On a Binary-Encoded ILP Coloring Formulation

Jon Lee
Department of Mathematical Sciences, IBM T.J. Watson Research Center,
P.O. Box 218, Yorktown Heights, NY 10598, U.S.A. Email: jonlee@us.ibm.com

François Margot
Tepper School of Business, Carnegie Mellon University,
Pittsburgh, PA 15213-3890, U.S.A. Email: 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 inequal-
ities, (iv) introduction of switched path inequalities and their efficient
separation, (v) a complete description for paths, and (vi) promising com-
putational results.

1. 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 $\delta(v) := \{ e \in E(G) : e$ is incident to $v \}$. Let $\Delta(G) := \max \{|\delta(v)| : v \in V(G)\}$. Let $e$ be a positive integer, and let
$C := \{0, 1, \ldots, e-1\}$.

A proper edge $c$-coloring of $G$ is a function $\Phi$ from $E(G)$ to $C$, so that $\Phi$ 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 (1964) proved that a proper edge $c$-coloring always exists when $c > \Delta(G)$. Holyer (1981) proved that it is NP- Complete to decide whether $G$ has a proper edge $\Delta(G)$-coloring (even when $\Delta(G) = 3$).

Lee (2002) 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 $x^j_e$ 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

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

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 Lee (2002). 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 \( \Phi \) 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 2 and 3 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 Lee (2002); 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 Coppersmith and Lee (2005).

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 2 and 3, we describe progress in this direction by addressing the separation of the General Block inequalities introduced in Lee (2002). We give an exact, LP-based, separation algorithm as well as a faster heuristic algorithm. In Sections 4 and 5, we describe progress for general graphs (i.e., not just stars) by introducing two families of inequalities, Matching inequalities and Switched Walk inequalities and separation algorithms for them. In Section 6, we describe our implementation and results of computational experiments using the separation algorithms described in the paper.

We note that a preliminary extended-abstract version of this work appeared as Lee and Margot (2004).

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

\[
x_e(S, S') := \sum_{j \in S} x_j^e + \sum_{j \in S'} (1 - x_j^e).
\]

This notation gives a compact way to write some of the inequalities that will be introduced in the remainder of the paper. When \( S \cup S' = N \), the ordered pair \((S, S')\) is a partition of \( N \). Separation algorithms for the inequalities of interest typically look for a partition \((S, S')\) minimizing the sum of the value of \((S, S')\) on a subset of edges. We thus generalize the above notation to subset of edges and give a name to edges \( e \) for which the value of \((S, S')\) on \( e \) is small: For \( E' \subseteq E(G) \), we define the value of \((S, S')\) on \( E' \) as

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

A \( t \)-light partition for \( x_e \in \mathbb{R}^N \) is a partition \((S, S')\) with \( x_e(S, S') < t \). An active partition for \( e \in E(G) \) is a 1-light partition.
2. 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 Lee (2002)):

$$(GBI) \quad \kappa(|L|, n) \leq x_L(S, S').$$

Moreover, any integer $X \in \mathbb{R}^{E(G) \times N}$ satisfying all GBIs with $|L| = 2$ corresponds to a valid coloring (Lee, 2002). General block inequalities are facet-describing for the all-different polytope when the $n$-binomial remainder of $|L|$ is not zero (Lee, 2002).

We will show that separating GBIs amounts to consider all partitions $(S, S')$ that are $(t+1)$-light for some $e \in M$, where $t$ is the $n$-binomial size of $m = |M|$. The crucial observation is that the total number of such partitions is polynomial in $m$ and $n$, as the following Lemma shows.

**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 $0 \leq x_e \leq 1$, at most $2(n+1)^2p^2$ 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

$$2(t + 1) > x_e(S_1, S'_1) + x_e(S_2, S'_2) \geq d,$$

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

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

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 (Avis and Fukuda, 1996): 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.

Using Lemma 1 with \(p = m\) leads to the following

**Separation Algorithm for GBIs**

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 \(\bigcup_{e \in M} T_e\):

   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 \(x_{e_i}(S, S') \leq x_{e_{i+1}}(S, S')\) for \(i = 1, \ldots, f - 1\).

