of the Color02 set verify $\chi \neq \chi_f$. Only three instances of the set DIMACS verify $\chi \neq \chi_f$. No instances of the set com-DIMACS verify $\chi \neq \chi_f$ hence, the gap closure does not exist for these instances.

		root node			Complete Branching				
Instances set	algorithm	CPU	# col	% gap	CPU	# col	# node	% faster	# cuts
Color02	Baseline	1.0	1.0	0.0	1.0	1.0	1.0	-	-
	Price and Cut	1.01	1.01	34.0	1.04	1.0	1.12	10.0	69
	k-alternation ($k = 1$)	1.49	1.01	28.99	1.25	0.44	0.57	7.5	47
	k-alternation ($k = 10$)	1.02	1.01	27.0	1.26	0.8	0.52	10.0	56
	k-alternation ($k = 50$)	1.01	1.01	34.0	1.09	0.92	1.02	17.5	64
	(α, k)-alternation	1.07	1.01	36.0	1.82	0.35	0.52	7.5	34
DIMACS	Baseline	1.0	1.0	0.0	1.0	1.0	1.0	-	-
	Price and Cut	1.02	1.0	0.0	1.01	1.0	1.0	-	0
	k-alternation ($k = 1$)	1.55	1.0	0.0	1.32	1.0	1.0	-	0
	k-alternation ($k = 10$)	1.03	1.0	0.0	1.02	1.0	1.0	-	0
	k-alternation ($k = 50$)	1.02	1.0	0.0	1.01	1.0	1.0	-	0
	(α, k)-alternation	1.08	1.0	0.0	1.05	1.0	1.0	-	0
comp-DIMACS	Baseline	1.0	1.0	-	1.0	1.0	1.0	-	-
	Price and Cut	1.11	1.0	-	1.05	1.0	0.95	14.29	11
	k-alternation ($k = 1$)	6.57	0.82	-	28.08	0.84	1.13	0.0	13
	k-alternation ($k = 10$)	1.16	0.96	-	2.83	1.01	0.97	14.29	12
	k-alternation ($k = 50$)	1.11	1.0	-	1.11	1.0	1.02	14.29	13
	(α,k)-alternation	6.46	0.96	-	37.45	0.66	0.61	14.29	5

Table 5.1: Results of the different formulations for all instances

SCIP Implementation details: The alternation between the three cutting strategies is achieved by having two SCIP object separators. The first is called when no column is generated at the end of the 3 pricers. This separator permits achieving a classic Price&Cut, that is, launching a cutting round when the pricing objects find no column. The second separator is called by our greedy pricer when the strategy is to alternate; in this case, it will not try to generate a column if a constraint is found.

Table 5.1 shows the results of our different algorithms on the three sets of instances. We do not present the result of the Baseline method as the results depend on the configuration of the computer and implementation details. We focus on the comparison between the Baseline method and the proposed one by giving a ratio between a given statistic for our strategies and the baseline. First, whatever alternation technique we consider or instances, the computation time needed for the root node is always greater than the one of the baseline. Note that the number of columns and computation time at the root node of the Price and Cut is necessarily greater or equal to the number of columns and computation time of the Baseline. This is not necessarily the case when alternating between cutting and pricing. On DIMACS, we can see that no constraint is generated. Perhaps this comes from the high relaxation value on these instances. Hence the Baseline is the best algorithm since all the others spend time separating nothing.

Significant differences on the improvements of the relaxation value: Table 5.1 shows that the relaxation value is a lot improved by Chvátal-Gomory constraints on Color02 instances, and we see that launching a lot of cutting rounds does not necessarily imply a better relaxation value as k-alternation with $k = 1$ closes the gap less than the Price&Cut or k-alternation with ($k = 50$). Still, the alternation that reduces most of the gap is the (α, k)-alternation. Surprisingly, the (α, k)-alternation and k-alternation with $k = 1$ generate fewer constraints than the other alternation procedure. A purely rhetorical explanation can be that the separating MILP generates constraints considering all the columns of the RMP: this implies that when the number of columns is high, it is harder to find the most violated constraint. In counterpart, the most violated constraint generated at the beginning of the column generation scheme may not be the most violated one by the optimal points of the linear relaxation, and may induce complicated optimal points that are harder to separate: this could be true for k-alternation ($k = 1$) but not for the (α, k)-alternation. This explanation seems hard to witness in practice, but we think that it somehow explains some observations...

Reducing the number of columns and branching nodes: The k-alternation with $k = 1, 10$ and (α, k)-alternation generates fewer columns on comp-DIMACS, showing a slight stabilization of the root node computation. When considering the whole branching process for Color02 instances, our alternation strategies, except k-alternation with $k = 50$, lowered the number of generated nodes drastically.

Explosion of the computation time because of the separator for (α, k) and $k = 1$ alternation: Unfortunately, these convergence stabilizations on Color02 did not induce a lower computation time, which explodes. To understand what causes this explosion, one can look at Table 5.2, where we give the repartition of the computation time. The column RMP is calculated as the total computation time minus the computation time of the pricers and separators. On Color02, most of the computation time of k-alternation with $k = 1$ and (α, k)-alternation is spent on the separators.

A hope to achieve faster convergence is to use the so-called DOI which is investigated in the next section.

Instance	Algorithm	CPU pricing	CPU separation	CPU RMP
Color02	Baseline	0.22	-	0.78
	Price and Cut	20	6	74
	k-alternation ($k = 1$)	12	81	7
	k-alternation ($k = 10$)	16	37	048
	k-alternation ($k = 50$)	19	11	7
	(α,k)-alternation	8	81	11
DIMACS	Baseline	99	-	1
	Price and Cut	98	1	1
	k-alternation ($k = 1$)	76	24	1
	k-alternation ($k = 10$)	97	2	1
	k-alternation ($k = 50$)	98	1	1
	(α,k)-alternation	94	5	1
comp-DIMACS	Baseline	65	-	35
	Price and Cut	62	4	34
	k-alternation ($k = 1$)	2	97	1
	k-alternation ($k = 10$)	23	64	13
	k-alternation ($k = 50$)	59	8	33
	(α,k)-alternation	1	99	1

Table 5.2: Repartition of the computation time in %

5.3 Dual optimal inequalities and elementary Chvátal closure

5.3.1 Cutting both primal and dual spaces

For the formulation (P), if T is a stable set whose closed neighborhood is included in the closed neighborhood of a vertex v, then by Theorem 4.5.2 $\chi^T - \chi^v$ is a DOI. Let $\mathcal{D}$ be a set of such pairs and y the associated primal variables. The stabilized formulation (P^{stab}) will then be :

$$\min \sum_{S\in\mathcal{S}} x_S \tag{5.5}$$

$$\sum_{S\in\mathcal{S}|v\in S} x_S - \sum_{(v,T)\in\mathcal{D}} y_{v,T} + \sum_{(w,T)\in\mathcal{D}|v\in T} y_{w,T} \geq 1 \qquad \forall\, v \in V \tag{5.6}$$

$$x_S \geq 0 \qquad \forall S \in \mathcal{S} \tag{5.7}$$

$$y_{w,T} \geq 0 \qquad \forall (w,T) \in \mathcal{D} \tag{5.8}$$

Adding CG constraints to P changes the dual structure by adding variables. Also, the constraint matrix may no longer have the (χ^v, χ^T)-exchange property. It depends on the coefficient of the variables y in the CG constraints. This implies that $\chi^T - \chi^v$ may not be a DOI anymore. We denote by Ψ the submatrix having its column set associated with DOI and its row set the CG constraints.

Let us denote by $\mathcal{C}$ the set of vector multiplier c associated with CG constraints. The resulting stabilized cut program (P_{cut}^{stab}) is as follows:

$$\min \sum_{S\in\mathcal{S}} x_S \tag{5.9}$$

$$\sum_{S\in\mathcal{S}|v\in S} x_S - \sum_{(v,T)\in\mathcal{D}} y_{v,T} + \sum_{(w,T)\in\mathcal{D}|v\in T} y_{w,T} \geq 1 \qquad \forall\, v \in V \tag{5.10}$$

$$\sum_{S\in\mathcal{S}} \lceil c(S) \rceil x_S + \sum_{w,T\in\mathcal{D}} \Psi_{w,T} \times y_{w,T} \geq \lceil \mathbb{1}c \rceil \qquad \forall c \in \mathcal{C} \tag{5.11}$$

$$x_S \geq 0 \qquad \forall S \in \mathcal{S} \tag{5.12}$$

$$y_{w,T} \geq 0 \qquad \forall (w,T) \in \mathcal{D} \tag{5.13}$$

Theoretical analysis of Ψ

In this section, we investigate several values of Ψ.

It is essential to notice that DOI cannot improve the relaxation value; the only hope we can have is to reduce the number of iterations or the CPU time from alternation strategies or finally help our separation procedure to find more constraints by changing its numerical behavior (remember that it is heuristic).

Let us consider $\widetilde{P}$ as defined in Section 5.2.2, the set covering formulation with CG constraints. We denote by d the variables associated with covering constraints and g the variables associated with CG constraints. The dual $\widetilde{D}$ of $\widetilde{P}$ is as follows:

$$(\widetilde{D}) \begin{cases} \max \sum_{v\in V} d_v + \sum_{c\in C} \lceil \mathbb{1}c \rceil g_c & \\ d(S) + \sum_{c\in C} \lceil c(S) \rceil g_c \leq 1 & \forall\, S \in \mathcal{S}, \\ d_v \geq 0 & \forall\, v \in V. \\ g_c \geq 0 & \forall\, c \in C. \end{cases}$$

Theorem 5.3.1. *Let (w,T) be a pair such that T is a stable set and $\overline{N}(T) \subseteq \overline{N}(w)$, the following inequalites are DOI for $\widetilde{P}$:*

(i) $d(T) + \sum_{c\in C} g_c(\lfloor c(T) \rfloor - \lceil c_w \rceil) \leq d_w$

(ii) $d(T) \leq d_w$ *if* $\forall c \in C, c_w \leq c(T)$

Proof. For the first statement, suppose that (d^*, g^*) is optimal for $\widetilde{D}$ and $d^*(T) + \sum_{c\in C} g_c^*(\lfloor c(T) \rfloor - \lceil c_w \rceil) > d_w^*$. Let S be a stable set containing w and $S' = S \setminus \{w\} \cup T$, S' also is a stable set by construction.

Since (d^*, g^*) is a solution of $\widetilde{D}$ we have:

$$1 \geq d^*(S') + \sum_{c\in C} g_c^* \lceil c(S') \rceil = d^*(T) + d^*(S \setminus \{w\}) + \sum_{c\in C} g_c^* \lceil c(T) + c(S \setminus \{w\}) \rceil.$$

Using the hypothesis we get:

$$1 > d^*(S) + \sum_{c \in C} g_c^*(\lceil c(T) + c(S \setminus \{w\}) \rceil + \lceil c_w \rceil - \lfloor c(T) \rfloor),$$

Now we claim that for any $a, b, c \in \mathcal{R}$: $\lceil a + c \rceil \leq \lceil a + b \rceil + \lceil c \rceil - \lfloor b \rfloor$. This comes from the fact that $\lceil a + c \rceil + \lfloor b \rfloor \leq \lceil a + b + c \rceil \leq \lceil a + b \rceil + \lceil c \rceil$.

Now, for a given $c \in C$, by setting $a = c(S \setminus w),\ b = c(T),\ c = c_w$ we obtain

$$\lceil c(S) \rceil \leq \lceil c(T) + c(S \setminus \{w\}) \rceil + \lceil c_w \rceil - \lfloor c(T) \rfloor.$$

Hence, we have:

$$1 > d^*(S) + \sum_{c \in C} g_c^*(\lceil c(T) + c(S \setminus \{w\}) \rceil + \lceil c_w \rceil - \lfloor c(T) \rfloor) \geq d^*(S) + \sum_{c \in C} \lceil c(S) \rceil g_c^*$$

Hence, no constraint of $\widetilde{D}$ having a positive component for d_w is tight, implying that (d^*, g^*) is not optimal.
For the second statement, suppose that all our CG constraints verify $c_w \leq c(T)\ \forall c \in C$.

Suppose that $d(T) > d_w$. We have analogously that:

$$1 \geq d^*(S') + \sum_{c \in C} g_c^* \lceil c(S') \rceil = d^*(T) + d^*(S \setminus \{w\}) + \sum_{c \in C} g_c^* \lceil c(T) + c(S \setminus \{w\}) \rceil$$

Hence, using the hypothesis and the fact that the rounding up function is increasing, we obtain:

$$d^*(T) + d^*(S \setminus \{w\}) + \sum_{c \in C} g_c^* \lceil c(T) + c(S \setminus \{w\}) > d^*(S) + \sum_{c \in C} g_c^* \lceil c(S) \rceil$$

We conclude that no inequality with a positive d_w coefficient is tight. □

We propose two methods using DOI and CG constraints simultaneously:

- DOICHG1 : $\forall (w, T)$ DOI, $\Psi_{w,T} = \lfloor c(T) \rfloor - \lceil c_w \rceil$,
- DOICHG2 : $c_w \leq c(T)$ imposed in the MILP separator (so every constraint satisfies Theorem 5.3.1 ii), and $\Psi = 0$.

Implementation details

Let (x^*, y^*) be a fractional solution we would like to cut.

With DOICHG1, to cut it with CG inequalities, we need two additional variables m_w and m_T for each pair (w, T) associated with a DOI of the primal and four constraints. Let $\mathcal{D}$ be the set of (w, T) associated with DOI added to the primal. The associated MILP is the following. The horizontal line permits to visually show what changed from SEPA (note that the objective function also changes):

$$
(SEPA1)\left\{
\begin{array}{rcll}
\multicolumn{4}{l}{\max \alpha_0 - \sum_{S\in\mathcal{S}^*} x^*_S\alpha_S - \sum_{w,T\in\mathcal{D}} y^*_{w,T}(m_T - m_w)} \\
f_S & = & \alpha_S - u(S) & \forall S \in \mathcal{S}^* \\
f_0 & = & \alpha_0 - \mathbb{1}u & \\
f_S & \geq & 0 & \forall S \in \mathcal{S}^* \\
f_S & \leq & 1-\delta & \forall S \in \mathcal{S}^* \\
u_v & \geq & 0 & \forall v \in V \\
u_v & \leq & 1-\delta & \forall v \in V \\
\alpha_S & \in & \mathbb{Z} & \forall\ S \in \mathcal{S}^* \\
\alpha_0 & \in & \mathbb{Z} & \\
\hline
m_w - u_w & \leq & 1-\delta & \forall (w,T) \in \mathcal{D} \\
m_w & \geq & u_w & \forall (w,T) \in \mathcal{D} \\
m_T & \leq & u(T) & \forall (w,T) \in \mathcal{D} \\
u(T) - m_T & \leq & 1-\delta & \forall (w,T) \in \mathcal{D} \\
u_w & \in & \{0,1\} & \forall (w,T) \in \mathcal{D} \\
u_T & \in & \mathbb{Z} & \forall (w,T) \in \mathcal{D}
\end{array}
\right.
$$

The two first new types of constraints ensure that $m_w = \lceil u_w \rceil$, the two second new types of constraints ensure that $m_T = \lfloor u(T) \rfloor$ and the last two define the variable space.

If one would like to price not only stable sets but also DOI, i.e. adding DOI dynamically, the reduced costs of the CG constraints would have to be taken into account. Still, since adding DOI at the beginning is already reducing the number of iterations significantly, we choose not to implement such a dynamic method as it also seems to harden a lot our framework as we would have to take into consideration the CG constraints while generating new DOIs.

With DOICHG2 We add the constraints $u_w \leq u(T)$ for each DOI associated with (w,T) in the separation MILP.

This implies that generating the DOI (u,T) dynamically should be done taking into account that some CG constraints already generated may not respect $u_w \leq u(T)$.

5.3.2 Preliminary computational experiments

We first decide what type of DOI to add to the formulation and then compare the root node of three formulations: the Baseline to which we added DOI, and two B&C&P with DOI implementing the two solutions proposed in Theorem 5.3.1.

For these experimental results, we only consider the subset of instances from the "Color02" which are solved by an algorithm in less than a day and for which the classic algorithm solves the linear relaxation in less than one hour and more than 0.1 seconds that contains the graph structure needed for the existence of the DOI as done in [73], it consists in 30 instances. The type of DOI we consider is introduced in [73] and corresponds to $\chi^S - \chi^v$ when S is a stable set such that $\overline{N}(S) \subseteq \overline{N}_v$.

Which set of DOI should we generate ?

Since the number of DOI proposed by Gschwind et al. [4] for the coloring problem is exponential, we have to choose a subset of these to add. The intuitive choice would be to consider DOI associated with couples (w, T) where T is a maximum stable set included in the neighborhood of w. Any DOI associated with a proper subset of T would lead to a weaker dual cut. To confirm this intuition, we consider the three following DOI types:

- type 1 : for each vertex w we construct one DOI with a maximum stable set S of its neighborhood such that $N(S) \subseteq \bar{N}_w$
- type 2 : for each vertex w we construct a DOI for each maximal stable sets $S_1, ..., S_n$ such that $\overline{N}(S_i) \subseteq \overline{N}_w$ and $\bigcup S_i = N_w$(these are computed greedily with the algorithm described in Table 1)
- type 3 : for each vertex w, $u \in N_w$ and $N_u \subset \overline{N}_w$ we add the DOI associated with $(w, \{u\})$

Algorithm 1 Greedy Algorithm to compute DOI of type 2

> $sol \leftarrow \emptyset$
> **for** $w \in V$ **do**
> $n_w \leftarrow \{v \in N_w \mid N_v \subseteq \overline{N_w}\}$
>   **for** $v \in n_w$ **do**
> $S \leftarrow \{v\}$
>   **for** $u \in n_w \setminus v$ **do**
>   **if** $N_u \cap S = \emptyset$ **then**
> $S \leftarrow S \cup \{u\}$
>   **end if**
>   **end for**
> $sol \leftarrow sol \cup (w, S)$
>   **end for**
> **end for**
> return sol

Algorithm 1 permits to cover the vertices of N_w by maximal stable sets of N_w, and so for each vertex $w \in V$. Here's the signification of the column titles :

- CPU is the proportion of the sum of resolution time for each instance between the corresponding algorithm and the classic one
- # ite is the proportion of the sum over these instances of the number of iterations between the corresponding algorithm and the classic one
- Price CPU is the proportion of the sum over these instances of the pricing time between the corresponding algorithm and the classic one.

The columns whose name starts with av. correspond to the average over all instances of the associated ratio. According to Table 5.3, even if the DOI of type 2 seems less effective in average CPU time than the DOI of type 1 and 3, it is the fastest in terms of total CPU time and total number of iterations. We will now consider only this type of DOI since it seems the most efficient on hard instances.

Algorithm	av. CPU	av. # Ite	av. Price CPU	CPU	#ite	Price CPU
Baseline	1	1	1	1	1	1
Baseline + DOI type 1	0.73	0.88	0.89	0.93	0.85	0.87
Baseline + DOI type 2	0.78	0.80	1.17	0.85	0.69	0.76
Baseline + DOI type 3	0.81	0.82	1.01	0.95	0.79	0.97

Table 5.3: Experimental results of the three different types of DOI on instances for which it exists DOI and that are solved in more than 0.1 seconds

Branching with DOI

Given a DOI-inducing structure (w, T), it happens that some branching decisions make that structure no longer DOI-inducing. For example, if the branching decision adds an edge between two vertices of T or between a vertex of T and some other vertex of the graph that is not adjacent to w. To address this issue, we propose two methods :

- AIBN (adapt inside branching nodes): in every branching node, we fix every DOI variable incompatible with the associated branching decision to zero (when the associated matrix does not have the associated exchange property because some of the columns of the original do not respect the branching decisions).
- DBB (delete before branching): after computing the first relaxation value, we delete the DOI variable from the primal (and complete the variable set such that no primal iteration has to be completed before branching)

algorithm	CPU	# ite	# nodes
Baseline	1	1	1
AIBN	0.7	0.65	0.34
DBB	1.15	1.08	1.25

Table 5.4: Experimental results for AIBN and DBB DOI + branching strategies

From Table 5.4, we see that the best is to keep the DOI all over the branching scheme and only remove them from the nodes where branching decisions are incompatible. Deleting the DOI before branching is even worse than the Baseline on this set of instances. It seems like having DOI at the beginning of the column generation process perturbates the whole B&P by changing the set of columns generated at the root node, hence implying different optimal solutions for each RMP. From now on, it is AIBN that we consider for all branching schemes.

5.3.3 Experiments using Chvátal-Gomory constraints, alternation and Dual Ooptimal Inequalities

It is essential to note that DOI cannot improve the relaxation value; the only hope we can have is to reduce the number of iterations or the CPU time. We only use DOI of type two, for which we compare the two following techniques suggested by Theorem 5.3.1:

- DOICHG1 : $\forall(w,T)$ DOI, the coefficient is $\lfloor \lambda(T) \rfloor - \lceil \lambda_w \rceil$
- DOICHG2 : $\lambda_w \leq \lambda(T)$ imposed in all separators

For these final experimentations, we consider the sets Color02, DIMACS, and DIMACS-comp containing DOI. Each set is composed of 30, 7, and 6 instances, respectively.

Baseline + DOI is the best in general: In Table 5.5, we see that the result of the Baseline + DOI all over the branching tree also performs significantly better than the Baseline on DIMACS and Color02.

k-alternation ($k = 10$) + DOICHG1 performs well on comp-DIMACS: Baseline + DOI is overall slower on comp-DIMACS for which the fastest algorithm is k-alternation ($k = 10$) + DOICHG1.

Poor relaxation value of DOICHG2: On Color02, none of the B&C&P with DOI and DOICHG2 improve the relaxation value of the Baseline, and the performances of this algorithm looks like the performances of the B&P + DOI in terms of the number of iterations. DOICHG2 does not seem to imply a different algorithm than the BP with DOI.

DOICHG1 yields mitigated relaxation values: Note that using DOICHG1 came with changing the structure of the separator MILP which explains why the DOICHG1 Price&Cut does not manage to close the gap at all when the other alternation, which launched more separation rounds managed to close a part of it. Remember that for numerical stability, the authors of [66] proposed to add an epsilon objective coefficient on some variables of the MILP. Similarly, our variant of this MILP (SEPA1) may be unstable numerically, and this could also explain why launching several separation rounds can help find interesting constraints.

DOICHG1 is the coefficient inducing a combination of primal and dual cuts: The method using both primal and dual constraints that tries to take advantage of both is DOICHG1, as DOICHG2 performs like the Baseline + DOI, but it is not efficient at all since it often comes with an explosion of the computation time.

(α, k)-alternation profits a lot from DOICHG2 On DIMACS and comp-DIMACS, surprisingly the fastest calculation of the root node is done by the

(α, k)-alternation, adding constraints to the separator of the form $u_w \leq u(T)$ may simplify the separator's structure which induces a huge stabilization of the root node calculation. This has more to do with the fact that the separator is heuristic than a real synergy between primal and dual cuts.

Conclusion: What seems clear in each set of instances, is that alternating AND using DOI is worse than the Price and Cut with DOI, which is worse than the Baseline + DOI. This section does not show a significant synergy between dual cuts and primal cuts, this may have to do with our choices on DOICHG1 and DOICHG2. From a more theoretical point of view, while cutting the dual restricts the dual space, adding cuts to the primal extends the dual space this may be the explanation for such poor experimental results.

Conclusion

In this chapter, we proposed a first B&C&P using DOI and alternation strategies and compared several settings and combinations of these techniques to try to improve an existing formulation. We investigated, in particular, how to branch with DOI and different sets of DOI for formulation P_1. All these experiments show that DOIs are a powerful tool that should be used all along the branching tree when solving a formulation. Alternation strategies do not seem to stabilize the column generation for this formulation, still, we do not think that the subject is closed. In fact, given a column generation formulation for which we know stronger cuts than the one we used for our problem and for which a fast separation exists, as (α, k)-alternation managed to close significantly the gap while generating fewer constraints than the other alternation strategies one can expect this alternation to work better than the Baseline.

A possible extension would be to reduce the separator's computation time which could be achieved by lowering the limit on its maximal number of branching nodes or also adding a minimal violation on a cut as a constraint of the separation MILP and adding the first Chvátal-Gomory constraint found (following the idea we used for the exact pricing procedure). We will also investigate another combination of primal and dual cuts in the next section for the 2-defective coloring.

		root node			Complete Branching				
Instances set	coeff	alg	CPU	# col	% gap	CPU	# col	# node	% faster
Color02	-	Baseline	1.0	1.0	-	1.0	1.0	1.0	0.0
	-	Baseline + DOI	0.85	0.69	-	0.7	0.65	0.34	78.57
	-	Price and Cut	1.1	1.01	53.0	11.8	1.89	4.78	7.14
	DOICHG1	Price and Cut	3.34	0.95	0.0	0.68	0.91	1.0	42.86
		k-alternation ($k = 1$)	9.06	0.74	62.0	4.87	0.78	1.14	14.29
		k-alternation ($k = 10$)	1.42	0.69	64.0	49.61	1.56	6.01	75.0
		(α, k)-alternation	20.93	0.71	35.0	23.84	0.71	0.64	39.29
	DOICHG2	Price and Cut	0.93	0.69	0.0	0.69	0.55	0.28	71.43
		k-alternation ($k = 1$)	1.42	0.69	0.0	1.05	0.57	0.32	39.29
		k-alternation ($k = 10$)	0.92	0.69	0.0	1.45	0.75	0.93	75.0
		(α, k)-alternation	0.93	0.65	0.0	2.28	0.59	0.34	53.57
DIMACS	-	Baseline	1.0	1.0	-	1.0	1.0	1.0	0.0
	-	Baseline + DOI	0.62	0.91	-	0.61	0.91	1.0	57.14
	-	Price and Cut	1.08	1.0	0.0	0.99	1.0	1.0	14.29
	DOICHG1	Price and Cut	3.32	0.76	0.0	0.68	0.91	1.0	42.86
		k-alternation ($k = 1$)	117.47	0.95	0.0	118.67	0.95	1.0	0.0
		k-alternation ($k = 10$)	3.36	0.94	0.0	1.46	0.93	1.0	28.57
		(α, k)-alternation	5.89	0.96	0.0	5.75	0.96	1.0	28.57
	DOICHG2	Price and Cut	0.66	0.91	-	0.67	0.91	1.0	28.57
		k-alternation ($k = 1$)	0.81	0.91	0.0	0.76	0.91	1.0	28.57
		k-alternation ($k = 10$)	0.68	0.91	0.0	0.68	0.91	1.0	14.29
		(α, k)-alternation	0.03	0.19	0.0	0.78	0.91	1.0	28.57
comp-DIMACS	-	Baseline	1.0	1.0	-	1.0	1.0	1.0	0.0
	-	Baseline + DOI	0.82	0.89	-	1.15	0.99	1.85	50.0
	-	Price and Cut	1.08	1.0	-	0.99	1.0	1.0	14.29
	DOICHG1	Price and Cut	1.04	0.73	-	2.67	1.2	2.3	33.33
		k-alternation ($k = 1$)	6.01	0.69	-	64.13	1.16	0.79	0.0
		k-alternation ($k = 10$)	1.11	0.91	-	0.31	0.39	0.65	50.0
		(α, k)-alternation	14.62	0.67	-	150.59	1.05	0.78	0.0
	DOICHG2	Price and Cut	1.33	1.01	-	2.35	1.66	2.04	16.67
		k-alternation ($k = 1$)	4.65	0.71	-	16.91	0.86	0.93	0.0
		k-alternation ($k = 10$)	1.32	1.01	-	2.82	1.56	0.91	16.67
		(α, k)-alternation	0.1	0.12	-	224.38	1.55	3.43	16.67

Table 5.5: Experimental result of the B&C&P with CG and DOI

Chapter 6

2-Defective coloring

This chapter aims to solve the co-2-plex partitioning problem, that is, to partition the vertex set of a graph with the minimum number of co-2-plexes. We introduce triangle constraints to reinforce the formulation. Secondly, we propose a set of *dual inequalities*, which are inequalities cutting the dual. These dual constraints are added to the primal as variables and help stabilize the column generation. Thirdly, we investigate how to use both techniques simultaneously. Each section will end with a set of computational experiments.

6.1 Covering by co-2-plexes polytope

In this section, we give simple corollaries of the results on the set covering polytope.

Corollary 6.1.1. *Let G be a connected graph, then, its 2-defective polytope is full dimensional if and only if G is not an edge or an isolated vertex.*

Proof. A vertex v of a graph G is contained in only one maximal co-2-plex S if and only if S contains all the vertices of G. Theorem 4.4.1 ends the proof. □

6.2 Experimental results of the baseline algorithm

6.2.1 Description of the baseline algorithm

The baseline algorithm corresponds to the algorithm of Furini et al. [2], see Section 4.1.2 for a detailed description of this algorithm. Two main features have been added for $k = 2$ to this baseline.

The first change that we tried using the exact pricer 'P' described in Chapter 3. Indeed, this algorithm challenges the algorithm 'N' proposed in [2].

The second change is a greedy pricing heuristic consisting in ordering the vertices and constructing greedily a co-2-plex with the vertices of higher costs.

When branching, the greedy heuristic is then adapted as follows. Consider for each vertex w, the list of vertices $L_w^{\cup}$ (resp. $L_w^{\cap}$) that have to be contained (resp. not contained) in the same co-2-plex as w according to all the branching decisions that correspond to the current branching node. Within the greedy heuristic procedure, every time a vertex w is added, the list $L_w^{\cup}$ is added, and the list $L_w^{\cap}$ is removed from the remaining candidates. At this point, it must be verified that the obtained set is still a co-2-plex of the graph.

6.2.2 Instances

We now introduce the instance sets used in this chapter to compare our different methods. To characterize the instances we will use the notion of dominance graph structure introduced in Definition 6.4.1. Indeed, if a graph has such a structure, we can use the dual inequalities (DI) defined in Section 6.4.

The instances come from two sources. We take the 30 instances from the vertex coloring benchmarks [59], called Color02, that our baseline BP has solved in less than three hours, and that for which the coloring problem is solved in less than an hour by some exact algorithm. This benchmark is a classical reference for partitioning problems. In [2], the authors proposed a BP framework to solve several partitioning problems to detect communities inside social networks. Their experiments are based on 9 instances from the 10th DIMACS Implementation challenge, called DIMACS. We take the 7 instances among these 9 ones that our baseline BP has solved in less than three hours and that have the dominance graph structure. We also consider the complement of the latter 7 instances, one of them does not contain the graph structure, and another is not solved in less than 3 hours, we do not consider ones.

Our 42 instances can be partitioned into 5 subsets to focus on the abilities of the proposed methods. The Color2 instances are first divided into two sets, depending on whether they have the dominance graph structure indexed by the letter D. The set having the dominance graph structure is then divided into two sets of balanced size depending on the computation time needed by the baseline BP: easy ones are indexed by the letter E and hard ones by the letter H. The resulting subsets are then:

- E-Color: 10 instances of Color02 not containing the dominance graph structure
- ED-Color: 10 instances of Color02 containing the dominance graph structure and taking the least computation time
- HD-Color: 10 instances of Color02 containing the graph dominance structure and taking the most computation time
- DIMACS: 7 instances from DIMACS containing the graph structure
- comp-DIMACS: complements of the 5 DIMACS
- others: 5 instances from the set color02 that the baseline did not solve

We tried to generate random graphs with respectively 70, 80, 85, and 90 vertices and a probability of 0.5, 0.7, and 0.9 for each edge, with 10 graphs for each combination. None of these contained the dominance graph structure. Hence, we do not consider them. A graph generated with this method seems to have few chances of containing such a structure. We were able to generate instances with the dominance structure with density 8%, the results on these instances will be discussed.

6.2.3 Experimental results of the baseline

We ran the experiments on a computer with processor intel core i5, 3.10 GHz, and 15.3 GB RAM. We used SCIP 7.0.3 for the column generation framework and CPLEX for the RMP resolutions and the two pricing heuristics using MILP.

Impact of formulation P: a preliminary experimental comparison has shown that pricing with formulation 'P' did not work better than using 'N' as the exact pricer. As said in Chapter 3, formulation 'P' works much better on random instances and DIMACS-comp but is slower on Color02 which composes most of the instance set of this Chapter. Tackling the problem of covering a graph with co-2-plexes using formulation (P) implies solving several maximum co-2-plex problems which implies solving 'easy' instances of maximum co-2-plex.

