

# Graph Coloring Facets from All-Different Systems

David Bergman, J. N. Hooker

Tepper School of Business, Carnegie Mellon University,  
Pittsburgh PA 15213, USA  
dbergman@andrew.cmu.edu, jh38@andrew.cmu.edu

**Abstract.** We explore the idea of obtaining valid inequalities for a 0-1 model from a constraint programming formulation of the problem. In particular, we formulate a graph coloring problem as a system of all-different constraints. By analyzing the polyhedral structure of alldiff systems, we obtain facet-defining inequalities that can be mapped to valid cuts in the classical 0-1 model of the problem. We focus on cuts corresponding to cyclic structures and show that they are stronger than known cuts. For example, when an existing separation algorithm identifies odd hole cuts, we can supply stronger cuts with no additional calculation. In addition, we generalize odd hole cuts to odd cycle cuts that are stronger than any collection of odd hole cuts.

## 1 Introduction

The vertex coloring problem is one of the best known optimization problems defined on a graph. It asks how many colors are necessary to color the vertices so that adjacent vertices receive different colors. The minimum number of colors is the chromatic number of the graph.

The problem can be given a 0-1 programming model or a constraint programming (CP) model. The 0-1 model benefits from several known classes of facet-defining inequalities that tighten its continuous relaxation. The CP model consists of all-different constraints and is normally solved without the help of a continuous relaxation.

Nonetheless, facet-defining inequalities can be derived for the CP model as well as for the 0-1 model, if its finite domain variables are interpreted as having numerical values. These inequalities can be mapped into the 0-1 model, using a simple change of variable, to obtain valid cuts that we call *finite domain cuts*. Because the CP model has a very different polyhedral structure than the 0-1 model, one might expect the finite domain cuts to be different from known 0-1 cuts. We find that at least one family of finite domain cuts, corresponding to cyclic structures, are not only different from but stronger than known cuts associated with cycles.

This is an instance of a general strategy: reformulate a given 0-1 model in terms of finite domain variables, study the resulting polyhedron, and map any cuts back into the 0-1 model. Binary variables frequently encode a choice that

might just as well be encoded by a single finite domain variable. For example, a 0-1 variable  $y_{ij}$  might represent whether job  $j$  is assigned to worker  $i$ , whether task  $i$  begins at time  $j$ , or whether stop  $j$  follows stop  $i$  on a bus route. These choices can be represented by a finite domain variable  $x_i$  that indicates which job is assigned worker  $i$ , at what time task  $i$  starts, or which stop follows stop  $i$ .

The polyhedral structure of some finite domain CP models, including all-different systems, has been studied. Yet the strength of the resulting cuts has not been directly compared with that of cuts in a 0-1 model. Furthermore, CP models commonly have nonlinear objective functions, so that there is no obvious way to solve a linear relaxation of the problem. However, linear inequalities derived for a CP model generally remain linear when mapped into a 0-1 model, which is more likely to have a linear objective function. This also allows finite domain cuts to be combined with 0-1 cuts that may have complementary strengths. To our knowledge, such a strategy has not previously been examined.

For purposes of testing the idea, we focus on *odd cycles* in the graph, which generalize odd holes. An odd cycle is a cycle of cliques with overlap between adjacent cliques. We identify a new family of *odd cycle cuts* for each odd cycle and show that they are facet-defining in the CP model. We then show that when mapped into 0-1 space, they are stronger than odd hole cuts and clique inequalities generated by the odd cycle.

The odd cycle cuts are parameterized by  $s$ , which is related to the size of the overlap of adjacent cliques. We show the following:

- If an existing algorithm discovers separating odd hole cuts, we can replace them with stronger odd cycle cuts with  $s = 1$ , without any additional calculation. This allows immediate improvement of solution methods that use odd hole cuts.
- A single odd cycle cut with  $s > 1$  strictly dominates all odd hole and clique inequalities that can be generated for the corresponding odd cycle (possibly hundreds or thousands of cuts). Adding a second odd cycle cut provides an even tighter bound.
- A (fast) polynomial-time separation algorithm finds separating odd cycle cuts for a given odd cycle whenever such cuts exist. To obtain the full benefit of odd cycle cuts for  $s > 1$ , existing solution methods can be augmented with a heuristic that identifies odd cycles and applies this algorithm.