   2.c) If one of the partial sums \(\sum_{i=1}^{k} x_{e_i}(S, S')\), 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 \(m\) and \(n\). We note that a much simpler algorithm can be implemented with complexity polynomial in \(m\) and \(2^n\) : Replace \(\bigcup_{e \in E'} T_e\) 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 \(x_e(S, S') < s + 1\) 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. \( \square \)

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 \(0 \leq x_e \leq 1\). 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: \[ 2 > x_e(S_1, S'_1) + x_e(S_2, S'_2) \geq |S_1 \Delta S_2| \] implies that \( |S_1 \Delta S_2| = 1 \). Moreover, we can not have more than two subsets of \( N \) so that the symmetric difference of each pair contains just one element. \( \square \)

Note that Lemma 2 gives a direct way to compute in \( O(n) \) the active partition(s) for an edge \( e \): The partition \((S_1, S'_1)\) minimizing \( x_e(S_1, S'_1) \) can be found greedily. If \((S_1, S'_1)\) is active and if a second active partition exists, then it is one of the \( n \) partitions obtained from \((S_1, S'_1)\) by moving one index from \( S_1 \) to \( S'_1 \) or vice versa. Motivated by Lemma 2, we devised the following heuristic as a simple alternative to the exact algorithm.

**Separation Heuristic for GBIs**

1. With respect to \( x_e \), compute its (at most) two active partitions and their values.
2. Then, for each partition \((S, S')\) of \( N\):
   1. Compute the set \( T \) of elements in \( M \) that have \((S, S')\) as an active partition.
   2. Sort the \( e \in T \) according to nondecreasing \( x_e(S, S') \), yielding the ordering \( T = \{e_1, \ldots, e_t\} \).
   3. If one of the partial sums \( \sum_{i=1}^{k} x_e(S, S') \), 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.

3. **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 \(-1 \leq \Pi \leq 1\). Clearly, if a valid inequality separating \( X \) 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 \( X \) 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 $\Pi$ 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

$$\sum_{j \in N} \pi_j^i \cdot \text{bit}_j[k],$$

where 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 $P$:

$$\max \sum_{i \in M} \sum_{j \in N} \sum_{k \in C} (\pi_j^i \cdot \text{bit}_j[k] \cdot z_{ik})$$

s.t. \[\sum_{k \in C} z_{ik} = 1, \quad \forall i \in M;\]

\[\sum_{i \in M} z_{ik} \leq 1, \quad \forall k \in C;\]

\[z_{ik} \geq 0, \quad \forall i \in M, \quad \forall k \in C,\]

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

The dual of $P$ is $D$:

$$\min \sum_{i \in M} \alpha_i + \sum_{k \in C} \beta_k$$

s.t. \[\alpha_i + \beta_k \geq \sum_{j \in N} \pi_j^i \cdot \text{bit}_j[k], \quad \forall i \in M, \quad \forall k \in C;\]

\[\beta_k \geq 0, \quad \forall k \in C.\]

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

$$\max \sum_{i \in M} \sum_{j \in N} \pi_j^i \bar x_j^i - \sigma$$

s.t. \[-1 \leq \Pi \leq 1;\]

\[\sum_{i \in M} \alpha_i + \sum_{k \in C} \beta_k \leq \sigma;\]

\[\alpha_i + \beta_k \geq \sum_{j \in N} \pi_j^i \cdot \text{bit}_j[k], \quad \forall i \in M, \quad \forall k \in C;\]

\[\beta_k \geq 0, \quad \forall k \in C.\]

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

This approach yields a practical and efficient algorithm for producing maximally violated normalized cuts if any such cut exists. In Section 6, we refer to each cut produced in this way as an LP cut (LPC). Note that in Lee and Margot (2004) we also proved Theorem 2 by constructing an efficient algorithm, but that algorithm is not practical for computation.

4. Matching Inequalities

Let \( S, S' \) be subsets of \( N \) with \( S \cap S' = \emptyset \). The optimal colors for \((S, S')\) are the colors \( x \in \{0, 1\}^N \) that yield \( x(S, S') = 0 \). The set of optimal colors for \((S, S')\) is denoted by \( B(S, S') \). 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 \( x \in \{0, 1\}^N \) is not an optimal color for \((S, S')\), then \( x(S, S') \geq 1 \).

**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 \( E' \))

\[
\text{(MI)} \quad x_{E'}(S, S') \geq |E' \setminus F|
\]

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. \( \square \)

When \( E' \) is an odd cycle, the matching inequalities reduce to the so-called “type-I odd-cycle inequalities” (see Lee (2002) who introduced these latter inequalities and Lee, Leung, and de Vries (2005) who provided an efficient separation algorithm for them).

