INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

On a Binary-Encoded ILP Coloring Formulation

Jon Lee, François Margot,

INFORMS Journal on Computing
Vol. 19, No. 3, Summer 2007, pp. 406–415
ISSN 1091-9856 | EISSN 1526-5528 | 07 | 1903 | 0406

DOI 10.1287/ijoc.1060.0178

On a Binary-Encoded ILP Coloring Formulation

Jon Lee
Department of Mathematical Sciences, IBM T. J. Watson Research Center,
Yorktown Heights, New York 10598, jonlee@us.ibm.com

François Margot
Tepper School of Business, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213,
fmargot@andrew.cmu.edu

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) an introduction of matching inequalities, (iv) an introduction of switched path inequalities and their efficient separation, (v) a complete description for paths, and (vi) the promising computational results.

Key words: integer programming; binary encoding; all different; edge coloring
History: Accepted by William J. Cook, Area Editor for Design and Analysis of Algorithms; received September 2004; revised May 2005, November 2005; accepted January 2006. Published online in *Articles in Advance* July 20, 2007.

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 $\delta(v) := \{e \in E(G)\colon e \text{ is incident to } v\}$. Let $\Delta(G) := \max\{|\delta(v)|\colon v \in V(G)\}$. 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 cannot 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_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

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

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 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 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 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 express effectively 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 $\mathbf{0} \le x_e \le \mathbf{1}$, and $S, S' \subseteq N$ with $S \cap S' = \varnothing$, we define the *value of* (S, S') *on* e as $\mathbf{x}_e(S, S') := \sum_{j \in S} x_e^j + \sum_{j \in S'} (1 - x_e^j)$. 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 a 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 $\mathbf{x}_{E'}(S, S') := \sum_{e \in E'} \mathbf{x}_e(S, S')$. 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.

2. Separation for General Block Inequalities

For $1 \le p \le 2^n$, p can be written uniquely as $p = h + \sum_{k=0}^{t} \binom{n}{k}$, with $0 \le h < \binom{n}{t+1}$. 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):

$$\text{(GBI)} \quad \kappa(|L|, n) \le \mathbf{x}_L(S, S').$$

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 considering 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 $1 \le p \le 2^n$, and let t be the n-binomial size of p. Then for $x_e \in \mathbb{R}^N$ with $\mathbf{0} \le x_e \le \mathbf{1}$, at most $2(n+1)^2 p^2$ 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 $S_1 = \varnothing$ by replacing x_e^j by $1 - x_e^j$ for all $j \in S_1$. As $2(t+1) > \mathbf{x}_e(S_1, S_1') + \mathbf{x}_e(S_2, S_2') \ge d$, we have that $d \le 2t+1$. The number T of possible $(t+1)$-light partitions satisfies $T \le \sum_{k=0}^{2t+1} \binom{n}{k}$.

Using that, for all $k \le n/2$, $\binom{n}{2k} \le \binom{n}{k}^2$ and $\binom{n}{2k-1} \le \binom{n}{k}^2$ and that, for nonnegative numbers a, b, we have $a^2 + b^2 \le (a+b)^2$, we get $T \le 2 \sum_{k=0}^{t+1} \binom{n}{k}^2 \le 2 \left(\sum_{k=0}^{t+1} \binom{n}{k}\right)^2$. By hypothesis, we have $\binom{n}{t+1} \le n\binom{n}{t} \le np$, and thus $\sum_{k=0}^{t+1} \binom{n}{k} \le (p-h) + np \le (n+1)p$. 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 (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 $\mathbf{x}_{e_i}(S, S') \le \mathbf{x}_{e_{i+1}}(S, S')$ 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 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 $\kappa(|L|, n) - \kappa(|L|-1, n) \le s+1$. 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 $s \le t$, 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 $\mathbf{0} \le x_e \le \mathbf{1}$. 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 cannot have more that two subsets of N so that the symmetric difference of each pair contains just one element. □

Note that Lemma 2 gives a direct way to compute in $O(n)$ time the active partition(s) for an edge e: The partition (S_1, S_1') minimizing $\mathbf{x}_e(S_1, S_1')$ 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

(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.

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 $-\mathbf{1} \leq \Pi \leq \mathbf{1}$. 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 $\sum_{j \in N} \pi_i^j \mathrm{bit}_j[k]$, 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:

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

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

The dual of P is D:

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

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

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

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 $\langle \Pi, \bar{X} \rangle \leq \sigma$ 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. □

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 $S \cap S' = \varnothing$. 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 $\mathscr{B}(S, S')$. 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 $|N \backslash (S \cup S')| = k$, 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 $\mathbf{x}(S, S') \geq 1$.

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')

$$\text{(MI)} \quad \mathbf{x}_{E'}(S, S') \geq |E' \backslash F|$$

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 Lee 2002 who introduced these latter inequalities, and Lee et al. 2005, who provided an efficient separation algorithm for them).