We make no claim that graph coloring problems are most efficiently solved using a purely polyhedral approach, although there have been efforts in this direction [11, 12]. Rather, we claim that if relaxation bounds play a role in the solution method, finite-domain cuts can provide tighter bounds than standard 0-1 cuts.

As it happens, the graph coloring problem has a linear objective function in both the CP and 0-1 models. Odd cycle cuts can therefore be added directly to a relaxation of the CP model, if desired, and a bound obtained by solving this relaxation. We find that the cuts yield the same tight bound as in the 0-1 model. If other families of finite domain cuts are developed, this suggests the possibility

of obtaining bounds from a relaxation of the CP model rather than from the much larger 0-1 relaxation.

We begin below with a definition of the problem and a survey of relevant literature. We then define odd cycles, identify an associated family of odd cycle cuts (which we call *x-cuts*), and prove that *x-cuts* are facet-defining. We show that any family of facet-defining inequalities for the graph coloring problem gives rise to a second family of facet-defining cuts that bound the chromatic number. We apply this result to the *x-cuts* just derived to obtain a new family of facet-defining *z-cuts*, which are crucial for obtaining tight bounds. At this point we map *x-cuts* and *z-cuts* into the 0-1 model and investigate computationally their strength relative to odd hole and clique cuts. This is followed by a separation algorithm and suggestions for future research.

## 2 The Problem

Given an undirected graph  $G$  with vertex set  $V$  and edge set  $E$ , the vertex coloring problem is to assign a color  $x_i$  to each vertex  $i \in V$  so that  $x_i \neq x_j$  for each  $(i, j) \in E$ . We seek a solution with the minimum number of colors; that is, a solution that minimizes  $|\{x_i \mid i \in V\}|$ .

The vertex coloring problem can be formulated as a system of all-different constraints. An all-different constraint  $\text{alldiff}(X)$  requires that the variables in set  $X$  take pairwise distinct values. Let  $\{V_k \mid k \in K\}$  be the vertex sets of maximal cliques of  $G$ , so that  $\bigcup_{k \in K} V_k = V$ . Let the colors be denoted by distinct nonnegative numbers  $v_j$  for  $j \in J$ , so that each variable  $x_i$  has the finite domain  $\{v_j \mid j \in J\}$ . Then the problem of minimizing the number of colors is

$$\begin{aligned} \min \quad & z \\ & z \geq x_i, \quad i \in V \\ & \text{alldiff}(X_k), \quad k = 1, \dots, K \\ & x_i \in D = \{v_j \mid j \in J\}, \quad i \in V \end{aligned} \tag{1}$$

where  $X_k = \{x_i \mid i \in V_k\}$ . It is convenient to assume that  $|V|$  colors are available, so that  $|J| = |V|$ . We also label the colors so that  $v_1 < \dots < v_n$ .

A standard 0-1 model for the problem uses binary variables  $y_{ij}$  to denote whether vertex  $i$  receives color  $j$ , and binary variables  $w_j$  that indicate whether color  $j$  is used. The model is

$$\begin{aligned} \min \quad & \sum_{j \in J} w_j \\ & \sum_{j \in J} y_{ij} = 1, \quad i \in V & (a) \\ & \sum_{i \in V_k} y_{ij} \leq w_j, \quad j \in J, \quad k \in K & (b) \\ & y_{ij} \in \{0, 1\}, \quad i \in V, \quad j \in J \end{aligned} \tag{2}$$

The finite domain variables  $x_i$  are readily expressed in terms of 0-1 variables  $y_{ij}$ :

$$x_i = \sum_{j \in J} v_j y_{ij} \quad (3)$$

This allows any valid inequality for model (1) to be mapped to a valid inequality for (2) by substituting the expression in (3) for each  $x_i$ .

### 3 Previous Work

All facets for a single all-different constraint are given in [5, 14]. The facial structure of a system of two all-different constraints is studied in [1, 2]. Facets for general all-different systems are derived for comb structures in [7, 8, 10] and for odd holes and webs in [9]. To our knowledge, the odd cycle cuts described in the present paper have not been previously identified.

It is natural to ask when all facets of an all-different system are facets of individual constraints in the system. It is shown in [10] that this occurs if and only if the all-different system has an *inclusion* property, which means that intersections of sets  $V_k$  in the alldiff constraints are ordered by inclusion. The cycle structures studied here lack the inclusion property and therefore generate new classes of facets.

