RC23324 (W0409-027) September 3, 2004
Mathematics

IBM Research Report

On a Binary-Encoded ILP Coloring Formulation

Jon Lee
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

François Margot
Carnegie Mellon University
Pittsburgh, PA 15213-3890

On a Binary-Encoded ILP Coloring Formulation

Jon Lee[1] and François Margot[2]

[1] IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA
jonlee@us.ibm.com
[2] Carnegie Mellon University, Pittsburgh, PA 15213-3890, USA
fmargot@andrew.cmu.edu

Abstract. We further develop the 0/1 ILP formulation of Lee for edge coloring where colors are encoded in binary. With respect to that formulation, our main contributions are: (i) an efficient separation algorithm for general block inequalities, (ii) an efficient LP-based separation algorithm for stars (i.e., the all-different polytope), (iii) introduction of matching inequalities, (iv) introduction of switched path inequalities and their efficient separation, (v) a complete description for paths, and (vi) promising computational results.

Introduction

Let G be a simple finite graph with vertex set $V(G)$ and edge set $E(G)$ and let $m := |E(G)|$. For $v \in V(G)$, let $\delta(v) := \{e \in E(G) \ : \ e \text{ is incident to } v\}$. Let $\Delta(G) := \max\{|\delta(v)| \ : \ v \in V(G)\}$. Let c be a positive integer, and let $C := \{0, 1, \ldots, c-1\}$.

A *proper edge c-coloring* of G is a function Φ from $E(G)$ to C, so that Φ restricted to $\delta(v)$ is an injection, for all $v \in V(G)$. Certainly, a proper edge c-coloring can not exist if $c < \Delta(G)$. Vizing [11] proved that a proper edge c-coloring always exists when $c > \Delta(G)$. Holyer [5] proved that it is NP-Complete to decide whether G has a proper edge $\Delta(G)$-coloring (even when $\Delta(G) = 3$).

Lee [6] developed a 0/1 integer linear programming formulation of the feasibility problem of determining whether G has a proper edge c-coloring based on the following variables: For each edge $e \in E(G)$, we use a string of n 0/1-variables to encode the color of that edge (i.e., the n-string is interpreted as the binary encoding of an element of C). Henceforth, we make no distinction between a color (i.e, an element of C) and its binary representation in $\{0, 1\}^N$.

Let $N := \{0, \ldots n-1\}$. For $X \in \mathbb{R}^{E(G) \times N}$, we let x_e denote the row of X indexed by e and x_e^j denote the entry of X in row e and column j ($e \in E(G)$, $j \in N$). We define the *n-bit edge coloring polytope* of G as

$$Q_n(G) := \operatorname{conv}\Big\{X \in \{0,1\}^{E(G) \times N} : x_e \neq x_f, \ \forall \text{ distinct } e, f \in \delta(v), \ \forall v \in V(G)\Big\}.$$

The graph G is a *star* if there is a $v \in V(G)$ such that $E(G) = \delta(v)$. If G is a star, then we call $Q(m, n) := Q_n(G)$ the *all-different polytope* (as defined in [6]). In this case, we let $M := E(G)$. For a general graph G, the all-different polytope is the fundamental "local" modeling object for encoding the constraint that Φ restricted to $\delta(v)$ is an injection, for all $v \in V(G)$. Note that this type of constraint is present in several combinatorial problems besides edge coloring: vertex coloring, timetabling, and some scheduling problems for example. Although the focus of this paper is on edge coloring, the results of Sections 1 and 2 are relevant in all such situations.

For determining whether G has a proper edge c-coloring, we choose $n := \lceil \log_2 c \rceil$, so we are using only $\sim m \log c$ variables, while the more obvious assignment-based formulation would require mc variables. A rudimentary method for allowing only c of the possible 2^n colors encoded by n bits is given in [6]; A much more sophisticated method for addressing the case where the number of colors c is not a power of two (i.e., $2^{n-1} < c < 2^n$) can be found in [3].

One difficulty with this binary-encoded model is to effectively express the all-different constraint at each vertex — that is, to give a computationally-effective description of the all-different polytope by linear inequalities. In Sections 1 and 2, we describe progress in this direction. In Sections 3 and 4, we describe progress for general graphs (i.e., not just stars). In Section 5, we describe our implementation and results of computational experiments.

We note that a preliminary extended-abstract version of this work appeared as [8].

In the remainder of this section, we set some notation and make some basic definitions. For $x_e \in \mathbb{R}^N$ with $\mathbf{0} \leq x_e \leq \mathbf{1}$, and $S, S' \subseteq N$ with $S \cap S' = \emptyset$, we define the *value of* (S, S') *on* e as

$$\mathbf{x_e}(S, S') := \sum_{j \in S} x_e^j + \sum_{j \in S'} (1 - x_e^j).$$

When $S \cup S' = N$, the ordered pair (S, S') is a partition of N. A *t-light partition* for $x_e \in \mathbb{R}^N$ is a partition (S, S') with $\mathbf{x_e}(S, S') < t$. An *active partition* for $e \in E(G)$ is a 1-light partition. For $E' \subseteq E(G)$, we define the *value of* (S, S') *on* E' as

$$\mathbf{x_{E'}}(S, S') := \sum_{e \in E'} \mathbf{x_e}(S, S').$$

1 Separation for General Block Inequalities

For $1 \leq p \leq 2^n$, p can be written uniquely as

$$p = h + \sum_{k=0}^{t} \binom{n}{k}, \text{ with } 0 \leq h < \binom{n}{t+1}.$$

The number t (resp., h) is the *n-binomial size* (resp., *remainder*) of p. Then let

$$\kappa(p, n) := (t+1)h + \sum_{k=0}^{t} k \binom{n}{k}.$$

Let S, S' be a partition of N and let L be a subset of M. Then $X \in Q(m,n)$, must satisfy the *general block inequalities* (see [6]):

$$\text{(GBI)} \qquad \kappa(|L|, n) \leq \mathbf{x_L}(S, S').$$

In fact, general block inequalities are facet-describing for the all-different polytope when the n-binomial remainder of $|L|$ is not zero [6].