An MI is *dominated* if it is implied by MIs on 2-connected nonbipartite subgraphs and by GBIs. The following proposition shows that it is enough to generate the nondominated 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 $\mathbf{x}_A(S, S') \geq |A| - 1$ 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'. 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 $\mathbf{x}_{E(G')}(S, S') \geq \sum_{i=1}^{k}(k_i - 1) = |E(G')| - k = |E(G') \backslash F|$, 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 nonbipartite block of the graph G' induced by T:

(1.b.i) Compute a maximum matching $F(G')$ in G'.

(1.b.ii) Check if $\mathbf{x}_{E(G')}(S, S') \geq |E(G') \backslash F(G')|$ 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 nonbipartite blocks take $O(m)$. For one partition (S, S'), computing the maximum matchings takes $O(\sqrt{|V(G)|}m)$ 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 $\bar{\mathbf{x}}_{E(G')}(S, S') < |E(G') \backslash F(G')|$. Then $\bar{\mathbf{x}}_{E(G')-e}(S, S') < |(E(G') - e) \backslash F(G' - e)|$ 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 MIs 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 nonbipartite subgraphs might thus be feasible.

5. Switched Walk Inequalities

Let $S, S' \subseteq N$ such that $S \cap S' = \varnothing$ 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 $|\mathscr{B}(S_1, S_1')| \leq 2$, that $|\mathscr{B}(S_2, S_2')| \leq 2$ and that $|\mathscr{B}(S_1, S_1') \cap \mathscr{B}(S_2, S_2')| \geq 1$.

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

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

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:

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

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 SWIs and the simple bound inequalities $\mathbf{0} \leq X \leq \mathbf{1}$.*

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 $x_i^j \geq 0$ or $-x_i^j \geq -1$. 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 $\mathbf{0} \in \mathbb{R}^N$ 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. We extend the definition of optimal colors given at the beginning of Section 3 to handle arbitrary coefficients: Let ϕ_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 $\bar{\mathbf{x}}_e$ yielding the minimum possible value for $\langle \phi_e, \bar{\mathbf{x}}_e \rangle$.

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 $e_1, e_2, \ldots, e_k$ be the smallest subpath of P containing all edges 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 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 Φ to S. Also, $\overline{X}$ will always be an integral matrix in $Q_n(P)$ satisfying $\langle \Phi, \overline{X}\rangle = \beta$.

Consider the set of all $\overline{X}$ satisfying $\langle \Phi, \overline{X}\rangle = \beta$ and the set χ of all colorings of P_1, of P_2 and of e_t that they induce. For $i = 1, 2$, let a_i be the optimal value of a coloring of P_i in χ, and let b_i be the second-best value of such a coloring (with $a_i < b_i$). Let χ^{a_i} (resp., χ^{b_i}) be the colorings of P_i in χ 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 χ^{a_i} (resp., χ^{b_i}). Note that $A_i \cap B_i = \varnothing$ as replacing a valid coloring of P_1 by another valid coloring of P_1 always gives a valid coloring of P if the color of e_{t-1} is not changed. Since all colorings $\overline{X}$ in χ have $\langle \Phi, \overline{X}\rangle = \beta$, 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 χ, 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 $\overline{X}$ inducing a coloring of P_1 that is not in χ^{a_1}. Let $\bar{c}$ be the color of e_t in $\overline{X}$. If there exists a coloring of P_1 in χ^{a_1} where e_{t-1} receives a color other than $\bar{c}$ then using this coloring for P_1 and the coloring induced by $\overline{X}$ for edges $\{e_t, \ldots, e_k\}$ would give a valid coloring $\overline{X}'$ with $\langle \Phi, \overline{X}'\rangle < \beta$, a contradiction. Thus all colorings in χ^{a_1} give to the shore of P_1 the color $\bar{c}$. 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_i in χ^{a_i} are in fact optimal colorings of P_i. 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. Indeed, any coloring c' of e_1 obtained by flipping a subset S of entries with $|S| > 1$ in its optimal coloring c_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 $\overline{X}$, 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 $\overline{X}$ induces a coloring on e_1 that is in c_1 or one of the colorings obtained from c_1 by flipping exactly one entry of c_1.

Suppose that there exists a coloring $\overline{X}$ inducing a coloring of P_1 in $\chi \backslash (\chi^{a_1} \cup \chi^{b_1})$. Let $\bar{c}$ be the color of e_t in $\overline{X}$. Since $A_1 \cap B_1 = \varnothing$, 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 $\overline{X}$ for edges $\{e_t, \ldots, e_k\}$ would give a valid coloring $\overline{X}'$ with $\langle \Phi, \overline{X}'\rangle < \beta$, a contradiction. Hence, the color of the shore in any coloring of P_i in χ is in $A_i \cup B_i$ for $i = 1, 2$.