Known facets for the 0-1 graph coloring model are discussed in [3, 11–13]. These include cuts based on odd holes, odd antiholes, cliques, paths, and odd wheels.

Finite domain cuts have been developed for a few global constraints other than alldiff systems. These include the element constraint [5], the circuit constraint [4], the cardinality constraint [6], cardinality rules [15], the sum constraint [16], and disjunctive and cumulative constraints [6].

### 4 Cycles

We will investigate valid inequalities that correspond to odd cycles. A cycle in graph  $G$  can be defined to be a subgraph of  $G$  induced by the vertices in  $V_1, \dots, V_q \in V$  (for  $q \geq 3$ ), where the subgraph induced by each  $V_k$  is a clique, and the only overlapping  $V_k$ 's are adjacent ones in the cycle  $V_1, \dots, V_q, V_1$ . Thus,

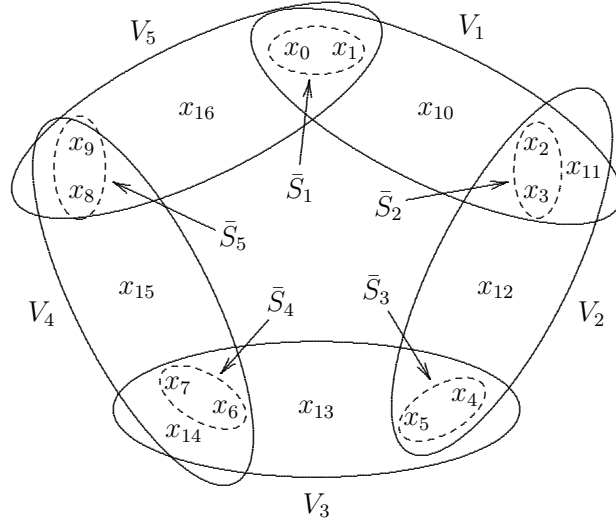
$$V_k \cap V_\ell = \begin{cases} S_k & \text{if } k+1 = \ell \text{ or } (k, \ell) = (q, 1) \\ \emptyset & \text{otherwise} \end{cases}$$

where  $S_k \neq \emptyset$ . A feasible vertex coloring on  $G$  must therefore satisfy

$$\text{alldiff}(X_k), \quad k = 1, \dots, q \quad (4)$$

where again  $X_k = \{x_i \mid i \in V_k\}$ . The cycle is *odd* if  $q$  is odd. If  $|V_k| = 2$  for each  $k$ , an odd cycle is an *odd hole*.

Figure 1 illustrates an odd cycle with  $q = 5$ . Each solid oval corresponds to a constraint  $\text{alldiff}(X_k)$ . Thus  $V_1 = \{0, 1, 2, 3, 10, 11\}$ , and similarly for  $V_2, \dots, V_5$ . All the vertices in a given  $V_k$  are connected by edges in  $G$ .



**Fig. 1.** A 5-cycle. The solid ovals correspond to constraints  $\text{alldiff}(X_k)$  for  $k = 1, \dots, 5$ . The sets  $\bar{S}_1, \dots, \bar{S}_5$  provide the basis for one possible valid cut with  $s = 2$ .

## 5 Valid Inequalities

We first identify valid inequalities that correspond to a given cycle. In the next section, we show that they are facet-defining.

**Lemma 1.** *Let  $V_1, \dots, V_q$  induce a cycle, and let  $\bar{S}_k \in S_k$  and  $|\bar{S}_k| = s \geq 1$  for  $k = 1, \dots, q$ . If  $q$  is odd and  $\bar{S} = \bar{S}_1 \cup \dots \cup \bar{S}_q$ , the following inequality is valid for (1):*

$$\sum_{i \in \bar{S}} x_i \geq \beta(q, s) \quad (5)$$

where

$$\beta(q, s) = \frac{q-1}{2} \sum_{j=1}^{L-1} v_j + \left( sq - \frac{q-1}{2}(L-1) \right) v_L$$

and

$$L = \left\lceil \frac{sq}{(q-1)/2} \right\rceil$$

*Proof.* Because  $q$  is odd, each color can be assigned to at most  $(q-1)/2$  vertices in the cycle. This means that the vertices must receive at least  $L$  distinct colors, and the variables in (4) must take at least  $L$  different values. Because  $v_1 < \dots < v_n$ , we have

$$\sum_{i \in \bar{S}} x_i \geq \frac{q-1}{2} v_1 + \frac{q-1}{2} v_2 + \dots + \frac{q-1}{2} v_{L-1} + \left( sq - \frac{q-1}{2}(L-1) \right) v_L = \beta(q, s)$$

where the coefficient of  $V_L$  is the number of vertices remaining to receive color  $v_L$  after colors  $v_1, \dots, v_{L-1}$  are assigned to  $(q-1)/2$  vertices each.  $\square$