Lemma 1. *Let p satisfy $1 \leq p \leq 2^n$, and let t be the n-binomial size of p. Then for $x_e \in \mathbb{R}^N$ with $\mathbf{0} \leq x_e \leq \mathbf{1}$, at most $4(n+1)^2p^2 + (n+1)p$ partitions are $(t+1)$-light for x_e.*

Proof. Let (S_1, S_1') and (S_2, S_2') be two $(t+1)$-light partitions with $d := |S_1 \Delta S_2|$ maximum. Without loss of generality, we can assume that $S_1 = \emptyset$ by replacing x_e^j by $1 - x_e^j$ for all $j \in S_1$. As

$$2(t+1) > \mathbf{x_e}(S_1, S_1') + \mathbf{x_e}(S_2, S_2') \geq d,$$

we have that $d \leq 2t+1$. The number T of possible $(t+1)$-light partitions satisfies

$$T \leq \sum_{k=0}^{2t+1} \binom{n}{k}.$$

Using that, for all $k \leq n/2$, $\binom{n}{2k} \leq \binom{n}{k}^2$ and $\binom{n}{2k-1} \leq \binom{n}{k}^2$ and that, for nonnegative numbers a, b, we have $a^2 + b^2 \leq (a+b)^2$, we get

$$T \leq \left(2 \sum_{k=0}^{t+1} \binom{n}{k} \right)^2 + \sum_{k=0}^{t+1} \binom{n}{k}$$

By hypothesis, we have $\binom{n}{t+1} \leq n\binom{n}{t} \leq np$, and thus

$$\sum_{k=0}^{t+1} \binom{n}{k} \leq (p-h) + np \leq (n+1)p \ .$$

The result follows. □

Note that computing all of the $(t+1)$-light partitions for x_e in the situation of Lemma 1 can be done in time polynomial in p and n using Reverse Search [1]: The number of partitions in the output is polynomial in p and n, and Reverse Search requires a number of operations polynomial in p and n for each partition in the output.

We are led then to the following

Separation Algorithm for GBI

(0) Let $X \in [0,1]^{M \times N}$, and let t be the n-binomial size of m.
(1) For each $e \in M$, compute the set T_e of all $(t+1)$-light partitions for x_e.

(2) Then, for each partition (S, S') in $\cup_{e \in M} T_e$:

(2.a) Compute $F \subseteq M$ such that, for each $e \in F$, (S, S') is a $(t+1)$-light partition for x_e.

(2.b) Order $F = \{e_1, \ldots, e_f\}$ such that $\mathbf{x_{e_i}}(S, S') \leq \mathbf{x_{e_{i+1}}}(S, S')$ for $i = 1, \ldots, f-1$.

(2.c) If one of the partial sums $\sum_{i=1}^{k} \mathbf{x_{e_i}}(S, S')$, for $k = 2, \ldots, f$ is smaller than $\kappa(k, n)$, then $L := \{e_1, \ldots, e_k\}$ and (S, S') generate a violated GBI for X.

By Lemma 1, it is easy to see that the algorithm is polynomial in p and n. We note that a much simpler algorithm can be implemented with complexity polynomial in p and 2^n : Replace $\cup_{e \in E'} T_e$ by the set of all 2^n possible partitions.

Theorem 1. *Let $X \in [0,1]^{M \times N}$. If the algorithm fails to return a violated GBI for X, then none exists.*

Proof. Suppose that $L \subseteq M$ generates a violated GBI for partition (S, S'). Let s be the n-binomial size of $|L|$. Observe that $\kappa(|L|, n) - \kappa(|L|-1, n) \leq s+1$. We may assume that no proper subset of L generates a GBI for partition (S, S'). Then $\mathbf{x_e}(S, S') < s+1$ for all $e \in L$. As $s \leq t$, this implies that (S, S') is a $(t+1)$-light partition for x_e, and the algorithm will find a violated GBI. □

We can sharpen Lemma 1 for the case of $t = 0$ to obtain the following result.

Lemma 2. *Let $e \in M$ and $x_e \in \mathbb{R}^N$ with $\mathbf{0} \leq x_e \leq \mathbf{1}$. Then there are at most two active partitions for e. Moreover, if two active partitions, say (S_1, S_1') and (S_2, S_2') exist, then $|S_1 \Delta S_2| = 1$.*

Proof. $2 > \mathbf{x_e}(S_1, S_1') + \mathbf{x_e}(S_2, S_2') \geq |S_1 \Delta S_2|$ implies that $|S_1 \Delta S_2| = 1$. Moreover, we can not have more that two subsets of N so that the symmetric difference of each pair contains just one element. □

Motivated by Lemma 2, we devised the following heuristic as a simple alternative to the exact algorithm.

Separation Heuristic for GBI

(0) With respect to x_e, compute its (at most) two active partitions and their values.

(1) Then, for each partition (S, S') of N:

(1.a) Compute the set T of elements in M that have (S, S') as an active partition.

(1.b) Sort the $e \in T$ according to nondecreasing $\mathbf{x_e}(S, S')$, yielding the ordering $T = \{e_1, \ldots, e_t\}$.

(1.c) If one of the partial sums $\sum_{i=1}^{k} \mathbf{x_{e_i}}(S, S')$, for $k = 2, \ldots, t$, is smaller than $\kappa(k, n)$, then $L := \{e_1, \ldots, e_k\}$ and (S, S') generate a violated GBI for X.

The complexity is $O(2^n m \log m)$. Note that the heuristic is an exact algorithm if the n-binomial size of m is zero.

2 Exact LP-Based Separation

In this section we describe an exact separation algorithm for the all-different polytope $Q(m, n)$. The algorithm is polynomial in m and 2^n. In many situations (e.g., edge coloring), we consider 2^n to be polynomial in the problem parameters (e.g., $\Delta(G)$); so the algorithm that we describe may be considered to be efficient in such situations.