A MI is dominated if it is implied by MIs on 2-connected non-bipartite subgraphs and by GBIs. The following proposition shows that it is enough to generate the non-dominated MIs, provided that the GBIs generated by the separation heuristic for GBIs of Section 2 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 inequality $x_A(S, S') \geq |A| - 1$ with $A = \delta(v) \cap E'$. Note that if this last inequality is violated, then so is the GBI for the edges in $A$ having $(S, S')$ as an active partition.

(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 MIs on $G_1$ and $G_2$.

(iv) By König's theorem, the cardinality of a minimum vertex cover of $G'$ is equal to the cardinality $k$ of a maximum matching $F$ of $G'$ 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 GBIs for the stars are all satisfied, then summing them up yields:

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

and the MI induced by $E'$ is also satisfied. Note that, similarly to point (ii), if one of the GBIs used above is violated, then so is one of the GBIs on the edges of each star having $(S, S')$ as an active partition. □

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

**Separation Heuristic for MIs**

(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 $x_{E(G')}^k(S, S') \geq |E(G') \setminus F(G')|$ 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 $O(\sqrt{|V(G)|})$ (Vazirani, 1994). 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 $x_{E(G')}^k(S, S') < |E(G') \setminus F(G')|$. Then $x_{E(G')}^k(S, S') < |(E(G') - e) \setminus F(G' - e)|$ is also violated, as the left-hand side has been reduced by more than 1, while the right-hand side has been reduced.
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 $|B(S_1, S'_1)| \leq 2$, that $|B(S_2, S'_2)| \leq 2$ and that $|B(S_1, S'_1) \cap B(S_2, S'_2)| \geq 1$.

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 $j \in S_{t'}$ (resp., $j \in S'_{t'}$) 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

$$(\text{SWI}) \quad \sum_{i=1}^{r} x_i(S_i, S'_i) \geq 1$$

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:

$$+x^0_1 \quad +x^0_2 \quad +x^1_2 \quad +x^2_2 \quad +x^3_2 \quad +x^0_3 \quad +x^0_3 \quad +x^1_3 \quad +x^2_3 \quad +x^3_3 \quad \geq 1.$$
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 SWIs and the simple bound inequalities $0 \leq X \leq 1$.

Theorem 3 was stated without proof in Lee and Margot (2004). The proof, which we present here, uses the following five lemmas.

**Lemma 3** Let $\langle \gamma, x \rangle \geq \beta$ describe a facet $F$ of a full dimensional 0,1 polytope $Q$ in $\mathbb{R}^q$. Assume that $\langle \gamma, x \rangle \geq \beta$ is not a positive multiple of a simple-bound inequality $x_i \geq 0$ or $-x_i \geq -1$. 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 $F$ satisfy $x_i = 0$ (resp., $x_i = 1$), and the inequality must be a positive multiple of one of the simple-bound inequalities, a contradiction. □

**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 $0 \in \mathbb{R}^N$ for all even $i$ and to color $1 \in \mathbb{R}^N$ 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 $\Phi$, define $\Phi_-$ as the sum of its negative entries. We extend the definition of optimal colors given at the beginning of Section 3 to handle arbitrary coefficients: Let $\phi_e$ be the vector of coefficients associated with the binary variables for edge $e$. The optimal colors for $e$ are all the 0,1 $n$-vectors $\bar{x}_e$ yielding the minimum possible value for $\langle \phi_e, \bar{x}_e \rangle$. 

10
Lemma 6 Let $f_1, \ldots, f_r$ be the ordered edges of path $P$. Let $\phi_i$ be the vector of coefficients associated with $f_i$ in a facet-describing inequality $\langle \Phi, X \rangle \geq \beta$. Suppose that $\phi_i$ 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 $X$ for which $\langle \Phi, X \rangle = \beta$.

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 $x_i^a \geq 0$ or $-x_i^a \geq -1$. Let $e_1, e_2, \ldots, e_k$ be the smallest subpath of $P$ containing all edges for which $\phi_i$ is not the zero vector. If $\min\{||\phi_i^a|| \ | \phi_i^a \neq 0\} = 1$, then all nonzero components of $\Phi$ are $\pm 1$, $\phi_1$ and $\phi_k$ each have no 0, and $\phi_i$ 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 GBIs give the convex hull of such colorings (see Lee (2002)). 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 $\Phi$ to $S$. Also, $X$ will always be an integral matrix in $Q_n(P)$ satisfying $\langle \Phi, X \rangle = \beta$.