We will refer to the valid inequality (5) as an  $x$ -cut because it contains the variables  $x_i$ . If the cycle is an odd hole, each  $|S_k| = 1$  and  $L = 3$ . So (5) becomes

$$\sum_{i \in \bar{S}} x_i \geq \frac{q-1}{2}(v_1 + v_2) + v_3 \quad (6)$$

If the domain  $\{v_1, \dots, v_n\}$  of each  $x_i$  is  $D_\delta = \{0, \delta, 2\delta, \dots, (n-1)\delta\}$  for some  $\delta > 0$ , inequality (5) becomes

$$\sum_{i \in \bar{S}} x_i \geq \left( sq - \frac{q-1}{4}L \right) (L-1)\delta \quad (7)$$

for a general cycle and

$$\sum_{i \in \bar{S}} x_i \geq \frac{q+3}{2}\delta$$

for an odd hole.

An example with  $q = 5$  appears in Fig. 1. By setting  $s = 2$  we can obtain 9 valid inequalities by selecting 2-element subsets  $\bar{S}_2$  and  $\bar{S}_4$  of  $S_2$  and  $S_4$ , respectively. Here  $L = 5$ , and if the colors are  $0, \dots, 9$ , the right-hand side of the cut is  $\beta(5, 2) = 20$ . The sets  $\bar{S}_1, \dots, \bar{S}_5$  illustrated in the figure give rise to the valid inequality

$$x_0 + \dots + x_9 \geq 20 \quad (8)$$

## 6 Facet-defining Inequalities

We now show that the valid inequalities identified in Lemma 1 are facet-defining. Let the variables  $x_i$  for  $i \in \bar{S}$  be indexed  $x_0, \dots, x_{qs-1}$ . We will say that a partial solution

$$(x_0, x_1, \dots, x_{qs-1}) = (\bar{x}_0, \bar{x}_1, \dots, \bar{x}_{qs-1}) \quad (9)$$

is feasible for (1) if it can be extended to a feasible solution of (1). That is, there is a complete solution  $(x_1, \dots, x_n)$  that is feasible in (1) and that satisfies (9). Because  $|V|$  colors are available, any partial solution (9) that satisfies (4) can be extended to a feasible solution simply by assigning the remaining vertices distinct unused colors. That is, assign vertices in  $V \setminus \{0, \dots, sq-1\}$  distinct colors from the set  $J \setminus \{\bar{x}_0, \dots, \bar{x}_{qs-1}\}$ .

**Theorem 1.** *If the graph coloring problem (1) is defined on a graph in which vertex sets  $V_1, \dots, V_q$  induce a cycle, where  $q$  is odd, then inequality (5) is facet defining for (1).*

*Proof.* Define

$$F = \{x \text{ feasible for (1)} \mid (x_0, \dots, x_{qs-1}) \text{ satisfies (5) at equality}\}$$

It suffices to show that if  $\mu x \geq \mu_{n+1}$  holds for all  $x \in F$ , then there is a scalar  $\lambda > 0$  such that

$$\mu_i = \begin{cases} \lambda & \text{for } i = 0, \dots, qs - 1 \\ \beta(q, s)\lambda & \text{for } i = n + 1 \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

We will construct a partial solution  $(\bar{x}_0, \dots, \bar{x}_{qs-1})$  that is feasible for (1) as follows. Domain values  $v_1, \dots, v_{L-1}$  will occur  $(q-1)/2$  times in the solution, and domain value  $v_L$  will occur  $r$  times, where

$$r = qs - \frac{q-1}{2}(L-1)$$

This will ensure that (5) is satisfied at equality. We form the partial solution by first cycling  $r$  times through the values  $v_1, \dots, v_L$ , and then by cycling through the values  $v_1, \dots, v_{L-1}$ . Thus

$$\bar{x}_i = \begin{cases} v_{(i \bmod L)+1} & \text{for } i = 0, \dots, rL - 1 \\ v_{((i-rL) \bmod (L-1))+1} & \text{for } i = rL, \dots, rs - 1 \end{cases} \quad (11)$$