We call an inequality $\langle \Pi, X\rangle = \sum_i \sum_j \pi_i^j x_i^j \leq \sigma$ *normalized* if $-\mathbf{1} \leq \Pi \leq \mathbf{1}$. Clearly, if a valid inequality separating $\bar{X}$ from $Q(m, n)$ exists, then a normalized inequality of this type exists as well. The following theorem shows how to find a most violated normalized inequality separating $\bar{X}$ from $Q(m, n)$.

Theorem 2. *Let $\bar{X}$ be a point in $[0, 1]^{M\times N}$. There is an efficient algorithm that checks whether $\bar{X}$ is in $Q(m, n)$, and if not, determines a hyperplane separating $\bar{X}$ from $Q(m, n)$.*

Proof. Consider first the problem of maximizing a linear function Π over $Q(m, n)$. It can be formulated as a maximum weight matching problem in a bipartite graph, with vertices on one side of the bipartition corresponding to the 2^n colors and vertices on the other side corresponding to the m rows of the matrix, with the additional constraint that the vertices corresponding to the rows must be all covered by the matching. If row i is assigned color k, then the contribution to the value of the solution is

$$\sum_{j\in N} \pi_i^j \cdot \mathrm{bit}_j[k] \ ,$$

where $\mathrm{bit}_j[k]$ denotes bit j of the binary representation of k. Hence, optimizing over $Q(m, n)$ may be expressed as the following linear program P:

$$\begin{aligned}
\max \ & \sum_{i\in M}\sum_{j\in N}\sum_{k\in\mathcal{C}} \left(\pi_i^j \cdot \mathrm{bit}_j[k]\ z_{ik}\right) \\
\text{s.t.} \ & \sum_{k\in\mathcal{C}} z_{ik} = 1, \quad \forall\ i \in M; \\
& \sum_{i\in M} z_{ik} \leq 1, \quad \forall\ k \in \mathcal{C}; \\
& z_{ik} \geq 0, \quad \forall\ i \in M,\ \forall\ k \in \mathcal{C},
\end{aligned}$$

where the binary variable z_{ik} indicates the assignment of color k to row i.

The dual of P is D:

$$\begin{aligned}
\min \ & \sum_{i\in M} \alpha_i + \sum_{k\in\mathcal{C}} \beta_k \\
\text{s.t.} \ & \alpha_i + \beta_k \geq \sum_{j\in N} \pi_i^j \cdot \mathrm{bit}_j[k], \quad \forall\ i \in M,\ \forall\ k \in \mathcal{C}; \\
& \beta_k \geq 0, \quad \forall\ k \in \mathcal{C}.
\end{aligned}$$

Consider now the separation problem for $\bar{X}$. We claim that it can be solved using the following LP with variables $\Pi \in \mathbb{R}^{M \times N},\ \sigma \in \mathbb{R},\ \alpha \in \mathbb{R}^M,\ \beta \in \mathbb{R}^{\mathcal{C}}$:

$$\begin{aligned} \max\ & \sum_{i \in M} \sum_{j \in N} \pi_i^j \bar{x}_i^j - \sigma \\ \text{s.t.}\ & -\mathbf{1} \le \Pi \le \mathbf{1}; \\ & \sum_{i \in M} \alpha_i + \sum_{k \in \mathcal{C}} \beta_k \le \sigma; \\ & \alpha_i + \beta_k \ge \sum_{j \in N} \pi_i^j \cdot \mathrm{bit}_j[k],\ \forall\, i \in M, \quad \forall\, k \in \mathcal{C}; \\ & \beta_k \ge 0, \quad \forall\, k \in \mathcal{C}. \end{aligned}$$

Indeed, let $(\Pi, \sigma, \alpha, \beta)$ be an optimal solution of this LP. Note that it has a positive value if and only if $\langle \Pi, \bar{X} \rangle > \sigma$. Moreover, (α, β) is a feasible solution of D with value at most σ if and only if P has an optimal value at most σ if and only if the halfspace $\langle \Pi, \bar{X} \rangle \le \sigma$ contains $Q(m, n)$.

It follows that this last inequality separates $\bar{X}$ from $Q(m, n)$ if and only if $(\Pi, \sigma, \alpha, \beta)$ is a feasible solution with positive value of the LP. □

This approach yields a practical and efficient algorithm for producing maximally violated normalized cuts if any such cut exists. In Section 5, we refer to cuts produced in this way as *LP cuts* (LPC). Note that in [8] we also proved Theorem 2 by constructing an efficient algorithm, but that algorithm is not practical for computation.

3 Matching Inequalities

Let S, S' be subsets of N with $S \cap S' = \emptyset$. The *optimal colors for* (S, S') are the colors $\mathbf{x} \in \{0,1\}^N$ that yield $\mathbf{x}(S, S') = 0$. The set of optimal colors for (S, S') is denoted by $\mathcal{B}(S, S')$. Note that if (S, S') is a partition of N, then there is a unique optimal color which is the characteristic vector of S'. In general, if $|N \setminus (S \cup S')| = k$, then the set of optimal colors for (S, S') has 2^k elements (it is the set of vertices of a k-dimensional face of $[0,1]^N$). Note that if $\mathbf{x} \in \{0,1\}^N$ is a not an optimal color for (S, S'), then $\mathbf{x}(S, S') \ge 1$.

Proposition 1. *Let $E' \subseteq E(G)$, and let $F \subseteq E'$ be a maximum matching in the graph induced by E'. Let (S, S') be a partition of N. The* matching inequality (induced by E')

$$\text{(MI)} \qquad \mathbf{x}_{\mathbf{E}'}(S, S') \ge |E' \setminus F|$$

is valid for $Q_n(G)$.

Proof. At most $|F|$ edges in E' can have the optimal color for (S, S'), and every other edge has a color contributing at least one to the left-hand side. □

When E' is an odd cycle, the matching inequalities reduce to the so-called "type-I odd-cycle inequalities" (see [6] which introduced these latter inequalities and [7] which provided an efficient separation algorithm for them).

A MI is *dominated* if it is implied by MI on 2-connected non-bipartite subgraphs and by GBI. The following proposition shows that it is enough to generate the non-dominated MI, provided that the GBI generated by the separation heuristic for GBI of Section 1 are all satisfied.