In Table 6.1, we present the results of the baseline BP on the 6 instance subsets. The entries in this table are as follows:

- CPU (h)	total solving time in hours,
- #nodes	total number of branching nodes,
- #col	total number of generated columns (at most one per iteration),
- M CPU p.it (ms)	total time spent solving RMP per iteration in milliseconds,
- SP CPU p.it (ms)	total time spent per iteration by the three pricers in milliseconds.

Instances	CPU	# col	# nodes	M CPU p.it	SP CPU p.it	% pricing
E-Color	0.028	7829	130	2.58	10.14	79.74
ED-Color	0.007	4322	88	1.16	4.24	78.52
HD-Color	7.605	201848	3169	116.04	17.5	13.1
DIMACS	0.546	5787	646	69.18	236.44	77.36
comp-DIMACS	5.626	189841	592	92.61	13.75	12.93
others	-	-	-	-	-	-

Table 6.1: Performances of the baseline BP

The number of iterations is then #col + #nodes, corresponding to the number of solved RMPs.

In Table **??** , we have the computational performances of our baseline. It is not significant when given alone. However, we will compare the performances of our algorithms by giving the ratio between a given statistic for our algorithm and that same statistic for the baseline.

The sets of instances we would like to solve are HD-color and comp-DIMACS since they are the hardest ones for the baseline. They are the sets of instances for which we generated the most branching nodes. We can see that in these sets of instances, most of the time is spent resolving master problems, this is certainly due to the huge number of generated columns and hence a witness of the tailing-off effect. In the sequel, we will try to stabilize by cutting both in primal and dual spaces.

6.3 Cutting the primal space

In this section, we introduce a new type of primal cuts associated with triangles of the graph and describe how we used them in the column generation framework.

6.3.1 Triangle inequalities

Given a *triangle* $K = \{u, v, w\}$ that is a clique of size 3 of the graph G, we need at least two co-2-plexes to cover it, which leads to the following triangle inequality:

$$\sum_{S \in \mathcal{S}_G | K \cap S \neq \emptyset} x_S \geq 2$$

Let $\mathcal{T}$ be the set of triangles associated with a set of separated triangle inequalities. We denote by $P_{\mathcal{T}}$ the LP obtained from P by adding triangle inequalities:

$$\min \sum_{S \in \mathcal{S}_G} x_S \tag{6.1}$$

$$\sum_{S \in \mathcal{S}_G | v \in S} x_S \geq 1 \qquad \forall\ v \in V \tag{6.2}$$

$$\sum_{S \in \mathcal{S}_G | S \cap K \neq \emptyset} x_S \geq 2 \qquad \forall K \in \mathcal{T} \tag{6.3}$$

$$x_S \geq 0 \qquad "\forall\ S \in \mathcal{S}_G \tag{6.4}$$

Now, we must consider the new dual variables μ_K, for each $K \in \mathcal{T}$ and adapt our pricing procedures.

Triangle inequalities are *non-robust* cuts to the primal, they are cuts changing the subproblem structure, to keep fast pricing heuristics, we will end up with two different MILPs: one for the heuristic root node pricer and the other for the exact pricer.

The new reduced cost of a variable associated with $S \in \mathcal{S}_G$ is:

$$c_S(\lambda) = 1 - \sum_{v \in S} \lambda_v - \sum_{K \in \mathcal{T} | K \cap S \neq \emptyset} \mu_K$$

Note that, the sum over triangles is not linear with respect to the vertices of S. As described in Chapter 5, we use two pricing heuristics and an exact one. For the two heuristic pricers, we propose a heuristic linearization:

$$\bar{c}_S(\lambda) = 1 - \sum_{v \in S} (\lambda_v + \frac{1}{2} \sum_{K \in \mathcal{T} | v \in K} \mu_K)$$

Now, the two pricing heuristics take the associated modified costs on the vertices, that is, for a vertex v:

$$\lambda_v + \frac{1}{2} \sum_{K \in \mathcal{T} | v \in K} \mu_K$$

Note that triangle constraints belong to the Chvátal elementary closure of (P_2), obtainable by multiplying the constraints associated with the vertices of a triangle by 0.5, summing them, and up-rounding every coefficient. For a set of elementary Chvátal closure constraints associated with coefficient $g \in \mathcal{G}$ and whose associated dual variable would be π_g, it would be natural to linearize the reduced cost of a variable the following way: $\bar{c}_S(\lambda) = 1 - \sum\limits_{v \in S} (\lambda_v + \sum_{g \in \mathcal{G}} \lambda \times \pi_g)$, this is where comes from the coefficient $\frac{1}{2}$ in $\bar{c}$. For the case of the exact pricing procedure, the new MILP now takes one binary variable ρ_K and one constraint for each $K \in \mathcal{T}$. This leads to the following formulation:

$$\max \sum_{v \in V} \lambda_v \pi_v + \sum_{K \in \mathcal{T}} \rho_K \times \mu_K$$

$$\begin{aligned}
(|N_u| - 1)\pi_u + \pi(N_u) \leq & \quad |N_u| \forall\ u \in V, \\
\rho_K \leq \pi(K) & \quad \forall K \in \mathcal{T} \\
\pi_v \in \{0, 1\} & \quad \forall\ v \in V, \\
\rho_K \in \{0, 1\} & \quad \forall\ K \in \mathcal{T},
\end{aligned}$$

6.3.2 Computational results

We perform our experiment with an exact separation procedure, enumerating the constraints associated with the triangles of the graph and adding the five first violated ones at each separation round. In the following table, the multicolumn 'Ratio' contains the ratio between a BCP's statistic divided by the baseline algorithm's corresponding statistic. For example, a value lower than 1 for the CPU means that the BCP is faster than the baseline. The following list gives the signification of the column titles in our tables:

- CPU root the time needed to solve the root node
-% sepa the proportion of solving time spent separating the constraints
-# cons the total number of constraints generated
- # faster the pourcentage of instances solved faster than the baseline algorithm

The triangle constraints do not improve the root node relaxation value, see Table 6.2, where we list the only instances for which adding triangle constraints improves the relaxation value. Preliminary experiments showed that separating the Chvátal elementary closure with the heuristic MILP proposed in [12] gives a lower relaxation value than when separating exactly triangle constraints. This is surprising since triangle constraints belong to the elementary Chvátal closure but understandable since the separation procedure proposed in [12] is a heuristic as it is MILP-based with a limit to the number of branching nodes.

Instances set	Name	baseline	BCP
HD-color	david	5.6304	5.6666
	mulsol.i.2	15.7667	16.0
	mulsol.i.3	15.7667	16.0
	mulsol.i.4	15.7667	16.0
	mulsol.i.5	15.7667	16.0
DIMACS	dolphins	17.9943	18.0
	jazz	47.5393	47.5421

Table 6.2: Linear relaxation value of the BCP

Differences over the solved instances. The BCP did not solve two instances that the baseline solved: zeroin.i.3 and fpsol2.i.1. The baseline did not solve three instances that the BCP managed to solve: mulsol.i.3 (0.6h) and mulsol.i.5 (1.4h). For the first of these instances, the relaxation value went from 15.7667 with the baseline to 16 with triangle constraints. On the other, both LPs have the same relaxation value.

	Ratio					Flat			
Instances	CPU	# col	# nodes	M CPU p.it	SP CPU p.it	% pricing	% sepa	# cons	# faster
E-Color	1.07	0.99	1.09	1.03	1.1	80.7	$< 10^{-2}$	2	0
ED-color	1.26	0.9	0.56	2.78	1.02	57.34	0.03	24	30
HD-color	0.92	0.84	0.69	1.09	1.16	13.85	$< 10^{-2}$	306	60
DIMACS	1.05	0.84	0.27	3.9	0.59	33.93	0.06	27	28
comp-DIMACS	1.15	1.09	1.31	1.04	1.12	28.57	$< 10^{-2}$	29	0
others	-	-	-	-	-	-	-	-	40

Table 6.3: Performances of the BCP

Triangle inequalities only speed up HD-color's resolution: In Table 6.3, we see that unless for the HD-color where we manage to solve faster 6 instances, which corresponds to a reduction of the total computation time by 8%, for the other instances, separating triangle constraints always increases the total computation time.

Triangle constraints do not harden pricing subproblems significantly: Adding cuts usually hardens the average master problem and the average pricing subproblem, but on the set DIMACS the average pricing subproblem gets easier and for the others, it remains comparable. Unless for comp-DIMACS and E-color, the number of iterations is still much lowered, which may increase the average RMP computation times and pricing. This augmentation of computation time does not come from the separation procedure since the latter always represents less than 1% of the total computation time.

For the set E-color, some triangle constraints are generated on only one instance, and it adds 25% of computation time for this instance.

The triangle constraints become useful later in the branching tree: The only set of instances where adding triangle constraints reduces the total computation time is the one for which adding triangle constraints improves the relaxation value in most instances. This improvement is insignificant since it does not permit inferring a higher lower bound at the root node. It may be explained by the fact that these constraints become useful after a few branching decisions.

6.4 Cutting the dual space

In this section, we adapt the DOI introduced by Gschwind et al. for the coloring problem to the co-2-plex covering problem. We introduce a set of extra columns in the co-2-plex covering formulation (P) but show that they are not DOI, that is, adding these inequalities may change the optimum of its linear relaxation. We will in the sequel call such new columns, *dual inequalities* (DI). However, we prove that for any optimal integer point of such obtained formulation, an optimal solution without these extra columns, and hence a solution of P may be recovered in polynomial time.

6.4.1 Dual inequalities

The DIs we will define in this section depend on a structure similar to the one given in [4]. Unfortunately, the constraint matrix of (P) does not have the exchange property as defined in [4]. As we will show, this structure implies that all optimal dual points can be cut off by adding such DI. As shown in the previous section, it seems hard to improve the relaxation value, particularly by heuristically separating the elementary Chvátal closure or triangle constraints. The hope is that adding these DIs would not decrease the relaxation value. Moreover, since these inequalities can cut all optimal dual points in some cases, we hope a better stabilization of the column generation by cutting 'significantly' the dual space. We first give a definition of the graph structure we will need to introduce our DIs.

Definition 6.4.1. *A pair (w, T) with $w \in V$ and $T \subseteq V \setminus w$ has the dominance property if:*

(i) $\overline{N}(T)$ is included in the neighborhood of $\overline{N}_w$

(ii) $\forall u, v \in T$, $\overline{N}_u \cap \overline{N}_v = \{w\}$

A graph having the dominance graph property contains at least one such pair.

We say that a dominance couple (v, T) is *maximal* if T is inclusion wise maximal with the above properties. Note that condition *(ii)* implies that T is a stable set.

Theorem 6.4.2. *If (w, T) is a dominance couple and S a co-k-plex containing w, then $S \cup T \setminus w$ is a co-k-plex.*

Proof. Given a co-k-plex S containing w, every vertex of $S \setminus w$ non-adjacent to w is not adjacent to T by condition *(i)*. Similarly, every vertex of $S \setminus w$ adjacent to w is adjacent to at most one vertex of T by condition *(ii)*. Also, for all $u \in T$, $N_u \cap S \subseteq N_w \cap S$ whose size is already at most $k-1$ since S is a co-k-plex. Hence, every vertex of $G[S \cup T \setminus w]$ has degree at most $k-1$, which implies that $S \cup T \setminus w$ is a co-k-plex of G. □

Since S may already contain vertices of T, the exchange property does not hold. That is, $\chi^S + \chi^T - \chi^w$ may not be the incidence vector of a co-2-plex since it may not be binary, nevertheless $\chi^S \cup \chi^T \setminus \chi^w$ is the incidence vector of a co-k-plex.

We will use a dominance couple (w, T) by adding a new column $\chi^T - \chi^w$ with objective coefficient 0, where χ^T (resp. χ^w) is the incidence vector of T (res. w). Let $\mathcal{D}$ be a set of couples (w, T) having the dominance property and $y_{w,T}$ their associated primal variables. The variables y associated with $\mathcal{D}$ will be called Dual Inequalities (DI). Hence the MILP we want to solve is the following $(\widetilde{P})$:

$$\min \sum_{S \in \mathcal{S}_G} x_S \tag{6.5}$$

$$\sum_{S \in \mathcal{S}_G | v \in S} x_S - \sum_{(v,T) \in D} y_{v,T} + \sum_{(w,T) \in \mathcal{D} | v \in T} y_{w,T} \geq 1 \qquad \forall\, v \in V \tag{6.6}$$

$$x_S \geq 0 \qquad \forall\, S \in \mathcal{S}_G \tag{6.7}$$

$$y_{w,T} \geq 0 \qquad \forall (w, T) \in \mathcal{D} \tag{6.8}$$

6.4.2 A Dual Inequality that cuts every dual optimal solutions

In this section, we will see that DIs with stable sets of size 1 are Deep Dual Optimal Inequalities (DDOI), that is, inequalities that do not cut every dual optimal solution, introduced in [74]. However, in the general case, a DI can cut all the dual optimal solutions.

Dual inequalities with stable sets of size 1

Let us denote by $\widetilde{P}_1$, the restrictions of $\widetilde{P}$ to a set of inequalities associated with nonmaximal dominance couples (w, T) satisfying $|T| = 1$, that is when the stable set T is of size 1.

Theorem 6.4.3. *From any fractional optimal solution $(\widetilde{x}, \widetilde{y})$ of $\widetilde{P}_1$, there exists a fractional solution x of P with a lower or equal objective value.*

Proof. Starting from $(x^0, y^0) = (\widetilde{x}, \widetilde{y})$, we build a sequence of solutions $(x^k, y^k),\ k = 0, \dots, N$ of $\widetilde{P}$ such that $\mathbb{1}y^{k+1} < \mathbb{1}y^k$ and $y^N = 0$.

Given (x^k, y^k), let G^k be the directed graph having V as vertex set and having an arc (u, v) if and only if there exists $(u, T) \in \mathcal{D}$ such that $y_{u,T} \neq 0$ and $v \in T$. Suppose that G^k contains a directed circuit $C = (w_1, w_2, \dots, w_p)$. Every arc (w_i, w_{i+1}) of that circuit is associated to the dominance couple $(w_i, \{w_{i+1}\})$. Now let us consider $m = \min_{i=1,\dots,p} y^k_{i,\{i+1\}}$ where $p + 1 = 1$. Let us set
$x^{k+1} = x^k$,

$$y^{k+1} = \begin{cases} y^k_{v,T} & \text{if } (v,T) \neq (w_i, \{w_{i+1}\}) \quad \forall i = 1, \dots, p \\ y^k_{w_i,w_{i+1}} - m & \text{otherwise} \end{cases}$$

Note that since the only composants of y^{k+1} different from the composants of y^k are associated with edges of $E(C)$, we have $G^k[V \setminus C] = G^{k+1}[V \setminus C]$. The point (x^{k+1}, y^{k+1}) verifies the inequalities of $\widetilde{P}$ associated with vertices of C and has the same objective value than (x^k, y^k). It also verifies $\mathbb{1}y^{k+1} < \mathbb{1}y^k$. The number of edges in a circuit is strictly lower in G^{k+1} than in G^k. The number of such edges is bounded by $|V|^2$, since it also bounds the number of DI (w, T) where $|T| = 1$. This results in a polynomial time algorithm (in the size of G) computing a solution (x^l, y^l) for which G^l does not contain any cycles.

Let us consider a non isolated leaf v of G^k and one of its predecessor u. First, let us suppose that there exists a dominance couple $(w, \{u\})$ such that $y^k_{w,\{u\}} \neq 0$. In that case, let us set $m = min\{y^k_{u,\{v\}}, y^k_{w,\{u\}}\}$ and

$$y^{k+1} = \begin{cases} y^k_{u,\{v\}} - m & \text{on component } (u, \{v\})) \\ y^k_{w,\{u\}} - m & \text{on component } (w, \{u\}) \\ y^k_{w,\{v\}} + m & \text{on component } (w, \{v\}) \\ y^k_{w,T} & \text{otherwise} \end{cases}$$

and $x^{k+1} = x^k$. Hence (x^{k+1}, y^{k+1}) is a solution of $\widetilde{P}_1$ where $\mathbb{1}y^{k+1} < \mathbb{1}y^k$. After applying recursively this procedure, we obtain a graph G^l with no path of size 2. Let us consider a non isolated leaf v of G^l and one of its predecessor u.
• If $\sum_{S \in \mathcal{S} \ | \ u \in S, v \notin S} x^l_S \geq y^l_{u,\{v\}}$, it is possible to build x^{l+1} by reducing the value of some positive components of x^l associated with co-2-plexes containing u but not v and by augmenting some components associated with co-2-plexes containing

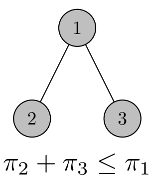


Figure 6.1: A dual inequality cutting all optimal points of the dual

v but not u such that the sum of the component associated with the latter co-2-plexes sum to 1, and by setting $y^{l+1}_{u,\{v\}}$ to 0.

- If $\sum_{S \in \mathcal{S} \ | \ u \in S, v \notin S} x^l_S < y^l_{u,\{v\}}$, by construction, since x^l is a solution we have

$$\sum_{S \in \mathcal{S} \ | \ u \in S, v \notin S} x^l_S + \sum_{S \in \mathcal{S} \ | \ u \in S, v \in S} x^l_S - \sum_{(u,\{w\}) \in \mathcal{D}, w \neq v} y^l_{u,\{w\}} - y^l_{u,\{v\}} \geq 1$$

and hence $\sum_{S \in \mathcal{S} \ | \ u \in S, v \in S} x^l_S > 1$, in which case $y^{l+1}_{u,\{v\}}$ can be set to 0 and all the other component of (x^{l+1}, y^{l+1}) can be equal to the corresponding component of (x^l, y^l) in order for (x^{l+1}, y^{l+1}) to be a solution of $\widetilde{P}$. □

Note that Theorem 6.4.3 shows that the inequalities associated with dominance couples (w, T) satisfying $|T| = 1$ are DDOI.

Dual inequalities with stable sets of at least 2 vertices

When their stable set contains two vertices, the following counter example shows that a DI can cut all optimal points of the dual. See Figure 6.1, where the dominance property couple associated with the dual inequality is $(1, \{2, 3\})$. The associated inequality cuts the only dual optimal point $\pi_1 = \pi_2 = \pi_3 = \frac{1}{2}$. Hence, the linear relaxation of $\widetilde{P}$ which is $\frac{4}{3}$ with the solution $x_{1,2} = x_{1,3} = \frac{2}{3}$ $y = \frac{1}{3}$ becomes strictly lower than the linear relaxation of P which is $\frac{3}{2}$ with the point $x_{1,2} = x_{1,3} = x_{2,3} = \frac{1}{2}$. This phenomenon only happens with dominance couple (w, T) where T is not a singleton. We also wonder if only adding DI associated with dominance couple $(w, \{v\})$ would stabilize the column generation. This DI would not necessarily be maximal and, hence, would be dominated by maximal ones when seen as constraints of the dual.

Therefore, program $(\tilde{P})$ can produce a solution (x^*, y^*) where x^* is not a solution of P. Fortunately, we present in the next section that we can construct in polynomial time an integer solution for P from any integer solution of $\tilde{P}$.

6.4.3 Pricing with Dual Inequalities

The following theorem tells that given an optimal integer solution (x^*, y^*) of $\widetilde{P}$ where $y^* > 0$, it is possible to reconstruct another integer solution $(x', 0)$ with the same optimal value.

Theorem 6.4.4. *From any integer optimal solution $(\widetilde{x}, \widetilde{y})$ of $\widetilde{P}$, an optimal solution of P with the same objective value can be recovered in polynomial time.*

Proof. Starting from $(x^0, y^0) = (\widetilde{x}, \widetilde{y})$, we build a sequence of optimal solutions (x^k, y^k) $k = 0, \ldots, N$ of $\widetilde{P}$ such that $\mathbb{1}y^{k+1} < \mathbb{1}y^k$ and $y^N = 0$, where N is polynomial in the size of G.

Given (x^k, y^k), let G^k be the directed graph having V as vertex set and having an arc (u, v) if and only if there exists $(u, T) \in \mathcal{D}$ such that $y_{u,T} \neq 0$ and $v \in T$.

Analogously to the proof of Theorem 6.4.4 one can suppose that G^k does not contain any directed circuit. Hence, suppose that G^k does not contain any circuit. Then, there exists a vertex v whose successors have no successors. Let (v, T_v) be a couple associated with an outgoing edge of v. If there exists a co-2-plex S_v containing v such that $x_{S_v} \geq 1$, by Theorem 6.4.2, $S_{T_v} = S_v \cup T_v \setminus v$ is a co-2-plex. Hence, let us set

$$x^{k+1} = \begin{cases} 1 & \text{for component } S_{T_v}, \\ x^k_{S_v} - 1 & \text{for component } S_v, \\ x^k_S & \text{otherwise} \end{cases}$$

and $y^{k+1} = \begin{cases} 0 & \text{for component } (v, T_v) \\ y^k_{w,U} & \text{for component } (w, U \setminus T_v) \text{ where } (w, U) \neq (v, T_v). \end{cases}$

Now, G^{k+1} contains $|T_v|$ more isolated vertices than G^k.

If every co-2-plex S containing v is such that $x^k_S = 0$. For (x^k, y^k) to be a solution of $\widetilde{P}$, there must exist an ongoing edge to v in G^k associated with, say (u, U) where $v \in U$. By definition of DI, for every vertex $w \in T$, $\overline{N}(w) \subseteq \overline{N}(v) \subseteq \overline{N}(u)$, and $wu \in E$ which necessarily implies that $T = \{w\}$, otherwise two vertices of T share at least two common neighbors which would contradict the DI definition. We show that $(u, U \cup w \setminus v)$ is a dominance couple. Since U is a stable set and $\overline{N}_w \subseteq \overline{N}_v$, $U \cup w \setminus v$ also is a stable set, and for every $a \in U \setminus v$, $\overline{N}(a) \cap \overline{N}(w) = \{u\}$ since $\overline{N}(a) \cap \overline{N}(v) = \{u\}$ and $\{u\} \subseteq \overline{N}(w) \subseteq \overline{N}(v)$. Let us set $x^{k+1} = x^k$ and $y^{k+1} = \begin{cases} 0 & \text{for component } (v, T) \\ y^k_{u,U} - 1 & \text{for component } (u, U) \\ y^k_{u,U \cup w \setminus v} + 1 & \text{for component } (u, U \cup w \setminus v) \\ y^k & \text{for the other components} \end{cases}$

Again, (x^{k+1}, y^{k+1}) is a solution of $\widetilde{P}$ for which $\mathbb{1}y^{k+1} < \mathbb{1}y^k$. This procedure is polynomial and can be applied at most $|V|^2$ times to get a solution (x^q, y^q) for which every connected component of G^q is a star whose center is contained in many different co-2-plexes S such that $x_S \geq 1$, now the previous procedure can be applied. This results in a polynomial time algorithm. $\square$

Theorem 6.4.4 implies that, even if adding these DIs extends the primal space, it does not add integer solutions with a lower objective value than the optimal solution of P.

Note that by Theorem 6.4.2, one could also use such DIs to solve the partitioning of a graph into co-k-plex for any $k \in \mathbb{Z}^+$.

However, whereas preserving the relaxation value was originally the idea behind adding DOIs, adding DIs should deteriorate the relaxation value. But note that, since adding variables can only reduce the primal's lower bound, the relaxation value of $\widetilde{P}$ remains a valid lower bound for P. Then the hope is that adding DIs can be sufficiently interesting in term of speed up ingredient, such that their bad impact over relaxation would not be significant.

6.5 Dealing with non-integral optimal solutions

In this section, we will see that, surprisingly, solution (x^*, y^*) of $\widetilde{P}$ for which $\sum_{S \in \mathcal{S}} x^*_S$ is equal to the optimum of P, even when x^* is integer, y^* is not necessarily integer. Note that this is important to be able to apply Theorem 6.4.4. Fortunately we present how to obtain such a integer point using dedicated Branching Rules.

6.5.1 Structure of the optimal solutions

We describe a family of graphs and the associated extreme point (x^*, y^*) where x^* is integer and y^* factional. This graph has 5 stages:

- The first stage only contains the vertex u,
- The second stage is a stable set of vertices $v_{ij},\ i \in \{1, \dots, n\},\ j \in \{1, \dots, m\}$, where n is odd and m is even, and each of its vertices is adjacent to vertex u,
- The third stage is a clique of vertices $w_k,\ k \in \{1, \dots, \frac{n(n-3)}{2}\}$. All w_k are adjacent to u. For each $j \in \{1, \dots, m\}$, for all $i \in \{1, \dots, n\}$ the vertex $v_{i,j}$ has exactly one common neighbor of the third stage with $v_{i+l,j}$, where $n+1 = 1$ and $l \in \{2, \dots, n-1\}$. Also, each vertex of the third stage has exactly two neighbors in each set $S_j = \{v_{1,j}, \dots, v_{n,j}\}$. This construction permits that $(u, \{v_{i,j}, v_{i+1,j}\})$ is a maximal dominance couple.
- The fourth stage is composed of one vertex z adjacent to u and to each w_k
- The fifth stage is composed of vertices $t_k,\ k \in \{1, \dots, \frac{n(n-3)}{2}\}$, where each t_k is adjacent to z and adjacent to all vertices of the third stage except w_k

The sets $\{u, w_k, t_k\}$ $\forall k \in \{1, \dots, \frac{n(n-3)}{2}\}$ form maximal co-2-plexes and $\{u, z\}$ is also a maximal co-2-plex. Every $(u, \{v_{ij}, v_{i+1j}\})$ forms a dominance couple where $\{v_{ij}, v_{i+1j}\}$ is inclusion wise maximal. By setting $x^*_{\{u,w_k,t_k\}} =$

$1\ \forall k \in \{1,\ldots,\frac{n(n-3)}{2}\}$, $x^*_{\{u,z\}} = 1$ and $y^*_{u,\{v_{ij},v_{i+1j}\}} = 0.5\ \forall i \in \{1,\ldots,n\},\ j \in \{1,\ldots,m\}$, we obtain that (x^*,y^*) is a solution of $(\widetilde{P})$ with same optimal value as its integer optimal since the three last stages form a 2-plex of size $n(n-3)+1$, which needs at least $\frac{n(n-3)}{2}+1$ co-2-plexes to be covered. Since the DIs we described have 0 coefficients in the rows associated with the three last stages, the relaxation value $(\widetilde{P})$ restricted to the constraint submatrix associated with these stages is a lower bound of the relaxation value of $\widetilde{P}$ and of P.

The relaxation value of $\widetilde{P}$ restricted to the constraint submatrix associated with these stages being $\frac{n(n-3)+1}{2}$ which is equal to the relaxation of P, we know that the relaxation of $\widetilde{P}$ on this graph is equal to $\frac{n(n-3)+1}{2}$.

We now give an example with $m = 2$ and $n = 3$. It does not completely represent the above construction since $\frac{3(3-3)}{2} = 0$, which should lead to an empty third stage, but in this case, the three last stages would not form a sufficiently big 2-plex for (x^*,y^*) to have the same objective value as the integer optimal point. Also, the dominance couples we will exhibit will not be maximal. Any example with $n \geq 5$ would satisfy these properties but be too huge for a clear comprehension. We assume that the upcoming example is sufficient to understand the idea behind the construction.

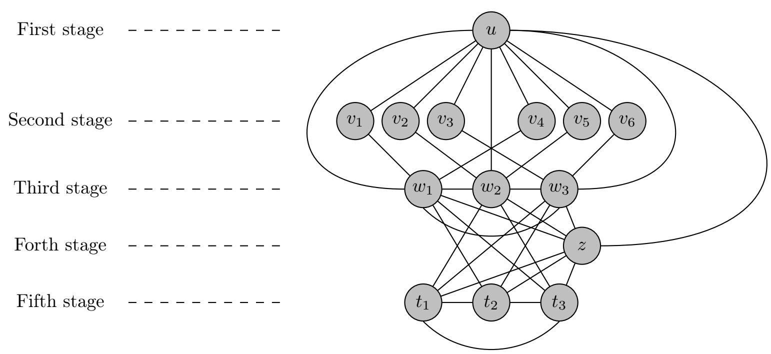


Figure 6.2: An example of graph G for which adding DI adds fractionnal vertices

$$
\begin{array}{r}
\text{First stage} \\
\\
\\
\text{Second stage} \\
\\
\\
\\
\\
\text{Third stage} \\
\\
\text{Fourth stage} \\
\text{Fifth stage} \\
\\
\\
\end{array}
\begin{pmatrix}
- & - & - & - & - & - & + & + & + & + \\
+ & 0 & + & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & + & + & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
+ & + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & + & 0 & + & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & + & + & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & + & + & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & + \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & + & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & + & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & + & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & + \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & + & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & + & 0 & 0
\end{pmatrix}
$$

Figure 6.3: Constraint submatrix

In Figure 6.2, a valid covering of co-2-plexes is as follows: $\{\{u,z\},\{v_1,v_2,v_3,v_4,v_5,v_6,t_1,t_2\},\{w_1,w_2\},\{w_3,t_3\}\}$. It is optimal since the third and fourth stages form a 2-plex of size 7, and every co-2-plex intersects at most 2 times a 2-plex. Let us now consider the point:

- $x^*_{\{u,w_1,t_1\}} = 1$
- $x^*_{\{u,w_2,t_2\}} = 1$
- $x^*_{\{u,w_3,t_3\}} = 1$
- $x^*_{\{u,z\}} = 1$
- all other components are 0

- $y^*_{u,\{v_1,v_2\}} = 0.5$
- $y^*_{u,\{v_2,v_3\}} = 0.5$
- $y^*_{u,\{v_1,v_3\}} = 0.5$
- $y^*_{u,\{v_4,v_5\}} = 0.5$
- $y^*_{u,\{v_5,v_6\}} = 0.5$
- $y^*_{u,\{v_4,v_6\}} = 0.5$

Note that the objective value of (x^*,y^*) is 4. The optimal value of the linear relaxations of $\widetilde{P}$ and P for this graph are 3.5. It may be more convenient to look at its constraint submatrix, see Figure 6.3 for an illustration where the'+' corresponds to 1 and '−' corresponds to -1. The first stage is the row associated with vertex u. The second stage contains two disjoint odd holes and a matrix of 0. The third stage is a matrix of zeros next to an identity matrix. The fourth stage is a vector with only one coefficient 1. The fifth stage is an identity matrix. From this matrix construction, we understand that (x^*, y^*) is an extreme point of the polyhedron since every constraint is satisfied with equality, and the constraint matrix has rank 10 (the trivial constraints associated with variables of value 0 form an identity matrix).

6.5.2 Branching rules

The fact that an optimal solution of $(\widetilde{P})$ can be fractional on y can be treated using the classical branching rule as described below but with an additional branching over y variable. However, before branching over y variables, the first branching rule branches on every pair of nodes u, v, stating that on the one hand, u and v must belong to the same co-2-plex and on the other hand, they never belong to the same co-2-plex. A leaf node is infeasible, or the branching decisions directly give the associated solution without solving any LP. In the meantime, we plan on considering the values of the variables associated with DI to guide the branching decisions as follows in order to deal with the DI variables.

At each branching node, we compute (u, v), a couple of vertices minimizing the following quantity:

$$\left| 0.5 - \sum_{S \in \mathcal{S}^* \mid \{u,v\} \subseteq S} x^*_S - \sum_{(w,T) \in \mathcal{D} \mid \{u,v\} \subseteq T} y^*_{w,T} \right|$$

The branching rule is then to create a first child node where u and v belong to the same co-2-plex of the solution and where each DI associated with (w, T) will be set to 0 if $u \in T,\ v \notin T$ or $v \in T,\ u \notin T$. Then, on the second child node, u and v never belong to the same co-2-plex, and each DI associated with (w, T) will be set to 0 if $\{u, v\} \subseteq T$.

6.5.3 Computational results

Adding DIs with a greedy algorithm

The dual inequalities have been added to the RMP before any LP resolution. Even if adding DOI dynamically has been experimentally successful [4], it is only slightly better than adding DOI as preprocessing on the coloring problem. Hence, for our DI variables, we choose only to implement the preprocessing and no dynamic generation. We use the following greedy algorithm to compute the dual inequalities:

For each vertex v, we greedily compute a cover $\{T_1, \ldots, T_k\}$ of $D_v \subseteq N_v$, where D_v is the set of vertices u verifying $\overline{N}_u \subseteq \overline{N}_v$, such that (v, T_i) has the dominance property and T_i is inclusion wise maximal for each $i \in \{1, \ldots, k\}$.