To show that this partial solution is feasible for the odd cycle, we must show

$$\begin{aligned} \text{alldiff}\{\bar{x}_i, i \in \bar{S}_k \cup \bar{S}_{k+1}\}, \text{ for } k = 1, \dots, q-1 & \quad (a) \\ \text{alldiff}\{\bar{x}_i, i \in \bar{S}_1 \cup \bar{S}_q\} & \quad (b) \end{aligned}$$

To show (a), we note that the definition of  $L$  implies  $L-1 \geq 2s$ . Therefore, any sequence of  $2s$  consecutive  $\bar{x}_i$ 's are distinct, and (a) is satisfied. To show (b), we note that the number of values  $\bar{x}_{rL}, \dots, \bar{x}_{rs-1}$  is

$$(rs-1) - rL + 1 = (L-1) \left( \frac{q-1}{2}L - qs \right)$$

from the definition of  $r$ . Because the number of values is a multiple of  $L-1$ , the values  $\bar{x}_i$  for  $i \in \bar{S}_q$  are  $(\bar{x}_{(q-1)s}, \dots, \bar{x}_{qs-1}) = (v_{L-s}, \dots, v_{L-1})$ , and they are all distinct. The values  $\bar{x}_i$  for  $i \in \bar{S}_1$  are  $(\bar{x}_0, \dots, \bar{x}_{s-1}) = (v_1, \dots, v_s)$  and are all distinct. But  $L-1 \geq 2s$  implies  $L-s > s$ , and (b) follows.

We now construct a partial solution  $(\tilde{x}_0, \dots, \tilde{x}_{qs-1})$  from the partial solution in (11) by swapping any two values  $\bar{x}_\ell, \bar{x}_{\ell'}$  for  $\ell, \ell' \in \bar{S}_k \cup \bar{S}_{k+1}$ , for any  $k \in \{1, \dots, q-1\}$ . That is,

$$\tilde{x}_i = \begin{cases} \bar{x}_{\ell'} & \text{if } i = \ell \\ \bar{x}_\ell & \text{if } i = \ell' \\ \bar{x}_i & \text{otherwise} \end{cases} \quad (12)$$

Extend the partial solutions (11) and (12) to complete solutions  $\bar{x}$  and  $\tilde{x}$ , respectively, by assigning values with

$$\bar{x}_i = \tilde{x}_i \text{ for } i \notin \{0, \dots, qs-1\}$$

such that the values assigned to  $\bar{x}_i$  for  $i \notin \{0, \dots, qs - 1\}$  are all distinct and do not belong to  $\{v_1, \dots, v_L\}$ . Because  $\bar{x}$  and  $\tilde{x}$  are feasible and satisfy (5) at equality, they satisfy  $\mu x = \mu_{n+1}$ . Subtracting, we have  $\mu_\ell = \mu_{\ell'}$  for  $\ell, \ell' \in \bar{S}_k \cup \bar{S}_{k+1}$  for any pair  $\ell, \ell' \in \bar{S}_k \cup \bar{S}_{k+1}$  and any  $k \in \{1, \dots, q - 1\}$ . This implies

$$\mu_\ell = \mu_{\ell'} \text{ for any } \ell, \ell' \in \bar{S} \quad (13)$$

Define  $\bar{x}'$  by letting  $\bar{x}' = \bar{x}$  except that for an arbitrary  $\ell \notin \{0, \dots, qs - 1\}$ ,  $\bar{x}'_\ell$  is assigned a value that does not appear in the tuple  $\bar{x}$ . Since  $\bar{x}$  and  $\bar{x}'$  are feasible and satisfy (5) at equality, we have  $\mu \bar{x} = \mu \bar{x}'$ . This and  $\bar{x}_\ell \neq \bar{x}'_\ell$  imply

$$\mu_i = 0, \quad i \in V \setminus \{0, \dots, qs - 1\} \quad (14)$$

Finally, (13) implies that for some  $\lambda > 0$ ,

$$\mu_i = \lambda, \quad i = 0, \dots, qs - 1 \quad (15)$$

Because  $\mu \bar{x} = \mu_{n+1}$ , we have from (15) that  $\mu_{n+1} = \beta(q, s)\lambda$ . This, (14), and (15) imply (10).  $\square$

In the example of Fig. 1, suppose that the vertices in  $V_1, \dots, V_5$  induce a cycle of  $G$ . That is, all vertices in each  $V_k$  are connected by edges, and there are no other edges of  $G$  between vertices in  $V_1 \cup \dots \cup V_5$ . Then (8) is facet-defining for (1).