Consider the set of all $X$ satisfying $\langle \Phi, X \rangle = \beta$ and the set $\chi$ of all colorings of $P_1$, of $P_2$ and of $e_1$ that they induce. For $i = 1, 2$, let $a_i$ be the optimal value of a coloring of $P_i$ in $\chi$, and let $b_i$ be the second best value of such a coloring (with $a_i < b_i$). Let $\chi^{a_i}$ (resp. $\chi^{b_i}$) be the colorings of $P_i$ in $\chi$ having value $a_i$ (resp. $b_i$). Let $A_i$ (resp., $B_i$) be the set of colors for the shore of $P_i$ in all colorings in $\chi^{a_i}$ (resp., $\chi^{b_i}$). Note that $A_i \cap B_i = \emptyset$ as replacing a valid coloring of $P_i$ by another valid coloring of $P_i$ always gives a valid coloring of $P$ if the color of $e_{t-1}$ is not changed. Since all colorings $X$ in $\chi$ have $\langle \Phi, X \rangle = \beta$, this would yield a contradiction. For edge $e_t$, let $a$, $b$ and $c$ be the three best values for a coloring of $e_t$ in $\chi$, with $a < b < c$ and with corresponding color sets $A$, $B$ and $C$.

As $\Phi$ induces a facet of $Q_n(P)$, there exists a coloring $\tilde{X}$ inducing a coloring of $P_1$ that is not in $\chi^{a_1}$. Let $\tilde{c}$ be the color of $e_t$ in $\tilde{X}$. If there exists a coloring of $P_1$ in $\chi^{a_1}$ where $e_{t-1}$ receives a color other than $\tilde{c}$ then using this coloring for $P_1$ and the coloring induced by $\tilde{X}$ for edges $\{e_t, \ldots, e_k\}$ would give a valid coloring $X'$ with $\langle \Phi, X' \rangle < \beta$, a contradiction. Thus all colorings in $\chi^{a_1}$ give to the shore of $P_1$ the color $\tilde{c}$. A similar remark holds for $P_2$. It follows that $|A_i| = 1$ for $i = 1, 2$. A similar reasoning shows that the colorings of $P_1$ in $\chi^{a_1}$.
are in fact optimal colorings of $P_t$. Taking $t = 2$ (resp., $t = k - 1$), this implies that $\phi_1$ (resp., $\phi_k$) has no 0. Lemma 3 shows that all entries in $\phi_1$ (resp., $\phi_k$) must have the same absolute value. Indeed, any coloring $c'$ of $e_1$ obtained by flipping a subset $S$ of entries with $|S| > 1$ in its optimal coloring $e_1$ has a value strictly larger than the value of any coloring obtained from $c_1$ by flipping a proper subset of $S$. Since the set $C_S$ of colorings obtained from $c_1$ by flipping exactly one entry of $S$ contains more than one color, if $c'$ is the color of $e_1$ in $X$, it is possible to replace $c'$ by a coloring in $C_S$ and obtain a feasible coloring of $P$ with a smaller value, a contradiction. It follows that $X$ induces a coloring on $e_1$ that is in $c_1$ or one of the colorings obtained from $e_1$ by flipping exactly one entry of $e_1$.

Suppose that there exists a coloring $\bar{X}$ inducing a coloring of $P_1$ in $\chi \setminus (\chi^a_1 \cup \chi^b_1)$. Let $\bar{c}$ be the color of $e_1$ in $\bar{X}$. Since $A_1 \cap B_1 = \emptyset$, there exist a coloring of $P_1$ in $\chi^a_1 \cup \chi^b_1$ giving a color other than $\bar{c}$ to its shore. Using this coloring for $P_1$ and the coloring induced by $X$ for edges $\{e_1, \ldots, e_k\}$ would give a valid coloring $X'$ with $\Phi(X') < \beta$, a contradiction. Hence, the color of the shore in any coloring of $P_1$ in $\chi$ is in $A_1 \cup B_1$ for $i = 1, 2$.