Experimental results on random instances

The baseline does not converge in less than 3 hours for the random instances considered in this section; hence, we only compare the root node.

instance type	nb DI	CPU	nb INst
1	1	1.06	58
2	2	0.9	26
3	3	0.96	12
4	4	0.79	4

Table 6.4: root node CPU on random instances with 50 vertices and their corresponding number of DI

Description of the random instances: The instances are connected graphs with a density of 8%, this low density was necessary to generate random instances having the dominant structure. Considering other values for the density made the try-and-reject algorithm have trouble generating instances with the dominance structure.

Table 6.4 lets us think that the random instances with more DI are easy to solve, but there may be a lot of bias because of the small number of instances for which we add 4 DIs.

To confirm this intuition we generated sets of 25 random instances with 1 to 8 DIs (rejecting the random instances not having the wanted number of DIs). These instances are relatively hard for the baseline, hence we only consider the root node but this should be sufficient to investigate the stabilization power of DIs. Unfortunately, the improvement appearing in Table 6.4 does not appear anymore in Table 6.5. For these instances, it seems that DI does not have a significant impact.

Algorithm	Instances type	CPU	# col	nb faster
$\widetilde{P}$	1	1.01	1.0	12
	2	1.07	0.99	10
	3	0.96	0.99	15
	4	1.02	1.0	10
	5	1.52	1.57	10
	6	0.97	1.0	15
	7	1.01	1.0	14
	8	1.01	1.0	13
$\widetilde{P}_1$	1	1.01	1.0	11
	2	1.07	0.99	10
	3	0.97	0.99	15
	4	1.02	1.0	10
	5	1.52	1.57	10
	6	0.97	1.0	15
	7	1.02	1.0	14
	8	1.49	1.65	15

Table 6.5: Experimental results for the root node on random balanced sets of instances with 100 vertices of $\widetilde{P}$ and $\widetilde{P}_1$

	$\widetilde{P}_1$			$\widetilde{P}$		
Instances	# nodes	# col	CPU	# node	# col	CPU
anna	201	73064	5.55e6	376	73378	1.08e7
huck	1	61	318.0	1	112	424.0
jean	71	4779	35683.0	1	79	456.0
david	711	28790	1.26e6	309	18235	471934.0
miles500	321	8209	205169.0	387	3578	123363.0
r125.1	7	702	2037.0	6	652	1817.0
dolphins.graph_comp	35	3945	26781.0	19	2172	11423.0
karate.graph_comp	9	392	827.0	4	176	290.0
lesmis.graph_comp	1	1078	5570.0	1	1164	5995.0
polbooks.graph_comp	632	94834	1.08e7	90	15148	282638.0
miles250	221	16450	401534.0	216	20590	564702.0

Table 6.6: Instances for which $\widetilde{P}_1 \neq \widetilde{P}$

		Ratio to baseline							
	Instances	CPU	# col	# nodes	M CPU p.it	CPU root	% pricing	# DI	#faster
$\widetilde{P}$	ED-color	503.42	25.25	5.35	80.42	3.8	14.71	3358	50
	HD-color	0.84	0.47	0.89	1.85	1.06	8.76	15182	80
	DIMACS	0.83	0.93	0.79	0.82	0.95	79.52	768	71
	comp-DIMACS	0.03	0.2	0.19	0.1	1.0	55.36	306	80
	others	0.21	0.19	0.56	1.07	0.01	15.49	3534	80
$\widetilde{P}_1$	ED-color	271.65	26.23	4.15	40.46	0.75	17.51	3273	40
	HD-color	0.88	0.55	0.99	1.64	1.1	1.13	15152	70
	DIMACS	0.83	0.93	0.79	0.81	0.95	79.77	768	71
	comp-DIMACS	1.15	1.07	1.15	1.05	0.99	28.44	273	40
	others	0.21	0.18	0.56	1.13	0.01	15.35	3528	80

Table 6.7: Performances of both BP with DI

Experimental results on literature instances

The column's titles of Tables 6.6 and 6.7 are listed in Section 5.2.3. Moreover, the column '# DI corresponds to the number of added DIs.

In Table 6.6, we compare the few instances for which there exists DI associated with couples (w, T) where $|T| > 1$. This table shows that it is unclear which is the best between $\widetilde{P}$ and $\widetilde{P}_1$. The algorithm based on $\widetilde{P}$ does not converge on the instance anna when the algorithm based on $\widetilde{P}_1$ does not converge on polbooks.graph_comp.

The linear relaxation value remains the same: Note that the linear relaxation of the baseline is always the same as $\widetilde{P}$ and $\widetilde{P}_1$ on these sets of instances. This is surprising as we showed that adding such DI can lower the primal's relaxation value from a theoretical point of view.

Bad behaviour of DIs for small instances: In Table 6.7 we give the

		Ratio to baseline							
	Instances	CPU	# col	# nodes	M CPU p.it	SP CPU p.it	% pricing	nb DI	#faster
$\widetilde{P}$	ED-color	1.15	0.94	1.0	0.55	1.56	84.54	3358	50
	HD-color	2.77	1.44	1.0	2.32	1.73	60.13	15182	50
	DIMACS	1.05	0.98	1.0	1.21	1.06	92.78	768	14.28
	comp-DIMACS	0.76	0.8	1.0	1.0	0.94	69.94	306	40
$\widetilde{P}_1$	ED-color	1.2	0.9	1.0	0.73	1.65	81.32	3273	50
	HD-color	2.76	1.39	1.0	2.39	1.78	60.1	15152	40
	DIMACS	1.06	0.98	1.0	1.13	1.08	93.27	768	0
	comp-DIMACS	0.83	0.82	1.0	0.93	1.03	73.23	273	80

Table 6.8: Performances of the BP with DI with knowledge of the optimal solution

aggregated results for each set of instances. First, note that on ED-color both algorithms perform much worse than the baseline. This is understandable since they are pretty easy instances and the set of DI added has half the size of the column set for the baseline, see Table 6.1, this implies that the RMPs are much harder, which is confirmed by looking at the column % pricing: when the baseline spend 79% of its computation time in solving pricing subproblems, our algorithms using DI spent 14% of their computation time in pricing subproblems.

DIs stabilize the column generation: For the other sets of instances, adding DI seems to stabilize the whole column generation process by decreasing the number of generated columns and branching nodes. The improvement in comp-DIMACS comes from the instance poolbocks_comp that takes most of the computation of the baseline and for which the BP with DI converges in 0.02% of the baseline's computation time.

Perturbating SCIP's heuristic: A surprising result that is not easy to exhibit in such tables, is the fact that adding DI perturbates SCIP's primal heuristics: sometimes DIs help one of SCIP's heuristics to find the optimal integer solution already at the root node and sometimes it makes the heuristics find it later than the baseline would. To notice that, we had to follow the resolution of some instances with and without DI. As preliminary experimentations, we noticed that the order in which the DIs have been added to the RMP also impacts these heuristics. Since doing column generation without primal heuristics seems to be limiting, we tried to compare this algorithm after giving the optimal solution of the whole BP as a prior knowledge, in that case, the BP only ensures the optimality of such solution. The associated results are given in Table 6.8. This table shows, for which instances, adding DI stabilizes the column generation process without considering the solutions given by the primal heuristics. In fact, for all sets but comp-DIMACS, the baseline with prior knowledge is faster but now, both BP with DI do not perform much worse than the baseline on ED-color: this proves that for this set of instances adding DI perturbates the primal heuristics in a bad way when for HD-color, and DIMACS, it perturbates the primal heuristics in a good way. Still, unless for HD-color, the number of generated columns is reduced when adding DIs which witness a slight stabilization. Unfortunately, it does not induce a smaller computation time. For comp-DIMACS, we see a true stabilization of the column generation process that does not rely on good primal solutions, and for this set, the DI that seemed to be the strongest from a dual point of view performs better: $\widetilde{P}$ performs better than $\widetilde{P}_1$.

$\widetilde{P}$ **is the fastest on the hardest sets of instances.** Remember that HD-color where the hardest sets of instances for the Baseline, and so by far. For these two sets of instances $\widetilde{P}$ performs better than the baseline.

6.6 Cutting both primal and dual spaces

This section investigates using triangle inequalities and DI simultaneously to improve the relaxation value and stabilize the column generation. Both techniques of the previous sections that are, adding primal and dual cuts seem to have a positive impact on the set HD-color, and we wonder if it is possible to use the benefits of both to solve this particular set of instances.

6.6.1 The triangle coefficient of dual inequalities

In this section, we investigate how to use triangle inequalities and DI simultaneously to improve the relaxation value and use DI simultaneously. To answer this question, we consider the following LP where Ψ is the matrix of y coefficient in triangle inequalities:

$$(\widetilde{P}^{\Psi}_{\mathcal{T}}) \left\{ \begin{array}{rcll} \min \sum\limits_{S \in \mathcal{S}_G} x_S & & & \\ \sum\limits_{S \in \mathcal{S}_G | v \in S} x_S - \sum\limits_{(v,T) \in D} y_{v,T} + \sum\limits_{(w,T) \in D | v \in S} y_{w,T} & \geq & 1 & \forall\ v \in V \\ \sum\limits_{S \in \mathcal{S}_G | S \cap K \neq \emptyset} x_S + \Psi_K y & \geq & 2 & \forall K \in \mathcal{T} \\ x_S & \geq & 0 & \forall\ S \in \mathcal{S}_G \\ y_{w,T} & \geq & 0 & \forall (w,T) \in \mathcal{D} \end{array} \right.$$

First, since every integer point x of P is a solution (not necessarily optimal) of $(P_{\mathcal{T}})$, we have that $(x, 0)$ is also a solution of $(\widetilde{P}^{\Psi}_{\mathcal{T}})$. Conversely, since $(\widetilde{P}^{\Psi}_{\mathcal{T}})$ is included in $(\widetilde{P})$, then every integer solution (x, y) of $(\widetilde{P}^{\Psi}_{\mathcal{T}})$ is an integer solution of $(\widetilde{P})$ and hence can be mapped using Theorem 6.4.4 to build a solution of P with the same objective value. It means that $(\widetilde{P}^{\Psi}_{\mathcal{T}})$ has the same optimal value as P. Note that this reasoning holds whatever Ψ is.

Intuitively, if the coefficients of y variables are large in triangle constraints *i.e.* Ψ has high coefficients, the triangle constraints become easy to satisfy, even for fractional points. Hence, the relaxation value of $(\widetilde{P}^{\Psi}_{\mathcal{T}})$ should be close to the one of $\widetilde{P}$. If Ψ is nonnegative, one can understand high coefficients as smaller or equal to 2 since it is the right-hand side of triangle inequalities. Having strictly higher coefficients would lead to trivially dominated constraints. Conversely, if Ψ has negative, low coefficients, having positive y variables makes it harder to satisfy the triangle inequalities, in which case the optimal value solution of $(\widetilde{P}_T^\Psi)$ probably verify $y = 0$, in which case we lose the benefits of DI.

Guided by this intuition and the recovery algorithm of Theorem 6.4.4, we propose that Ψ depends on how the triangle inequality intersects the dominance property associated pairs. Given a triangle K and a dominance couple (w, T) we propose the following coefficients:

- $\Psi_K^{w,T} = 1$ if $w \notin K$ and $T \cap K \neq \emptyset$
- $\Psi_K^{w,T} = -1$ if $w \in K$ and $T \cap K = \emptyset$
- $\Psi_k^{w,T} = 0$ otherwise.

In the first case, having $y_{w,T} \neq 0$ means that some co-2-plex not necessarily intersecting the triangle K will be replaced by the recovery algorithm by a co-2-plex intersecting it for sure. In the second case, some co-2-plex intersecting it will be replaced by a co-2-plex that may not intersect it.

To investigate the impact of this dedicated coefficient Ψ, we use this same procedure with $\Psi = 0$.

6.6.2 Computational results

The branching rule for these algorithms is the same as the one used for the BP with DI.

In Table 6.9 we give the aggregated results of the two different BCPs using DI. First note that the time spent separating constraints is ridiculous. On ED-color, the BCP with $\Psi = 0$ is much worse than the baseline, remember that the same phenomenon happened when considering the BP with DI based on formulation $\widetilde{P}$.

BCPs with DI perform well on HD-color: Both BCPs with DI solve HD-color faster than both BP with DI based on formulations $\widetilde{P}$ and $\widetilde{P}_1$. The root node takes generally a little bit more time to be computed because of the additional constraints, even if the addition of DI permits to compute it slightly faster on comp-DIMACS. Also, the number of generated constraints is really low for both BCPs on comp-DIMACS. The BCP with DI that seems to work the best on HD-color is with $\Psi = 0$.

The two BCPs with DI manage to solve 2 instances from 'others' when both BP with DI solve 4 of them. On this hard set of instances for the baseline, the BPs with DI seem are the best algorithms.

The time spent solving the pricing subproblems is low: it represents in general less than one-third of the total computation time. On HD-color, the number of columns generated and branching nodes is lowered, in particular with $\Psi = 0$, for this set of instances it seems like taking $\Psi = 0$ is the coefficient that combines the most effectively primal and dual inequalities.

Both BCPs with DI perform badly on DIMACS: For the instance set DIMACS, the number of iterations is a little higher than for the baseline, but the average Master problem becomes suddenly really hard while the average pricing subproblem becomes a little bit harder. This results in two BCP algorithms that are close to 6 times slower than the baseline.

The relaxation value remains the same as the classic BCP Table 6.10 shows the relaxation value of the baseline, the classic BCP and the two BCP using DI on the only instances where they differ. Both BCP with DI yield the same relaxation value that may be lower than the classic BCP, this is surprising since $\Psi = 0$ should imply stronger primal constraints but weaker dual constraints and when going back to Table 6.9 we see that $\Psi = 0$ is faster in general.

Among all algorithm, B&C&P with $\Psi = 0$ is the fastest on HD-color. This shows that for this problem, using primal and dual inequalities permitted to solve 2 times faster HD-color than the state-of-the-art method.

		Ratio from the baseline					
Alg	Instances	CPU	# col	# nodes	M CPU p.it	SP CPU p.it	CPU root
$\Psi = 0$	ED-color	122.55	11.29	2.41	42.79	2.34	3.04
	HD-color	0.49	0.38	0.8	1.21	1.56	1.39
	DIMACS	5.52	1.07	1.61	15.67	1.77	2.68
	comp-DIMACS	1.15	1.04	1.24	1.08	1.17	0.98
	others	0.62	0.66	0.62	0.94	0.92	1
$\Psi \neq 0$	ED-color	8.09	2.74	0.66	6.46	2.05	3.69
	HD-color	0.65	0.46	0.95	1.31	1.89	1.54
	DIMACS	5.54	1.07	1.63	16.02	1.67	2.6
	comp-DIMACS	1.15	1.04	1.26	1.1	1.11	0.98
	others	0.44	0.53	0.7	0.83	0.82	1

Alg	Instances	% pricing	% sepa	# cons	# DOI	% faster
$\Psi = 0$	ED-color	16.63	10^{-2}	28	3358	50
	HD-color	16.22	10^{-2}	1780	15182	70
	DIMACS	27.78	0.07	126	768	43
	comp-DIMACS	28.67	10^{-2}	13	306	40
	others	15.33	10^{-2}	1352	3534	40
$\Psi \neq 0$	ED-color	53.67	10^{-2}	35	3358	50
	HD-color	17.87	10^{-2}	2312	15182	80
	DIMACS	26.27	0.07	130	768	43
	comp-DIMACS	27.28	10^{-2}	19	306	60
	others	15.54	10^{-2}	894	3534	40

Table 6.9: Computational results of the BCP with DI

		relaxation			
Instances set	Name	baseline	BCP	BCP+ DI $\Psi = 0$	BCP + DI $\Psi \neq 0$
HD-color	david	5.63	5.67	5.67	5.67
	mulsol.i.2	15.7667	16.0	15.7667	15.7667
	mulsol.i.4	15.7667	16.0	15.7667	15.7667
	mulsol.i.5	15.7667	16.0	15.7667	15.7667
DIMACS	dolphins	17.99	18.0	18.0	18.0
	jazz	47.5393	47.5421	47.5393	47.5393

Table 6.10: Linear relaxation value of the BCP with DI

6.7 Conclusion

In this chapter, we proposed to improve a formulation used to solve the 2-defective coloring by adding primal and dual inequalities. These new dual inequalities sometimes cut off all optimal dual points. We show that this type of dual inequality reduces the computation time to solve this problem in some instances. Moreover, we investigated how to use both primal and dual constraints simultaneously. We proposed two new approaches and gave a set of instances for which the best algorithm uses both primal and dual inequalities. We showed that even if some instances are hard for the baseline, our dual inequalities could help solve some of them faster. We showed that they managed to stabilize the column generation on some instance sets. But they also are often slower
Since we generalized that type of DI to the partitioning into co-k-plex for every positive integer k, one could investigate if it also helps for k strictly greater than 2. Furthermore, since every DI adds an integer variable to the primal, it could be interesting to dynamically generate these DI to avoid manipulating non-useful integer variables as done in [4], still, by doing that, we may loose the potential benefit of the interactions between such variables and SCIP's primal heuristics.

As we have seen that the DI help SCIP's heuristics to find great primal solutions and that it induces an important reduction of the computation time, it would be interesting to find dedicated primal heuristics for these problems, that is a heuristic able to give a good quality primal solution from a set of dual inequalities, branching decisions and co-2-plexes.

Another perspective, since triangle constraints do not show significant improvements, someone may be interested in using general odd 2-plex constraints. In fact, every 2-plex of size $2n + 1$ must be covered by at least $n + 1$ co-2-plexes. This remark led us to triangle constraints that can be enumerated exactly. Separating general odd 2-plex constraints may improve the relaxation by exhibiting rank constraints with high right-hand sides.

Conclusion and perspectives

Conclusion

In this thesis, we first introduced a new subclass of perfect graphs and characterized them in several manners. This theoretical study permitted us to give a new polynomial graph class for the problem of finding a maximum weighted co-2-plex and strong extended formulations based on the weak perfect graphs theorem. After deducing that our new formulations perform quite well on some sets of instances we tried to solve the 2-defective coloring problem. Towards new algorithms for the 2-defective problem, we studied a well-known formulation for the coloring problem. We first tried to add primal cuts and investigated different alternation strategies between cutting and pricing. This formulation has been improved in the literature by considering dual optimal inequalities at the root node. We showed that using these dual inequalities during the whole branching scheme yielded a better algorithm and proposed two methods combining primal and dual inequalities and experimentally compared them to the baseline. We deduced using only dual inequalities is the best improvement over the coloring problem. The study of this formulation permitted us to give new dual inequalities for the 2-defective coloring problem's analogous formulation. We also considered adding Chvátal-Gomory or triangle inequalities to the primal and tried to use that same framework considering primal and dual inequalities. This time, using both primal and dual inequalities seemed to be pertinent on a set of instances but with only a low improvement compared to the use of dual inequalities alone.

Perspectives about the maximum weighted stable set/clique problem on contraction perfect graphs

No combinatorial algorithm is known to solve the problem of weighted stable sets/cliques on perfect graphs. It is a pretty challenging topic, in particular, the perfect graph decomposition does not yield such an algorithm.

A simpler perspective would be to find such an algorithm for contraction-perfect graphs. As the complements of contraction-perfect graphs are not necessarily contraction-perfect, unlike for perfect graphs, both problems of finding a maximum clique and a maximum stable set are not exactly the same. Moreover, contracting an edge, can increase, decrease, or let unchanged the size of a largest clique when it can only decrease or let unchanged the size of a largest stable set.

Analogously to the perfect graph decomposition, a first idea would be to investigate a contraction-perfect graph decomposition. For sure, some of the operations used in the decomposition of perfect graphs preserve contraction-perfectness (the replication lemma for contraction perfect and clique identification should do the job), but I am not sure about the so-called 2-amalgam [37] as I am not sure how it works. Moreover, it should also not be too hard to find the contraction-perfect base cases.

The last idea for this perspective is to investigate the combinatorial structure of the stable set/clique polytope and in particular, their 1-skeleta that is the graph having as vertex set, the extreme points of the stable set polytope (incidence vectors of stable sets) and where two vertices are adjacent if their corresponding extreme points belong to the same one-dimensional face. Farid Aliniaeifard et al. [80] prove the following result for a class of polytopes including stable set/clique polytopes:

Theorem 6.7.1 ([80]). *Two vertices of the stable set polytope χ^{S_1}, χ^{S_2} are adjacent if and only if $\chi^{S_1} + \chi^{S_2} = \chi^{S_3} + \chi^{S_4} \Rightarrow \{\chi^{S_1}, \chi^{S_2}\} = \{\chi^{S_3}, \chi^{S_4}\}$*

As a corollary of Theorem we obtain that the diameter of the 1-skeleta of the stable set polytope is polynomial. This is a first step to try finding a simplex rule following a polynomial path between an arbitrary extreme point and an extreme point corresponding to a maximal size stable set. Note that these results also hold for the clique polytope.

Finally investigating how the 1-skeleta of perfect graphs behave when applying the decomposition/composition operations used in the strong perfect graph theorem's proof could be another track and contractions. Moreover, the edges of the 1-skeleta of the stable set polytope correspond to subsets of cliques and vertices as they correspond to subsets of $|V| - 1$ linear independent constraints.

Perspectives about the co-2-plex polytope of richer graph classes

The result we obtained for the co-2-plex polytope of trees on the natural variable set is not satisfying to me. In particular, I feel like we used heavy weapons (the perfect graphs theorem, Fourier-Motzkin elimination) to obtain a weak result characterizing this polytope on a tiny graph class. I have come to the point that I do not expect us to obtain the natural variable space description of this polytope on split graphs nor trivially perfect. However, we could probably go further than a characterization for only trees as our result does not involve, particularly new proof methods and is more or less a direct consequence of Fourier-Motzkin elimination and our extended space characterization of the co-2-polytope of chordal graphs.

Perspectives about alternating between cutting and pricing

The Chvátal-Gomory constraints we added for the coloring problem did not improve the relaxation value much and were costly to separate. Looking for a problem, formulated as a column generation formulation, for which strong reinforcement inequalities can be computed easily (a polynomial exact separation or a fast heuristic) would be a good candidate for a computational improvement using alternation strategies. As alternating strategies may induce cutting before primal convergence, several cuts may be added and hence would be preferably robust.

A good candidate could be a problem that is naturally formulated with an exponential number of constraints and columns and for which the dual is highly degenerated. I am thinking of an operational problem combining several known problems that would inherit from both combinatorial structures. Unfortunately, I am not aware of an application of this last problem which would be the first important feature to look for.

Perspectives about the power of our DI for k-defective coloring

The first obvious perspective on this topic would be to investigate the power of our DIs for $k > 2$ and k-plexes inequalities, but we believe that the correct way to use them should be first investigated. As these cuts did not stabilize the column generation but helped SCIP's heuristic find good solutions, a natural way to utilize such cuts would be to add them after primal convergence, and then let SCIP try to find a good primal solution, and hence remove them at each branching node. This would permit to use of the benefits of such cuts without hardening the master problems.

That being said, another natural question would be to find a dedicated quality primal heuristic able to take into account the branching decisions. A lot of work has been done to compute heuristically maximum co-k-plexes [16, 17, 18, 19, 20], but recall that the branching decisions we used are nonrobust, so these heuristics would have to be adapted to take into account a set of branching decisions. One could even try to find primal heuristics profiting from the graph structure inducing our dual cuts.

Using the bound given in Theorem 1.1.5 on the size of the intersection of a co-k-plex and a k-plex, we deduce that the number of co-k-plexes covering a k-plex K is at least $\lceil \frac{|K|}{2k-2(kmod2)} \rceil$. This yields a family of constraints containing triangle constraints. The last perspective of this Chapter would then be to add such inequalities in a B&C&P scheme. Computing such constraints is probably hard, and hence it would be necessary to find a heuristic separation.

Bibliography

Bibliography

[1] M. Grötschel, L. Lovász, and A. Schrijver. Polynomial algorithms for perfect graphs. In C. Berge and V. Chvátal, editors, *Topics on Perfect Graphs*, volume 88 of *North-Holland Mathematics Studies*, pages 325–356. North-Holland, 1984.

[2] Fabio Furini, Timo Gschwind, Stefan Irnich, and Roberto Wolfler Calvo. Decomposition of social networks into relaxed cliques. *Informs*, 2021.

[3] Anuj Mehrotra and Michael A. Trick. A column generation approach for graph coloring. *INFORMS Journal on Computing*, 8: 344–354, 1996.

[4] Timo Gschwind and Stefan Irnich. Dual inequalities for stabilized column generation revisited. *INFORMS Journal on Computing*, 28:175–194, 02 2016.

[5] G. Cornuejols, Xinming Liu, and K. Vuskovic. A polynomial algorithm for recognizing perfect graphs. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 20–27, 2003.

[6] Stephen Seidman and Brian Foster. A graph-theoretic generalization of the clique concept*. *Journal of Mathematical Sociology*, 6:139–154, 01 1978.

[7] Balabhaskar Balasundaram, Sergiy Butenko, and Illya Hicks. Clique relaxations in social network analysis: The maximum k-plex problem. *Operations Research*, 59(1):133–142, 2011.

[8] Steffen Rebennack. *Ellipsoid methodEllipsoid Method*, pages 890–899. Springer US, Boston, MA, 2009.

[9] Michael R. Garey and David S. Johnson. Computers and intractability. a guide to the theory of np-completeness. *Journal of Symbolic Logic*, 48(2):498–500, 1983.

[10] F.R. Giles and W.R. Pulleyblank. Total dual integrality and integer polyhedra. *Linear Algebra and its Applications*, 25:191–196, 1979.

[11] A. J. Hoffman and J. B. Kruskal. *13. Integral Boundary Points of Convex Polyhedra*, pages 223–246. Princeton University Press, Princeton, 1957.

[12] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4(4):305–337, 1973.

[13] Egon Balas. Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89(1):3–44, 1998.

[14] Kenneth Appel and Wolfgang Haken. *The Four-Color Problem*, pages 153–180. Springer New York, New York, NY, 1978.

[15] Balabhaskar Balasundaram. Cohesive subgroup model for graph-based text mining. In *2008 IEEE International Conference on Automation Science and Engineering*, pages 989–994, 2008.

[16] Wayne Pullan. Local search for the maximum k-plex problem. *J. Heuristics*, 27(3):303–324, 2021.

[17] Peilin Chen, Hai Wan, Shaowei Cai, Weilin Luo, and Jia Li. Combining reinforcement learning and configuration checking for maximum k-plex problem. *ArXiv*, abs/1906.02578, 2019.

[18] Sun-Yuan Hsieh, Shih-Shun Kao, and Yu-Sheng Lin. A swap-based heuristic algorithm for the maximum k -plex problem. *IEEE Access*, 7:110267–110278, 2019.

[19] Yanqi Jin, John H. Drake, Una Benlic, and Kun He. Effective reinforcement learning based local search for the maximum k-plex problem. *ArXiv*, abs/1903.05537, 2019.

[20] Peilin Chen, Hai Wan, Shaowei Cai, Jia Li, and Haicheng Chen. Local search with dynamic-threshold configuration checking and incremental neighborhood updating for maximum k-plex problem. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(03):2343–2350, Apr. 2020.

[21] Kuixian Wu, Jian Gao, Rong Chen, and Xianji Cui. Vertex selection heuristics in branch-and-bound algorithms for the maximum k-plex problem. *International Journal on Artificial Intelligence Tools*, 28(05):1950015, 2019.

[22] Yoshiaki Okubo, Masanobu Matsudaira, and Makoto Haraguchi. Detecting maximum k-plex with iterative proper l-plex search. In Sašo Džeroski, Panče Panov, Dragi Kocev, and Ljupčo Todorovski, editors, *Discovery Science*, pages 240–251, Cham, 2014. Springer International Publishing.

[23] Jiongzhi Zheng, Mingming Jin, Yan Jin, and Kun He. Relaxed graph color bound for the maximum k-plex problem. *CoRR*, abs/2301.07300, 2023.

[24] Yue Wang, Xun Jian, Zhenhua Yang, and Jia Li. Query optimal k-plex based community in graphs. *Data Sci. Eng.*, 2(4):257–273, 2017.

[25] Hua Jiang, Dongming Zhu, Zhichao Xie, Shaowen Yao, and Zhang-Hua Fu. A new upper bound based on vertex partitioning for the maximum k-plex problem. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1689–1696. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Main Track.

[26] Lijun Chang, Mouyi Xu, and Darren Strash. Efficient maximum k-plex computation over large sparse graphs. *Proc. VLDB Endow.*, 16(2):127–139, oct 2022.

[27] Yun-Ya Huang and Chih-Ya Shen. Learning computation bounds for branch-and-bound algorithms to k-plex extraction. *CoRR*, abs/2208.05763, 2022.

[28] P.I. Stetsyuk, O.M. Khomiak, Y.A. Blokhin, and A. A. Suprun. Optimization problems for the maximum k-plex*. *Cybernetics and Systems Analysis*, 58:530–541, 2022.

[29] Qiangqiang Dai, Rong-Hua Li, Hongchao Qin, Meihao Liao, and Guoren Wang. Scaling up maximal k-plex enumeration. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 345–354, New York, NY, USA, 2022. Association for Computing Machinery.

[30] Zhuo Wang, Qun Chen, Boyi Hou, Bo Suo, Zhanhuai Li, Wei Pan, and Zachary G. Ives. Parallelizing maximal clique and k-plex enumeration over graph data. *Journal of Parallel and Distributed Computing*, 106:79–91, 2017.

[31] Saïd Jabbour, Nizar Mhadhbi, Badran Raddaoui, and Lakhdar Sais. A declarative framework for maximal k-plex enumeration problems. In *Adaptive Agents and Multi-Agent Systems*, 2022.

[32] Benjamin McClosky and Illya Hicks. The co-2-plex polytope and integral systems. *Siam Journal on Discrete Mathematics - SIAMDM*, 23:1135–1148, 2009.

[33] Lovász László. Normal hypergraphs and the perfect graph conjecture. *Discrete Mathematics*, 2:253–267, 05 1972.

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

[35] Jack Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards Section B Mathematics and Mathematical Physics*, page 125, 1965.

[36] L Lovász. A characterization of perfect graphs. *Journal of Combinatorial Theory, Series B*, 13(2):95–98, 1972.

[37] Gérard Cornuéjols. *Combinatorial Optimization: Packing and Covering*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, 2001.

[38] Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. The strong perfect graph theorem. *Annals of Mathematics*, 164:51–229, 2006.

[39] J. Fonlupt and J.P. Uhry. Transformations which preserve perfectness and h-perfectness of graphs. In Achim Bachem, Martin Grötschel, and Bemhard Korte, editors, *Bonn Workshop on Combinatorial Optimization*, volume 66 of *North-Holland Mathematics Studies*, pages 83–95. North-Holland, 1982.

[40] H. Meyniel. A new property of critical imperfect graphs and some consequences. *European Journal of Combinatorics*, 8(3):313–316, 1987.

[41] Dan Bienstock. On the complexity of testing for odd holes and induced odd paths. *Discrete Mathematics*, 90(1):85–92, 1991.

[42] Maria Chudnovsky, Gérard Cornuéjols, Xinming Liu, Paul D. Seymour, and Kristina Vuskovic. Recognizing berge graphs. *Comb.*, 25(2):143–186, 2005.

[43] Frédéric Maffray and Nicolas Trotignon. A class of perfectly contractile graphs. *Journal of Combinatorial Theory, Series B*, 96(1):1–19, 2006.

[44] László Lovász. Normal hypergraphs and the perfect graph conjecture. *Discrete Mathematics*, 2(3):253–267, 1972.

[45] Hanif Sherali and Warren Adams. Reformulation–linearization techniques for discrete optimization problems. *Handbook of Combinatorial Optimization*, 07 2013.

[46] H. N. de Ridder et al. Information System on Graph Classes and their Inclusions (ISGCI). `https://www.graphclasses.org`.

[47] https://porta.zib.de/.

[48] Benoît Legat. Polyhedral computation. In *JuliaCon*, July 2023.

[49] Jian Gao, Jiejiang Chen, Minghao Yin, Rong Chen, and Yiyuan Wang. An exact algorithm for maximum k-plexes in massive graphs. pages 1449–1455, 07 2018.