Similarly, for some $\overline{X}$, the color of e_t in $\overline{X}$ does not have value a. Let A' be the union of A and of the set of all optimal colorings of e_t. We have that $|A'| < 3$ (as otherwise the color of e_t can be changed to a color in A' in any valid coloring of P). Thus, if $|A| = 1$ then ϕ_t has no 0, and if $|A| = 2$ then ϕ_t has exactly one 0 and A is the set of optimal colors for e_t. Also, we have that the color of e_t in $\overline{X}$ is in $A \cup B$ if $|A| + |B| \geq 3$ and in $A \cup B \cup C$ if $|A| = |B| = 1$.

We say that $\overline{X}$ *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 $\overline{X}$ 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 $\overline{X}$ 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_i is always optimally colored in all $\overline{X}$, 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 ϕ_t has no 0 entries.

Case Ia. $A_1 \neq A$ and $A_2 \neq A$. Then any $\overline{X}$ 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 $\overline{X}$ induces on (e_{t-1}, e_t, e_{t+1}) one of the patterns (B_1, A, B_2), (A_1, B, A_2), or (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 $\gamma = (b_1 - a_1)/(b - a)$. Observe that each $\overline{X}$ satisfies the inequality obtained on $P_1 \cup e_t$ using the restriction of Φ to P_1 and using $\gamma \phi_t$ 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 $\overline{X}$ induces on (e_{t-1}, e_t, e_{t+1}) one of the patterns (A_1, A, B_2), (A_1, B, A_2), or (B_1, C, A_2). Note that solutions inducing the third pattern are worse than solutions with the second pattern, implying that no $\overline{X}$ induces the third pattern. Then all $\overline{X}$ induce a coloring of P_1 that is in χ^{a_1}, a contradiction.

Case Ib3. $|B| = 1$ and $A_1 = B$. Then any $\overline{X}$ induces on (e_{t-1}, e_t, e_{t+1}) one of the patterns (A_1, A, B_2), (B_1, B, A_2), or (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 $\bar{c} \notin A \cup B = A_2 \cup A_1$ and thus replacing C by $\bar{c}$ is possible in an $\overline{X}$ 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 $\bar{X}$ 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), or $(A_1, B-A_1, A_2)$. Note that solutions with the second pattern are worse than solutions with the third pattern, implying that no $\bar{X}$ induces the second pattern. Then all $\bar{X}$ induce a coloring of P_1 that is in χ^{a_1}, a contradiction.

Case II. $|A|=2$. Let $A=\{U, V\}$ with $|U|=|V|=1$. As shown earlier, we have that ϕ_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 $\bar{X}$ on e_t. Lemma 3 implies that $|B|=2(n-1)$.

Case IIa. $A_1 \cap A = \varnothing$ (or, symmetrically, $A_2 \cap A = \varnothing$). 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), or (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 the coloring induced on P_1 is always in χ^{a_1}.

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), or (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), or (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 nonzero 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. Indeed, as ϕ_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 $x_i^j = 1$ and $j \in S_i'$ if and only if $x_i^j = -1$. 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 $\phi_t^j = \phi_{t'+1}^j$ if $t'-t$ is even and $\phi_t^j = -\phi_{t'+1}^j$ if $t'-t$ is odd. □

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_1, P_2 and e_t. But this is impossible, as $A_1 \cup A_2 = A$. □

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 $\mathbf{x}_e(S, S') < 1$;
(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 two 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 $O(mn^3 \log(mn))$.

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 and 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, 8 KB 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 (*of*5_14_7 (degree 5) and *of*7_18_9 (degree 7));

(e) an overfull graph wih 9 vertices (*ofsub*9) obtained as a subgraph of *of*7_18_9 ($\Delta = 8$);

(f) the 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 cannot 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 nonregular 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 nonbipartite 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. 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, use of SWIs does not seem 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,

Table 1 Number of Nodes

	B&C			
	1	2	3	4
*g*4_20	18	16	19	7
*g*4_30	32	29	27	16
*g*4_40	58	43	44	37
*g*4_50	75	51	71	52
*g*4_60	75	81	72	61
*g*4_70	87	81	86	98
*g*4_80	662	85	92	78
*g*4_90	139	126	121	100
*g*4_100	1,168	120	127	119
*g*8_20	133	106	100	106
*g*8_30	*	212	172	171
*g*8_40	302	239	218	224
*k*5	3	3	3	3
peter	3	3	3	3
*of*5_14_7	58	47	50	56
*of*7_18_9	50,328	24,255	24,299	24,607
*ofsub*9	125,934	83,373	82,987	88,237
*jgt*18	680	445	474	629
*jgt*30	91,515	58,246	57,806	85,553

Note. The asterisk denotes that the problem was not solved in one hour of CPU time.