Similarly, for some $\bar{X}$, the color of $e_1$ in $\bar{X}$ does not have value $a$. Let $A'$ be the union of $A$ and of the set of all optimal colorings of $e_1$. We have that $|A'| < 3$ (as otherwise the color of $e_1$ can be changed to a color in $A'$ in any valid coloring of $P$). Thus, if $|A| = 1$ then $\phi_t$ has no 0, and if $|A| = 2$ then $\phi_t$ has exactly one 0 and $A$ is the set of optimal colors for $e_1$. Also, we have that the color of $e_1$ 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 $X$ induces a pattern $(H_1, H_2)$ 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., $H_2$). Lemmas 4 and 5 imply that the projection on $(e_{t-1}, e_t, e_{t+1})$ of all the points $X$ should span an affine space of dimension at least $3n - 1$, i.e. there should be at least $3n$ affinely independent such projections. Note that $A_i \subseteq (A \cup B \cup C)$, as otherwise $P_t$ is always optimally colored in all $X$, a contradiction. As we have shown above that $|A_i| = 1$ for $i = 1, 2$, we have that $A_i \subseteq A, A_i \subseteq B$ or $A_i \subseteq C$.

Case I: $|A| = 1$. As shown above, we have that $\phi_t$ has no 0 entries.

Case Ia: $A_1 \neq A$ and $A_2 \neq A$. Then any $X$ induces on $(e_{t-1}, e_t, e_{t+1})$ the pattern $(A_1, A_2)$, 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 $X$ induces on $(e_{t-1}, e_t, e_{t+1})$ one of the patterns $(B_1, A, B_2), (A_1, B, A_2), (A_1, C, A_2)$. 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 $\gamma = \frac{b_1 - a_1}{b_2 - a_2}$. Observe that each $X$ satisfies the inequality obtained on $P_t \cup e_t$ using the restriction of $\Phi$ to $P_t$ and using $\gamma \cdot \phi_t$ for $e_t$ with right hand side $\beta - (1 - \gamma)a = \beta - a_2 - (1 - \gamma)b$, a contradiction.

Case Ib2: $A_1 \subseteq C$. Then any $X$ 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 $X$ induce the third pattern. Then all $X$ induce a coloring of $P_1$ that is in $\chi^{a_1}$, a
contradiction.

Case Ib3: \(|B| = 1\) and \(A_1 = B\). Then any \(X\) 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 \(A\) must be the optimal color for \(e_t\). Otherwise, let \(\bar{c}\) be the optimal coloring of \(e_t\). Then \(\bar{c} \not\in A \cup B = A_2 \cup A_1\) and thus replacing \(C\) by \(\bar{c}\) is possible in an \(X\) inducing the pattern \((A_1, C, A_2)\), a contradiction. A similar reasoning shows that \(B\) (resp., \(C\)) must be the set of “second best” (resp., “third best”) colorings of \(e_t\). It follows that \(|C| \leq n - 1\) 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 \(A\). 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 \(X\) 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 solutions with the third pattern, implying that no \(X\) induces the second pattern. Then all \(X\) induce a coloring of \(P_t\) that is in \(\chi^{a_1}\), a contradiction.

Case II: \(|A| = 2\). Let \(A = \{U, V\}\) with \(|U| = |V| = 1\). As shown earlier, we have that \(\phi_t\) has exactly one 0 entry and that \(A\) is the set of optimal colors for \(e_t\). Then only \(A\) and \(B\) may appear in the projection of \(X\) 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 \(X\) induces on \((e_{t-1}, e_t, e_{t+1})\) one of the patterns \((A_1, U, A_2)\), \((A_1, V, A_2)\), \((A_1, B - A_1, A_2)\) and \((B_1, A_1, A_2)\). 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 the coloring induced on \(P_t\) is always in \(\chi^{a_1}\).

Case IIb: \(A_1 = A_2 = U\). Then any \(X\) induces on \((e_{t-1}, e_t, e_{t+1})\) one of the patterns \((B_1, U, A_2)\), \((A_1, V, A_2)\) 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 \(X\) 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 \(\phi_t\) 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 nonzero entries in \(\Phi\) 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 \(\Phi\) and \(\beta\) in Case IIc of Lemma 7 force the inequality to be a SWI. Indeed, as \(\phi_i\) is a 0, ±1-vector for all edges \(e_i\), we can associate the subpartition \((S_i, S'_i)\) with \(j \in S_i\) if and only if \(x_{ij} = 1\) and \(j \in S'_i\) if and only if \(x_{ij} = -1\). It is clear that conditions (a) and (b) of the definition of a SWI are satisfied. 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. \(\square\)

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

**Proof:** If \(k = 2\), the SWI is a GBI and thus is valid. Consider a SWI generated on a path \(P\) with edges \(\{e_1, e_2, \ldots, e_k\}\) 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 \(P_t\), \(P_2\) and \(e_t\). But this is impossible, as \(A_1 \cup A_2 = A\). \(\square\)

We separate the SWIs 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 \(x_e(S, S') < 1\);

(d) an indicator \(\text{ind}\) 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 \(\text{ind}\) of that node.