[50] https://mat.tepper.cmu.edu/color02/.

[51] Elizabeth D. Dolan and Jorge J. Moré. Benchmarking optimization software with performance profiles. *CoRR*, cs.MS/0102001, 2001.

[52] D. R. Woodall L. J. Cowen, R. H. Cowen. Defective colorings of graphs in surfaces: Partitions into subgraphs of bounded valency. *Journal of graph theory*, 1986.

[53] Belmonte1 Rémy, Lampis Michael, and Mitsou Valia. Defective coloring on classes of perfect graphs. *Discrete Mathematics & Theoretical Computer Science*, 2017.

[54] H. Hadwiger. Uber eine klassifikation der streckenkomplexe. *Naturforsch. Ges*, 1943.

[55] Neil Robertson, Paul Seymour, and Robin Thomas. Hadwiger's conjecture fork6-free graphs. *Combinatorica*, 1993.

[56] Katherine Edwards, Dong Yeap Kang, Jaehoon Kim, Sang-il Oum, and Paul Seymour. A relative of hadwiger's conjecture. *SIAM Journal on Discrete Mathematics*, 29(4):2385–2388, 2015.

[57] Jan Mycielski. Sur le coloriage des graphs. *Colloquium Mathematicae*, 3(2):161–162, 1955.

[58] Stefano Gualandi and Federico Malucelli. Exact solution of graph coloring problems via constraint programming and column generation. *INFORMS Journal on Computing*, 24, 2010.

[59] Graph coloring benchmarks. `https://sites.google.com/site/graphcoloring/vertex-coloring`. Accessed: 2021-09-30.

[60] Enrico Malaguti, Michele Monaci, and Paolo Toth. An exact approach for the vertex coloring problem. *Discrete Optimization*, 8(2): 174–190, 2011.

[61] Stephan Held, William Cook, and Edward C. Sewell. Maximum-weight stable sets and safe lower bounds for graph coloring. *Mathematical Programming Computation*, 4, 2012.

[62] David Morrison, Jason Sauppe, Edward Sewell, and Sheldon Jacobson. A wide branching strategy for the graph coloring problem. *INFORMS Journal on Computing*, 26:704–717, 11 2014.

[63] P. Hansen, M. Labbé, and D. Schindl. Set covering and packing formulations of graph coloring: Algorithms and first polyhedral results. *Discrete Optimization*, 6(2): 135–147, 2009.

[64] Ricardo Fukasawa, Humberto Longo, Jens Lysgaard, Marcus Poggi, Marcelo Reis, Eduardo Uchoa, and Renato Werneck. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming*, 106: 491–511, 2006.

[65] Michael Larsen, James Gary Propp, and Daniel H. Ullman. The fractional chromatic number of mycielski's graphs. *J. Graph Theory*, 19:411–416, 1995.

[66] Matteo Fischetti and Andrea Lodi. Optimizing over the first chvátal closure. *Mathematical Programming*, 2007.

[67] A Sassano. On the facial structure of the set covering polytope. *Mathematical Programming*, 44, 1989.

[68] Shu Ming Ng Balas Egon. On the set covering polytope: I. all the facets with coefficients in 0, 1, 2. *Mathematical Programming 43*, pages 57–69, 1989.

[69] Shu Ming Ng Balas Egon. On the set covering polytope: Ii. lifting the facets with coefficients in 0, 1, 2. *Mathematical Programming 45*, pages 1–20, 1989.

[70] M. Sánchez-García, M. I. Sobrón, and B. Vitoriano. On the set covering polytope:facets with coefficients in $\{0, 1, 2, 3\}$. *Annals of Operations Research*, 81:343–356, 1998.

[71] Gérard Cornuéjols and Antonio Sassano. On the 0, 1 facets of the set covering polytope. *Math. Program.*, 43(1-3):45–55, 1989.

[72] Artur Pessoa, Ruslan Sadykov, Eduardo Uchoa, and François Vanderbeck. Automation and combination of linear-programming based stabilization techniques in column generation. *INFORMS Journal on Computing*, 30, 05 2018.

[73] José M. Valério de Carvalho. Using extra dual cuts to accelerate column generation. *INFORMS Journal on Computing 17*, pages 175–182, 2005.

[74] José Manuel Valério de Carvalho Hatem Ben Amor, Jacques Desrosiers. Dual-optimal inequalities for stabilized column generation. *Operations Research 54*, pages 454–463, 2006.

[75] Cláudio Alves and J.M. Valério de Carvalho. A stabilized branch-and-price-and-cut algorithm for the multiple length cutting stock problem. *Computers & Operations Research*, pages 1315–1328, 2008.

[76] François Clautiaux, Cláudio Alves, José Valério de Carvalho, and Jürgen Rietz. New stabilization procedures for the cutting stock problem. *INFORMS Journal on Computing 23*, 59(4):530–545, 2010.

[77] Simon Spoorendonk. *Cut and column generation : chapter 6*. PhD thesis, Faculty of science university of copenhagen, 2008.

[78] Gerald Gamrath, Daniel Anderson, Ksenia Bestuzheva, Wei-Kun Chen, Leon Eifler, Maxime Gasse, Patrick Gemander, Ambros Gleixner, Leona Gottwald, Katrin Halbig, Gregor Hendel, Christopher Hojny, Thorsten Koch, Pierre Le Bodic, Stephen J. Maher, Frederic Matter, Matthias Miltenberger, Erik Mühmer, Benjamin Müller, Marc E. Pfetsch, Franziska Schlösser, Felipe Serrano, Yuji Shinano, Christine Tawfik, Stefan Vigerske, Fabian Wegscheider, Dieter Weninger, and Jakob Witzig. The SCIP Optimization Suite 7.0. ZIB-Report 20-10, Zuse Institute Berlin, March 2020.

[79] https://www.ibm.com/docs/en/icos/20.1.0?topic=cplex-c-reference-manual.

[80] Farid Aliniaeifard, Carolina Benedetti, Nantel Bergeron, Shu Xiao Li, and Franco Saliola. Stable set polytopes and their 1-skeleta, 2021.

Appendix

Complete tables for the maximum co-2-plex problem

Algorithm	Instances set	name	CPU	# nodes	# cuts
N	Color02	1-FullIns_3	0.81	271	0
		1-FullIns_4	23.76	13549	15
		1-FullIns_5	3600.0	440093	703
		1-Insertions_4	9.65	6032	16
		1-Insertions_5	1330.43	339974	595
		1-Insertions_6	3600.0	210598	649
		2-FullIns_3	3.75	4518	5
		2-FullIns_4	1612.62	421961	48
		2-FullIns_5	3600.0	57759	2
		2-Insertions_3	1.05	378	2
		2-Insertions_4	379.37	182677	104
		2-Insertions_5	3600.0	189415	448
		3-FullIns_3	12.85	7852	7
		3-FullIns_4	3600.0	178940	279
		3-Insertions_3	4.16	4034	34
		3-Insertions_4	3600.0	468244	531
		3-Insertions_5	3600.0	19050	18
		4-FullIns_3	102.85	80555	127
		4-FullIns_4	3600.0	22078	7
		4-Insertions_3	27.39	46291	242
		4-Insertions_4	3600.0	79992	66
		anna	1.83	220	219
		david	1.56	570	411
		DSJC125.1	3600.0	4678016	142253
		DSJC125.5	3600.0	3401525	111212
		DSJC125.9	257.15	35000	89770
		DSJC250.1	3600.0	1383769	26356
		fpsol2.i.1	89.76	78831	18228
		homer	214.22	545252	55551

		huck	2.05	1275	120
		inithx.i.1	52.52	23050	27063
		inithx.i.2	14.49	1	0
		inithx.i.3	13.03	1	0
		jean	2.29	1904	731
		miles1000	10.34	3360	1345
		miles1500	3.26	37	310
		miles250	30.14	47284	4421
		miles500	4.44	13145	1803
		miles750	3.74	3205	812
		mug100_1	2.09	1	0
		mug100_25	2.08	1	42
		mug88_1	1.35	1	25
		mug88_25	1.28	1	34
		mulsol.i.1	11.73	1761	531
		mulsol.i.2	0.41	1	0
		mulsol.i.3	0.37	1	0
		mulsol.i.4	0.25	1	33
		mulsol.i.5	0.27	1	25
		myciel3	0.01	1	0
		myciel4	0.44	365	0
		myciel5	2.43	1989	8
		myciel6	25.96	23797	44
		myciel7	531.52	240259	310
		queen10_10	3600.0	12352964	430633
		queen11_11	3600.0	8005249	451381
		queen12_12	3600.0	5649691	204536
		queen13_13	3600.0	4560284	154732
		queen14_14	3600.0	3027060	113904
		queen15_15	3600.0	2828203	106755
		queen16_16	3600.0	2030277	62597
		queen5_5	0.9	3549	644
		queen6_6	2.77	7608	1547
		queen7_7	15.73	96060	6022
		queen8_12	3600.0	11530239	543401
		queen8_8	255.14	1386539	50427
		queen9_9	3389.33	13610897	526666
		r125.1c	0.04	1	0
		r125.1	2.92	2755	130
		r125.5	68.2	44961	6813
		r250.1c	2.11	2033	965
		r250.1	3600.0	2700059	557958
		zeroin.i.1	5.75	393	1355
		zeroin.i.2	1.14	164	231
		zeroin.i.3	1.1	146	248

		70_1177_num_9	247.34	904881	48219
		70_1193_num_2	262.44	967617	45304
		70_1197_num_5	244.9	864624	48384
		70_1201_num_8	180.56	564469	37902
		70_1216_num_6	212.78	747661	42079
		70_1216_num_7	211.88	699869	44191
		70_1218_num_0	145.23	393398	31732
		70_1219_num_3	202.19	676925	40682
		70_1241_num_1	141.52	359511	31216
		70_1245_num_4	191.09	561732	42981
		70_1657_num_9	99.77	171921	33757
		70_1665_num_2	68.27	84076	25191
		70_1676_num_8	87.85	127183	31086
		70_1680_num_3	83.65	134308	29764
		70_1681_num_1	108.76	138308	39585
		70_1682_num_4	80.75	114198	29055
		70_1691_num_5	68.97	55372	26900
		70_1706_num_6	96.19	108692	35372
		70_1712_num_0	68.34	80554	26035
		70_1712_num_7	105.18	165449	39211
		70_669_num_7	441.42	1299754	27256
		70_689_num_3	458.57	1504316	35633
		70_700_num_6	727.93	2618918	59628
		70_703_num_2	682.07	2397051	59460
		70_705_num_5	628.53	2159141	54876
		70_712_num_9	434.63	1430006	41591
		70_731_num_0	291.1	881183	20138
		70_735_num_4	577.34	1965454	45420
		70_738_num_1	616.46	2223312	64889
		70_743_num_8	601.69	2074172	55681
	random instances	80_1552_num_2	358.76	926695	59371
		80_1566_num_6	454.56	1411564	64309
		80_1569_num_3	426.88	1276818	67929
		80_1569_num_4	390.5	1076744	62504
		80_1573_num_5	441.27	1307404	68484
		80_1577_num_1	659.47	2256433	95174
		80_1584_num_0	406.98	1085194	66292
		80_1592_num_9	374.85	953655	66520
		80_1602_num_8	381.63	1020940	62247
		80_1627_num_7	362.38	897755	61614
		80_2169_num_4	168.59	177192	47635
		80_2200_num_8	254.28	367153	68838
		80_2202_num_5	186.53	182646	55201
		80_2203_num_2	243.37	345161	67333
		80_2207_num_1	232.13	311312	63032
		80_2215_num_7	242.26	303036	67767

		80_2217_num_0	228.73	280595	64440
		80_2220_num_3	220.83	261577	62840
		80_2232_num_9	228.17	294180	64439
		80_2258_num_6	222.16	264042	65843
		80_881_num_3	1149.15	3248857	73964
		80_907_num_7	3600.0	5633111	142953
		80_910_num_4	3600.0	5463889	140659
		80_912_num_9	3063.59	9890234	232482
		80_928_num_1	2378.02	7447972	170590
		80_938_num_0	1027.32	2828160	57200
		80_941_num_2	3600.0	12098654	310037
		80_949_num_6	1584.16	4850641	111032
		80_973_num_8	3199.14	10308566	264192
		80_973_num_5	3600.0	5900603	150673
		85_1039_num_1	3600.0	10639892	248299
		85_1047_num_0	3539.94	9673876	259402
		85_1052_num_8	3600.0	11172053	281244
		85_1055_num_7	3600.0	10978077	280324
		85_1057_num_4	3600.0	10894840	283422
		85_1059_num_5	3600.0	10901452	274126
		85_1072_num_6	3600.0	10182077	237902
		85_1075_num_3	3600.0	11068889	274420
		85_1088_num_9	3600.0	10634636	249169
		85_1093_num_2	3600.0	10616522	265427
		85_1735_num_3	1058.42	3372918	125616
		85_1763_num_4	530.7	1296383	72413
		85_1771_num_1	677.22	1840640	88849
		85_1781_num_7	468.5	969506	72036
		85_1785_num_5	743.09	2191119	99084
		85_1804_num_6	697.76	1836094	96044
		85_1818_num_8	612.38	1646019	84301
		85_1823_num_0	795.52	2154282	112083
		85_1872_num_9	710.75	1945512	98979
		85_1889_num_2	691.98	1855961	97335
		85_2452_num_7	267.27	260989	66051
		85_2453_num_0	266.62	262453	68160
		85_2471_num_9	345.95	488773	83385
		85_2472_num_6	361.15	445015	88548
		85_2480_num_5	320.59	425559	79840
		85_2485_num_2	313.34	467116	77544
		85_2507_num_1	305.73	344671	80554
		85_2512_num_4	321.07	396738	80631
		85_2521_num_3	303.26	349693	79311
		85_2524_num_8	315.95	409428	80622
		90_1158_num_2	3600.0	10038325	233480
		90_1168_num_9	3600.0	9143239	204933

		90_1173_num_0	3600.0	10369489	277123
		90_1177_num_5	3600.0	10517441	260123
		90_1199_num_7	3600.0	10563140	268206
		90_1200_num_1	3600.0	9688057	222462
		90_1204_num_8	3600.0	9548422	249569
		90_1206_num_6	3600.0	9462666	216890
		90_1214_num_3	3600.0	10589544	284130
		90_1218_num_4	3600.0	10524587	265560
		90_1985_num_9	904.88	2114077	107229
		90_1992_num_4	1279.37	3313346	136543
		90_1993_num_3	1076.15	2665139	122987
		90_1994_num_5	1212.45	3093574	137756
		90_2003_num_2	1138.48	2851473	127573
		90_2010_num_6	1135.87	2905049	125614
		90_2019_num_8	1093.53	2649323	121277
		90_2030_num_0	992.05	2403399	115701
		90_2036_num_7	945.55	2379172	110534
		90_2042_num_1	1066.83	2524025	123685
		90_2739_num_6	384.91	351477	87292
		90_2772_num_2	348.55	308739	80566
		90_2781_num_5	452.7	545444	103206
		90_2806_num_3	359.79	344629	83556
		90_2813_num_7	423.94	589981	96113
		90_2817_num_4	420.17	474794	98741
		90_2818_num_8	421.58	484569	98461
		90_2830_num_9	426.29	524150	96827
		90_2846_num_0	260.07	185819	64887
		90_2853_num_1	355.02	311460	87848
	inst_DIMACS	adjnoun.graph	11.15	4856	7350
		chesapeake.graph	0.7	55	232
		dolphins.graph	0.88	478	704
		football.graph	14.46	7211	14266
		jazz.graph	153.82	86090	32061
		karate.graph	0.11	6	57
		lesmis.graph	0.88	1051	80
		polbooks.graph	4.67	5240	3114
	comp_dimacs	adjnoun.graph_comp	73.43	148203	1440
		chesapeake.graph_comp	0.05	1	0
		dolphins.graph_comp	4.7	9152	424
		football.graph_comp	3600.0	5257279	477067
		jazz.graph_comp	3600.0	1720512	26280
		karate.graph_comp	0.03	1	0
		lesmis.graph_comp	2.1	2340	335
		polbooks.graph_comp	46.92	79706	4148
		1-FullIns_3	0.29	1	0

		1-FullIns_4	29.23	1	0
		1-FullIns_5	3600.0	1	2
		1-Insertions_4	7.43	1	3
		1-Insertions_5	272.3	30	37
		1-Insertions_6	3600.0	3071	1057
		2-FullIns_3	1.61	1	0
		2-FullIns_4	446.88	1	56
		2-FullIns_5	3600.0	1	0
		2-Insertions_3	0.29	1	3
		2-Insertions_4	63.06	51	35
		2-Insertions_5	3600.0	2877	531
		3-FullIns_3	6.18	1	8
		3-FullIns_4	3600.0	1	300
		3-Insertions_3	1.04	1	4
		3-Insertions_4	268.81	1178	419
		3-Insertions_5	3600.0	1253	423
		4-FullIns_3	25.84	1	24
		4-FullIns_4	3600.0	1	169
		4-Insertions_3	2.67	1	11
		4-Insertions_4	1095.93	986	289
		anna	0.67	1	0
		david	0.98	1	0
		DSJC125.1	3600.0	785847	5841
		DSJC125.5	3600.0	29765	5424
		DSJC125.9	3600.0	1	76
		DSJC250.1	3600.0	16311	868
		fpsol2.i.1	752.95	1	0
		homer	101.5	1	0
		huck	0.7	1	0
		inithx.i.1	302.86	1	0
		inithx.i.2	213.83	1	0
		inithx.i.3	200.13	1	0
		jean	0.63	1	0
		miles1000	528.4	1	0
		miles1500	1258.26	1	0
		miles250	1.32	1	0
		miles500	15.18	1	0
		miles750	83.04	1	0
		mug100_1	0.47	1	0
		mug100_25	0.46	1	0
		mug88_1	0.39	1	0
		mug88_25	0.36	1	0
		mulsol.i.1	738.03	1	0
		mulsol.i.2	0.48	1	0
		mulsol.i.3	0.54	1	0
		mulsol.i.4	0.29	1	0

		mulsol.i.5	0.32	1	0
		myciel3	0.02	1	0
		myciel4	0.2	1	0
		myciel5	5.28	1	22
		myciel6	89.22	1	53
		myciel7	3555.46	1	191
		queen10_10	764.7	18429	81069
		queen11_11	330.71	3370	1013
		queen12_12	2202.55	57452	17223
		queen13_13	3318.6	34001	7870
		queen14_14	3600.0	16158	1734
		queen15_15	3600.0	31812	466
		queen16_16	3600.0	10923	424
		queen5_5	3.7	1	201
		queen6_6	13.21	1	191
		queen7_7	25.49	19	1707
		queen8_12	870.1	11214	71633
		queen8_8	51.21	396	5661
		queen9_9	506.82	19225	71620
		r125.1c	0.14	1	0
		r125.1	0.86	1	0
		r125.5	3003.32	1	1
		r250.1c	17.61	1	3
		r250.1	9.15	1	0
		zeroin.i.1	12.49	1	0
		zeroin.i.2	10.67	1	0
		zeroin.i.3	10.26	1	0
		70_1177_num_9	1746.77	51555	61575
		70_1193_num_2	1748.78	49579	62809
		70_1197_num_5	1912.51	46598	62779
		70_1201_num_8	1007.36	16084	33027
		70_1216_num_6	1416.21	41410	50476
		70_1216_num_7	1445.42	40698	51034
		70_1218_num_0	706.81	5913	14487
		70_1219_num_3	1521.69	38811	52228
		70_1241_num_1	590.07	1626	4783
		70_1245_num_4	1307.7	26078	45228
		70_1657_num_9	1101.48	693	4601
		70_1665_num_2	931.87	1	828
		70_1676_num_8	1039.83	31	2087
		70_1680_num_3	1049.7	918	4737
		70_1681_num_1	1290.52	475	3396
		70_1682_num_4	1101.19	42	1370
		70_1691_num_5	893.92	1	317
		70_1706_num_6	1234.59	294	2551

		70_1712_num_0	963.76	1	453
		70_1712_num_7	1231.58	1856	4941
		70_669_num_7	346.8	19894	12983
		70_689_num_3	396.07	25857	13601
		70_700_num_6	641.38	57515	19835
		70_703_num_2	575.71	48200	19394
		70_705_num_5	530.91	47983	17575
		70_712_num_9	460.27	32288	16357
		70_731_num_0	351.4	16750	12908
		70_735_num_4	555.68	48553	15246
		70_738_num_1	565.1	40721	20173
		70_743_num_8	734.75	55514	24479
		80_1552_num_2	3600.0	46017	81325
		80_1566_num_6	3600.0	59580	78805
		80_1569_num_3	3600.0	53453	76867
		80_1569_num_4	3600.0	56061	75194
		80_1573_num_5	3600.0	47701	69133
		80_1577_num_1	3600.0	38812	54505
		80_1584_num_0	3600.0	53026	67991
		80_1592_num_9	3600.0	47694	78535
		80_1602_num_8	3600.0	42752	66816
		80_1627_num_7	3600.0	43041	85830
		80_2169_num_4	2206.71	7	1186
		80_2200_num_8	3600.0	21218	51737
		80_2202_num_5	2535.63	545	3639
		80_2203_num_2	3600.0	20236	41401
		80_2207_num_1	3559.93	11379	29809
		80_2215_num_7	3170.21	8978	26088
		80_2217_num_0	3356.57	7414	22320
		80_2220_num_3	3506.01	7657	19849
		80_2232_num_9	3107.94	7300	21929
		80_2258_num_6	3598.38	2913	11523
		80_881_num_3	1442.33	83212	27977
		80_907_num_7	3600.0	438646	30289
		80_910_num_4	3600.0	383643	41057
		80_912_num_9	3600.0	362581	37604
		80_928_num_1	3600.0	336603	39023
		80_938_num_0	1322.69	60662	32643
		80_941_num_2	3600.0	403266	31382
		80_949_num_6	2676.94	191760	44264
		80_973_num_8	3600.0	273419	39615
		80_973_num_5	2854.96	133295	51165
		85_1039_num_1	3600.0	311802	26702
		85_1047_num_0	3600.0	238458	33785
		85_1052_num_8	3600.0	385500	21709
		85_1055_num_7	3600.0	378023	21356

		85_1057_num_4	3600.0	374708	18168
		85_1059_num_5	3600.0	365935	22077
		85_1072_num_6	3600.0	285068	24237
		85_1075_num_3	3600.0	334686	18627
		85_1088_num_9	3600.0	298197	25342
		85_1093_num_2	3600.0	255065	36775
		85_1735_num_3	3600.0	29986	37849
		85_1763_num_4	3600.0	45685	64874
		85_1771_num_1	3600.0	29546	39970
		85_1781_num_7	3600.0	40388	68324
		85_1785_num_5	3600.0	32065	43064
		85_1804_num_6	3600.0	26294	45463
		85_1818_num_8	3600.0	32234	52172
		85_1823_num_0	3600.0	27265	46040
		85_1872_num_9	3600.0	21569	39787
		85_1889_num_2	3600.0	24644	42493
		85_2452_num_7	3600.0	3543	12130
		85_2453_num_0	3600.0	794	5760
		85_2471_num_9	3600.0	27299	49926
		85_2472_num_6	3600.0	28053	46383
		85_2480_num_5	3600.0	23290	48210
		85_2485_num_2	3600.0	24906	50765
		85_2507_num_1	3600.0	14081	33307
		85_2512_num_4	3600.0	17911	38565
		85_2521_num_3	3600.0	15834	34560
		85_2524_num_8	3600.0	20773	37338
		90_1158_num_2	3600.0	371507	14238
		90_1168_num_9	3600.0	243158	24774
		90_1173_num_0	3600.0	385789	11692
		90_1177_num_5	3600.0	382126	12178
		90_1199_num_7	3600.0	363135	15123
		90_1200_num_1	3600.0	300658	14081
		90_1204_num_8	3600.0	268111	18090
		90_1206_num_6	3600.0	286787	16461
		90_1214_num_3	3600.0	321229	14413
		90_1218_num_4	3600.0	326761	13429
		90_1985_num_9	3600.0	36456	45667
		90_1992_num_4	3600.0	27063	31235
		90_1993_num_3	3600.0	29918	34946
		90_1994_num_5	3600.0	20159	33377
		90_2003_num_2	3600.0	26075	32996
		90_2010_num_6	3600.0	33298	36034
		90_2019_num_8	3600.0	27168	38070
		90_2030_num_0	3600.0	26555	41548
		90_2036_num_7	3600.0	20873	35796
		90_2042_num_1	3600.0	19712	34906

		90_2739_num_6	3600.0	4549	16328
		90_2772_num_2	3600.0	388	4058
		90_2781_num_5	3600.0	22421	42173
		90_2806_num_3	3600.0	1080	5302
		90_2813_num_7	3600.0	21514	41201
		90_2817_num_4	3600.0	20158	41893
		90_2818_num_8	3600.0	19851	41866
		90_2830_num_9	3600.0	21255	35634
		90_2846_num_0	3600.0	1102	4832
		90_2853_num_1	3600.0	404	3861
	inst_DIMACS	adjnoun.graph	731.47	1	26
		chesapeake.graph	3.78	1	0
		dolphins.graph	6.45	1	3
		football.graph	927.03	1	3
		jazz.graph	3600.0	1	9
		karate.graph	0.84	1	0
		lesmis.graph	3.29	1	0
		polbooks.graph	133.78	1	3
	comp_dimacs	adjnoun.graph_comp	43.38	313	98
		chesapeake.graph_comp	0.08	1	0
		dolphins.graph_comp	0.65	1	0
		football.graph_comp	24.29	157	194
		jazz.graph_comp	2621.26	8356	2151
		karate.graph_comp	0.02	1	0
		lesmis.graph_comp	1.18	1	0
		polbooks.graph_comp	6.24	1	0
		1-FullIns_3	0.17	1	0
		1-FullIns_4	24.76	1	0
		1-FullIns_5	1781.02	1	0
		1-Insertions_4	4.02	1	0
		1-Insertions_5	304.97	1	0
		1-Insertions_6	3600.0	5270	0
		2-FullIns_3	1.05	1	0
		2-FullIns_4	322.47	1	0
		2-FullIns_5	3600.0	1	0
		2-Insertions_3	0.29	1	0
		2-Insertions_4	20.83	1	0
		2-Insertions_5	3600.0	5931	0
		3-FullIns_3	3.61	1	0
		3-FullIns_4	3600.0	1	0
		3-Insertions_3	1.12	1	0
		3-Insertions_4	218.36	1	0
		3-Insertions_5	3600.0	2126	0
		4-FullIns_3	12.63	1	0
	Color02	4-FullIns_4	3600.0	1	0

		4-Insertions_3	3.12	1	0
		4-Insertions_4	1006.56	31	0
		anna	0.36	1	0
		david	0.37	1	0
		DSJC125.1	3600.0	4340649	0
		DSJC125.5	3600.0	811442	0
		DSJC125.9	3600.0	111656	0
		DSJC250.1	3600.0	365528	0
		fpsol2.i.1	217.64	1	0
		homer	67.44	1	0
		huck	0.25	1	0
		inithx.i.1	104.08	1	0
		inithx.i.2	138.69	1	0
		inithx.i.3	127.63	1	0
		jean	0.26	1	0
		miles1000	58.52	1	0
		miles1500	3600.0	59052	0
		miles250	0.68	1	0
		miles500	9.05	1	0
		miles750	24.98	1	0
		mug100_1	0.43	1	0
		mug100_25	0.49	1	0
		mug88_1	0.43	1	0
		mug88_25	0.39	1	0
		mulsol.i.1	126.13	1	0
		mulsol.i.2	0.35	1	0
		mulsol.i.3	0.36	1	0
		mulsol.i.4	0.48	1	0
		mulsol.i.5	0.59	1	0
		myciel3	0.04	1	0
		myciel4	0.16	1	0
		myciel5	3.96	1	0
		myciel6	59.55	1	0
		myciel7	1039.68	1	0
		queen10_10	597.48	22149	0
		queen11_11	75.83	199	0
		queen12_12	202.41	14902	0
		queen13_13	221.51	4623	0
		queen14_14	255.02	2162	0
		queen15_15	544.6	39164	0
		queen16_16	544.31	3020	0
		queen5_5	2.11	1	0
		queen6_6	8.31	1	0
		queen7_7	49.26	1	0
		queen8_12	571.72	87226	0
		queen8_8	103.62	1	0

		queen9_9	305.99	18836	0
		r125.1c	0.13	1	0
		r125.1	1.09	1	0
		r125.5	199.84	1	0
		r250.1c	7.81	1	0
		r250.1	5.25	1	0
		zeroin.i.1	1.58	1	0
		zeroin.i.2	1.63	1	0
		zeroin.i.3	1.57	1	0
	random instances	70_1177_num_9	1099.64	72524	0
		70_1193_num_2	1066.75	72276	0
		70_1197_num_5	901.87	70204	0
		70_1201_num_8	894.16	19202	0
		70_1216_num_6	834.79	59644	0
		70_1216_num_7	848.12	62463	0
		70_1218_num_0	691.61	5620	0
		70_1219_num_3	821.36	59526	0
		70_1241_num_1	605.4	1481	0
		70_1245_num_4	831.24	48583	0
		70_1657_num_9	619.03	1	0
		70_1665_num_2	492.82	1	0
		70_1676_num_8	600.97	1	0
		70_1680_num_3	664.04	1	0
		70_1681_num_1	640.67	1	0
		70_1682_num_4	603.51	1	0
		70_1691_num_5	496.64	1	0
		70_1706_num_6	628.2	1	0
		70_1712_num_0	597.72	1	0
		70_1712_num_7	614.4	1	0
		70_669_num_7	278.79	18497	0
		70_689_num_3	340.61	29685	0
		70_700_num_6	360.84	57296	0
		70_703_num_2	374.82	59368	0
		70_705_num_5	369.65	60499	0
		70_712_num_9	329.96	21658	0
		70_731_num_0	345.46	19625	0
		70_735_num_4	387.65	49674	0
		70_738_num_1	388.2	34907	0
		70_743_num_8	396.05	56060	0
		80_1552_num_2	1140.66	59979	0
		80_1566_num_6	1362.88	107528	0
		80_1569_num_3	1373.56	103662	0
		80_1569_num_4	1450.56	102718	0
		80_1573_num_5	1418.23	105071	0
		80_1577_num_1	1594.77	171265	0

		80_1584_num_0	1356.96	93624	0
		80_1592_num_9	1361.33	93206	0
		80_1602_num_8	1341.42	87723	0
		80_1627_num_7	1377.51	79203	0
		80_2169_num_4	2916.3	5307	0
		80_2200_num_8	2590.75	8037	0
		80_2202_num_5	2814.85	3798	0
		80_2203_num_2	3600.0	14951	0
		80_2207_num_1	2927.62	1	0
		80_2215_num_7	3493.78	3169	0
		80_2217_num_0	3434.55	10703	0
		80_2220_num_3	3466.89	1	0
		80_2232_num_9	3564.81	9347	0
		80_2258_num_6	3257.92	10597	0
		80_881_num_3	585.83	49485	0
		80_907_num_7	1606.98	868863	0
		80_910_num_4	774.93	218238	0
		80_912_num_9	867.89	244083	0
		80_928_num_1	823.03	200101	0
		80_938_num_0	744.85	62225	0
		80_941_num_2	1176.22	565238	0
		80_949_num_6	765.29	118520	0
		80_973_num_8	923.65	215782	0
		80_973_num_5	861.95	82342	0
		85_1039_num_1	1630.45	474600	0
		85_1047_num_0	1061.19	152938	0
		85_1052_num_8	2232.43	1147035	0
		85_1055_num_7	1749.97	688842	0
		85_1057_num_4	2321.92	1157032	0
		85_1059_num_5	2617.91	1494446	0
		85_1072_num_6	1864.16	555970	0
		85_1075_num_3	3600.0	2129909	0
		85_1088_num_9	1401.39	351061	0
		85_1093_num_2	1151.9	217210	0
		85_1735_num_3	1791.3	204890	0
		85_1763_num_4	1745.37	101900	0
		85_1771_num_1	1933.14	171729	0
		85_1781_num_7	1676.89	92370	0
		85_1785_num_5	1855.34	149379	0
		85_1804_num_6	1833.17	147910	0
		85_1818_num_8	1800.92	144178	0
		85_1823_num_0	1725.95	140145	0
		85_1872_num_9	2180.85	176265	0
		85_1889_num_2	2241.75	169359	0
		85_2452_num_7	3600.0	179340	0
		85_2453_num_0	3600.0	19502	0