Proposition 2. *Let G' be the graph induced by E'. The MI induced by E' is dominated in the following cases:*

(i) G' is not connected;
(ii) G' has a vertex v saturated by every maximum matching in G';
(iii) G' has a cut vertex v;
(iv) G' is bipartite.

Proof. (i) The MI is implied by those induced by the components of G'.

(ii) The MI is implied by the MI on $G' - v$ and the GBI for $\delta(v) \cap E'$.

(iii) Let G_1 and G_2 be a partition of E' sharing only vertex v. By (ii), we can assume that there exists a maximum matching F of G with v not saturated by F. Then $E(F) \cap E(G_i)$ is a maximum matching in G_i for $i = 1, 2$. The MI is thus implied by the MI on G_1 and G_2.

(iv) By the König's theorem, the cardinality of a minimum vertex cover of G' is equal to the cardinality k of a maximum matching F of G'. It is then possible to partition the edges of G' into k stars, such that star i has k_i edges. If the GBI inequalities for the stars are all satisfied, then summing them up yields:

$$\mathbf{x_{E(G')}}(S, S') \geq \sum_{i=1}^{k} (k_i - 1) = |E(G')| - k = |E(G') \setminus M|,$$

and the MI induced by E' is also satisfied. □

Recall that a *block* of a graph is a maximal 2-connected subgraph. Proposition 2 is the justification of the following:

Separation Heuristic for MI

(0) Let $\bar{X}$ be a point in $[0, 1]^{E(G) \times N}$.
(1) For each partition (S, S') of N:
- (1.a) Compute the edges T for which (S, S') is an active partition.
- (1.b) For each non-bipartite block of the graph G' induced by T:
  - (1.b.i) Compute a maximum matching $F(G')$ in G'.
  - (1.b.ii) Check if $\mathbf{x_{E(G')}}(S, S') \geq |E(G') \setminus F(G')|$ is a violated matching inequality.

Complexity: Since each edge of G has at most two active partitions, all computations of active partitions take $O(nm)$ and all computations of non-bipartite blocks take $O(m)$. For one partition (S, S'), computing the maximum matchings takes $O(\sqrt{|V(G)|}\ m)$ [10]. The overall complexity is thus $O(2^n \sqrt{|V(G)|}\ m)$.

Note that ignoring edges e for which (S, S') is not an active partition does not prevent generation of violated matching inequalities: Suppose that e appears in a violated matching inequality $\bar{\mathbf{x}}_{\mathbf{E(G')}}(S, S') < |E(G') \setminus F(G')|$. Then $\bar{\mathbf{x}}_{\mathbf{E(G')-e}}(S, S') < |(E(G') - e) \setminus F(G' - e)|$ is also violated, as the left-hand side has been reduced by more than 1, while the right-hand side has been reduced by at most 1. The algorithm is nevertheless not exact, as we should generate MI for all 2-connected subgraphs, not only for blocks. In practice, the blocks are very sparse and rarely contain more than a few odd cycles. Enumerating the 2-connected non-bipartite subgraphs might thus be feasible.

4 Switched Walk Inequalities

Let $S, S' \subseteq N$ such that $S \cap S' = \emptyset$ and $|S \cup S'| \geq n - 1$. Then (S, S') is a *subpartition* of N.

Let (S_1, S_1') be a subpartition of N. Let (S_2, S_2') be a subpartition obtained from (S_1, S_1') by performing the following two steps:

(1) adding the only element not in $S_1 \cup S_1'$ (if any) either to S_1 or to S_1'; call the resulting partition (P_2, P_2').
(2) removing at most one element from P_2 or at most one element from P_2'.

Then (S_2, S_2') is a *switch* of (S_1, S_1'). Observe that $|\mathcal{B}(S_1, S_1')| \leq 2$, that $|\mathcal{B}(S_2, S_2')| \leq 2$ and that $|\mathcal{B}(S_1, S_1') \cap \mathcal{B}(S_2, S_2')| \geq 1$.

Let $(f_1, \ldots, f_r)$ be the ordered set of edges of a walk in G with $r \geq 2$. For $i = 1, \ldots, r$, let (S_i, S_i') be subpartitions of N such that

(a) $|S_i \cup S_i'| = \begin{cases} n, & \text{if } i = 1, \text{ or } i = r; \\ n - 1, & \text{otherwise.} \end{cases}$
(b) For $i = 1, \ldots, r - 1$, (S_{i+1}, S_{i+1}') is a switch of (S_i, S_i').
(c) For all $j \in S_t$ (resp., $j \in S_t'$), if t' is maximum such that for all $t + 1 \leq i \leq t'$ we have $N - (S_i \cup S_i') = \{j\}$, then $j \in S_{t'+1}$ (resp., $j \in S_{t'+1}'$) if and only if $t' - t$ is even.

Then the walk and the set of subpartitions $(S_1, S_1'), \ldots, (S_r, S_r')$ form a *switched walk*.

Given a switched walk, the inequality

$$\text{(SWI)} \qquad \sum_{i=1}^{r} \mathbf{x}_{\mathbf{f_i}}(S_i, S_i') \geq 1$$

is a *switched walk inequality*.

Example 1. Let $N := \{0, 1, 2\}$. Consider the path of edges $(f_1, f_2, f_3, f_4, f_5)$. Associated with the sequence of edges of the path is the switched walk: $(\{0\}, \{1, 2\})$, $(\{0\}, \{2\})$, $(\{1\}, \{2\})$, $(\{1\}, \{0\})$, $(\{1, 2\}, \{0\})$. The given switched walk gives rise to the SWI:

$$\begin{array}{llll} +x_1^0 & +(1-x_1^1) & +(1-x_1^2) & \\ +x_2^0 & & +(1-x_2^2) & \\ & +x_3^1 & +(1-x_3^2) & \\ +(1-x_4^0) & +x_4^1 & & \\ +(1-x_5^0) & +x_5^1 & +x_5^2 & \geq 1 \ . \end{array}$$

The only possibility for a 0/1 solution to violate this is to have each edge colored with one of its optimal colors. This implies that the color of f_1 must be 011. Then, of the two optimal colors for f_2, the only one that is different from the color of f_1 is 001. Similarly, f_3 must get color 101 and f_4 gets 100. But this is not different from the only optimal color for f_5.