The weight associated with the node is \(x_e(S, S')\). There is an arc from node \((e_1 = (u_1, v_1), (S_1, S'_1), \text{ind}_1)\) to node \((e_2 = (u_2, v_2), (S_2, S'_2), \text{ind}_2)\) 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 \(\text{ind}_1 \neq \text{ind}_2\) or (II) \(j_1 \neq j_2\) and \(j_1\) is in the set \(\text{ind}_1\) of the second node, and \(\text{ind}_2 = S'_2\) 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 \(8n(n + 1)^2m\) 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 \(O(mn^3 \log(mn))\).
6. Computational Results

We report computational results for Branch-and-Cut (B&C) algorithms using the GBIs, LPCs, MIs, and SWIs. The results that we present improve upon the preliminary results first reported in Lee and Margot (2004). The code is based on the open-source codes BCP (Branch, Cut & Price) and CLP (an LP solver), which are freely available from COIN-OR (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 $g_{4,p}$ on $p$ nodes, for $p = 20, 30, \ldots, 100$;
(b) three 8-regular graphs $g_{8,p}$ 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 ($of_{5,14,7}$ (degree 5) and $of_{7,18,9}$ (degree 7));
(e) an overfull graph with 9 vertices ($of_{sub9}$) obtained as a subgraph of $of_{7,18,9}$ ($\Delta = 8$);
(f) Graphs from Chetwynd and Wilson (1983) on 18 vertices, 33 edges and 30 vertices, 57 edges ($jgt_{30}$). Both graphs have maximum degree 4.

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. The problems solved by the B&C are just the feasibility problems, i.e. deciding if the graph can be colored with $\Delta(G)$ colors or not.

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 and trade-offs between the different types of cuts, we report results of three B&C algorithms based on the binary formulation and one B&C using the usual (unary) formulation. The separation algorithms for the different types of cuts are: the separation heuristic for GBIs of Section 2, the exact LPC separation algorithm alluded to at the end of Section 3, the heuristic MI separation algorithm of Section 4 (except that blocks are not computed, using the non bipartite connected components instead), and the separation algorithm for SWIs of Section 5.

Optimizing the cut management is not an obvious task, since we have four different types of cuts, two of which are similar (GBIs and LPCs). We settled for the following reasonable scheme, although it is probably not optimum. No
more than six rounds of cutting is done at each node, each type of cut being considered. However, in each round of cutting, LPCs are used only if no GBI could be obtained by the separation heuristic. Moreover, at most one round of SWIs is used at each node of the B&C. Cuts that are not tight for the current optimal solution of the LP relaxation are immediately removed.

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 only GBIs. B&C 2 uses, in addition, LPCs, and MIs and B&C 3 uses all four types of cuts. For B&C 4 (using the unary formulation), no cuts are generated and the above branching scheme is used. 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 LPCs and MIs seems to be important. (The effect attributable to LPCs is much larger than the one for MIs.) On the other hand, the use of SWIs does not seems to help much on these problems.

Table 2 shows that using SWIs 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 SWIs at most, the separation algorithm returns a large number of violated SWIs. A better understanding of these cuts might help generate “useful” ones more efficiently. The separation times are very small for GBIs and MIs. The LPCs, however take significant time (more than 75% of the total time for the 4-regular graphs, about 50% of the total time for the 8-regular graphs and more than 50% for of7,18,9). The SWI separation is also time consuming, taking roughly 70% of the time difference between B&C 2 and B&C 3. We separate LPCs by looking at each node of the graph, and solving the LP of Section 3 on the edges incident to the node. Significant savings could probably be achieved by constructing a priority order for the nodes and stopping after a given number of LPCs have been generated in each iteration. Any faster way to generate these cuts would help.

Comparing B&C 2 with B&C 4 (which uses the traditional unary formulation), it appears that B&C 4 is roughly 8 times faster than B&C 2 on the most difficult problems, but the number of nodes is generally comparable to that of B&C 2. On the other hand, the time comparison between B&C 1 and B&C 4 is much closer, with a node advantage to B&C 4. We note that for jgt30 (which is a very difficult instance), B&C 2 and B&C 3 have a clear advantage over B&C4 with respect to nodes, and B&C1 has a small advantage over B&C4 with respect to time. It is a challenge to be able to get the promise of the node decreases of B&C 2 and B&C 3 to translate to faster solution times. Overall, we believe that this is enough evidence of potential for our methods to justify further investigation.