		85_2471_num_9	3600.0	54054	0
		85_2472_num_6	3600.0	53020	0
		85_2480_num_5	3600.0	166480	0
		85_2485_num_2	3600.0	50644	0
		85_2507_num_1	3600.0	149927	0
		85_2512_num_4	3600.0	162622	0
		85_2521_num_3	3600.0	152791	0
		85_2524_num_8	3600.0	162161	0
		90_1158_num_2	3600.0	1982501	0
		90_1168_num_9	1625.16	334181	0
		90_1173_num_0	3600.0	2681958	0
		90_1177_num_5	3600.0	2548747	0
		90_1199_num_7	3600.0	2460100	0
		90_1200_num_1	3342.94	1395841	0
		90_1204_num_8	2446.35	657422	0
		90_1206_num_6	3375.85	1421037	0
		90_1214_num_3	3600.0	2172342	0
		90_1218_num_4	3600.0	2216155	0
		90_1985_num_9	2565.47	161299	0
		90_1992_num_4	2702.77	246502	0
		90_1993_num_3	2498.23	216085	0
		90_1994_num_5	2648.27	277362	0
		90_2003_num_2	2580.8	243275	0
		90_2010_num_6	2729.97	225512	0
		90_2019_num_8	2873.05	206894	0
		90_2030_num_0	2727.52	207710	0
		90_2036_num_7	2600.74	199601	0
		90_2042_num_1	2703.36	206049	0
		90_2739_num_6	3600.0	133740	0
		90_2772_num_2	3600.0	139446	0
		90_2781_num_5	3600.0	145532	0
		90_2806_num_3	3600.0	112721	0
		90_2813_num_7	3600.0	150535	0
		90_2817_num_4	3600.0	118911	0
		90_2818_num_8	3600.0	122280	0
		90_2830_num_9	3600.0	134771	0
		90_2846_num_0	3600.0	116602	0
		90_2853_num_1	3600.0	114681	0
	inst_DIMACS	adjnoun.graph	232.16	8228	0
		chesapeake.graph	4.67	1	0
		dolphins.graph	5.39	1	0
		football.graph	485.02	40897	0
		jazz.graph	3600.0	33696	0
		karate.graph	0.58	1	0
		lesmis.graph	1.81	1	0

		polbooks.graph	67.76	1	0
	comp_dimacs	adjnoun.graph_comp	16.99	1	0
		chesapeake.graph_comp	0.1	1	0
		dolphins.graph_comp	0.58	1	0
		football.graph_comp	13.27	1	0
		jazz.graph_comp	1052.91	648	0
		karate.graph_comp	0.03	1	0
		lesmis.graph_comp	0.51	1	0
		polbooks.graph_comp	3.44	1	0
P	Color02	1-FullIns_3	0.61	1	30
		1-FullIns_4	41.76	1	485
		1-FullIns_5	2934.5	1790	6530
		1-Insertions_4	12.12	1	413
		1-Insertions_5	349.57	374	3527
		1-Insertions_6	3600.0	2581	2979
		2-FullIns_3	5.78	1	247
		2-FullIns_4	581.05	24	2737
		2-FullIns_5	3600.0	591	621
		2-Insertions_3	0.65	1	74
		2-Insertions_4	88.75	353	1706
		2-Insertions_5	3600.0	2348	4175
		3-FullIns_3	25.41	3	852
		3-FullIns_4	3600.0	238	5050
		3-Insertions_3	2.3	1	108
		3-Insertions_4	613.79	2736	3807
		3-Insertions_5	3600.0	1306	1148
		4-FullIns_3	95.36	472	1901
		4-FullIns_4	3600.0	731	1611
		4-Insertions_3	4.81	1	140
		4-Insertions_4	3015.9	4079	6054
		anna	1.69	1	133
		david	3.61	1	104
		DSJC125.1	3600.0	250095	49976
		DSJC125.5	3600.0	44829	24494
		DSJC125.9	3600.0	6997	6189
		DSJC250.1	3600.0	31455	17554
		fpsol2.i.1	3180.91	4449	2322
		homer	1177.47	58751	2881
		huck	15.86	168	18
		inithx.i.1	1021.78	3748	2124
		inithx.i.2	91.79	7	276
		inithx.i.3	46.93	1	270
		jean	7.34	7	143
		miles1000	386.88	168	2092
		miles1500	672.85	58	1909

		miles250	27.9	1569	313
		miles500	30.69	20	1191
		miles750	71.27	30	623
		mug100_1	3.39	1	2
		mug100_25	5.45	76	166
		mug88_1	2.01	1	16
		mug88_25	2.4	1	93
		mulsol.i.1	1800.98	645	474
		mulsol.i.2	1.29	1	0
		mulsol.i.3	1.67	1	0
		mulsol.i.4	1.03	1	33
		mulsol.i.5	1.15	1	25
		myciel3	0.03	1	10
		myciel4	0.35	1	46
		myciel5	12.08	1	832
		myciel6	84.77	1	1258
		myciel7	999.89	687	4210
		queen10_10	627.07	11376	12009
		queen11_11	3600.0	5014	103599
		queen12_12	2464.13	13206	17249
		queen13_13	3600.0	11926	15831
		queen14_14	3600.0	6410	9901
		queen15_15	3600.0	4296	6231
		queen16_16	3600.0	1907	4192
		queen5_5	6.6	872	1351
		queen6_6	18.46	1947	2475
		queen7_7	50.66	5226	4865
		queen8_12	489.56	9377	10524
		queen8_8	152.28	8636	7820
		queen9_9	316.8	11409	10362
		r125.1c	0.13	1	12
		r125.1	5.3	1	2
		r125.5	3600.0	1125	6214
		r250.1c	27.82	780	3322
		r250.1	3600.0	157026	7945
		zeroin.i.1	43.58	254	293
		zeroin.i.2	13.68	1	196
		zeroin.i.3	12.75	13	191
		70_1177_num_9	857.95	67307	38215
		70_1193_num_2	762.93	58721	33875
		70_1197_num_5	817.38	61939	35357
		70_1201_num_8	797.16	58753	33746
		70_1216_num_6	904.62	66114	36894
		70_1216_num_7	844.33	61830	35402
		70_1218_num_0	729.3	51351	30120

		70_1219_num_3	900.5	62816	36349
		70_1241_num_1	815.05	54399	31579
		70_1245_num_4	863.54	60731	34662
		70_1657_num_9	1196.73	44415	26155
		70_1665_num_2	1062.94	39128	23353
		70_1676_num_8	1214.53	43477	25510
		70_1680_num_3	1164.31	39974	23856
		70_1681_num_1	1342.99	52580	29608
		70_1682_num_4	1241.52	44544	26057
		70_1691_num_5	1151.52	38416	23761
		70_1706_num_6	1335.95	46791	27224
		70_1712_num_0	1174.2	37788	23229
		70_1712_num_7	1345.17	47321	27283
		70_669_num_7	401.13	48828	26079
		70_689_num_3	382.24	45733	24841
		70_700_num_6	506.18	61155	32279
		70_703_num_2	424.66	50081	27776
		70_705_num_5	393.7	45320	25435
		70_712_num_9	395.44	44463	24725
		70_731_num_0	363.38	36740	21279
		70_735_num_4	485.84	57315	29588
		70_738_num_1	429.32	50835	27071
		70_743_num_8	459.5	53224	28881
		80_1552_num_2	2156.12	110877	63160
		80_1566_num_6	1944.33	105324	58601
		80_1569_num_3	2107.74	109653	62384
		80_1569_num_4	2544.38	124723	72187
		80_1573_num_5	2435.3	117662	68989
		80_1577_num_1	2423.71	120915	67832
		80_1584_num_0	2638.13	126385	72360
		80_1592_num_9	2190.55	111451	62570
		80_1602_num_8	2277.58	113017	64596
		80_1627_num_7	2054.83	98594	57520
		80_2169_num_4	2802.39	80178	44904
		80_2200_num_8	3236.85	81003	47426
		80_2202_num_5	2811.67	74027	42857
		80_2203_num_2	3107.51	81550	46756
		80_2207_num_1	3351.82	84119	48201
		80_2215_num_7	2726.94	74351	42040
		80_2217_num_0	3361.21	83968	47493
		80_2220_num_3	2855.7	78406	44015
		80_2232_num_9	2911.66	74727	42796
		80_2258_num_6	3234.44	79110	44234
		80_881_num_3	925.56	83745	42516
		80_907_num_7	1245.59	112816	57885
		80_910_num_4	1291.66	109791	57514

		80_912_num_9	1353.09	115790	59930
		80_928_num_1	1462.32	127860	64486
		80_938_num_0	1162.1	94364	51073
		80_941_num_2	1624.13	127295	70746
		80_949_num_6	1188.11	99372	52356
		80_973_num_8	1638.0	135637	70777
		80_973_num_5	1557.12	118662	66066
		85_1039_num_1	2432.34	163672	87667
		85_1047_num_0	2305.25	157587	84244
		85_1052_num_8	2462.52	166171	90408
		85_1055_num_7	2965.08	189680	106708
		85_1057_num_4	2421.32	163489	88222
		85_1059_num_5	2262.6	154286	82710
		85_1072_num_6	2929.54	181634	104804
		85_1075_num_3	3005.87	181007	104553
		85_1088_num_9	2301.47	147727	82442
		85_1093_num_2	2122.49	139994	76937
		85_1735_num_3	3600.0	146159	85906
		85_1763_num_4	3600.0	139989	84808
		85_1771_num_1	3600.0	140484	82653
		85_1781_num_7	3600.0	148217	85494
		85_1785_num_5	3600.0	134528	81677
		85_1804_num_6	3600.0	133525	80849
		85_1818_num_8	3239.36	126195	69710
		85_1823_num_0	2831.09	113026	62318
		85_1872_num_9	3600.0	132919	76573
		85_1889_num_2	2959.0	111300	59801
		85_2452_num_7	3600.0	77682	45960
		85_2453_num_0	3600.0	82231	47090
		85_2471_num_9	3600.0	78227	44978
		85_2472_num_6	3600.0	79750	44977
		85_2480_num_5	3600.0	77601	43585
		85_2485_num_2	3600.0	77609	44399
		85_2507_num_1	3600.0	74737	42468
		85_2512_num_4	3600.0	71511	41034
		85_2521_num_3	3600.0	71526	41835
		85_2524_num_8	3600.0	72859	41514
		90_1158_num_2	3600.0	203966	116044
		90_1168_num_9	3596.1	203244	114930
		90_1173_num_0	3600.0	199135	114366
		90_1177_num_5	3600.0	190926	113741
		90_1199_num_7	3600.0	191770	118427
		90_1200_num_1	3600.0	191062	114478
		90_1204_num_8	3600.0	188087	112272
		90_1206_num_6	3600.0	191875	111630
		90_1214_num_3	3600.0	191018	110477

		90_1218_num_4	3600.0	189642	114089
		90_1985_num_9	3600.0	114216	64307
		90_1992_num_4	3600.0	108199	70849
		90_1993_num_3	3600.0	112022	70873
		90_1994_num_5	3600.0	113523	65139
		90_2003_num_2	3600.0	111058	67149
		90_2010_num_6	3600.0	111507	65451
		90_2019_num_8	3600.0	110098	68950
		90_2030_num_0	3600.0	113362	63697
		90_2036_num_7	3600.0	115655	63860
		90_2042_num_1	3600.0	116707	63967
		90_2739_num_6	3600.0	68408	39451
		90_2772_num_2	3600.0	65356	36862
		90_2781_num_5	3600.0	64540	35497
		90_2806_num_3	3600.0	61376	35973
		90_2813_num_7	3600.0	63205	35333
		90_2817_num_4	3600.0	60256	34887
		90_2818_num_8	3600.0	64374	36101
		90_2830_num_9	3600.0	59595	32799
		90_2846_num_0	3600.0	61599	35437
		90_2853_num_1	3600.0	54117	32346
	inst_DIMACS	adjnoun.graph	759.33	4621	6558
		chesapeake.graph	18.27	355	1513
		dolphins.graph	16.98	507	1575
		football.graph	697.45	6049	8129
		jazz.graph	3600.0	1888	1498
		karate.graph	2.37	19	285
		lesmis.graph	9.04	141	910
		polbooks.graph	193.19	3155	5570
	comp_dimacs	adjnoun.graph_comp	66.62	1149	1877
		chesapeake.graph_comp	0.09	1	5
		dolphins.graph_comp	8.55	24	169
		football.graph_comp	1103.0	124644	20158
		jazz.graph_comp	3600.0	54734	24149
		karate.graph_comp	0.04	1	0
		lesmis.graph_comp	9.1	74	221
		polbooks.graph_comp	26.18	1	423
		1-FullIns_3	0.64	1	0
		1-FullIns_4	18.6	1	0
		1-FullIns_5	1419.16	1	0
		1-Insertions_4	6.41	1	0
		1-Insertions_5	305.37	11	0
		1-Insertions_6	3600.0	1511	0
		2-FullIns_3	4.71	1	0
		2-FullIns_4	387.6	1	0

		2-FullIns_5	3600.0	493	0
		2-Insertions_3	0.71	1	0
		2-Insertions_4	59.49	21	0
		2-Insertions_5	3600.0	1065	0
		3-FullIns_3	12.53	1	0
		3-FullIns_4	3600.0	873	0
		3-Insertions_3	2.46	1	0
		3-Insertions_4	609.64	1771	0
		3-Insertions_5	3600.0	295	0
		4-FullIns_3	49.91	1	0
		4-FullIns_4	3600.0	543	0
		4-Insertions_3	5.27	1	0
		4-Insertions_4	2922.94	3386	0
		anna	2.91	1	0
		david	4.93	1	0
		DSJC125.1	3600.0	818698	0
		DSJC125.5	3600.0	1767500	0
		DSJC125.9	1954.1	58368	0
		DSJC250.1	3600.0	190966	0
		fpsol2.i.1	542.69	2698	0
		homer	1103.28	1107813	0
		huck	6.59	1	0
		inithx.i.1	482.28	394	0
		inithx.i.2	76.18	1	0
		inithx.i.3	53.59	1	0
		jean	6.51	1	0
		miles1000	211.04	3513	0
		miles1500	278.88	847	0
		miles250	31.27	1714	0
		miles500	18.3	1	0
		miles750	54.69	877	0
		mug100_1	3.8	1	0
		mug100_25	5.15	1	0
		mug88_1	2.41	1	0
		mug88_25	3.16	1	0
		mulsol.i.1	152.48	1	0
		mulsol.i.2	1.82	1	0
		mulsol.i.3	2.37	1	0
		mulsol.i.4	1.84	1	0
		mulsol.i.5	1.99	1	0
		myciel3	0.03	1	0
		myciel4	0.42	1	0
		myciel5	4.35	1	0
		myciel6	40.41	1	0
		myciel7	483.48	1	0
		queen10_10	3600.0	6441542	0

		queen11_11	3600.0	3698265	0
		queen12_12	3600.0	1625200	0
		queen13_13	3600.0	929582	0
		queen14_14	3600.0	692228	0
		queen15_15	3600.0	475407	0
		queen16_16	3600.0	301347	0
		queen5_5	2.34	1	0
		queen6_6	3.64	1	0
		queen7_7	21.64	1	0
		queen8_12	3600.0	5814920	0
		queen8_8	160.24	261068	0
		queen9_9	1391.34	2033809	0
		r125.1c	0.14	1	0
		r125.1	4.12	1	0
		r125.5	3600.0	1677634	0
		r250.1c	10.62	1	0
		r250.1	3600.0	598880	0
		zeroin.i.1	18.18	1	0
		zeroin.i.2	11.4	1	0
		zeroin.i.3	11.09	1	0
		70_1177_num_9	152.17	1933	0
		70_1193_num_2	146.1	2289	0
		70_1197_num_5	148.65	1	0
		70_1201_num_8	137.92	1	0
		70_1216_num_6	153.81	1993	0
		70_1216_num_7	141.19	1	0
		70_1218_num_0	134.68	1	0
		70_1219_num_3	158.69	1557	0
		70_1241_num_1	130.35	1	0
		70_1245_num_4	136.59	1	0
		70_1657_num_9	197.45	736	0
		70_1665_num_2	173.78	477	0
		70_1676_num_8	182.44	1	0
		70_1680_num_3	192.88	895	0
		70_1681_num_1	208.41	2200	0
		70_1682_num_4	169.66	933	0
		70_1691_num_5	178.16	1	0
		70_1706_num_6	184.82	687	0
		70_1712_num_0	200.94	2086	0
		70_1712_num_7	195.19	1130	0
		70_669_num_7	131.06	70514	0
		70_689_num_3	153.03	93508	0
		70_700_num_6	222.23	217928	0
		70_703_num_2	219.68	219880	0
		70_705_num_5	224.66	218964	0

random instances

		70_712_num_9	147.74	81851	0
		70_731_num_0	114.15	28312	0
		70_735_num_4	181.69	125283	0
		70_738_num_1	192.82	155971	0
		70_743_num_8	201.14	154699	0
		80_1552_num_2	271.42	12005	0
		80_1566_num_6	315.68	48192	0
		80_1569_num_3	304.52	30941	0
		80_1569_num_4	337.17	45790	0
		80_1573_num_5	370.18	101015	0
		80_1577_num_1	671.43	399716	0
		80_1584_num_0	320.1	52674	0
		80_1592_num_9	271.73	19435	0
		80_1602_num_8	322.81	59367	0
		80_1627_num_7	285.43	38350	0
		80_2169_num_4	376.89	33126	0
		80_2200_num_8	443.72	85728	0
		80_2202_num_5	426.54	36696	0
		80_2203_num_2	416.84	63821	0
		80_2207_num_1	391.18	51591	0
		80_2215_num_7	412.24	72909	0
		80_2217_num_0	418.92	55786	0
		80_2220_num_3	425.44	54882	0
		80_2232_num_9	430.14	50889	0
		80_2258_num_6	397.22	57054	0
		80_881_num_3	464.2	281140	0
		80_907_num_7	1082.97	735497	0
		80_910_num_4	1014.08	646275	0
		80_912_num_9	1003.83	647716	0
		80_928_num_1	805.54	543582	0
		80_938_num_0	439.29	251330	0
		80_941_num_2	1194.16	821187	0
		80_949_num_6	554.43	419268	0
		80_973_num_8	1048.23	632592	0
		80_973_num_5	716.24	401493	0
		85_1039_num_1	1397.86	750749	0
		85_1047_num_0	1398.03	691090	0
		85_1052_num_8	1554.44	904033	0
		85_1055_num_7	1669.08	1017078	0
		85_1057_num_4	1655.51	980675	0
		85_1059_num_5	1712.38	1043099	0
		85_1072_num_6	1633.06	914120	0
		85_1075_num_3	1759.35	1083605	0
		85_1088_num_9	1487.66	777035	0
		85_1093_num_2	1315.8	680837	0
		85_1735_num_3	916.62	434951	0

		85_1763_num_4	406.75	23907	0
		85_1771_num_1	736.51	230411	0
		85_1781_num_7	462.01	52378	0
		85_1785_num_5	813.81	253742	0
		85_1804_num_6	665.14	157692	0
		85_1818_num_8	514.9	280091	0
		85_1823_num_0	605.26	400896	0
		85_1872_num_9	631.41	192195	0
		85_1889_num_2	515.8	297740	0
		85_2452_num_7	563.29	49992	0
		85_2453_num_0	541.56	20667	0
		85_2471_num_9	654.22	136096	0
		85_2472_num_6	537.6	32407	0
		85_2480_num_5	560.12	48092	0
		85_2485_num_2	561.57	125963	0
		85_2507_num_1	598.94	60041	0
		85_2512_num_4	609.03	81545	0
		85_2521_num_3	580.97	78467	0
		85_2524_num_8	633.65	99988	0
		90_1158_num_2	2173.63	1177766	0
		90_1168_num_9	1667.07	716350	0
		90_1173_num_0	2600.79	1806907	0
		90_1177_num_5	2528.04	1705655	0
		90_1199_num_7	2464.71	1629450	0
		90_1200_num_1	2169.96	1082383	0
		90_1204_num_8	2154.05	1048830	0
		90_1206_num_6	2072.26	1010182	0
		90_1214_num_3	2334.79	1394870	0
		90_1218_num_4	2411.08	1506214	0
		90_1985_num_9	1230.57	499539	0
		90_1992_num_4	1506.72	527780	0
		90_1993_num_3	1449.59	503979	0
		90_1994_num_5	1189.45	642058	0
		90_2003_num_2	1378.19	642642	0
		90_2010_num_6	1337.63	635474	0
		90_2019_num_8	1012.32	245826	0
		90_2030_num_0	838.49	423694	0
		90_2036_num_7	775.76	409359	0
		90_2042_num_1	1035.82	490698	0
		90_2739_num_6	876.49	92703	0
		90_2772_num_2	747.78	21487	0
		90_2781_num_5	908.45	115052	0
		90_2806_num_3	806.25	70188	0
		90_2813_num_7	963.36	178884	0
		90_2817_num_4	935.41	155164	0
		90_2818_num_8	776.32	39622	0

		90_2830_num_9	781.67	50868	0
		90_2846_num_0	775.07	60745	0
		90_2853_num_1	660.19	16268	0
	inst_DIMACS	adjnoun.graph	103.16	139	0
		chesapeake.graph	2.55	1	0
		dolphins.graph	3.93	1	0
		football.graph	50.05	1	0
		jazz.graph	3600.0	35632	0
		karate.graph	1.83	1	0
		lesmis.graph	2.79	1	0
		polbooks.graph	21.94	1	0
	comp_dimacs	adjnoun.graph_comp	23.28	1	0
		chesapeake.graph_comp	0.16	1	0
		dolphins.graph_comp	5.58	1	0
		football.graph_comp	2198.85	522907	0
		jazz.graph_comp	2390.87	114206	0
		karate.graph_comp	0.09	1	0
		lesmis.graph_comp	4.11	1	0
		polbooks.graph_comp	22.54	1	0

Complete tables for the coloring problem

Algorithm	Instance	CPU	# col	# node	% pricing	% separation	# cuts	# DOI
		1-FullIns_3	5.0	12	1	0.02	0.0	0
		2-FullIns_3	6.0	17	1	0.03	0.0	0
		3-FullIns_3	5.0	19	1	0.04	0.0	0
		4-FullIns_3	6.0	26	1	0.05	0.0	0
		5-FullIns_3	7.0	35	1	0.03	0.0	0
		anna	24.0	70	1	0.02	0.0	0
		david	16.0	62	1	0.02	0.0	0
		DSJC125.9	95972.0	428	142	0.1	0.0	0
		fpsol2.i.1	2040.0	510	3	0.08	0.0	0
	Color02	games120	43563.0	2102	604	0.01	0.0	0
		huck	54.0	71	14	0.0	0.0	0
		jean	5.0	49	1	0.0	0.0	0
		miles1000	239.0	201	8	0.05	0.0	0
		miles1500	267.0	225	1	0.08	0.0	0
		miles250	15073.0	1269	380	0.0	0.0	0
		miles500	281.0	163	12	0.01	0.0	0
		miles750	41.0	156	1	0.03	0.0	0
		mulsol.i.1	103.0	214	6	0.01	0.0	0
		mulsol.i.2	25.0	161	1	0.01	0.0	0
		mulsol.i.3	37.0	175	1	0.01	0.0	0
		mulsol.i.4	24.0	166	1	0.01	0.0	0
baseline		mulsol.i.5	36.0	162	1	0.01	0.0	0
		myciel3	29.0	19	5	0.02	0.0	0
		myciel4	5312.0	1173	829	0.04	0.0	0

		queen5_5	6.0	16	1	0.03	0.0	0
		queen6_6	218.0	130	3	0.07	0.0	0
		queen7_7	100.0	139	1	0.05	0.0	0
		queen8_12	149637.0	4902	371	0.07	0.0	0
		r125.1c	10.0	51	1	0.08	0.0	0
		fpsol2.i.2	4.28128e6	45060	277	0.01	0.0	0
		fpsol2.i.3	5.9248e6	43984	423	0.01	0.0	0
		r125.1	3.03222e6	17327	3956	0.0	0.0	0
		r125.5	3257.0	554	11	0.08	0.0	0
		r250.1c	421.0	117	1	0.09	0.0	0
		r250.5	154891.0	2290	112	0.07	0.0	0
		zeroin.i.1	90.0	164	1	0.07	0.0	0
		zeroin.i.2	18.0	134	1	0.03	0.0	0
		zeroin.i.3	18.0	134	1	0.02	0.0	0
	DIMACS	adjnoun.graph	431.0	132	1	0.09	0.0	0
		celegans_metabolic.graph	761.0	261	1	0.1	0.0	0
		chesapeake.graph	10.0	39	1	0.03	0.0	0
		dolphins.graph	7.0	40	1	0.07	0.0	0
		football.graph	398.0	135	1	0.09	0.0	0
		jazz.graph	58755.0	451	10	0.1	0.0	0
		karate.graph	4.0	22	1	0.05	0.0	0
		lesmis.graph	3.0	33	1	0.03	0.0	0
		polbooks.graph	15.0	61	1	0.08	0.0	0
	comp-DIMACS	adjnoun.graph_comp	58194.0	4481	26	0.08	0.0	0
		chesapeake.graph_comp	8.0	20	1	0.04	0.0	0
		dolphins.graph_comp	394.0	296	44	0.01	0.0	0
		jazz.graph_comp	370942.0	13034	67	0.06	0.0	0
		karate.graph_comp	3.0	14	1	0.03	0.0	0

		lesmis.graph_comp	68.0	91	3	0.03	0.0	0
		polbooks.graph_comp	15046.0	2444	35	0.06	0.0	0
		1-FullIns_3	5.0	12	1	0.0	0.0	0
		2-FullIns_3	6.0	17	1	0.03	0.0	0
		3-FullIns_3	5.0	19	1	0.04	0.0	0
		4-FullIns_3	6.0	26	1	0.05	0.0	0
		5-FullIns_3	7.0	35	1	0.03	0.0	0
		anna	74.0	70	1	0.01	0.07	0
		david	124.0	62	1	0.0	0.08	0
		DSJC125.9	97470.0	428	142	0.1	0.0	0
		fpsol2.i.1	2640.0	510	3	0.06	0.02	0
	Color02	games120	2.40302e6	9460	5110	0.0	0.01	0
		huck	718.0	62	3	0.0	0.1	0
		jean	5.0	49	1	0.02	0.0	0
		miles1000	445.0	201	8	0.03	0.05	0
		miles1500	277.0	225	1	0.08	0.0	0
		miles250	29464.0	2176	303	0.01	0.01	0
		miles500	856.0	163	12	0.0	0.07	0
		miles750	137.0	156	1	0.01	0.06	0
		mulsol.i.1	368.0	214	6	0.0	0.07	0
		mulsol.i.2	37.0	161	1	0.01	0.0	0
		mulsol.i.3	40.0	175	1	0.01	0.0	0
		mulsol.i.4	24.0	166	1	0.01	0.0	0
Price and Cut		mulsol.i.5	34.0	162	1	0.01	0.0	0
		myciel3	276.0	16	1	0.03	0.06	0
		myciel4	10623.0	732	555	0.04	0.04	0
		queen5_5	8.0	16	1	0.06	0.0	0
		queen6_6	1386.0	142	11	0.02	0.07	0

		queen7_7	86.0	139	1	0.04	0.0	0
		queen8_12	156027.0	4902	371	0.07	0.0	0
		r125.1	481600.0	8961	873	0.01	0.03	0
		r125.1c	11.0	51	1	0.07	0.0	0
		fpsol2.i.2	4.60412e6	45060	277	0.01	0.01	0
		fpsol2.i.3	6.62844e6	43984	423	0.01	0.01	0
		r125.5	3794.0	554	11	0.07	0.01	0
		r250.1c	426.0	117	1	0.09	0.0	0
		r250.5	160429.0	2290	112	0.07	0.0	0
		zeroin.i.1	86.0	164	1	0.07	0.0	0
		zeroin.i.2	19.0	134	1	0.03	0.0	0
		zeroin.i.3	27.0	134	1	0.02	0.0	0
	DIMACS	adjnoun.graph	412.0	132	1	0.09	0.0	0
		celegans_metabolic.graph	763.0	261	1	0.1	0.0	0
		chesapeake.graph	10.0	39	1	0.03	0.0	0
		dolphins.graph	8.0	40	1	0.05	0.0	0
		football.graph	408.0	135	1	0.09	0.0	0
		jazz.graph	59398.0	451	10	0.1	0.0	0
		karate.graph	4.0	22	1	0.05	0.0	0
		lesmis.graph	3.0	33	1	0.03	0.0	0
		polbooks.graph	16.0	61	1	0.07	0.0	0
	comp-DIMACS	adjnoun.graph_comp	63740.0	4481	26	0.07	0.01	0
		chesapeake.graph_comp	7.0	20	1	0.04	0.0	0
		dolphins.graph_comp	2554.0	200	36	0.0	0.09	0
		jazz.graph_comp	385297.0	13034	67	0.06	0.0	0
		karate.graph_comp	4.0	14	1	0.02	0.0	0
		lesmis.graph_comp	194.0	91	3	0.01	0.06	0

		polbooks.graph_comp	16339.0	2444	35	0.06	0.01	0
		1-FullIns_3	403.0	14	1	0.0	0.1	0
		2-FullIns_3	668.0	17	1	0.0	0.1	0
		3-FullIns_3	411.0	19	1	0.0	0.1	0
		4-FullIns_3	1169.0	26	1	0.0	0.1	0
		5-FullIns_3	277.0	35	1	0.0	0.1	0
		anna	515.0	70	1	0.0	0.09	0
		david	485.0	62	1	0.0	0.1	0
		DSJC125.9	105746.0	428	142	0.09	0.01	0
		fpsol2.i.1	16999.0	507	15	0.01	0.08	0
	Color02	games120	1.29935e6	5172	2097	0.0	0.07	0
		huck	289.0	62	1	0.0	0.1	0
		jean	305.0	49	1	0.0	0.1	0
		miles1000	2239.0	201	8	0.0	0.09	0
		miles1500	3717.0	225	1	0.01	0.09	0
		miles250	184295.0	2135	860	0.0	0.06	0
		miles500	2413.0	163	12	0.0	0.09	0
		miles750	1142.0	156	1	0.0	0.1	0
		mulsol.i.1	4165.0	214	6	0.0	0.1	0
		mulsol.i.2	1603.0	162	2	0.0	0.09	0
		mulsol.i.3	1869.0	175	1	0.0	0.1	0
		mulsol.i.4	1454.0	154	1	0.0	0.1	0
k-alternation ($k = 1$)		mulsol.i.5	1491.0	163	1	0.0	0.1	0
		myciel3	205.0	16	1	0.02	0.08	0
		myciel4	74466.0	1030	619	0.01	0.09	0
		queen5_5	776.0	16	1	0.0	0.1	0
		queen6_6	696.0	87	1	0.0	0.09	0
		queen7_7	1308.0	178	1	0.01	0.09	0

		queen8_12	65550.0	1399	34	0.02	0.08	0
		r125.1	1.44807e6	6584	844	0.0	0.08	0
		r125.1c	156.0	51	1	0.0	0.09	0
		fpsol2.i.2	1.0803e7	13829	147	0.0	0.09	0
		fpsol2.i.3	1.08037e7	13446	52	0.0	0.1	0
		r125.5	16743.0	554	11	0.02	0.08	0
		r250.1c	1259.0	117	1	0.03	0.07	0
		r250.5	312251.0	2316	123	0.03	0.05	0
		zeroin.i.1	2721.0	164	1	0.0	0.1	0
		zeroin.i.2	1401.0	134	1	0.0	0.1	0
		zeroin.i.3	1409.0	134	1	0.0	0.1	0
	DIMACS	adjnoun.graph	2808.0	132	1	0.01	0.08	0
		celegans_metabolic.graph	2171.0	261	1	0.04	0.06	0
		chesapeake.graph	292.0	39	1	0.0	0.1	0
		dolphins.graph	91.0	40	1	0.0	0.09	0
		football.graph	3010.0	136	1	0.01	0.09	0
		jazz.graph	70798.0	451	10	0.08	0.02	0
		karate.graph	41.0	22	1	0.0	0.09	0
		lesmis.graph	4.0	33	1	0.08	0.0	0
		polbooks.graph	415.0	61	1	0.0	0.1	0
	comp-DIMACS	adjnoun.graph_comp	665728.0	3012	16	0.0	0.09	0
		chesapeake.graph_comp	268.0	20	1	0.0	0.1	0
		dolphins.graph_comp	2929.0	179	23	0.0	0.09	0
		jazz.graph_comp	1.08169e7	9795	60	0.0	0.1	0
		karate.graph_comp	153.0	14	1	0.0	0.1	0
		lesmis.graph_comp	1231.0	97	1	0.0	0.09	0
		polbooks.graph_comp	1.00043e6	3908	98	0.0	0.1	0
		1-FullIns_3	5.0	12	1	0.0	0.0	0