Table 2 CPU Time

	B&C			
	1	2	3	4
*g*4_20	0.00	0.20	0.20	0.00
*g*4_30	0.00	0.60	0.60	0.00
*g*4_40	0.10	1.20	1.60	0.10
*g*4_50	0.20	1.50	3.60	0.10
*g*4_60	0.20	3.00	4.40	0.20
*g*4_70	0.30	3.50	7.20	0.40
*g*4_80	4.10	4.20	9.70	0.40
*g*4_90	0.70	7.50	20.20	0.50
*g*4_100	11.30	8.20	20.60	1.00
*g*8_20	0.50	4.20	5.80	0.20
*g*8_30	*	17.50	20.60	0.80
*g*8_40	3.40	31.50	46.60	1.40
*k*5	0.00	0.00	0.00	0.00
peter	0.00	0.00	0.00	0.00
*of*5_14_7	0.20	0.70	1.00	0.10
*of*7_18_9	304.70	1,027.30	1,352.00	132.40
*ofsub*9	187.20	820.30	1,176.10	123.40
*jgt*18	0.70	4.70	7.00	0.60
*jgt*30	143.50	1,086.70	1,787.20	152.40

Notes. The CPU time is in seconds. The asterisk denotes that the problem was not solved in one hour of CPU time.

about 50% of the total time for the 8-regular graphs, and more than 50% for *of*7_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 eight 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 *jgt*30 (which is a very difficult instance), B&C 2 and B&C 3 have a clear advantage over B&C 4 with respect to nodes, and B&C 1 has a small advantage over B&C 4 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) vs. 2^n variables (for the unary formulation) 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.

Table 3 Number of Generated Cuts

	B&C							
	1	2			3			
	GBI	LPC	GBI	MI	LPC	GBI	MI	SWI
*g*4_20	68	27	98	4	26	104	5	326
*g*4_30	192	57	222	4	64	200	4	469
*g*4_40	404	98	330	9	112	420	9	1,210
*g*4_50	682	135	576	15	197	768	11	3,094
*g*4_60	430	195	674	7	184	752	9	2,998
*g*4_70	638	214	734	20	209	976	30	3,509
*g*4_80	7,416	183	960	21	264	1,338	15	5,022
*g*4_90	1,338	323	1,266	51	302	1,814	32	6,098
*g*4_100	24,380	375	1,238	31	421	1,806	19	8,035
*g*8_20	1,302	1,452	1,934	15	1,437	2,056	38	2,006
*g*8_30	*	4,029	5,796	61	3,556	4,226	30	5,592
*g*8_40	4,050	5,416	5,074	33	5,772	5,974	21	7,594
*k*5	6	5	6	0	5	6	0	10
peter	0	5	0	4	5	0	4	0
*of*5_14_7	1,146	256	842	39	307	954	46	779
*of*7_18_9	1,405,372	377,279	936,644	48,487	376,199	966,400	49,291	1,162,716
*ofsub*9	281,339	231,743	217,871	15,370	231,818	227,901	15,960	670,033
*jgt*18	1,053	555	1,075	215	602	1,103	212	7,122
*jgt*30	249,030	105,896	201,736	33,838	104,187	219,596	37,628	1,368,017

Note. The asterisk denotes that the problem was not solved in one hour of CPU time.

Acknowledgments
The research of the second author was supported in part by Office of Naval Research Grant N00014-03-1-0188.

References
Avis, D., K. Fukuda. 1996. Reverse search for enumeration. *Discrete Appl. Math.* **65** 21–46.

Chetwynd, A. G., R. J. Wilson. 1983. The rise and fall of the critical graph conjecture. *J. Graph Theory* **7** 153–157.

Coppersmith, D., J. Lee. 2005. Parsimonious binary-encoding in integer programming. *Discrete Optim.* **2** 190–200.

Holyer, I. 1981. The NP-completeness of edge-coloring. *SIAM J. Comput.* **10** 718–720.

Lee, J. 2002. All-different polytopes. *J. Combin. Optim.* **6** 335–352.

Lee, J., F. Margot. 2004. More on a binary-encoded coloring formulation. D. Bienstock, G. Nemhauser, eds. *Integer Programming and Combinatorial Optimization, Lecture Notes in Computer Science*, Vol. 3064. Springer-Verlag, Berlin, Germany, 271–282.

Lee, J., J. Leung, S. de Vries. 2005. Separating type-I odd-cycle inequalities for a binary encoded edge-coloring formulation. *J. Combin. Optim.* **9** 59–67.

Vazirani, V. V. 1994. 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.

Vizing, V. G. 1964. On an estimate of the chromatic class of a p-graph. *Diskret. Analiz No.* **3** 25–30.