The speed advantage of the unary formulation is not completely unexpected as the advantage (in both time and memory) obtained from working with $n$ variables (for the binary formulation) versus $2^n$ variables (for the unary formu-
lation) associated with each edge is limited for small values of $n$ (our examples had $n \leq 3$). We looked at testing with instances with larger degrees, but large feasible instances require efficient coloring heuristics (adding a lot of randomness in the results) and Class 2 graphs are not easy to produce. Most known constructions for Class 2 graphs with large degree either yield highly symmetrical graphs or extremely large instances. Both are very difficult to solve using any ILP formulation.

Acknowledgement

The research of François Margot was supported in part by ONR grant N00014-03-1-0188.

References


17
List of Tables

1. Number of nodes. ........................................... 19
2. cpu time in seconds. ........................................ 20
3. Number of generated cuts. .............................. 21
Table 1: Number of nodes.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(g_{4,20})</td>
<td>18</td>
<td>16</td>
<td>19</td>
<td>7</td>
</tr>
<tr>
<td>(g_{4,30})</td>
<td>32</td>
<td>29</td>
<td>27</td>
<td>16</td>
</tr>
<tr>
<td>(g_{4,40})</td>
<td>58</td>
<td>43</td>
<td>44</td>
<td>37</td>
</tr>
<tr>
<td>(g_{4,50})</td>
<td>75</td>
<td>51</td>
<td>71</td>
<td>52</td>
</tr>
<tr>
<td>(g_{4,60})</td>
<td>75</td>
<td>81</td>
<td>72</td>
<td>61</td>
</tr>
<tr>
<td>(g_{4,70})</td>
<td>87</td>
<td>81</td>
<td>86</td>
<td>98</td>
</tr>
<tr>
<td>(g_{4,80})</td>
<td>662</td>
<td>85</td>
<td>92</td>
<td>78</td>
</tr>
<tr>
<td>(g_{4,90})</td>
<td>139</td>
<td>126</td>
<td>121</td>
<td>100</td>
</tr>
<tr>
<td>(g_{4,100})</td>
<td>1168</td>
<td>120</td>
<td>127</td>
<td>119</td>
</tr>
<tr>
<td>(g_{8,20})</td>
<td>133</td>
<td>106</td>
<td>100</td>
<td>106</td>
</tr>
<tr>
<td>(g_{8,30})</td>
<td>*</td>
<td>212</td>
<td>172</td>
<td>171</td>
</tr>
<tr>
<td>(g_{8,40})</td>
<td>302</td>
<td>239</td>
<td>218</td>
<td>224</td>
</tr>
<tr>
<td>(k_5)</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>(pete)</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>(af_{5,14,7})</td>
<td>58</td>
<td>47</td>
<td>50</td>
<td>56</td>
</tr>
<tr>
<td>(af_{7,18,9})</td>
<td>50328</td>
<td>24255</td>
<td>24299</td>
<td>24607</td>
</tr>
<tr>
<td>(af_{sub})</td>
<td>125934</td>
<td>83373</td>
<td>82987</td>
<td>88237</td>
</tr>
<tr>
<td>(jgt_{18})</td>
<td>680</td>
<td>445</td>
<td>474</td>
<td>629</td>
</tr>
<tr>
<td>(jgt_{30})</td>
<td>91515</td>
<td>58246</td>
<td>57806</td>
<td>85553</td>
</tr>
</tbody>
</table>
Table 2: cpu time in seconds.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>g4_20</td>
<td>0.00</td>
<td>0.20</td>
<td>0.20</td>
<td>0.00</td>
</tr>
<tr>
<td>g4_30</td>
<td>0.00</td>
<td>0.60</td>
<td>0.60</td>
<td>0.00</td>
</tr>
<tr>
<td>g4_40</td>
<td>0.10</td>
<td>1.20</td>
<td>1.60</td>
<td>0.10</td>