Color02

		2-FullIns_3	7.0	17	1	0.06	0.0	0
		3-FullIns_3	5.0	19	1	0.06	0.0	0
		4-FullIns_3	6.0	26	1	0.03	0.0	0
		5-FullIns_3	7.0	35	1	0.03	0.0	0
		anna	72.0	70	1	0.01	0.07	0
		david	107.0	62	1	0.0	0.08	0
		DSJC125.9	97831.0	428	142	0.1	0.0	0
		fpsol2.i.1	3598.0	510	3	0.04	0.05	0
		games120	57387.0	1804	421	0.0	0.05	0
		huck	645.0	62	3	0.0	0.1	0
		jean	5.0	49	1	0.02	0.0	0
		miles1000	635.0	201	8	0.02	0.06	0
		miles1500	263.0	225	1	0.08	0.0	0
		miles250	17980.0	849	212	0.0	0.06	0
		miles500	1112.0	163	12	0.0	0.08	0
		miles750	221.0	156	1	0.01	0.08	0
		mulsol.i.1	433.0	214	6	0.0	0.07	0
		mulsol.i.2	24.0	161	1	0.01	0.0	0
		mulsol.i.3	51.0	175	1	0.01	0.0	0
		mulsol.i.4	25.0	166	1	0.01	0.0	0
		mulsol.i.5	24.0	162	1	0.01	0.0	0
		myciel3	246.0	16	1	0.04	0.05	0
		myciel4	37453.0	1109	843	0.01	0.08	0
		queen5_5	8.0	16	1	0.06	0.0	0
		queen6_6	1146.0	104	3	0.01	0.09	0
		queen7_7	491.0	139	1	0.01	0.08	0
		queen8_12	183066.0	4902	371	0.06	0.02	0
		r125.1	12007.0	1447	152	0.01	0.04	0

		r125.1c	11.0	51	1	0.07	0.0	0
		fpsol2.i.2	8.16029e6	40995	436	0.01	0.02	0
		fpsol2.i.3	1.08476e7	38747	145	0.0	0.08	0
		r125.5	4053.0	554	11	0.06	0.02	0
		r250.1c	457.0	117	1	0.08	0.01	0
		r250.5	163498.0	2290	112	0.06	0.0	0
		zeroin.i.1	200.0	164	1	0.03	0.06	0
		zeroin.i.2	110.0	134	1	0.0	0.08	0
		zeroin.i.3	127.0	134	1	0.0	0.09	0
	DIMACS	adjnoun.graph	411.0	132	1	0.09	0.0	0
		celegans_metabolic.graph	896.0	261	1	0.09	0.01	0
		chesapeake.graph	11.0	39	1	0.02	0.0	0
		dolphins.graph	8.0	40	1	0.06	0.0	0
		football.graph	897.0	135	1	0.04	0.05	0
		jazz.graph	59450.0	451	10	0.1	0.0	0
		karate.graph	3.0	22	1	0.03	0.0	0
		lesmis.graph	3.0	33	1	0.03	0.0	0
		polbooks.graph	16.0	61	1	0.08	0.0	0
	comp-DIMACS	adjnoun.graph_comp	78067.0	4607	31	0.06	0.02	0
		chesapeake.graph_comp	7.0	20	1	0.04	0.0	0
		dolphins.graph_comp	2462.0	301	34	0.0	0.08	0
		jazz.graph_comp	1.15483e6	13034	67	0.02	0.07	0
		karate.graph_comp	4.0	14	1	0.02	0.0	0
		lesmis.graph_comp	174.0	91	3	0.01	0.06	0
		polbooks.graph_comp	23680.0	2444	35	0.04	0.04	0
		1-FullIns_3	4.0	12	1	0.0	0.0	0
		2-FullIns_3	6.0	17	1	0.05	0.0	0
		3-FullIns_3	5.0	19	1	0.04	0.0	0

Color02

		4-FullIns_3	6.0	26	1	0.05	0.0	0
		5-FullIns_3	8.0	35	1	0.04	0.0	0
		anna	87.0	70	1	0.01	0.07	0
		david	108.0	62	1	0.0	0.08	0
		DSJC125.9	97629.0	428	142	0.1	0.0	0
		fpsol2.i.1	2618.0	510	3	0.06	0.02	0
		games120	3.54893e6	9460	5110	0.0	0.04	0
		huck	596.0	62	3	0.0	0.1	0
		jean	6.0	49	1	0.02	0.0	0
		miles1000	425.0	201	8	0.02	0.05	0
		miles1500	262.0	225	1	0.08	0.0	0
		miles250	3677.0	223	75	0.0	0.06	0
		miles500	876.0	163	12	0.0	0.07	0
		miles750	136.0	156	1	0.01	0.07	0
		mulsol.i.1	373.0	214	6	0.0	0.07	0
		mulsol.i.2	37.0	161	1	0.01	0.0	0
		mulsol.i.3	39.0	175	1	0.01	0.0	0
		mulsol.i.4	24.0	166	1	0.01	0.0	0
		mulsol.i.5	23.0	162	1	0.01	0.0	0
		myciel3	221.0	16	1	0.03	0.06	0
		myciel4	19505.0	948	683	0.02	0.06	0
		queen5_5	6.0	16	1	0.03	0.0	0
		queen6_6	1313.0	144	9	0.02	0.07	0
		queen7_7	90.0	139	1	0.05	0.0	0
		queen8_12	181256.0	4902	371	0.06	0.02	0
		r125.1	12974.0	1329	126	0.0	0.05	0
		r125.1c	10.0	51	1	0.07	0.0	0
		fpsol2.i.2	4.73613e6	45060	277	0.01	0.01	0

		fpsol2.i.3	6.93329e6	43984	423	0.01	0.01	0
		r125.5	3750.0	554	11	0.07	0.01	0
		r250.1c	425.0	117	1	0.09	0.0	0
		r250.5	163008.0	2290	112	0.06	0.0	0
		zeroin.i.1	90.0	164	1	0.07	0.0	0
		zeroin.i.2	19.0	134	1	0.03	0.0	0
		zeroin.i.3	18.0	134	1	0.03	0.0	0
	DIMACS	adjnoun.graph	416.0	132	1	0.09	0.0	0
		celegans_metabolic.graph	766.0	261	1	0.1	0.0	0
		chesapeake.graph	10.0	39	1	0.02	0.0	0
		dolphins.graph	9.0	40	1	0.04	0.0	0
		football.graph	388.0	135	1	0.09	0.0	0
		jazz.graph	59258.0	451	10	0.1	0.0	0
		karate.graph	4.0	22	1	0.05	0.0	0
		lesmis.graph	3.0	33	1	0.07	0.0	0
		polbooks.graph	16.0	61	1	0.07	0.0	0
	comp-DIMACS	adjnoun.graph_comp	64273.0	4481	26	0.07	0.01	0
		chesapeake.graph_comp	7.0	20	1	0.04	0.0	0
		dolphins.graph_comp	2876.0	314	47	0.0	0.08	0
		jazz.graph_comp	407677.0	13034	67	0.06	0.01	0
		karate.graph_comp	4.0	14	1	0.05	0.0	0
		lesmis.graph_comp	179.0	91	3	0.01	0.06	0
		polbooks.graph_comp	18656.0	2444	35	0.05	0.02	0
		1-FullIns_3	1035.0	26	1	0.0	0.1	0
		2-FullIns_3	816.0	17	1	0.0	0.1	0
		3-FullIns_3	554.0	19	1	0.0	0.1	0
		4-FullIns_3	1282.0	26	1	0.0	0.1	0
		5-FullIns_3	461.0	35	1	0.0	0.1	0

Color02

		anna	247.0	71	1	0.0	0.09	0
		david	157.0	62	1	0.0	0.09	0
		DSJC125.9	123640.0	428	142	0.08	0.02	0
		fpsol2.i.1	3745.0	510	3	0.04	0.05	0
		games120	7.35294e6	4704	1519	0.0	0.1	0
		huck	530.0	60	1	0.0	0.1	0
		jean	243.0	49	1	0.0	0.1	0
		miles1000	745.0	215	1	0.01	0.08	0
		miles1500	633.0	225	1	0.03	0.06	0
		miles250	2.76814e6	2971	1026	0.0	0.09	0
		miles500	2122.0	163	12	0.0	0.09	0
		miles750	343.0	156	1	0.0	0.08	0
		mulsol.i.1	888.0	214	6	0.0	0.09	0
		mulsol.i.2	314.0	160	1	0.0	0.09	0
		mulsol.i.3	382.0	175	1	0.0	0.09	0
		mulsol.i.4	287.0	166	1	0.0	0.09	0
		mulsol.i.5	296.0	162	1	0.0	0.09	0
		myciel3	253.0	18	1	0.01	0.08	0
		myciel4	294655.0	1063	565	0.0	0.1	0
		queen5_5	610.0	16	1	0.0	0.1	0
		queen6_6	2517.0	76	1	0.0	0.1	0
		queen7_7	4107.0	109	1	0.0	0.1	0
		queen8_12	56123.0	1589	60	0.04	0.06	0
		r125.1	355142.0	1115	102	0.0	0.1	0
		r125.1c	10.0	51	1	0.07	0.0	0
		fpsol2.i.2	1.08029e7	10357	41	0.0	0.1	0
		fpsol2.i.3	1.08005e7	11105	49	0.0	0.1	0
		r125.5	6448.0	554	11	0.0	0.05	0

baseline+ DOI		r250.1c	617.0	117	1	0.06	0.03	0
		r250.5	55654.0	1465	13	0.08	0.01	0
		zeroin.i.1	724.0	164	1	0.01	0.09	0
		zeroin.i.2	249.0	134	1	0.0	0.09	0
		zeroin.i.3	295.0	134	1	0.0	0.09	0
	DIMACS	adjnoun.graph	986.0	132	1	0.04	0.06	0
		celegans_metabolic.graph	758.0	261	1	0.1	0.0	0
		chesapeake.graph	220.0	39	1	0.0	0.1	0
		dolphins.graph	109.0	40	1	0.0	0.09	0
		football.graph	925.0	135	1	0.04	0.06	0
		jazz.graph	60471.0	451	10	0.1	0.0	0
		karate.graph	4.0	22	1	0.05	0.0	0
		lesmis.graph	4.0	33	1	0.05	0.0	0
		polbooks.graph	86.0	61	1	0.01	0.08	0
	comp-DIMACS	adjnoun.graph_comp	5.8161e6	4212	34	0.0	0.1	0
		chesapeake.graph_comp	221.0	19	1	0.0	0.1	0
		dolphins.graph_comp	18597.0	256	26	0.0	0.1	0
		jazz.graph_comp	1.08024e7	7921	33	0.0	0.1	0
		karate.graph_comp	153.0	14	1	0.0	0.1	0
		lesmis.graph_comp	730.0	109	3	0.01	0.08	0
		polbooks.graph_comp	14951.0	999	10	0.01	0.08	0
	Color02	1-FullIns_3	9.0	16	1	0.02	0.0	4
		2-FullIns_3	8.0	14	1	0.06	0.0	9
		3-FullIns_3	7.0	17	1	0.04	0.0	16
		4-FullIns_3	8.0	20	1	0.06	0.0	25
		5-FullIns_3	6.0	24	1	0.05	0.0	36
		anna	13.0	55	1	0.04	0.0	104
		david	8.0	49	1	0.05	0.0	119

	fpsol2.i.1	22.0	169	1	0.03	0.0	2191
	games120	10910.0	1715	176	0.03	0.0	4
	huck	5.0	46	1	0.02	0.0	234
	jean	30.0	41	1	0.05	0.0	143
	miles1000	228.0	176	1	0.08	0.0	668
	miles1500	33.0	124	1	0.06	0.0	2170
	miles250	792.0	182	52	0.0	0.0	119
	miles500	21.0	127	1	0.02	0.0	268
	miles750	16.0	118	1	0.02	0.0	397
	mulsol.i.1	11.0	94	1	0.04	0.0	1194
	mulsol.i.2	18.0	95	1	0.03	0.0	587
	mulsol.i.3	8.0	94	1	0.04	0.0	583
	mulsol.i.4	13.0	98	1	0.05	0.0	586
	mulsol.i.5	8.0	94	1	0.04	0.0	570
	r125.1c	12.0	50	1	0.04	0.0	1004
	fpsol2.i.2	45.0	176	1	0.05	0.0	839
	fpsol2.i.3	44.0	176	1	0.05	0.0	839
	r125.5	1865.0	397	2	0.08	0.0	554
	r250.1c	156.0	95	1	0.09	0.0	623
	r250.5	140333.0	1861	144	0.06	0.0	1793
	zeroin.i.1	11.0	86	1	0.04	0.0	975
	zeroin.i.2	13.0	87	1	0.05	0.0	394
	zeroin.i.3	13.0	87	1	0.05	0.0	394
DIMACS	adjnoun.graph	414.0	108	1	0.09	0.0	1101
	celegans_metabolic.graph	292.0	241	1	0.07	0.0	12932
	chesapeake.graph	14.0	36	1	0.04	0.0	26
	dolphins.graph	12.0	40	1	0.06	0.0	194
	karate.graph	3.0	21	1	0.03	0.0	210

		lesmis.graph	7.0	33	1	0.04	0.0	966
		polbooks.graph	10.0	55	1	0.05	0.0	650
	comp-DIMACS	adjnoun.graph_comp	7249.0	1096	1	0.08	0.0	4
		chesapeake.graph_comp	9.0	20	1	0.06	0.0	3
		dolphins.graph_comp	231.0	218	27	0.01	0.0	7
		karate.graph_comp	3.0	13	1	0.03	0.0	10
		lesmis.graph_comp	41.0	58	1	0.05	0.0	101
		polbooks.graph_comp	77194.0	5889	172	0.05	0.0	31
Price and Cut + DOICHG1	Color02	1-FullIns_3	9.0	16	1	0.03	0.0	4
		2-FullIns_3	7.0	14	1	0.04	0.0	9
		3-FullIns_3	6.0	17	1	0.03	0.0	16
		4-FullIns_3	7.0	20	1	0.04	0.0	25
		5-FullIns_3	6.0	24	1	0.05	0.0	36
		anna	13.0	55	1	0.04	0.0	104
		david	8.0	49	1	0.02	0.0	119
		fpsol2.i.1	33.0	169	1	0.02	0.0	2191
		games120	1.08012e7	13998	11150	0.0	0.02	4
		huck	6.0	46	1	0.02	0.0	234
		jean	26.0	41	1	0.02	0.0	143
		miles1000	233.0	176	1	0.07	0.0	668
		miles1500	43.0	124	1	0.04	0.0	2170
		miles250	4656.0	158	35	0.0	0.09	119
		miles500	22.0	127	1	0.02	0.0	268
		miles750	18.0	118	1	0.02	0.0	397
		mulsol.i.1	16.0	94	1	0.02	0.0	1194
		mulsol.i.2	12.0	95	1	0.02	0.0	587
		mulsol.i.3	12.0	94	1	0.02	0.0	583
		mulsol.i.4	16.0	98	1	0.04	0.0	586

	mulsol.i.5	11.0	94	1	0.02	0.0	570
	r125.1c	18.0	50	1	0.03	0.0	1004
	fpsol2.i.2	49.0	176	1	0.05	0.0	839
	fpsol2.i.3	53.0	176	1	0.05	0.0	839
	r125.5	2147.0	397	2	0.07	0.01	554
	r250.1c	154.0	95	1	0.08	0.0	623
	r250.5	102452.0	1309	30	0.05	0.04	1793
	zeroin.i.1	15.0	86	1	0.02	0.0	975
	zeroin.i.2	13.0	87	1	0.03	0.0	394
	zeroin.i.3	15.0	87	1	0.04	0.0	394
DIMACS	adjnoun.graph	422.0	108	1	0.09	0.0	1101
	celegans_metabolic.graph	365.0	241	1	0.06	0.0	12932
	chesapeake.graph	12.0	36	1	0.02	0.0	26
	dolphins.graph	14.0	40	1	0.06	0.0	194
	karate.graph	5.0	21	1	0.02	0.0	210
	lesmis.graph	11.0	33	1	0.01	0.0	966
	polbooks.graph	14.0	55	1	0.04	0.0	650
comp-DIMACS	adjnoun.graph_comp	7766.0	1096	1	0.07	0.01	4
	chesapeake.graph_comp	8.0	20	1	0.05	0.0	3
	dolphins.graph_comp	2512.0	246	32	0.0	0.09	7
	karate.graph_comp	4.0	13	1	0.02	0.0	10
	lesmis.graph_comp	42.0	58	1	0.05	0.0	101
	polbooks.graph_comp	186264.0	7393	217	0.03	0.03	31
	1-FullIns_3	221.0	22	1	0.0	0.09	4
	2-FullIns_3	301.0	20	1	0.0	0.1	9
	3-FullIns_3	543.0	17	1	0.0	0.1	16
	4-FullIns_3	546.0	20	1	0.0	0.1	25
	5-FullIns_3	219.0	24	1	0.0	0.1	36

Color02

k-alternation ($k = 1$) + DOICHG1

		anna	462.0	56	1	0.0	0.09	104
		david	1775.0	67	1	0.0	0.1	119
		fpsol2.i.1	32.0	169	1	0.02	0.0	2191
		games120	471243.0	3293	1271	0.0	0.06	4
		huck	67.0	46	1	0.0	0.09	234
		jean	191.0	41	1	0.0	0.09	143
		miles1000	9462.0	158	1	0.0	0.1	668
		miles1500	2729.0	125	1	0.0	0.1	2170
		miles250	5337.0	114	10	0.0	0.1	119
		miles500	14089.0	131	1	0.0	0.1	268
		miles750	2355.0	125	1	0.0	0.1	397
		mulsol.i.1	17.0	94	1	0.02	0.0	1194
		mulsol.i.2	1336.0	100	1	0.0	0.1	587
		mulsol.i.3	417.0	94	1	0.0	0.1	583
		mulsol.i.4	897.0	98	1	0.0	0.1	586
		mulsol.i.5	300.0	97	1	0.0	0.09	570
		r125.1c	516.0	50	1	0.0	0.1	1004
		fpsol2.i.2	3908.0	176	1	0.0	0.1	839
		fpsol2.i.3	2639.0	172	1	0.0	0.1	839
		r125.5	34896.0	377	1	0.0	0.1	554
		r250.1c	5763.0	109	1	0.01	0.09	623
		r250.5	520395.0	1525	26	0.01	0.09	1793
		zeroin.i.1	16.0	86	1	0.02	0.0	975
		zeroin.i.2	448.0	87	1	0.0	0.1	394
		zeroin.i.3	437.0	87	1	0.0	0.1	394
		adjnoun.graph	8303.0	128	1	0.01	0.09	1101
		celegans_metabolic.graph	136341.0	243	1	0.0	0.1	12932
	DIMACS	chesapeake.graph	524.0	39	1	0.0	0.1	26

		dolphins.graph	190.0	39	1	0.0	0.09	194
		karate.graph	6.0	21	1	0.03	0.0	210
		lesmis.graph	12.0	33	1	0.02	0.0	966
		polbooks.graph	704.0	55	1	0.0	0.1	650
	comp-DIMACS	adjnoun.graph_comp	4.29493e6	6095	35	0.0	0.1	4
		chesapeake.graph_comp	494.0	20	1	0.0	0.1	3
		dolphins.graph_comp	5992.0	174	25	0.0	0.1	7
		karate.graph_comp	104.0	14	1	0.0	0.1	10
		lesmis.graph_comp	1810.0	80	1	0.0	0.09	101
		polbooks.graph_comp	424218.0	2126	24	0.0	0.1	31
k-alternation ($k = 10$) + DOICHG1	Color02	1-FullIns_3	9.0	16	1	0.03	0.0	4
		2-FullIns_3	7.0	14	1	0.06	0.0	9
		3-FullIns_3	7.0	17	1	0.06	0.0	16
		4-FullIns_3	8.0	20	1	0.05	0.0	25
		5-FullIns_3	6.0	24	1	0.03	0.0	36
		anna	13.0	55	1	0.03	0.0	104
		david	7.0	49	1	0.03	0.0	119
		fpsol2.i.1	32.0	169	1	0.02	0.0	2191
		games120	1.0807e7	10602	6942	0.0	0.06	4
		huck	6.0	46	1	0.03	0.0	234
		jean	8.0	41	1	0.04	0.0	143
		miles1000	228.0	176	1	0.07	0.0	668
		miles1500	43.0	124	1	0.04	0.0	2170
		miles250	4034.0	170	24	0.0	0.09	119
		miles500	23.0	127	1	0.02	0.0	268
		miles750	19.0	118	1	0.03	0.0	397
		mulsol.i.1	16.0	94	1	0.02	0.0	1194
		mulsol.i.2	11.0	95	1	0.03	0.0	587

	mulsol.i.3	12.0	94	1	0.02	0.0	583
	mulsol.i.4	16.0	98	1	0.04	0.0	586
	mulsol.i.5	11.0	94	1	0.03	0.0	570
	r125.1c	17.0	50	1	0.03	0.0	1004
	fpsol2.i.2	49.0	176	1	0.05	0.0	839
	fpsol2.i.3	51.0	176	1	0.05	0.0	839
	r125.5	2120.0	397	2	0.07	0.01	554
	r250.1c	157.0	95	1	0.09	0.0	623
	r250.5	127918.0	1473	33	0.05	0.04	1793
	zeroin.i.1	16.0	86	1	0.02	0.0	975
	zeroin.i.2	13.0	87	1	0.03	0.0	394
	zeroin.i.3	14.0	87	1	0.04	0.0	394
DIMACS	adjnoun.graph	1380.0	120	1	0.06	0.03	1101
	celegans_metabolic.graph	359.0	241	1	0.06	0.0	12932
	chesapeake.graph	13.0	36	1	0.02	0.0	26
	dolphins.graph	13.0	40	1	0.05	0.0	194
	karate.graph	5.0	21	1	0.04	0.0	210
	lesmis.graph	11.0	33	1	0.02	0.0	966
	polbooks.graph	14.0	55	1	0.04	0.0	650
comp-DIMACS	adjnoun.graph_comp	7766.0	1096	1	0.07	0.01	4
	chesapeake.graph_comp	8.0	20	1	0.05	0.0	3
	dolphins.graph_comp	3944.0	202	45	0.0	0.09	7
	karate.graph_comp	4.0	13	1	0.02	0.0	10
	lesmis.graph_comp	40.0	58	1	0.06	0.0	101
	polbooks.graph_comp	11063.0	1455	22	0.03	0.04	31
	1-FullIns_3	389.0	24	1	0.0	0.09	4
	2-FullIns_3	724.0	16	1	0.0	0.1	9
	3-FullIns_3	327.0	17	1	0.0	0.1	16

Color02

		4-FullIns_3	325.0	21	1	0.0	0.1	25
		5-FullIns_3	523.0	24	1	0.0	0.1	36
		anna	383.0	57	1	0.0	0.09	104
		david	165.0	49	1	0.0	0.1	119
		fpsol2.i.1	32.0	169	1	0.02	0.0	2191
		games120	1.16821e6	1925	513	0.0	0.1	4
		huck	5.0	46	1	0.0	0.0	234
		jean	174.0	41	1	0.0	0.1	143
		miles1000	4253.0	162	1	0.0	0.1	668
		miles1500	44.0	124	1	0.04	0.0	2170
		miles250	104447.0	502	99	0.0	0.1	119
		miles500	1722.0	126	1	0.0	0.1	268
		miles750	417.0	121	1	0.0	0.09	397
		mulsol.i.1	16.0	94	1	0.02	0.0	1194
		mulsol.i.2	12.0	95	1	0.02	0.0	587
		mulsol.i.3	11.0	94	1	0.03	0.0	583
		mulsol.i.4	15.0	98	1	0.04	0.0	586
		mulsol.i.5	11.0	94	1	0.03	0.0	570
		r125.1c	16.0	50	1	0.03	0.0	1004
		fpsol2.i.2	49.0	176	1	0.05	0.0	839
		fpsol2.i.3	49.0	176	1	0.05	0.0	839
		r125.5	47880.0	380	1	0.0	0.1	554
		r250.1c	511.0	100	1	0.04	0.05	623
		r250.5	3.92772e6	1955	108	0.0	0.1	1793
		zeroin.i.1	15.0	86	1	0.02	0.0	975
		zeroin.i.2	14.0	87	1	0.04	0.0	394
		zeroin.i.3	15.0	87	1	0.05	0.0	394
		adjnoun.graph	6174.0	135	1	0.02	0.08	1101
	DIMACS							

		celegans_metabolic.graph	354.0	241	1	0.06	0.0	12932
		chesapeake.graph	363.0	39	1	0.0	0.09	26
		dolphins.graph	153.0	40	1	0.01	0.09	194
		jazz.graph	39660.0	370	2	0.09	0.01	58
		karate.graph	5.0	21	1	0.02	0.0	210
		lesmis.graph	12.0	33	1	0.02	0.0	966
		polbooks.graph	13.0	55	1	0.04	0.0	650
	comp-DIMACS	adjnoun.graph_comp	7.93188e6	4575	30	0.0	0.1	4
		chesapeake.graph_comp	1110.0	27	1	0.0	0.1	3
		dolphins.graph_comp	12320.0	183	15	0.0	0.1	7
		karate.graph_comp	212.0	14	1	0.0	0.1	10
		lesmis.graph_comp	1251.0	84	1	0.0	0.09	101
		polbooks.graph_comp	3.15398e6	2811	38	0.0	0.1	31
Price and Cut + DOICHG2	Color02	1-FullIns_3	9.0	16	1	0.03	0.0	4
		2-FullIns_3	8.0	14	1	0.05	0.0	9
		3-FullIns_3	9.0	17	1	0.06	0.0	16
		4-FullIns_3	7.0	20	1	0.04	0.0	25
		5-FullIns_3	5.0	24	1	0.04	0.0	36
		anna	14.0	55	1	0.04	0.0	104
		david	34.0	49	1	0.04	0.0	119
		fpsol2.i.1	23.0	169	1	0.03	0.0	2191
		games120	5242.0	780	106	0.01	0.03	4
		huck	6.0	46	1	0.02	0.0	234
		jean	8.0	41	1	0.05	0.0	143
		miles1000	247.0	176	1	0.07	0.0	668
		miles1500	34.0	124	1	0.06	0.0	2170
		miles250	1228.0	182	52	0.0	0.05	119
		miles500	50.0	127	1	0.01	0.0	268

		miles750	17.0	118	1	0.03	0.0	397
		mulsol.i.1	11.0	94	1	0.04	0.0	1194
		mulsol.i.2	10.0	95	1	0.03	0.0	587
		mulsol.i.3	9.0	94	1	0.03	0.0	583
		mulsol.i.4	13.0	98	1	0.05	0.0	586
		mulsol.i.5	9.0	94	1	0.03	0.0	570
		r125.1c	14.0	50	1	0.04	0.0	1004
		fpsol2.i.2	46.0	176	1	0.05	0.0	839
		fpsol2.i.3	50.0	176	1	0.05	0.0	839
		r125.5	1911.0	397	2	0.08	0.0	554
		r250.1c	153.0	95	1	0.09	0.0	623
		r250.5	142177.0	1861	144	0.06	0.0	1793
		zeroin.i.1	18.0	86	1	0.03	0.0	975
		zeroin.i.2	14.0	87	1	0.05	0.0	394
		zeroin.i.3	12.0	87	1	0.04	0.0	394
	DIMACS	adjnoun.graph	426.0	108	1	0.09	0.0	1101
		celegans_metabolic.graph	305.0	241	1	0.07	0.0	12932
		chesapeake.graph	14.0	36	1	0.04	0.0	26
		dolphins.graph	28.0	40	1	0.05	0.0	194
		karate.graph	4.0	21	1	0.02	0.0	210
		lesmis.graph	27.0	33	1	0.02	0.0	966
		polbooks.graph	22.0	55	1	0.04	0.0	650
	comp-DIMACS	adjnoun.graph_comp	90732.0	5930	28	0.07	0.01	4
		chesapeake.graph_comp	9.0	20	1	0.06	0.0	3
		dolphins.graph_comp	1827.0	287	21	0.01	0.08	7
		karate.graph_comp	5.0	13	1	0.04	0.0	10
		lesmis.graph_comp	42.0	58	1	0.05	0.0	101

		polbooks.graph_comp	80646.0	5889	172	0.05	0.0	31
k-alternation ($k=1$) + DOICHG2	Color02	1-FullIns_3	192.0	16	1	0.0	0.09	4
		2-FullIns_3	231.0	20	1	0.0	0.09	9
		3-FullIns_3	208.0	21	1	0.0	0.09	16
		4-FullIns_3	238.0	24	1	0.0	0.1	25
		5-FullIns_3	114.0	24	1	0.0	0.09	36
		anna	44.0	55	1	0.01	0.07	104
		david	10.0	49	1	0.0	0.03	119
		fpsol2.i.1	24.0	169	1	0.03	0.0	2191
		games120	25889.0	1016	154	0.0	0.08	4
		huck	6.0	46	1	0.0	0.03	234
		jean	48.0	41	1	0.0	0.09	143
		miles1000	1882.0	176	1	0.01	0.09	668
		miles1500	80.0	124	1	0.02	0.06	2170
		miles250	3506.0	182	54	0.0	0.08	119
		miles500	912.0	127	1	0.0	0.1	268
		miles750	246.0	118	1	0.0	0.09	397
		mulsol.i.1	11.0	94	1	0.04	0.0	1194
		mulsol.i.2	32.0	95	1	0.01	0.07	587
		mulsol.i.3	25.0	94	1	0.01	0.06	583
		mulsol.i.4	28.0	98	1	0.01	0.05	586
		mulsol.i.5	22.0	94	1	0.01	0.06	570
		r125.1c	13.0	50	1	0.04	0.01	1004
		fpsol2.i.2	72.0	176	1	0.03	0.03	839
		fpsol2.i.3	72.0	176	1	0.03	0.03	839
		r125.5	4567.0	397	2	0.03	0.06	554
		r250.1c	169.0	95	1	0.08	0.01	623
		r250.5	193754.0	1861	144	0.05	0.03	1793

		zeroin.i.1	18.0	86	1	0.03	0.0	975
		zeroin.i.2	27.0	87	1	0.03	0.02	394
		zeroin.i.3	19.0	87	1	0.04	0.03	394
	DIMACS	adjnoun.graph	450.0	108	1	0.09	0.0	1101
		celegans_metabolic.graph	331.0	241	1	0.06	0.01	12932
		chesapeake.graph	119.0	36	1	0.01	0.09	26
		dolphins.graph	15.0	40	1	0.04	0.01	194
		jazz.graph	49304.0	393	3	0.08	0.02	58
		karate.graph	4.0	21	1	0.02	0.0	210
		lesmis.graph	7.0	33	1	0.03	0.0	966
		polbooks.graph	12.0	55	1	0.03	0.01	650
	comp-DIMACS	adjnoun.graph_comp	1.0339e6	3750	22	0.0	0.09	4
		chesapeake.graph_comp	89.0	20	1	0.0	0.09	3
		dolphins.graph_comp	6713.0	261	52	0.0	0.09	7
		karate.graph_comp	37.0	13	1	0.0	0.09	10
		lesmis.graph_comp	112.0	58	1	0.02	0.06	101
		polbooks.graph_comp	205901.0	2252	25	0.0	0.09	31
	Color02	1-FullIns_3	9.0	16	1	0.03	0.0	4
		2-FullIns_3	6.0	14	1	0.02	0.0	9
		3-FullIns_3	8.0	17	1	0.05	0.0	16
		4-FullIns_3	7.0	20	1	0.06	0.0	25
		5-FullIns_3	5.0	24	1	0.04	0.0	36
		anna	13.0	55	1	0.05	0.0	104
		david	7.0	49	1	0.03	0.0	119
		fpsol2.i.1	24.0	169	1	0.03	0.0	2191
		games120	165034.0	2577	847	0.0	0.05	4
		huck	6.0	46	1	0.02	0.0	234
		jean	31.0	41	1	0.04	0.0	143