Next, we state a result indicating the importance of the switched walk inequalities.

Theorem 3. *If P is a path and $n \geq 2$, then $Q_n(P)$ is described by the SWI and the simple bound inequalities $\mathbf{0} \leq X \leq \mathbf{1}$.*

Theorem 3 was stated without proof in [8]. The proof, which we present here, uses the following five lemmas.

Lemma 3. *If a 0,1 polytope Q in $\mathbb{R}^q$ is full dimensional and $\langle \gamma, x \rangle \geq \beta$ describes one of its facets, then, for each $i = 1, \ldots, q$, there exists a 0,1 point $\bar{x} \in Q$ with $\bar{x}_i = 1$ (resp., $\bar{x}_i = 0$) satisfying $\langle \gamma, \bar{x} \rangle = \beta$.*

Proof. If this is not the case, then all points in Q satisfy $x_i = 0$ (resp., $x_i = 1$), a contradiction with Q full dimensional. □

Lemma 4. *If a polytope Q in $\mathbb{R}^q$ is full dimensional and $\langle \gamma, x \rangle \geq \beta$ describes one of its facets F, then the orthogonal projection of F onto any subset S of the variables has dimension $|S|$ or $|S| - 1$.*

Proof. If this is not the case, then all points in F satisfy at least two linearly independent inequalities. One of these inequalities is not a positive multiple of $\langle \gamma, x \rangle \geq \beta$, a contradiction. □

Lemma 5. *If $n \geq 2$, then $Q_n(P)$ is full dimensional.*

Proof. Let $f_1, \ldots, f_r$ be the ordered edges of path P. Set the color of f_i to $\mathbf{0} \in \mathbb{R}^N$ for all even i and to color $\mathbf{1} \in \mathbb{R}^N$ for all odd i. Flipping any single bit of this valid coloring gives another valid coloring, yielding $1 + n|E(P)|$ affinely-independent valid colorings of P. □

For a matrix Φ, define Φ_- as the sum of its negative entries.

Lemma 6. *Let $f_1, \ldots, f_r$ be the ordered edges of path P. Let ϕ_i be the vector of coefficients associated with f_i in a facet-describing inequality $\langle \Phi, X \rangle \geq \beta$. Suppose that ϕ_i has at least one zero, for all $i = 2, \ldots, r$. Then $\beta = \Phi_-$, and each edge receives one of its optimal colors in any coloring $\bar{X}$ for which $\langle \Phi, \bar{X} \rangle = \beta$.*

Proof. Each edge, except possibly f_1 has at least two optimal colors. Hence, starting by coloring f_1 with one of its optimal colors, there exists a valid coloring such that each edge is colored with one of its optimal colors. □

Lemma 7. *Let $n \geq 2$, and let $\langle \Phi, X \rangle \geq \beta$ be a facet-describing inequality for $Q_n(P)$. Assume that $\langle \Phi, X \rangle \geq \beta$ is not a positive multiple of a simple-bound inequality $x_i^j \geq 0$ or $-x_i^j \geq -1$. Let e_1 (resp., e_k) be the first (resp., last) edge of P for which ϕ_i is not the zero vector. If $\min\{|\phi_i^j| \mid \phi_i^j \neq 0\} = 1$, then all nonzero components of Φ are ± 1, ϕ_1 and ϕ_k each have no 0, and ϕ_i has exactly one 0 for all $i = 2, \ldots, k-1$. Moreover, any two consecutive edges e_i and e_{i+1} share at least one optimal color, and $\beta = 1 + \Phi_-$.*

Proof. Note that $k = 1$ is impossible, as there exists a valid coloring of P with edge e_1 receiving an arbitrary color. If $k = 2$, all colorings with e_1 and e_2 receiving distinct colors satisfy the inequality. Then the inequality must be a GBI for the pair e_1, e_2, as the GBI give the convex hull of such colorings (see [6]). The result thus holds. Otherwise, let $e_t \in \{e_2, \ldots, e_{k-1}\}$, $P_1 = \{e_1, \ldots, e_{t-1}\}$, and $P_2 = \{e_{t+1}, \ldots, e_k\}$. We call e_{t-1} (resp., e_{t+1}) the *shore* of P_1 (resp., P_2).

In this proof, the value of a coloring of any subset S of edges is always computed with respect to the cost function obtained as the restriction of Φ to S. Also, $\bar{X}$ will always be an integral matrix in $Q_n(P)$ satisfying $\langle \Phi, \bar{X} \rangle = \beta$.

For $i = 1, 2$, let a_i be the optimal value of a coloring of P_i, and let b_i be the second best value of such a coloring (with $a_i < b_i$). Let A_i (resp., B_i) be the set of colors for the shore of P_i in all $\bar{X}$ achieving value a_i (resp., b_i). For edge e_t, let a, b and c be the three best values for a coloring, with $a < b < c$ and with corresponding color sets A, B and C.

As Φ induces a facet of $Q_n(P)$, there exists a coloring $\bar{X}$ that does not induce an optimal coloring of P_1. Thus every optimal colorings of P_1 give to its shore the color that e_t has in $\bar{X}$. A similar remark holds for P_2. It follows that $|A_i| = 1$ for $i = 1, 2$ and that the color of the shore in any $\bar{X}$ is in $A_i \cup B_i$ for $i = 1, 2$. Taking $t = 2$ (resp., $t = k - 1$), this implies that ϕ_1 (resp., ϕ_k) has no 0. Lemma 3 shows that all entries in ϕ_1 (resp., ϕ_k) must have the same absolute value.

Similarly, for some $\bar{X}$, the color of e_t in $\bar{X}$ does not have value a. This implies that $|A| < 3$ and thus that ϕ_t has at most one 0. Also, we have that the color of e_t in $\bar{X}$ is in $A \cup B$ if $|A| + |B| \geq 3$ and in $A \cup B \cup C$ if $|A| = |B| = 1$.