</tr>
<tr>
<td>g4_50</td>
<td>0.20</td>
<td>1.50</td>
<td>3.60</td>
<td>0.10</td>
</tr>
<tr>
<td>g4_60</td>
<td>0.20</td>
<td>3.00</td>
<td>4.40</td>
<td>0.20</td>
</tr>
<tr>
<td>g4_70</td>
<td>0.30</td>
<td>3.50</td>
<td>7.20</td>
<td>0.40</td>
</tr>
<tr>
<td>g4_80</td>
<td>4.10</td>
<td>4.20</td>
<td>9.70</td>
<td>0.40</td>
</tr>
<tr>
<td>g4_90</td>
<td>0.70</td>
<td>7.50</td>
<td>20.20</td>
<td>0.50</td>
</tr>
<tr>
<td>g4_100</td>
<td>11.30</td>
<td>8.20</td>
<td>20.60</td>
<td>1.00</td>
</tr>
<tr>
<td>g8_20</td>
<td>0.50</td>
<td>4.20</td>
<td>5.80</td>
<td>0.20</td>
</tr>
<tr>
<td>g8_30</td>
<td>*</td>
<td>17.50</td>
<td>20.60</td>
<td>0.80</td>
</tr>
<tr>
<td>g8_40</td>
<td>3.40</td>
<td>31.50</td>
<td>46.60</td>
<td>1.40</td>
</tr>
<tr>
<td>k5</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>peter</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>o5f_14_7</td>
<td>0.20</td>
<td>0.70</td>
<td>1.00</td>
<td>0.10</td>
</tr>
<tr>
<td>of7_18_9</td>
<td>304.70</td>
<td>1027.30</td>
<td>1352.00</td>
<td>132.40</td>
</tr>
<tr>
<td>of_subb9</td>
<td>187.20</td>
<td>820.30</td>
<td>1176.10</td>
<td>123.40</td>
</tr>
<tr>
<td>jgt18</td>
<td>0.70</td>
<td>4.70</td>
<td>7.00</td>
<td>0.60</td>
</tr>
<tr>
<td>jgt30</td>
<td>143.50</td>
<td>1086.70</td>
<td>1787.20</td>
<td>152.40</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>GBI</td>
<td>LPC</td>
<td>GBI</td>
<td>MI</td>
</tr>
<tr>
<td>$g_{4,20}$</td>
<td>68</td>
<td>27</td>
<td>98</td>
<td>4</td>
</tr>
<tr>
<td>$g_{4,30}$</td>
<td>192</td>
<td>57</td>
<td>222</td>
<td>4</td>
</tr>
<tr>
<td>$g_{4,40}$</td>
<td>404</td>
<td>98</td>
<td>330</td>
<td>9</td>
</tr>
<tr>
<td>$g_{4,50}$</td>
<td>682</td>
<td>135</td>
<td>576</td>
<td>15</td>
</tr>
<tr>
<td>$g_{4,60}$</td>
<td>430</td>
<td>195</td>
<td>674</td>
<td>7</td>
</tr>
<tr>
<td>$g_{4,70}$</td>
<td>638</td>
<td>214</td>
<td>734</td>
<td>20</td>
</tr>
<tr>
<td>$g_{4,80}$</td>
<td>716</td>
<td>183</td>
<td>960</td>
<td>21</td>
</tr>
<tr>
<td>$g_{4,90}$</td>
<td>1338</td>
<td>323</td>
<td>1266</td>
<td>51</td>
</tr>
<tr>
<td>$g_{4,100}$</td>
<td>24380</td>
<td>375</td>
<td>1238</td>
<td>31</td>
</tr>
<tr>
<td>$g_{8,20}$</td>
<td>1302</td>
<td>1452</td>
<td>1934</td>
<td>15</td>
</tr>
<tr>
<td>$g_{8,30}$</td>
<td>4029</td>
<td>5796</td>
<td>61</td>
<td>3556</td>
</tr>
<tr>
<td>$g_{8,40}$</td>
<td>4050</td>
<td>5416</td>
<td>5074</td>
<td>33</td>
</tr>
<tr>
<td>$k_5$</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>$peter$</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>$o_{5,14,7}$</td>
<td>1146</td>
<td>256</td>
<td>842</td>
<td>39</td>
</tr>
<tr>
<td>$o_{7,18,9}$</td>
<td>1405372</td>
<td>377279</td>
<td>936644</td>
<td>48487</td>
</tr>
<tr>
<td>$o_{sub9}$</td>
<td>281339</td>
<td>231743</td>
<td>217871</td>
<td>15370</td>
</tr>
<tr>
<td>$jgt18$</td>
<td>1053</td>
<td>555</td>
<td>1075</td>
<td>215</td>
</tr>
<tr>
<td>$jgt30$</td>
<td>249030</td>
<td>105896</td>
<td>201736</td>
<td>33838</td>
</tr>
</tbody>
</table>