k-alternation ($k = 10$) + DOICHG2

	miles1000	222.0	176	1	0.07	0.0	668
	miles1500	35.0	124	1	0.05	0.0	2170
	miles250	2970.0	257	73	0.0	0.07	119
	miles500	23.0	127	1	0.02	0.0	268
	miles750	16.0	118	1	0.03	0.0	397
	mulsol.i.1	11.0	94	1	0.04	0.0	1194
	mulsol.i.2	10.0	95	1	0.03	0.0	587
	mulsol.i.3	9.0	94	1	0.03	0.0	583
	mulsol.i.4	13.0	98	1	0.05	0.0	586
	mulsol.i.5	14.0	94	1	0.03	0.0	570
	r125.1c	12.0	50	1	0.04	0.0	1004
	fpsol2.i.2	49.0	176	1	0.05	0.0	839
	fpsol2.i.3	48.0	176	1	0.05	0.0	839
	r125.5	1893.0	397	2	0.08	0.0	554
	r250.1c	149.0	95	1	0.09	0.0	623
	r250.5	149168.0	1861	144	0.06	0.0	1793
	zeroin.i.1	12.0	86	1	0.03	0.0	975
	zeroin.i.2	13.0	87	1	0.05	0.0	394
	zeroin.i.3	14.0	87	1	0.04	0.0	394
DIMACS	adjnoun.graph	444.0	108	1	0.09	0.0	1101
	celegans_metabolic.graph	299.0	241	1	0.07	0.0	12932
	chesapeake.graph	12.0	36	1	0.03	0.0	26
	dolphins.graph	49.0	40	1	0.06	0.0	194
	karate.graph	4.0	21	1	0.05	0.0	210
	lesmis.graph	8.0	33	1	0.02	0.0	966
	polbooks.graph	22.0	55	1	0.04	0.0	650
comp-DIMACS	adjnoun.graph_comp	130941.0	5930	28	0.05	0.04	4
	chesapeake.graph_comp	19.0	20	1	0.05	0.0	3

		dolphins.graph_comp	973.0	54	2	0.0	0.09	7
		karate.graph_comp	5.0	13	1	0.04	0.0	10
		lesmis.graph_comp	40.0	58	1	0.06	0.0	101
		polbooks.graph_comp	76158.0	5356	67	0.06	0.02	31
(α, k)-alternation + DOICHG2	Color02	1-FullIns_3	430.0	23	1	0.0	0.09	4
		2-FullIns_3	871.0	17	1	0.0	0.1	9
		3-FullIns_3	255.0	24	1	0.0	0.09	16
		4-FullIns_3	954.0	25	1	0.0	0.1	25
		5-FullIns_3	410.0	24	1	0.0	0.1	36
		anna	193.0	55	1	0.0	0.09	104
		david	9.0	49	1	0.02	0.01	119
		fpsol2.i.1	24.0	169	1	0.03	0.0	2191
		games120	343455.0	1180	205	0.0	0.1	4
		huck	26.0	46	1	0.04	0.0	234
		jean	47.0	41	1	0.0	0.06	143
		miles1000	576.0	176	1	0.03	0.06	668
		miles1500	38.0	124	1	0.04	0.0	2170
		miles250	2443.0	130	24	0.0	0.09	119
		miles500	249.0	127	1	0.0	0.09	268
		miles750	116.0	118	1	0.0	0.09	397
		mulsol.i.1	12.0	94	1	0.03	0.0	1194
		mulsol.i.2	10.0	95	1	0.03	0.0	587
		mulsol.i.3	9.0	94	1	0.03	0.0	583
		mulsol.i.4	14.0	98	1	0.04	0.0	586
		mulsol.i.5	9.0	94	1	0.03	0.0	570
		r125.1c	12.0	50	1	0.04	0.0	1004
		fpsol2.i.2	49.0	176	1	0.05	0.0	839
		fpsol2.i.3	49.0	176	1	0.05	0.0	839

		r125.5	2307.0	397	2	0.07	0.02	554
		r250.1c	152.0	95	1	0.08	0.0	623
		r250.5	149742.0	1861	144	0.06	0.0	1793
		zeroin.i.1	11.0	86	1	0.03	0.0	975
		zeroin.i.2	22.0	87	1	0.05	0.0	394
		zeroin.i.3	21.0	87	1	0.04	0.0	394
	DIMACS	adjnoun.graph	446.0	108	1	0.09	0.0	1101
		celegans_metabolic.graph	295.0	241	1	0.07	0.0	12932
		chesapeake.graph	149.0	36	1	0.0	0.09	26
		dolphins.graph	53.0	40	1	0.06	0.01	194
		karate.graph	5.0	21	1	0.04	0.0	210
		lesmis.graph	7.0	33	1	0.01	0.0	966
		polbooks.graph	11.0	55	1	0.05	0.0	650
	comp-DIMACS	adjnoun.graph_comp	5.72456e6	4243	44	0.0	0.1	4
		chesapeake.graph_comp	704.0	30	1	0.0	0.1	3
		dolphins.graph_comp	11584.0	169	20	0.0	0.1	7
		karate.graph_comp	114.0	13	1	0.0	0.1	10
		lesmis.graph_comp	58.0	58	1	0.04	0.03	101
		polbooks.graph_comp	1.0803e7	6893	310	0.0	0.1	31

Complete Tables for the 2-defective coloring

Algorithm	Instances	Name	CPU	# columns	# nodes	Pricing CPU	Sep CPU	NB ct	NB DI	linear rel
Baseline	E-color	1-Insertions_4	12.04	1330	1	9686.67	-	-	-	2.41
		2-Insertions_3	0.42	292	1	280.61	-	-	-	2.0
		3-Insertions_3	5.39	1110	1	3971.77	-	-	-	2.0
		myciel3	0.0	8	1	1.08	-	-	-	2.0
		myciel4	0.24	74	1	214.01	-	-	-	2.56
		myciel5	3.26	434	1	2992.26	-	-	-	3.16
		queen5_5	0.18	99	4	149.01	-	-	-	5.0
		queen6_6	0.5	184	1	389.86	-	-	-	5.25
		queen7_7	43.38	1159	118	41190.41	-	-	-	5.84
		4-Insertions_3	35.79	3139	1	21823.2	-	-	-	2.0
	ED-color	1-FullIns_3	0.8	122	1	756.21	-	-	-	2.63
		2-FullIns_3	5.36	1618	13	4170.36	-	-	-	3.0
		anna	2.0	551	1	1457.3	-	-	-	5.5
		huck	0.51	274	1	398.25	-	-	-	5.5
		jean	0.28	150	1	107.48	-	-	-	5.0
		miles1500	11.38	729	65	8821.58	-	-	-	36.5
		r125.1c	2.67	502	2	2376.67	-	-	-	35.0
		3-FullIns_3	0.62	226	1	512.01	-	-	-	3.5
		4-FullIns_3	0.08	53	2	57.56	-	-	-	4.0
		5-FullIns_3	0.1	97	1	46.49	-	-	-	4.5
	HD-color	david	1113.7	30953	473	426919.27	-	-	-	5.63
		miles1000	53.63	2654	132	29978.83	-	-	-	21.5
		miles500	225.31	8897	310	55448.24	-	-	-	10.5
		miles750	190.48	5680	428	54885.17	-	-	-	16.5
		mulsol.i.1	79.7	3173	88	26532.17	-	-	-	25.0
		mulsol.i.2	9663.56	40748	837	1.21379314e6	-	-	-	15.77
		r125.1	1330.92	40826	90	352898.74	-	-	-	3.0
		zeroin.i.1	28.79	1770	23	18339.8	-	-	-	24.5
		zeroin.i.3	5789.68	46175	408	724116.45	-	-	-	15.33
		fpsol2.i.1	8901.25	20972	380	683950.84	-	-	-	33.0
	DIMACS	adjnoun.graph	5.83	700	1	5433.32	-	-	-	33.11
		chesapeake.graph	0.21	123	3	173.95	-	-	-	11.0
		dolphins.graph	0.9	296	1	826.24	-	-	-	17.99
		jazz.graph	1954.56	3646	635	1.51045203e6	-	-	-	47.54
		karate.graph	0.42	83	3	394.65	-	-	-	11.75
		lesmis.graph	1.08	291	2	983.66	-	-	-	20.5
		polbooks.graph	3.08	648	1	2773.64	-	-	-	27.08
	comp_DIMACS	chesapeake_comp	1.42	676	1	1132.43	-	-	-	3.73
		dolphins_comp	7.96	1738	10	6204.96	-	-	-	3.0
		karate_comp	0.32	137	2	266.4	-	-	-	3.0
		lesmis_comp	6.8	1261	1	5371.28	-	-	-	6.33
		polbooks_comp	9438.7	90834	577	2.54444934e6	-	-	-	4.22
	other	miles250	-	82777	545	2.28105746e6	-	-	-	4.5
		mulsol.i.3	-	43941	732	1.66574976e6	-	-	-	15.77
		mulsol.i.5	-	52580	593	2.55668674e6	-	-	-	15.77
		zeroin.i.2	-	66715	167	1.89016744e6	-	-	-	15.33

		1-FullIns_4	-	94589	1	74507.4	-	-	-	3.13
	Randoms	50_86_num_0_1	12.84	1583	5	11113.95	0.0	0	0	2.0
		50_86_num_0_2	8.5	1490	2	6914.34	0.0	0	0	2.0
		50_86_num_0_3	4.44	1123	1	3413.98	0.0	0	0	2.0
		50_86_num_0_4	3.2	754	1	2626.9	0.0	0	0	2.0
		50_86_num_10_1	2.4	706	1	1877.72	0.0	0	0	2.05
		50_86_num_10_2	6.33	1112	1	5319.62	0.0	0	0	2.0
		50_86_num_10_3	3.07	720	1	2532.32	0.0	0	0	2.04
		50_86_num_10_4	2.95	787	1	2329.77	0.0	0	0	2.0
		50_86_num_11_1	9.94	1459	1	8391.21	0.0	0	0	2.0
		50_86_num_11_2	9.52	1373	2	7956.32	0.0	0	0	2.0
		50_86_num_11_3	10.84	1565	6	9275.06	0.0	0	0	2.0
		50_86_num_11_4	3.32	820	1	2668.04	0.0	0	0	2.08
		50_86_num_1_1	5.08	1257	1	3926.94	0.0	0	0	2.1
		50_86_num_12_1	3.06	693	1	2571.3	0.0	0	0	2.1
		50_86_num_12_2	4.02	928	1	3204.68	0.0	0	0	2.0
		50_86_num_12_3	3.01	811	1	2347.41	0.0	0	0	2.07
		50_86_num_12_4	5.49	1212	2	4318.71	0.0	0	0	2.0
		50_86_num_1_2	2.61	753	1	2017.65	0.0	0	0	2.0
		50_86_num_13_1	5.45	961	1	4588.84	0.0	0	0	2.0
		50_86_num_13_2	4.85	900	1	4073.5	0.0	0	0	2.09
		50_86_num_13_3	2.64	718	1	2134.99	0.0	0	0	2.0
		50_86_num_13_4	4.58	938	1	3801.38	0.0	0	0	2.0
		50_86_num_1_3	3.03	915	1	2291.72	0.0	0	0	2.0
		50_86_num_14_1	8.04	1153	1	6952.31	0.0	0	0	2.0
		50_86_num_14_2	1.78	674	1	1360.38	0.0	0	0	2.0
		50_86_num_14_3	1.0	468	1	704.96	0.0	0	0	2.14
		50_86_num_14_4	3.83	735	1	3255.24	0.0	0	0	2.05
		50_86_num_1_4	1.72	556	1	1370.33	0.0	0	0	2.12
		50_86_num_15_1	6.3	1180	4	5138.67	0.0	0	0	2.0
		50_86_num_15_2	8.85	1226	2	7508.92	0.0	0	0	2.0
		50_86_num_15_3	1.5	553	1	1167.01	0.0	0	0	2.0
		50_86_num_15_4	8.69	1368	1	7241.62	0.0	0	0	2.0
		50_86_num_16_1	6.2	1104	2	5162.19	0.0	0	0	2.0
		50_86_num_16_2	4.26	881	1	3537.68	0.0	0	0	2.09
		50_86_num_16_3	6.64	1037	1	5646.31	0.0	0	0	2.05
		50_86_num_16_4	4.22	1000	1	3375.97	0.0	0	0	2.15
		50_86_num_17_1	0.95	363	1	733.87	0.0	0	0	2.11
		50_86_num_17_2	4.99	885	1	4230.02	0.0	0	0	2.0
		50_86_num_17_3	9.55	1439	1	7987.32	0.0	0	0	2.0
		50_86_num_17_4	1.96	612	1	1564.38	0.0	0	0	2.0
		50_86_num_18_1	3.14	814	1	2515.49	0.0	0	0	2.06
		50_86_num_18_2	3.88	989	1	2921.47	0.0	0	0	2.05
		50_86_num_18_3	2.35	727	1	1827.9	0.0	0	0	2.08
		50_86_num_18_4	1.92	684	1	1435.28	0.0	0	0	2.14
		50_86_num_19_1	4.25	828	1	3569.23	0.0	0	0	2.0
		50_86_num_19_2	0.52	335	1	359.49	0.0	0	0	2.25
		50_86_num_19_3	0.32	217	1	209.79	0.0	0	0	2.33
		50_86_num_19_4	3.15	743	1	2579.34	0.0	0	0	2.06
		50_86_num_20_1	1.04	455	1	776.62	0.0	0	0	2.0
		50_86_num_20_2	5.85	1125	1	4760.73	0.0	0	0	2.08
		50_86_num_20_3	3.1	811	1	2450.43	0.0	0	0	2.0
		50_86_num_20_4	4.66	868	1	3916.34	0.0	0	0	2.06
		50_86_num_21_1	2.5	918	1	1765.0	0.0	0	0	2.11
		50_86_num_21_2	1.9	705	1	1368.45	0.0	0	0	2.2
		50_86_num_21_3	5.55	924	1	4761.37	0.0	0	0	2.05

		50_86_num_21_4	2.84	732	1	2263.49	0.0	0	0	2.07
		50_86_num_2_1	9.25	1471	1	7739.6	0.0	0	0	2.0
		50_86_num_22_1	1.55	617	1	1139.02	0.0	0	0	2.0
		50_86_num_22_2	1.45	524	1	1113.74	0.0	0	0	2.0
		50_86_num_22_3	2.75	716	1	2195.9	0.0	0	0	2.07
		50_86_num_22_4	2.13	686	1	1649.34	0.0	0	0	2.09
		50_86_num_2_2	4.57	773	1	3938.43	0.0	0	0	2.04
		50_86_num_23_1	16.84	2126	7	14365.82	0.0	0	0	2.0
		50_86_num_23_2	9.43	1407	3	7869.23	0.0	0	0	2.0
		50_86_num_23_3	3.76	800	1	3126.23	0.0	0	0	2.04
		50_86_num_23_4	3.78	740	1	3236.04	0.0	0	0	2.03
		50_86_num_2_3	2.86	761	1	2306.22	0.0	0	0	2.13
		50_86_num_24_1	3.44	805	1	2799.22	0.0	0	0	2.03
		50_86_num_24_2	3.07	728	1	2495.78	0.0	0	0	2.08
		50_86_num_24_3	5.16	1065	2	4175.72	0.0	0	0	2.0
		50_86_num_24_4	5.03	1036	1	4094.09	0.0	0	0	2.07
		50_86_num_2_4	4.18	753	1	3598.95	0.0	0	0	2.06
		50_86_num_3_1	3.62	1107	1	2638.33	0.0	0	0	2.06
		50_86_num_3_2	2.94	780	1	2339.7	0.0	0	0	2.16
		50_86_num_3_3	6.12	1024	1	5241.4	0.0	0	0	2.0
		50_86_num_3_4	3.52	839	1	2868.14	0.0	0	0	2.1
		50_86_num_4_1	3.09	676	1	2598.94	0.0	0	0	2.0
		50_86_num_4_2	7.83	1291	1	6495.44	0.0	0	0	2.0
		50_86_num_4_3	7.14	1179	1	5942.36	0.0	0	0	2.04
		50_86_num_4_4	3.22	791	1	2574.22	0.0	0	0	2.11
		50_86_num_5_1	3.26	891	1	2532.3	0.0	0	0	2.0
		50_86_num_5_2	3.74	948	1	2910.13	0.0	0	0	2.06
		50_86_num_5_3	3.43	862	1	2686.79	0.0	0	0	2.1
		50_86_num_5_4	4.56	1057	1	3572.44	0.0	0	0	2.04
		50_86_num_6_1	1.35	590	1	996.74	0.0	0	0	2.0
		50_86_num_6_2	15.72	2392	7	12677.45	0.0	0	0	2.0
		50_86_num_6_3	2.13	631	1	1740.01	0.0	0	0	2.13
		50_86_num_6_4	5.07	975	1	4246.1	0.0	0	0	2.0
		50_86_num_7_1	10.23	1495	1	8482.68	0.0	0	0	2.0
		50_86_num_7_2	6.13	966	1	5255.41	0.0	0	0	2.0
		50_86_num_7_3	1.14	609	1	719.99	0.0	0	0	2.25
		50_86_num_7_4	4.44	918	1	3650.27	0.0	0	0	2.03
		50_86_num_8_1	6.52	1106	1	5473.89	0.0	0	0	2.0
		50_86_num_8_2	14.42	2232	5	12015.43	0.0	0	0	2.0
		50_86_num_8_3	1.39	460	1	1091.74	0.0	0	0	2.17
		50_86_num_8_4	3.88	790	1	3250.38	0.0	0	0	2.06
		50_86_num_9_1	9.93	1333	2	8478.24	0.0	0	0	2.0
		50_86_num_9_2	5.59	1236	4	4422.04	0.0	0	0	2.0
		50_86_num_9_3	7.63	1271	2	6452.46	0.0	0	0	2.0
		50_86_num_9_4	1.86	730	1	1353.37	0.0	0	0	2.08
	ED-color	1-FullIns_3.col	0.5	106	1	354.02	0.0	0	2	2.63
		2-FullIns_3.col	8.36	1854	17	6814.64	0.0	0	2	3.0
		anna.col	5559.86	73064	201	811689.66	0.0	0	304	5.5
		huck.col	0.32	61	1	294.43	0.0	0	293	5.5
		jean.col	35.68	4779	71	19576.24	0.0	0	184	5.0
		miles1500.col	0.53	182	1	454.68	0.0	0	1967	36.5
		r125.1c.col	3.46	434	1	3313.2	0.0	0	515	35.0
		3-FullIns_3.col	0.04	69	1	17.6	0.0	0	2	3.5
		4-FullIns_3.col	0.08	48	1	60.99	0.0	0	2	4.0
		5-FullIns_3.col	862.65	32861	71	290714.28	0.0	0	2	4.5

HD-color	david.col	1268.25	28790	711	407474.09	0.0	0	243	5.63
	miles1000.col	9.86	716	46	5961.75	0.0	0	669	21.5
	miles500.col	205.17	8209	321	57891.91	0.0	0	262	10.5
	miles750.col	45.62	2085	201	19241.87	0.0	0	448	16.5
	mulsol.i.1.col	0.33	172	1	248.81	0.0	0	2073	25.0
	mulsol.i.2.col	793.22	15957	201	258756.15	0.0	0	1199	15.77
	r125.1.col	2.04	702	7	1020.42	0.0	0	123	3.0
	zeroin.i.1.col	19.62	1433	6	13135.81	0.0	0	2181	24.5
	zeroin.i.3.col	10816.4	43480	869	1.18132872e6	0.0	0	959	15.33
	fpsol2.i.1.col	10815.0	9158	776	303263.26	0.0	0	6995	33.0
DIMACS	adjnoun.graph.txt	7.4	640	1	6994.92	0.0	0	226	33.11
	chesapeake.graph.txt	0.18	92	2	141.69	0.0	0	26	11.0
	dolphins.graph.txt	1.42	269	1	1322.8	0.0	0	78	17.99
	jazz.graph.txt	1616.61	3513	497	1.28781597e6	0.0	0	75	47.54
	karate.graph.txt	0.34	77	5	309.85	0.0	0	89	11.75
	lesmis.graph.txt	0.64	211	1	580.3	0.0	0	254	20.5
	polbooks.graph.txt	2.8	598	1	2514.24	0.0	0	20	27.08
comp-DIMACS	chesapeake.graph_comp.txt	1.24	579	1	999.29	0.0	0	3	3.73
	dolphins.graph_comp.txt	26.78	3945	35	20270.83	0.0	0	12	3.0
	karate.graph_comp.txt	0.83	392	9	688.6	0.0	0	29	3.0
	lesmis.graph_comp.txt	5.57	1078	1	4469.19	0.0	0	184	6.33
	polbooks.graph_comp.txt	10800.3	94834	632	3.05449847e6	0.0	0	45	4.22
others	miles250.col	401.53	16450	221	104899.96	0.0	0	130	4.5
	1-FullIns_4.col	22.28	2251	1	15898.18	0.0	0	2	3.13
	zeroin.i.2.col	14.75	1502	1	10881.44	0.0	0	959	15.33
	mulsol.i.3.col	10820.9	41004	926	1.59162482e6	0.0	0	1208	15.77
	mulsol.i.5.col	7.5	899	1	5619.9	0.0	0	1229	15.77
	50_86_num_0_1.txt	19.08	2302	7	16326.04	0.0	0	1	2.0
	50_86_num_0_2.txt	9.64	1639	5	7915.14	0.0	0	2	2.0
	50_86_num_0_3.txt	11.49	1702	1	9417.07	0.0	0	3	2.0
	50_86_num_0_4.txt	5.2	1252	3	4116.52	0.0	0	4	2.0
	50_86_num_10_1.txt	2.44	686	1	1921.64	0.0	0	1	2.05
	50_86_num_10_2.txt	2.76	766	1	2155.43	0.0	0	2	2.0
	50_86_num_10_3.txt	3.13	695	1	2633.37	0.0	0	3	2.04
	50_86_num_10_4.txt	2.03	651	1	1572.75	0.0	0	4	2.0
	50_86_num_11_1.txt	11.9	1412	1	10320.8	0.0	0	1	2.0
	50_86_num_11_2.txt	8.97	1373	2	7407.81	0.0	0	2	2.0
	50_86_num_11_3.txt	8.13	1433	3	6835.19	0.0	0	3	2.0
	50_86_num_11_4.txt	2.67	676	1	2189.52	0.0	0	4	2.08
	50_86_num_1_1.txt	3.62	1080	1	2699.36	0.0	0	1	2.1
	50_86_num_12_1.txt	2.47	710	1	1974.62	0.0	0	1	2.1
	50_86_num_12_2.txt	6.2	1029	1	5293.47	0.0	0	2	2.0
	50_86_num_12_3.txt	2.53	754	1	1958.74	0.0	0	3	2.07
	50_86_num_12_4.txt	2.47	865	1	1831.11	0.0	0	4	2.0
	50_86_num_1_2.txt	3.52	815	2	2819.82	0.0	0	2	2.0
	50_86_num_13_1.txt	5.11	961	1	4259.7	0.0	0	1	2.0
	50_86_num_13_2.txt	6.86	1162	1	5753.66	0.0	0	2	2.09
	50_86_num_13_3.txt	1.55	577	1	1180.32	0.0	0	3	2.0
	50_86_num_13_4.txt	5.88	1014	2	4993.19	0.0	0	4	2.0
	50_86_num_1_3.txt	4.06	942	1	3242.73	0.0	0	3	2.0
	50_86_num_14_1.txt	10.86	1531	1	9194.0	0.0	0	1	2.0
	50_86_num_14_2.txt	4.65	1105	2	3720.49	0.0	0	2	2.0
	50_86_num_14_3.txt	1.11	418	1	854.25	0.0	0	3	2.14

50_86_num_14_4.txt	4.57	852	1	3858.49	0.0	0	4	2.05
50_86_num_1_4.txt	1.3	469	1	999.85	0.0	0	4	2.12
50_86_num_15_1.txt	11.66	1735	5	9868.26	0.0	0	1	2.0
50_86_num_15_2.txt	11.49	1412	2	9966.9	0.0	0	2	2.0
50_86_num_15_3.txt	1.28	523	1	959.33	0.0	0	3	2.0
50_86_num_15_4.txt	6.24	1183	1	5180.06	0.0	0	4	2.0
50_86_num_16_1.txt	7.23	1158	2	6121.26	0.0	0	1	2.0
50_86_num_16_2.txt	4.06	844	1	3343.61	0.0	0	2	2.09
50_86_num_16_3.txt	4.43	801	1	3803.62	0.0	0	3	2.05
50_86_num_16_4.txt	3.62	927	1	2859.98	0.0	0	4	2.15
50_86_num_17_1.txt	0.88	337	1	687.64	0.0	0	1	2.11
50_86_num_17_2.txt	4.96	1005	1	4035.35	0.0	0	2	2.0
50_86_num_17_3.txt	8.98	1354	1	7582.22	0.0	0	3	2.0
50_86_num_17_4.txt	2.84	826	1	2199.76	0.0	0	4	2.0
50_86_num_18_1.txt	2.6	773	1	2047.92	0.0	0	1	2.06
50_86_num_18_2.txt	2.89	745	1	2301.57	0.0	0	2	2.05
50_86_num_18_3.txt	2.59	753	1	2044.36	0.0	0	3	2.08
50_86_num_18_4.txt	1.55	566	1	1162.28	0.0	0	4	2.14
50_86_num_19_1.txt	4.26	828	1	3578.54	0.0	0	1	2.0
50_86_num_19_2.txt	1.26	681	1	806.23	0.0	0	2	2.25
50_86_num_19_3.txt	0.31	219	1	203.89	0.0	0	3	2.33
50_86_num_19_4.txt	2.89	706	1	2376.48	0.0	0	4	2.06
50_86_num_20_1.txt	0.69	369	1	490.91	0.0	0	1	2.0
50_86_num_20_2.txt	4.51	835	1	3839.37	0.0	0	2	2.08
50_86_num_20_3.txt	5.88	1086	2	4884.42	0.0	0	3	2.0
50_86_num_20_4.txt	5.32	921	1	4525.8	0.0	0	4	2.06
50_86_num_21_1.txt	2.49	898	1	1746.9	0.0	0	1	2.11
50_86_num_21_2.txt	2.17	717	1	1672.59	0.0	0	2	2.2
50_86_num_21_3.txt	5.13	907	1	4363.54	0.0	0	3	2.05
50_86_num_21_4.txt	3.53	749	1	2959.52	0.0	0	4	2.07
50_86_num_2_1.txt	9.3	1471	1	7783.07	0.0	0	1	2.0
50_86_num_22_1.txt	1.53	617	1	1130.22	0.0	0	1	2.0
50_86_num_22_2.txt	3.95	895	1	3212.55	0.0	0	2	2.0
50_86_num_22_3.txt	3.23	815	1	2607.27	0.0	0	3	2.07
50_86_num_22_4.txt	2.69	768	1	2093.32	0.0	0	4	2.09
50_86_num_2_2.txt	5.45	887	1	4696.86	0.0	0	2	2.04
50_86_num_23_1.txt	16.82	2126	7	14343.16	0.0	0	1	2.0
50_86_num_23_2.txt	9.43	1433	1	7763.03	0.0	0	2	2.0
50_86_num_23_3.txt	3.39	722	1	2874.66	0.0	0	3	2.04
50_86_num_23_4.txt	3.87	738	1	3288.34	0.0	0	4	2.03
50_86_num_2_3.txt	3.09	905	1	2317.28	0.0	0	3	2.13
50_86_num_24_1.txt	3.89	840	1	3207.21	0.0	0	1	2.03
50_86_num_24_2.txt	2.31	590	1	1912.14	0.0	0	2	2.08
50_86_num_24_3.txt	3.9	904	1	3154.11	0.0	0	3	2.0
50_86_num_24_4.txt	4.76	911	1	3987.03	0.0	0	4	2.07
50_86_num_2_4.txt	5.92	1014	1	4942.99	0.0	0	4	2.06
50_86_num_3_1.txt	3.19	1005	1	2346.35	0.0	0	1	2.06
50_86_num_3_2.txt	2.89	780	1	2291.68	0.0	0	2	2.16
50_86_num_3_3.txt	4.93	927	1	4160.67	0.0	0	3	2.0
50_86_num_3_4.txt	4.27	912	1	3529.77	0.0	0	4	2.1
50_86_num_4_1.txt	5.01	903	2	4202.84	0.0	0	1	2.0
50_86_num_4_2.txt	7.08	1142	1	5979.66	0.0	0	2	2.0
50_86_num_4_3.txt	8.74	1292	1	7401.91	0.0	0	3	2.04
50_86_num_4_4.txt	4.64	985	1	3740.56	0.0	0	4	2.11
50_86_num_5_1.txt	19.04	2562	5	15709.06	0.0	0	1	2.0
50_86_num_5_2.txt	2.69	784	1	2079.08	0.0	0	2	2.06