We say that $\bar{X}$ *induces a pattern* (H_1, H, H_2) *on* (e_{t-1}, e_t, e_{t+1}) if the color of e_{t-1} (resp., e_t, e_{t+1}) in $\bar{X}$ is in H_1 (resp., H, H_2). Lemmas 4 and 5 imply that the projection on (e_{t-1}, e_t, e_{t+1}) of all the points $\bar{X}$ should span an affine space of dimension at least $3n - 1$, i.e. there should be at least $3n$ affinely independent such projections.

Case I: $|A| = 1$. This implies that ϕ_t has no 0 entries. It follows that $|B| \leq n$ as any color obtained from A by flipping more than one entry has a value worse than any color obtained by flipping a single entry in A. Moreover, the same reasoning implies that if $|B| = 1$ then $|C| \leq n - 1$.

Case Ia: $A_1 \neq A$ and $A_2 \neq A$. Then any $\bar{X}$ induces on (e_{t-1}, e_t, e_{t+1}) the pattern (A_1, A, A_2), a contradiction with the fact that there should be $3n$ affinely independent such projections.

Case Ib: $A_2 = A$ (the case $A_1 = A$ is symmetrical).

Case Ib1: $A_1 = A$. Then any $\bar{X}$ induces on (e_{t-1}, e_t, e_{t+1}) one of the pattern (B_1, A, B_2), (A_1, B, A_2), (A_1, C, A_2). Since any solution with the last pattern has a value strictly worse than a solution with the second pattern, only the first two patterns may occur. Moreover, we have $b - a = (b_1 - a_1) + (b_2 - a_2)$. let $\gamma = \frac{b_1 - a_1}{b - a}$. Observe that each $\bar{X}$ satisfies the inequality obtained on $P_1 \cup e_t$ using the restriction of Φ to P_1 and using $\gamma \cdot \phi_t$ for e_t with right hand side $\beta - b_2 - (1 - \gamma)a = \beta - a_2 - (1 - \gamma)b$ with equality, a contradiction.

Case Ib2: $A_1 \subseteq C$. Then any $\bar{X}$ induces on (e_{t-1}, e_t, e_{t+1}) one of the patterns (A_1, A, B_2), (A_1, B, A_2) and (B_1, C, A_2). Note that solutions inducing the third pattern are worse than solution with the second pattern, implying that no $\bar{X}$ induce the third pattern. Then all $\bar{X}$ optimally color P_1, a contradiction.

Case Ib3: $|B| = 1$ and $A_1 = B$. Then any $\bar{X}$ induces on (e_{t-1}, e_t, e_{t+1}) one of the patterns (A_1, A, B_2), (B_1, B, A_2) and (A_1, C, A_2). Note that at most n points with the first (resp., second) pattern may be affinely independent, and at most $n - 1$ points with the third pattern may be affinely independent. Thus, at most $3n - 1$ of the points are affinely independent, a contradiction.

Case Ib4: $|B| > 1$ and $A_1 \subseteq B$. Then any optimal $\bar{X}$ induces on (e_{t-1}, e_t, e_{t+1}) one of the patterns (A_1, A, B_2), (B_1, A_1, A_2) and $(A_1, B - A_1, A_2)$. Note that solutions with the second pattern are worse than solution with the third pattern, implying that no $\bar{X}$ induces the second pattern. Then all $\bar{X}$ optimally color P_1, a contradiction.

Case II: $|A| = 2 = \{U, V\}$. Then ϕ_t has exactly one 0, and only A and B may appear in the projection of $\bar{X}$ on e_t. Lemma 3 implies that $|B| = 2(n - 1)$.

Case IIa : $A_1 \cap A = \emptyset$ (or, symmetrically, $A_2 \cap A = \emptyset$). Then any $\bar{X}$ induces on (e_{t-1}, e_t, e_{t+1}) one of the patterns $(A_1, U, A_2$ or $B_2)$, $(A_1, V, A_2$ or $B_2)$, $(A_1, B - A_1, A_2$ or $B_2)$ and $(B_1, A_1, A_2$ or $B_2)$. One of the first two patterns occurs with A_2 on e_{t+1} and it is better than the last two, yielding a contradiction, as P_1 is always optimally colored.

Case IIb: $A_1 = A_2 = U$. Then any $\bar{X}$ induces on (e_{t-1}, e_t, e_{t+1}) one of the patterns (B_1, U, B_2), (A_1, V, A_2) and (A_1, B, A_2). But the second pattern is strictly better than the other two patterns. All the projections inducing the second pattern generate an affine space of dimension 0, a contradiction.

Case IIc: $A_1 = U, A_2 = V$. Then any $\bar{X}$ induces on (e_{t-1}, e_t, e_{t+1}) one of the pattern (B_1, U, A_2), (A_1, V, B_2) and (A_1, B, A_2), each contributing for at most n affinely independent projections. The first two patterns show that $b_1 - a_1 = b_2 - a_2$ and the last two show that $b_2 - a_2 = b - a$. Lemma 3 shows that all nonzero entries in ϕ_t must have the same absolute value.

Over all the above cases, only Case IIc may occur, so it holds for all t. Using induction on t, we can then show that all entries in Φ must have the same absolute value (± 1 without loss of generality) using the fact that $b_1 - a_1 = b - a$. Lemma 6 and the pattern (A_1, B, A_2) yields $\beta = \Phi_- + (b - a) = \Phi_- + 1$. □

Proof (Theorem 3). The conditions spelled out for Φ and β in Case IIc of Lemma 7 force the inequality to be a SWI. This is clear for conditions (a) and (b) of the definition of a SWI. To see that the inequality satisfies (c), let P_q be the path consisting of $e_1, \ldots, e_q$, for $q = t, \ldots, t' + 1$ with t' maximum with $N - (S_i \cup S'_i) = \{j\}$ for all $i = t+1, \ldots, t'$. Let U and V be the two optimal colors for e_{t+1}. By Case IIc of Lemma 7, all optimal colorings of P_t have e_t with color U or V, say U. (Colors U and V only differ in bit j.) Then, for $s = 1, \ldots, t' - t$, all optimal colorings of P_{t+s} have e_{t+s} with color V if s is odd and U if s is even. Hence the color of $e_{t'}$ in an optimal coloring of $P_{t'}$ must have color U if $t' - t$ is even and color V otherwise. Since that color must be a color that is optimal for $e_{t'+1}$, we must have $\phi^j_t = \phi^j_{t'+1}$ if $t' - t$ is even and $\phi^j_t = -\phi^j_{t'+1}$ if $t' - t$ is odd. □

Theorem 4. *If $n \geq 2$, the SWI are valid for $Q_n(G)$.*

Proof. If $k = 2$, the SWI is a GBI and thus is valid. Consider a SWI with $k \geq 3$ and let $t = 2$. Using notation similar to the proof of Lemma 7, Case IIc above shows that a valid coloring of P violating the SWI must optimally color P_1, P_2 and e_t. But this is impossible, as $A_1 \cup A_2 = A$. □

We separate the SWI by solving m shortest path problems on a directed graph G' with nonnegative node weights constructed as follows: A node of G' is identified by:

(a) an edge $e \in G$;
(b) a travel direction on e;
(c) a subpartition (S, S') such that $\mathbf{x_e}(S, S') < 1$;
(d) an indicator *ind* with value S or S' with the meaning that the next time $j = N - (S \cup S')$ is in $S \cup S'$, it must be in the set *ind* of that node.

The weight associated with the node is $\mathbf{x_e}(S, S')$. There is an arc from node $(e_1 = (u_1, v_1), (S_1, S'_1), ind_1)$ to node $(e_2 = (u_2, v_2), (S_2, S'_2), ind_2)$ if and only if the sum of their weights is less than 1, $v_1 = u_2$, (S_2, S'_2) is a switch of (S_1, S'_1) and for $j_1 = N - (S_1 \cup S'_1)$, $j_2 = N - (S_2 \cup S'_2)$, either (I) $j_1 = j_2$ and $ind_1 \neq ind_2$ or (II) $j_1 \neq j_2$, j_1 is in the set ind_1 of the second node, and $ind_2 = S'_2$ if and only if $j_2 \in S_1$.

Observe that the number of nodes in G' is at most $8(n+1)m$: For each edge $e \in G$, there are 2 choices for (b), two choices for (d), $n + 1$ possibilities for the choice of $N - (S_1 \cup S'_1)$ and, by Lemma 2 at most two subpartitions for each of these $(n + 1)$ possibilities. The number of edges is bounded by $8n(n + 1)^2 m$ as the degree of a node in G' is bounded by $n(n + 1)$.

Any directed path (with at least one edge) in G' of weight strictly less than 1 starting and ending at a node of G' whose subpartition is indeed a partition yields a violated SWI. If a violated SWI exists, then one can be found by at most m calls to a shortest-path algorithm. The overall complexity of the separation algorithm is thus $O(mn^3 \log(mn))$.

5 Computational Results

We report computational results for Branch-and-Cut (B&C) algorithms using the GBI, LPC, MI, SWI and Gomory Cuts. The results that we present improve upon the preliminary results first reported in [8]. The code is based on the open-source codes BCP (Branch, Cut & Price) and CLP (an LP solver), which are freely available at `www.coin-or.org`. It was run on a Dell Precision 650 (Intel Xeon processor, 8KB level-1 cache). Test problems consist of

(a) nine 4-regular graphs $g4\_p$ on p nodes, for $p = 20, 30, \ldots, 100$;
(b) three 8-regular graphs $g8\_p$ on p nodes, for $p = 20, 30, \ldots, 40$;
(c) the Petersen graph ($peter$);
(d) two regular graphs on 14 and 18 vertices having overfull subgraphs ($of5\_14\_7$ and $of7\_18\_9\_5$);
(e) an overfull graph wih 9 vertices ($ofsub9$) obtained as a subgraph of $of7\_18\_9\_5$;
(f) graphs from [2] on 18 vertices and 33 edges ($jgt18$) and 30 vertices and 57 edges ($jgt30$).

Graphs in (a) and (b) are randomly generated and can be colored with 4 or 8 colors respectively. It is likely that most heuristics would be able to color them optimally, but our B&C algorithms have no such heuristic, i.e. they will find a feasible solution only if the solution of the LP is integer. The remaining graphs are "Class 2" graphs, i.e. graphs G that can not be colored with $\Delta(G)$ colors.

A subgraph H of a graph G is an *overfull* subgraph if $|V(H)|$ is odd, $\Delta(H) = \Delta(G)$, and $|E(H)| > \Delta(H) \cdot (|V(H)| - 1)/2$. If G has an overfull subgraph, then G is a Class 2 graph. Graphs in (d) were randomly generated and have overfull subgraphs, but are not overfull themselves. The graph in (e) is a small non-regular Class 2 graph.

To illustrate the benefits of and trade-offs between the different types of cuts, we report results of three B&C algorithms. The separation algorithms for the different types of cuts are: the separation heuristic for GBI of Section 1, the exact LPC separation algorithm alluded to at the end of Section 2, the heuristic separation for MI algorithm of Section 3 (except that blocks are not computed), and the separation algorithm for SWI of Section 4. No more than six rounds of cutting is done at each node, each type of cut being considered. The branching is done as follows: At the beginning, the edges of the graph are ordered in Breadth-First Search fashion, starting from a vertex of maximum degree. When a branching decision is made, the algorithm chooses to branch on the first edge for which one of the associated variables is fractional. The children are created by assigning to the chosen edge all (still) feasible colors.

B&C 1 uses GBI, MI and Gomory Cuts. B&C 2 uses, in addition, LPC, and B&C 3 uses all five types of cuts. Table 1 gives the number of nodes in the enumeration tree. As expected, in general, the number of nodes is smaller when more cuts are in use, but for some problems, there is a big drop between variant 1 and 2, i.e. the use of LPC seems to be important. Most of these problems have relatively large degree, which is also expected, as GBI give a good approximation of the all-different polytope when the number of rows in X is small. On the other hand, the use of SWI does not seems to help much on these problems.