		50_86_num_5_3.txt	3.38	822	1	2700.28	0.0	0	3	2.1
		50_86_num_5_4.txt	4.13	869	1	3345.54	0.0	0	4	2.04
		50_86_num_6_1.txt	5.87	1174	3	4734.34	0.0	0	1	2.0
		50_86_num_6_2.txt	6.58	1168	2	5496.52	0.0	0	2	2.0
		50_86_num_6_3.txt	2.49	812	1	1867.01	0.0	0	3	2.13
		50_86_num_6_4.txt	5.06	1032	1	4172.0	0.0	0	4	2.0
		50_86_num_7_1.txt	6.62	1229	1	5388.63	0.0	0	1	2.0
		50_86_num_7_2.txt	2.12	687	1	1654.58	0.0	0	2	2.0
		50_86_num_7_3.txt	0.97	504	1	689.63	0.0	0	3	2.25
		50_86_num_7_4.txt	4.48	911	1	3650.93	0.0	0	4	2.03
		50_86_num_8_1.txt	6.02	975	1	5161.74	0.0	0	1	2.0
		50_86_num_8_2.txt	4.49	1114	1	3528.28	0.0	0	2	2.0
		50_86_num_8_3.txt	1.45	533	1	1086.58	0.0	0	3	2.17
		50_86_num_8_4.txt	3.52	783	1	2936.8	0.0	0	4	2.06
		50_86_num_9_1.txt	7.62	1166	2	6453.7	0.0	0	1	2.0
		50_86_num_9_2.txt	2.78	774	2	2085.65	0.0	0	2	2.0
		50_86_num_9_3.txt	6.45	1039	1	5481.72	0.0	0	3	2.0
		50_86_num_9_4.txt	1.79	717	1	1241.03	0.0	0	4	2.08
	ED-color	1-FullIns_3	0.42	106	1	375.02	0.0	0	2	2.63
		2-FullIns_3	12.75	1854	17	11054.64	0.0	0	2	3.0
		anna	10817.3	73378	376	1.41956934e6	0.0	0	356	5.5
		huck	0.42	112	1	372.85	0.0	0	312	5.5
		jean	0.46	79	1	423.77	0.0	0	198	5.0
		miles1500	0.6	182	1	523.19	0.0	0	1967	36.5
		r125.1c	4.37	434	1	3862.28	0.0	0	515	35.0
		3-FullIns_3	0.32	69	1	304.93	0.0	0	2	3.5
		4-FullIns_3	0.42	48	1	412.22	0.0	0	2	4.0
		5-FullIns_3	1154.84	32861	71	327707.45	0.0	0	2	4.5
	HD-color	david	471.93	18235	309	219038.0	0.0	0	269	5.63
		miles1000	9.9	716	46	5989.64	0.0	0	669	21.5
		mulsol.i.1	0.33	172	1	248.81	0.0	0	2073	25.0
		miles500	123.36	3578	387	23042.39	0.0	0	263	10.5
		miles750	45.53	2085	201	18807.56	0.0	0	448	16.5
		mulsol.i.2	793.96	15957	201	258583.5	0.0	0	1199	15.77
		r125.1	1.82	652	6	1090.92	0.0	0	126	3.0
		zeroin.i.1	19.5	1433	6	13256.56	0.0	0	2181	24.5
		zeroin.i.3	10824.8	43480	870	1.18032593e6	0.0	0	959	15.33
$\tilde{P}_1$		fpsol2.i.1	10818.2	9167	779	303408.95	0.0	0	6995	33.0
	DIMACS	adjnoun.graph	7.42	640	1	7024.05	0.0	0	226	33.11
		chesapeake.graph	0.21	92	2	170.73	0.0	0	26	11.0
		dolphins.graph	1.23	269	1	1048.87	0.0	0	78	17.99
		jazz.graph	1617.86	3513	497	1.28486483e6	0.0	0	75	47.54
		karate.graph	0.35	77	5	324.63	0.0	0	89	11.75
		lesmis.graph	0.67	211	1	603.23	0.0	0	254	20.5
		polbooks.graph	2.78	598	1	2494.69	0.0	0	20	27.08
	comp-DIMACS	chesapeake.graph_comp	1.22	579	1	988.42	0.0	0	3	3.73
		dolphins.graph_comp	11.42	2172	19	8346.07	0.0	0	14	3.0
		karate.graph_comp	0.29	176	4	234.4	0.0	0	33	3.0
		lesmis.graph_comp	6.0	1164	1	4786.4	0.0	0	207	6.33
		polbooks.graph_comp	282.64	15148	90	152590.77	0.0	0	49	4.22
	others	miles250	564.7	20590	216	136663.14	0.0	0	136	4.5
		1-FullIns_4	22.15	2251	1	15744.25	0.0	0	2	3.13
		zeroin.i.2	14.94	1502	1	10951.04	0.0	0	959	15.33

	mulsol.i.3	10812.0	41141	932	1.59998722e6	0.0	0	1208	15.77
	mulsol.i.5	7.49	899	1	5605.52	0.0	0	1229	15.77
Randoms	50_86_num_0_1	19.14	2302	7	16388.13	0.0	0	1	2.0
	50_86_num_0_2	9.68	1639	5	7968.55	0.0	0	2	2.0
	50_86_num_0_3	11.58	1702	1	9492.5	0.0	0	3	2.0
	50_86_num_0_4	5.31	1252	3	4218.89	0.0	0	4	2.0
	50_86_num_10_1	2.35	686	1	1852.46	0.0	0	1	2.05
	50_86_num_10_2	2.82	766	1	2218.06	0.0	0	2	2.0
	50_86_num_10_3	3.14	695	1	2642.04	0.0	0	3	2.04
	50_86_num_10_4	2.0	651	1	1538.76	0.0	0	4	2.0
	50_86_num_11_1	12.01	1412	1	10417.38	0.0	0	1	2.0
	50_86_num_11_2	9.09	1373	2	7528.06	0.0	0	2	2.0
	50_86_num_11_3	8.08	1433	3	6787.03	0.0	0	3	2.0
	50_86_num_11_4	2.7	676	1	2217.29	0.0	0	4	2.08
	50_86_num_1_1	3.58	1080	1	2658.49	0.0	0	1	2.1
	50_86_num_12_1	2.45	710	1	1955.92	0.0	0	1	2.1
	50_86_num_12_2	5.91	1029	1	5009.64	0.0	0	2	2.0
	50_86_num_12_3	2.55	754	1	1985.73	0.0	0	3	2.07
	50_86_num_12_4	2.52	865	1	1880.92	0.0	0	4	2.0
	50_86_num_1_2	3.54	815	2	2823.56	0.0	0	2	2.0
	50_86_num_13_1	5.23	961	1	4377.18	0.0	0	1	2.0
	50_86_num_13_2	7.04	1162	1	5924.53	0.0	0	2	2.09
	50_86_num_13_3	1.57	577	1	1201.54	0.0	0	3	2.0
	50_86_num_13_4	5.92	1014	2	5023.27	0.0	0	4	2.0
	50_86_num_1_3	4.05	942	1	3244.5	0.0	0	3	2.0
	50_86_num_14_1	10.94	1531	1	9257.92	0.0	0	1	2.0
	50_86_num_14_2	4.7	1105	2	3771.06	0.0	0	2	2.0
	50_86_num_14_3	1.11	418	1	858.42	0.0	0	3	2.14
	50_86_num_14_4	4.68	852	1	3968.85	0.0	0	4	2.05
	50_86_num_1_4	1.32	469	1	1023.25	0.0	0	4	2.12
	50_86_num_15_1	11.66	1735	5	9846.22	0.0	0	1	2.0
	50_86_num_15_2	11.33	1412	2	9815.58	0.0	0	2	2.0
	50_86_num_15_3	1.27	523	1	953.05	0.0	0	3	2.0
	50_86_num_15_4	6.29	1183	1	5233.73	0.0	0	4	2.0
	50_86_num_16_1	7.26	1158	2	6147.78	0.0	0	1	2.0
	50_86_num_16_2	4.19	844	1	3470.95	0.0	0	2	2.09
	50_86_num_16_3	4.51	801	1	3882.99	0.0	0	3	2.05
	50_86_num_16_4	3.53	927	1	2770.88	0.0	0	4	2.15
	50_86_num_17_1	0.86	337	1	668.11	0.0	0	1	2.11
	50_86_num_17_2	4.96	1005	1	4037.5	0.0	0	2	2.0
	50_86_num_17_3	8.87	1354	1	7470.05	0.0	0	3	2.0
	50_86_num_17_4	2.8	826	1	2160.49	0.0	0	4	2.0
	50_86_num_18_1	2.65	773	1	2099.47	0.0	0	1	2.06
	50_86_num_18_2	2.89	745	1	2308.84	0.0	0	2	2.05
	50_86_num_18_3	2.54	753	1	1989.43	0.0	0	3	2.08
	50_86_num_18_4	1.5	566	1	1114.45	0.0	0	4	2.14
	50_86_num_19_1	4.16	828	1	3481.7	0.0	0	1	2.0
	50_86_num_19_2	1.24	681	1	788.42	0.0	0	2	2.25
	50_86_num_19_3	0.3	219	1	198.8	0.0	0	3	2.33
	50_86_num_19_4	2.9	706	1	2384.22	0.0	0	4	2.06
	50_86_num_20_1	0.69	369	1	493.47	0.0	0	1	2.0
	50_86_num_20_2	4.43	835	1	3755.2	0.0	0	2	2.08
	50_86_num_20_3	5.91	1086	2	4911.62	0.0	0	3	2.0
	50_86_num_20_4	5.36	921	1	4561.92	0.0	0	4	2.06
	50_86_num_21_1	2.48	898	1	1734.64	0.0	0	1	2.11
	50_86_num_21_2	2.16	717	1	1659.32	0.0	0	2	2.2

		50_86_num_21_3	5.14	907	1	4365.48	0.0	0	3	2.05
		50_86_num_21_4	3.49	749	1	2915.2	0.0	0	4	2.07
		50_86_num_2_1	9.23	1471	1	7723.32	0.0	0	1	2.0
		50_86_num_22_1	1.56	617	1	1153.56	0.0	0	1	2.0
		50_86_num_22_2	3.96	895	1	3222.75	0.0	0	2	2.0
		50_86_num_22_3	3.2	815	1	2575.5	0.0	0	3	2.07
		50_86_num_22_4	2.7	768	1	2103.23	0.0	0	4	2.09
		50_86_num_2_2	5.43	887	1	4667.2	0.0	0	2	2.04
		50_86_num_23_1	16.81	2126	7	14333.22	0.0	0	1	2.0
		50_86_num_23_2	9.55	1433	1	7893.02	0.0	0	2	2.0
		50_86_num_23_3	3.34	722	1	2820.89	0.0	0	3	2.04
		50_86_num_23_4	3.9	738	1	3311.52	0.0	0	4	2.03
		50_86_num_2_3	3.16	905	1	2385.22	0.0	0	3	2.13
		50_86_num_24_1	3.9	840	1	3218.58	0.0	0	1	2.03
		50_86_num_24_2	2.29	590	1	1890.03	0.0	0	2	2.08
		50_86_num_24_3	3.87	904	1	3117.6	0.0	0	3	2.0
		50_86_num_24_4	4.72	911	1	3948.13	0.0	0	4	2.07
		50_86_num_2_4	5.85	1014	1	4877.56	0.0	0	4	2.06
		50_86_num_3_1	3.18	1005	1	2341.67	0.0	0	1	2.06
		50_86_num_3_2	2.93	780	1	2330.92	0.0	0	2	2.16
		50_86_num_3_3	4.99	927	1	4220.59	0.0	0	3	2.0
		50_86_num_3_4	4.25	912	1	3507.48	0.0	0	4	2.1
		50_86_num_4_1	4.96	903	2	4155.2	0.0	0	1	2.0
		50_86_num_4_2	7.05	1142	1	5955.76	0.0	0	2	2.0
		50_86_num_4_3	5.18	980	1	4369.39	0.0	0	4	2.04
		50_86_num_4_4	4.59	985	1	3698.29	0.0	0	4	2.11
		50_86_num_5_1	18.92	2562	5	15601.97	0.0	0	1	2.0
		50_86_num_5_2	2.76	784	1	2142.58	0.0	0	2	2.06
		50_86_num_5_3	3.37	822	1	2695.89	0.0	0	3	2.1
		50_86_num_5_4	4.09	869	1	3310.09	0.0	0	4	2.04
		50_86_num_6_1	5.92	1174	3	4782.4	0.0	0	1	2.0
		50_86_num_6_2	6.53	1168	2	5443.54	0.0	0	2	2.0
		50_86_num_6_3	2.52	812	1	1888.55	0.0	0	3	2.13
		50_86_num_6_4	5.06	1032	1	4168.91	0.0	0	4	2.0
		50_86_num_7_1	6.55	1229	1	5307.75	0.0	0	1	2.0
		50_86_num_7_2	2.1	687	1	1634.72	0.0	0	2	2.0
		50_86_num_7_3	1.0	504	1	717.37	0.0	0	3	2.25
		50_86_num_7_4	4.7	927	1	3847.26	0.0	0	5	2.03
		50_86_num_8_1	5.98	975	1	5125.41	0.0	0	1	2.0
		50_86_num_8_2	4.54	1114	1	3577.16	0.0	0	2	2.0
		50_86_num_8_3	1.46	533	1	1098.22	0.0	0	3	2.17
		50_86_num_8_4	3.48	783	1	2894.56	0.0	0	5	2.06
		50_86_num_9_1	7.57	1166	2	6407.44	0.0	0	1	2.0
		50_86_num_9_2	2.82	774	2	2125.41	0.0	0	2	2.0
		50_86_num_9_3	6.41	1039	1	5436.51	0.0	0	4	2.0
		50_86_num_9_4	1.79	717	1	1244.58	0.0	0	4	2.08
BCP	E-color	1-Insertions_4	11.61	1332	1	9351.01	0.0	0	-	2.41
		2-Insertions_3	0.57	292	1	284.04	0.0	0	-	2.0
		3-Insertions_3	5.11	1110	1	3802.78	0.0	0	-	2.0
		myciel3	0.01	8	1	1.45	0.0	0	-	2.0
		myciel4	0.36	74	1	348.04	0.0	0	-	2.56
		myciel5	3.21	434	1	2945.82	0.0	0	-	3.16
		queen5_5	0.31	99	4	259.69	0.0	0	-	5.0
		queen6_6	0.52	184	1	426.77	0.0	0	-	5.25
		queen7_7	54.68	1179	130	51726.61	0.43	2	-	5.84
		4-Insertions_3	31.75	3004	1	18114.65	0.0	0	-	2.0

	ED-color	1-FullIns_3	1.02	122	1	987.83	0.0	0	-	2.63
		2-FullIns_3	4.51	1393	10	3590.01	0.0	1	-	3.0
		anna	1.93	551	1	1401.48	0.0	0	-	5.5
		huck	0.68	274	1	563.85	0.0	0	-	5.5
		jean	0.4	150	1	354.37	0.0	0	-	5.0
		miles1500	14.98	541	29	6220.34	7.58	21	-	36.5
		r125.1c	4.51	502	2	2513.84	1.77	0	-	35.0
		3-FullIns_3	0.58	226	1	480.5	0.0	0	-	3.5
		4-FullIns_3	0.8	53	2	753.29	0.0	0	-	4.0
		5-FullIns_3	0.57	97	1	332.55	0.0	2	-	4.5
	HD-color	david	855.01	26036	333	395019.42	21.93	33	-	5.67
		miles1000	41.43	2259	58	23128.45	7.66	21	-	21.5
		miles500	153.46	7862	192	53901.89	8.45	34	-	10.5
		miles750	149.09	4136	339	46949.7	8.48	26	-	16.5
		mulsol.i.1	2.82	325	11	1481.47	0.82	4	-	25.0
		mulsol.i.2	313.53	9390	114	117727.79	20.72	16	-	16.0
		r125.1	1881.18	46977	95	534748.09	1.75	18	-	3.0
		zeroin.i.1	42.9	1519	36	16344.3	11.95	14	-	24.5
		zeroin.i.3	-	54381	534	1.6015516e6	874.7	61	-	15.33
		fpsol2.i.1	-	16307	489	681387.2	2220.13	79	-	33.0
	DIMACS	adjnoun.graph	5.9	700	1	5483.92	0.0	0	-	33.11
		chesapeake.graph	0.18	118	1	155.73	0.0	1	-	11.0
		dolphins.graph	0.92	296	1	840.37	0.0	0	-	18.0
		jazz.graph	2053.49	2785	164	690011.16	1265.33	19	-	47.54
		karate.graph	0.62	75	3	583.29	0.01	4	-	11.75
		lesmis.graph	0.94	258	1	728.49	0.13	3	-	20.5
		polbooks.graph	3.28	648	1	2939.98	0.0	0	-	27.08
	comp_DIMACS	chesapeake_comp	1.52	676	1	1224.15	0.0	0	-	3.73
		dolphins_comp	34.8	5437	42	24198.55	0.06	6	-	3.0
		karate_comp	0.9	104	2	859.72	0.0	1	-	3.0
		lesmis_comp	43.16	4116	48	30373.9	0.95	10	-	6.33
		polbooks_comp	-	92568	681	3.05677385e6	94.41	12	-	4.22
	other	miles250	-	81773	555	2.63706592e6	109.04	40	-	4.5
		mulsol.i.3	2479.6	27150	272	704283.53	364.51	40	-	16.0
		mulsol.i.5	5300.42	35873	425	1.0736947e6	773.73	46	-	16.0
		zeroin.i.2	-	92452	33	897344.7	43.16	18	-	15.33
		1-FullIns_4	-	98844	1	72281.9	0.0	0	-	3.13
BP + DI	ED-color	1-FullIns_3	0.46	106	1	427.88	-	-	2	2.63
		2-FullIns_3	7.47	1854	17	5973.77	-	-	2	3.0
		anna	599.01	25024	60	234967.79	-	-	214	5.5
		huck	0.44	133	1	370.54	-	-	254	5.5
		jean	0.18	92	1	147.99	-	-	156	5.0
		miles1500	0.6	187	1	528.52	-	-	1967	36.5
		r125.1c	3.38	388	1	3240.21	-	-	515	35.0
		3-FullIns_3	0.04	69	1	23.71	-	-	2	3.5
		4-FullIns_3	0.15	48	1	48.26	-	-	2	4.0
		5-FullIns_3	824.7	32861	71	289135.95	-	-	2	4.5
	HD-color	david	290.07	13452	241	132482.1	-	-	190	5.63
		miles1000	20.42	861	128	9248.05	-	-	669	21.5
		miles500	162.09	7271	340	49452.87	-	-	261	10.5
		miles750	0.8	328	1	322.6	-	-	448	16.5
		mulsol.i.1	13.58	184	67	1989.64	-	-	2073	25.0

	mulsol.i.2	771.85	13930	201	277393.95	-	-	1199	15.77
	r125.1	546.88	25143	110	150723.34	-	-	120	3.0
	zeroin.i.1	24.36	1567	12	14466.7	-	-	2181	24.5
	zeroin.i.3	713.83	14419	148	227955.89	-	-	959	15.33
	fpsol2.i.1	984.74	3508	114	98710.19	-	-	6995	33.0
DIMACS	adjnoun.graph	7.76	623	1	7354.99	-	-	226	33.11
	chesapeake.graph	0.15	92	2	124.37	-	-	26	11.0
	dolphins.graph	1.19	279	1	1107.48	-	-	78	17.99
	jazz.graph	2690.75	4080	922	1.99999299e6	-	-	75	47.54
	karate.graph	0.32	76	5	298.29	-	-	89	11.75
	lesmis.graph	0.98	252	1	901.27	-	-	254	20.5
	polbooks.graph	3.0	598	1	2698.4	-	-	20	27.08
comp_DIMACS	chesapeake_comp	1.34	579	1	1086.2	-	-	3	3.73
	dolphins_comp	22.07	3481	31	16218.69	-	-	10	3.0
	karate_comp	0.13	55	1	122.93	-	-	24	3.0
	lesmis_comp	6.06	1136	1	4777.86	-	-	156	6.33
	polbooks_comp	-	95901	623	2.6523055e6	-	-	41	4.22
other	miles250	146.07	9986	90	57070.53	-	-	123	4.5
	mulsol.i.3	8980.61	43066	909	1.61343634e6	-	-	1208	15.77
	mulsol.i.5	413.07	10958	119	168310.75	-	-	1229	15.77
	zeroin.i.2	12.66	1314	1	9366.9	-	-	959	15.33
	1-FullIns_4	23.04	2251	1	16409.78	-	-	2	3.13
Randoms	50_86_num_0_1	19.08	2302	7	16326.04	0.0	0	1	2.0
	50_86_num_0_2	9.64	1639	5	7915.14	0.0	0	2	2.0
	50_86_num_0_3	11.49	1702	1	9417.07	0.0	0	3	2.0
	50_86_num_0_4	5.2	1252	3	4116.52	0.0	0	4	2.0
	50_86_num_10_1	2.44	686	1	1921.64	0.0	0	1	2.05
	50_86_num_10_2	2.76	766	1	2155.43	0.0	0	2	2.0
	50_86_num_10_3	3.13	695	1	2633.37	0.0	0	3	2.04
	50_86_num_10_4	2.03	651	1	1572.75	0.0	0	4	2.0
	50_86_num_11_1	11.9	1412	1	10320.8	0.0	0	1	2.0
	50_86_num_11_2	8.97	1373	2	7407.81	0.0	0	2	2.0
	50_86_num_11_3	8.13	1433	3	6835.19	0.0	0	3	2.0
	50_86_num_11_4	2.67	676	1	2189.52	0.0	0	4	2.08
	50_86_num_1_1	3.62	1080	1	2699.36	0.0	0	1	2.1
	50_86_num_12_1	2.47	710	1	1974.62	0.0	0	1	2.1
	50_86_num_12_2	6.2	1029	1	5293.47	0.0	0	2	2.0
	50_86_num_12_3	2.53	754	1	1958.74	0.0	0	3	2.07
	50_86_num_12_4	2.47	865	1	1831.11	0.0	0	4	2.0
	50_86_num_1_2	3.52	815	2	2819.82	0.0	0	2	2.0
	50_86_num_13_1	5.11	961	1	4259.7	0.0	0	1	2.0
	50_86_num_13_2	6.86	1162	1	5753.66	0.0	0	2	2.09
	50_86_num_13_3	1.55	577	1	1180.32	0.0	0	3	2.0
	50_86_num_13_4	5.88	1014	2	4993.19	0.0	0	4	2.0
	50_86_num_1_3	4.06	942	1	3242.73	0.0	0	3	2.0
	50_86_num_14_1	10.86	1531	1	9194.0	0.0	0	1	2.0
	50_86_num_14_2	4.65	1105	2	3720.49	0.0	0	2	2.0
	50_86_num_14_3	1.11	418	1	854.25	0.0	0	3	2.14
	50_86_num_14_4	4.57	852	1	3858.49	0.0	0	4	2.05
	50_86_num_1_4	1.3	469	1	999.85	0.0	0	4	2.12
	50_86_num_15_1	11.66	1735	5	9868.26	0.0	0	1	2.0
	50_86_num_15_2	11.49	1412	2	9966.9	0.0	0	2	2.0
	50_86_num_15_3	1.28	523	1	959.33	0.0	0	3	2.0

50_86_num_15_4	6.24	1183	1	5180.06	0.0	0	4	2.0
50_86_num_16_1	7.23	1158	2	6121.26	0.0	0	1	2.0
50_86_num_16_2	4.06	844	1	3343.61	0.0	0	2	2.09
50_86_num_16_3	4.43	801	1	3803.62	0.0	0	3	2.05
50_86_num_16_4	3.62	927	1	2859.98	0.0	0	4	2.15
50_86_num_17_1	0.88	337	1	687.64	0.0	0	1	2.11
50_86_num_17_2	4.96	1005	1	4035.35	0.0	0	2	2.0
50_86_num_17_3	8.98	1354	1	7582.22	0.0	0	3	2.0
50_86_num_17_4	2.84	826	1	2199.76	0.0	0	4	2.0
50_86_num_18_1	2.6	773	1	2047.92	0.0	0	1	2.06
50_86_num_18_2	2.89	745	1	2301.57	0.0	0	2	2.05
50_86_num_18_3	2.59	753	1	2044.36	0.0	0	3	2.08
50_86_num_18_4	1.55	566	1	1162.28	0.0	0	4	2.14
50_86_num_19_1	4.26	828	1	3578.54	0.0	0	1	2.0
50_86_num_19_2	1.26	681	1	806.23	0.0	0	2	2.25
50_86_num_19_3	0.31	219	1	203.89	0.0	0	3	2.33
50_86_num_19_4	2.89	706	1	2376.48	0.0	0	4	2.06
50_86_num_20_1	0.69	369	1	490.91	0.0	0	1	2.0
50_86_num_20_2	4.51	835	1	3839.37	0.0	0	2	2.08
50_86_num_20_3	5.88	1086	2	4884.42	0.0	0	3	2.0
50_86_num_20_4	5.32	921	1	4525.8	0.0	0	4	2.06
50_86_num_21_1	2.49	898	1	1746.9	0.0	0	1	2.11
50_86_num_21_2	2.17	717	1	1672.59	0.0	0	2	2.2
50_86_num_21_3	5.13	907	1	4363.54	0.0	0	3	2.05
50_86_num_21_4	3.53	749	1	2959.52	0.0	0	4	2.07
50_86_num_2_1	9.3	1471	1	7783.07	0.0	0	1	2.0
50_86_num_22_1	1.53	617	1	1130.22	0.0	0	1	2.0
50_86_num_22_2	3.95	895	1	3212.55	0.0	0	2	2.0
50_86_num_22_3	3.23	815	1	2607.27	0.0	0	3	2.07
50_86_num_22_4	2.69	768	1	2093.32	0.0	0	4	2.09
50_86_num_2_2	5.45	887	1	4696.86	0.0	0	2	2.04
50_86_num_23_1	16.82	2126	7	14343.16	0.0	0	1	2.0
50_86_num_23_2	9.43	1433	1	7763.03	0.0	0	2	2.0
50_86_num_23_3	3.39	722	1	2874.66	0.0	0	3	2.04
50_86_num_23_4	3.87	738	1	3288.34	0.0	0	4	2.03
50_86_num_2_3	3.09	905	1	2317.28	0.0	0	3	2.13
50_86_num_24_1	3.89	840	1	3207.21	0.0	0	1	2.03
50_86_num_24_2	2.31	590	1	1912.14	0.0	0	2	2.08
50_86_num_24_3	3.9	904	1	3154.11	0.0	0	3	2.0
50_86_num_24_4	4.76	911	1	3987.03	0.0	0	4	2.07
50_86_num_2_4	5.92	1014	1	4942.99	0.0	0	4	2.06
50_86_num_3_1	3.19	1005	1	2346.35	0.0	0	1	2.06
50_86_num_3_2	2.89	780	1	2291.68	0.0	0	2	2.16
50_86_num_3_3	4.93	927	1	4160.67	0.0	0	3	2.0
50_86_num_3_4	4.27	912	1	3529.77	0.0	0	4	2.1
50_86_num_4_1	5.01	903	2	4202.84	0.0	0	1	2.0
50_86_num_4_2	7.08	1142	1	5979.66	0.0	0	2	2.0
50_86_num_4_3	8.74	1292	1	7401.91	0.0	0	3	2.04
50_86_num_4_4	4.64	985	1	3740.56	0.0	0	4	2.11
50_86_num_5_1	19.04	2562	5	15709.06	0.0	0	1	2.0
50_86_num_5_2	2.69	784	1	2079.08	0.0	0	2	2.06
50_86_num_5_3	3.38	822	1	2700.28	0.0	0	3	2.1
50_86_num_5_4	4.13	869	1	3345.54	0.0	0	4	2.04
50_86_num_6_1	5.87	1174	3	4734.34	0.0	0	1	2.0
50_86_num_6_2	6.58	1168	2	5496.52	0.0	0	2	2.0
50_86_num_6_3	2.49	812	1	1867.01	0.0	0	3	2.13

		50_86_num_6_4	5.06	1032	1	4172.0	0.0	0	4	2.0
		50_86_num_7_1	6.62	1229	1	5388.63	0.0	0	1	2.0
		50_86_num_7_2	2.12	687	1	1654.58	0.0	0	2	2.0
		50_86_num_7_3	0.97	504	1	689.63	0.0	0	3	2.25
		50_86_num_7_4	4.48	911	1	3650.93	0.0	0	4	2.03
		50_86_num_8_1	6.02	975	1	5161.74	0.0	0	1	2.0
		50_86_num_8_2	4.49	1114	1	3528.28	0.0	0	2	2.0
		50_86_num_8_3	1.45	533	1	1086.58	0.0	0	3	2.17
		50_86_num_8_4	3.52	783	1	2936.8	0.0	0	4	2.06
		50_86_num_9_1	7.62	1166	2	6453.7	0.0	0	1	2.0
		50_86_num_9_2	2.78	774	2	2085.65	0.0	0	2	2.0
		50_86_num_9_3	6.45	1039	1	5481.72	0.0	0	3	2.0
		50_86_num_9_4	1.79	717	1	1241.03	0.0	0	4	2.08
BCP + DI $\Psi = 0$	ED-color	1-FullIns_3	0.46	106	1	426.35	0.0	0	2	2.63
		2-FullIns_3	1.36	385	3	1166.88	0.0	1	2	3.0
		anna	1.57	415	1	1067.01	0.04	13	214	5.5
		huck	1.19	174	1	1038.43	0.02	64	254	5.5
		jean	0.19	92	1	157.71	0.0	0	156	5.0
		miles1500	0.73	187	1	640.0	0.0	0	1967	36.5
		r125.1c	3.3	388	1	3141.0	0.0	0	515	35.0
		3-FullIns_3	0.34	69	1	319.04	0.0	0	2	3.5
		4-FullIns_3	0.3	48	1	283.69	0.0	0	2	4.0
		5-FullIns_3	0.82	48	1	794.8	0.0	3	2	4.5
	HD-color	david	1161.08	25844	577	490441.3	28.13	138	190	5.67
		miles1000	33.72	1207	105	18412.18	1.94	256	669	21.5
		miles500	349.91	9151	438	105849.72	5.56	206	261	10.5
		miles750	1.25	328	1	797.44	0.18	72	448	16.5
		mulsol.i.1	1.11	173	1	925.77	0.05	96	2073	25.0
		mulsol.i.2	9847.02	35876	1149	2.3765158e6	16.16	1164	1199	15.77
		r125.1	586.23	21726	122	248010.03	0.39	47	120	3.0
		zeroin.i.1	47.83	1378	1	20844.72	22.63	721	2181	24.5
		zeroin.i.3	865.14	12092	201	318406.59	0.87	294	959	15.33
		fpsol2.i.1	7491.91	8913	538	626143.5	2.1	240	6995	33.0
	DIMACS	adjnoun.graph	7.79	623	1	7374.71	0.0	0	226	33.11
		chesapeake.graph	0.21	92	1	176.06	0.01	11	26	11.0
		dolphins.graph	1.21	279	1	1123.93	0.0	0	78	18.0
		jazz.graph	-	4256	993	2.82543962e6	7240.38	103	75	47.54
		karate.graph	0.22	53	1	186.56	0.0	6	89	11.75
		lesmis.graph	1.01	252	1	915.93	0.0	0	254	20.5
		polbooks.graph	2.95	598	1	2625.08	0.0	0	20	27.08
	comp_DIMACS	chesapeake_comp	1.32	579	1	1054.11	0.0	0	3	3.73
		karate_comp	0.24	55	1	229.61	0.0	0	24	3.0
		lesmis_comp	6.48	1136	1	5033.73	0.16	15	156	6.33
		polbooks_comp	-	98482	588	3.19722503e6	17.15	7	41	4.22
	other	miles250	-	52137	1616	1.9057239e6	181.65	681	123	4.5
		mulsol.i.3	-	44462	892	2.513178e6	6.22	604	1208	15.77
		mulsol.i.5	169.8	5937	50	101973.02	0.21	162	1229	15.77
		zeroin.i.2	-	100717	132	2.24204773e6	0.99	257	959	15.33
		1-FullIns_4	-	88577	1	86143.9	0.0	0	2	3.13
BCP + DI $\Psi \neq 0$	ED-color	1-FullIns_3	0.47	106	1	339.69	0.0	0	2	2.63
		2-FullIns_3	1.56	385	3	1373.98	0.0	1	2	3.0
		anna	1.63	417	1	1129.84	0.04	11	214	5.5
		huck	1.12	172	1	947.18	0.03	53	254	5.5

		jean	0.1	92	1	69.93	0.0	0	156	5.0
		miles1500	0.66	187	1	566.69	0.0	0	1967	36.5
		r125.1c	3.4	388	1	3241.58	0.0	0	515	35.0
		3-FullIns_3	0.24	69	1	199.69	0.0	0	2	3.5
		4-FullIns_3	0.3	48	1	290.19	0.0	0	2	4.0
		5-FullIns_3	0.3	48	1	273.79	0.0	3	2	4.5
	HD-color	david	3472.41	45329	831	1.3094185e6	119.26	165	190	5.67
		miles1000	27.11	1347	59	18126.3	0.9	150	669	21.5
		miles500	2142.07	16393	1373	326464.4	117.88	881	261	10.5
		miles750	70.78	2200	261	29655.77	0.55	132	448	16.5
		mulsol.i.1	1.13	173	1	501.34	0.01	36	2073	25.0
		mulsol.i.2	-	41890	840	2.4019553e6	16.5	564	1199	15.77
		r125.1	686.0	22437	150	316726.74	0.72	56	120	3.0
		zeroin.i.1	28.96	1273	1	16990.89	8.55	522	2181	24.5
		zeroin.i.3	-	37498	946	2.0003662e6	16.61	831	959	15.33
		fpsol2.i.1	-	8959	752	1.1405904e6	19.43	579	6995	33.0
	DIMACS	adjnoun.graph	7.84	623	1	7427.54	0.0	0	226	33.11
		chesapeake.graph	0.22	92	1	181.42	0.01	11	26	11.0
		dolphins.graph	1.23	279	1	1145.85	0.0	0	78	18.0
		jazz.graph	-	4256	993	2.86362773e6	7258.75	103	75	47.54
		karate.graph	0.43	53	1	422.01	0.0	6	89	11.75
		lesmis.graph	1.03	252	1	941.14	0.0	0	254	20.5
		polbooks.graph	2.91	598	1	2599.23	0.0	0	20	27.08
	comp_DIMACS	chesapeake_comp	1.64	579	1	1380.87	0.0	0	3	3.73
		dolphins_comp	16.04	2730	23	12158.27	0.01	4	10	3.0
		karate_comp	0.16	55	1	151.21	0.0	0	24	3.0
		lesmis_comp	146.48	8405	201	77771.07	6.43	84	156	6.33
		polbooks_comp	-	98355	587	3.1970079e6	17.23	7	41	4.22
	other	miles250	300.53	12882	159	107809.61	0.64	44	123	4.5
		mulsol.i.3	8980.61	43066	909	1.61343634e6	0.0	0	1208	15.77
		mulsol.i.5	-	43339	850	3.0840438e6	20.72	878	1229	15.77
		zeroin.i.2	708.69	13345	131	305671.26	0.6	186	959	15.33
		1-FullIns_4	-	89321	1	86574.5	0.0	0	2	3.13