Table 2 shows that for problems with low maximum degree, using SWI increases the overall cpu time. This (and Table 3) illustrates the difficulties for separating these inequalities efficiently. Even with the restricted use of one round of SWI cuts at most, the separation algorithm returns a large number of violated SWI cuts. A better understanding of these cuts might help generate "useful" ones more efficiently. The separation times are very small for GBI, MI and Gomory Cuts. The LPC, however take significant time (more than 50% of the total time for the 4-regular graphs, about 25% of the total time for the 8-regular graphs and 15% for of7_18_9). The SWI separation is also time consuming, taking roughly 10-15% of the total time.

Table 1. Number of nodes.

	1	2	3
$g4\_20$	11	13	7
$g4\_30$	25	29	26
$g4\_40$	40	36	39
$g4\_50$	56	44	54
$g4\_60$	190	70	70
$g4\_70$	88	69	82
$g4\_80$	89	86	74
$g4\_90$	122	97	33
$g4\_100$	178	111	108
$g8\_20$	114	109	105
$g8\_30$	216	178	170
$g8\_40$	1511	225	226
$peter$	1	1	1
$of5\_14\_7$	31	30	28
$of7\_18\_9$	734	3535	3851
$ofsub9$	93223	63882	65363
$jgt18$	260	247	230
$jgt30$	42225	37480	37340

Table 2. cpu time in seconds.

	1	2	3
$g4\_20$	0.10	0.20	0.10
$g4\_30$	0.30	0.50	1.00
$g4\_40$	0.40	1.50	1.80
$g4\_50$	0.60	2.00	3.60
$g4\_60$	4.60	3.50	6.10
$g4\_70$	1.80	4.80	9.80
$g4\_80$	2.30	6.80	9.70
$g4\_90$	4.10	12.70	17.30
$g4\_100$	8.80	11.70	63.10
$g8\_20$	2.50	10.90	14.20
$g8\_30$	20.80	45.10	62.20
$g8\_40$	136.80	120.20	166.00
$peter$	0.00	0.00	0.00
$of5\_14\_7$	0.70	1.10	1.30
$of7\_18\_9$	238.90	1365.80	2195.00
$ofsub9$	805.70	1283.20	1695.30
$jgt18$	6.30	9.30	9.60
$jgt30$	649.60	1333.20	1871.00

Table 3. Number of generated cuts.

	1			2				3				
	GBI	MI	GOM	GBI	MI	GOM	LPC	GBI	MI	GOM	LPC	SWI
g4_20	100	4	194	78	4	140	17	66	3	82	10	135
g4_30	168	10	245	160	4	138	50	196	7	302	73	565
g4_40	350	5	305	338	4	302	94	316	6	95	84	1034
g4_50	560	15	288	464	8	223	113	690	14	38	137	2088
g4_60	1706	31	1461	722	13	291	169	908	30	266	177	2800
g4_70	994	21	493	864	21	353	196	1118	15	264	256	4048
g4_80	1116	21	513	952	15	274	223	1110	10	158	196	3260
g4_90	1478	43	788	1318	29	738	232	862	6	106	55	2805
g4_100	2636	51	1508	1120	13	381	298	1612	30	95	250	7364
g8_20	2110	48	252	1668	11	60	1773	1616	30	19	1920	2524
g8_30	3974	51	693	3174	19	92	4306	3246	29	67	4159	4646
g8_40	33170	489	5972	4592	32	152	6849	4712	39	120	7098	8316
peter	0	2	16	0	2	16	8	0	2	16	8	0
of5_14_7	704	42	526	628	36	294	267	564	19	287	275	609
of7_18_9	23406	2142	34536	69224	4788	90295	56235	82034	5571	106189	63893	151031
ofsub9	283192	13771	486194	113321	10071	88597	208805	108951	10036	77333	197433	596291
jgt18	1428	182	9113	1129	171	6397	612	1109	152	5402	572	5040
jgt30	180598	24211	563261	151658	24420	407837	76088	134531	24345	333986	69574	1189512

References

1. David Avis and Komei Fukuda. Reverse search for enumeration. *Discrete Appl. Math.*, 65(1-3):21–46, 1996. First International Colloquium on Graphs and Optimization (GOI), 1992 (Grimentz).
2. Amanda G. Chetwynd and Robin J. Wilson. The rise and fall of the critical graph conjecture. *J. Graph Theory*, 7(2):153–157, 1983.
3. Don Coppersmith and Jon Lee. Parsimonious binary-encoding in integer programming. *IBM Research Report RC23838*, 2003.
4. Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, Volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, second edition, 1993.
5. Ian Holyer. The NP-completeness of edge-coloring. *SIAM J. Comput.*, 10(4):718–720, 1981.
6. Jon Lee. All-different polytopes. *J. Comb. Optim.*, 6(3):335–352, 2002.
7. Jon Lee, Janny Leung, and Sven de Vries. Separating type-I odd-cycle inequalities for a binary encoded edge-coloring formulation. To appear in: *Journal of Combinatorial Optimization*.
8. Jon Lee and François Margot. More on a binary-encoded coloring formulation. *Integer Programming and Combinatorial Optimization (New York, NY, 2004)*. D. Bienstock and G. Nemhauser, Eds. Lecture Note in Computer Science, 3064:271–282, 2004. Springer-Verlag.
9. R. Kipp Martin. Using separation algorithms to generate mixed integer model reformulations. *Oper. Res. Lett.*, 10(3):119–128, 1991.
10. Vijay V. Vazirani. A theory of alternating paths and blossoms for proving correctness of the $O(\sqrt{V}E)$ general graph maximum matching algorithm. *Combinatorica*, 14:71–109, 1994.
11. Vadim G. Vizing. On an estimate of the chromatic class of a p-graph. *Diskret. Analiz No.*, 3:25–30, 1964.