## 7 Bounds on the Objective Function

We can write a facet-defining inequality involving the objective function variable  $z$  if the domain of each  $x_i$  is  $D_\delta$  for  $\delta > 0$ . To do so we rely on the following:

**Theorem 2.** *If  $ax \geq \beta$  is facet-defining for a graph coloring problem (1) in which each  $x_i$  has domain  $D_\delta$  for  $\delta > 0$ , then*

$$aez \geq ax + \beta \quad (16)$$

*is also facet defining, where  $e = (1, \dots, 1)$ .*

*Proof.* To show that (16) is valid, note that for any  $x \in D_\delta^n$ ,  $z - x_i \in D_\delta$  for all  $i$ , where  $z = \max_i \{x_i\}$ . Because  $ax \geq \beta$  is valid for all  $x \in D_\delta^n$  and  $z - x_i \in D_\delta$ ,  $ax \geq \beta$  holds when  $z - x_i$  is substituted for each  $x_i$ . This implies (16) because  $z$  in (1) satisfies  $z \geq x_i$  for each  $i$ .

To show (16) is facet-defining, let

$$F = \{(z, x) \text{ feasible for (1)} \mid aez = ax + \beta\}$$

It suffices to show that if  $\mu_z z = \mu x + \mu_0$  is satisfied by all  $(z, x) \in F$ , then there is a  $\lambda > 0$  with

$$\begin{aligned} \mu_z &= \lambda a e \\ \mu &= \lambda a \\ \mu_0 &= \lambda \beta \end{aligned} \quad (17)$$

Let  $F' = \{x \text{ feasible for (1)} \mid ax = \beta\}$ .  $F'$  is nonempty because  $ax \geq \beta$  is facet defining.  $F$  is therefore nonempty, because for any  $x \in F'$ , we have  $(\bar{z}, \bar{x}) \in F$  where  $\bar{z} = \max_i \{x_i\}$  and  $\bar{x} = ze - x$ . But for any point  $(z, x) \in F$ , we also have  $(z + \delta, x + \delta e) \in F$ . So  $\mu_z z = \mu x + \mu_0$  and  $\mu_z(z + \delta) = \mu(x + \delta e) + \mu_0$ . Subtracting, we get  $\mu_z = \mu e$ . We now claim that any  $(ez - x) \in F'$  satisfies  $\mu(ez - x) = \mu_0$ . This is because  $(ez - x) \in F'$  implies  $(z, x) \in F$ , which implies  $\mu ez = \mu x + \mu_0$ , which implies  $\mu(ez - x) = \mu_0$ . But because  $ax \geq \beta$  is facet defining, there is a  $\lambda > 0$  for which  $\mu = \lambda a$  and  $\mu_0 = \lambda \beta$ . Because  $\mu_z = \mu e$ , this same  $\lambda$  satisfies (17).  $\square$

Inequality (5) and Theorem 2 imply

**Corollary 1.** *If the graph coloring problem (1) is defined on a graph in which vertex sets  $V_1, \dots, V_q$  induce a cycle, where  $q$  is odd and each  $x_i$  has domain  $D_\delta$  with  $\delta > 0$ , then*

$$z \geq \frac{1}{qs} \sum_{i \in \bar{S}} x_i + \gamma(q, s) \quad (18)$$

is facet defining for (1), where

$$\gamma(q, s) = \left(1 - \frac{q-1}{4qs}L\right) (L-1)\delta$$

We will refer to cuts of the form (18) as  $z$ -cuts. In the case of an odd hole ( $s = 1$ ), the  $z$ -cut is

$$z \geq \frac{1}{q} \sum_{i \in \bar{S}} x_i + \frac{q+3}{2q} \delta$$

In the example of Fig. 1, the  $z$ -cut is

$$z \geq \frac{1}{5} (x_0 + \dots + x_9) + 2 \quad (19)$$

## 8 Mapping to 0-1 Cuts

The 0-1 model for a coloring problem on a cycle has the following continuous relaxation:

$$\sum_{j \in J} y_{ij} = 1, \quad i = 1, \dots, q \quad (a)$$

$$\sum_{i \in V_k} y_{ij} \leq w_j, \quad j \in J, k = 1, \dots, q \quad (b) \quad (20)$$

$$0 \leq y_{ij}, w_j \leq 1, \quad \text{all } i, j \quad (c)$$

Because constraints (b) appear for each maximal clique, the relaxation implies all clique inequalities  $\sum_{i \in V_k} y_{ij} \leq 1$ . Nonetheless, we will see that two finite-domain cuts strengthen the relaxation more than the collection of all odd hole cuts.

To simplify discussion, let each  $x_i$  have domain  $D_1 = \{0, 1, \dots, n-1\}$ . Then valid cuts for (1) become valid 0-1 cuts for (2) after the substitution

$$x_i = \sum_{j=1}^{n-1} j y_{ij}$$

The  $x$ -cut (10) therefore maps into the cut

$$\sum_{i \in \bar{S}} \sum_{j=1}^{n-1} j y_{ij} \geq \left( sq - \frac{q-1}{4} L \right) (L-1) \quad (21)$$

To map the  $z$ -cut (18) into 0-1 space, we must note that the objective function in the finite domain model (1) is the largest color number  $z$ , while the objective function in the 0-1 model (2) is the number  $\sum_j w_j$  of colors. Because the colors are numbered  $0, \dots, n-1$ , the minimum  $z$  is one less than the minimum number of colors. We therefore replace  $z$  with  $\sum_j w_j - 1$  and obtain the 0-1 cut

$$\sum_{j=0}^{n-1} w_j - 1 \geq \frac{1}{q} \sum_{i \in \bar{S}} \sum_{j=1}^{n-1} j y_{ij} + \frac{q+3}{2q} \quad (22)$$

We will compare cuts (21)–(22) with classical odd hole cuts, which have the form

$$\sum_{i \in H} y_{ij} \leq \frac{q-1}{2} w_j, \quad j = 0, \dots, n-1 \quad (23)$$

where  $H$  is the vertex set for an odd hole.

We first note that when  $s = 1$ , the  $x$ -cut (21) is redundant of odd hole cuts.

**Lemma 2.** *If  $s = 1$ , the  $x$ -cut (21) is implied by the 0-1 model (20) with odd hole cuts (23).*

*Proof.* When  $s = 1$ , the cut (21) becomes

$$\sum_{i \in \bar{S}} \sum_{j=0}^{n-1} j y_{ij} \geq \frac{q+3}{2} \quad (24)$$

It suffices to show that (24) is dominated by a nonnegative linear combination of (20) and (23), where  $H = \bar{S}$  in (23). Assign multiplier 2 to each constraint in (20a); multipliers 2 and 1, respectively, to constraints (23) with  $j = 0, 1$ ; and multipliers  $q-1$  and  $(q-1)/2$ , respectively, to the constraints  $w_0 \leq 1$  and  $w_1 \leq 1$ . The resulting linear combination is

$$\sum_{i \in \bar{S}} y_{i1} + 2 \sum_{j=2}^{n-1} \sum_{i \in \bar{S}} y_{ij} \geq 2q - \frac{q-1}{2} - (q-1) = \frac{q+3}{2}$$

This dominates (24) because the left-hand side coefficients are less than or equal to the corresponding coefficients in (24).  $\square$

However, the two finite domain cuts (21) and (22), when combined, strictly dominate the  $n$  odd hole cuts (23) even when  $s = 1$ . For example, when  $q = 5$ , the 10 odd hole cuts provide a lower bound of 2.5 on the chromatic number, while the two finite domain cuts provide a bound of 2.6. The improvement is modest, but 10 cuts are replaced by only two cuts. Comparisons for larger  $q$  appear in the next section.

Furthermore, when  $s > 1$ , the single  $z$ -cut (22) strictly dominates the collection of all odd hole cuts, which have no effect in this case. There are  $s^q$  odd hole cuts (23) for each color  $j$ , one for every  $H$  that selects one element from each  $\bar{S}_k$ ,  $k = 1, \dots, q$ . For example, when  $q = 5$  and  $s = 2$ , there are  $ns^q = 320$  odd hole cuts. The lower bound on the chromatic number is 4.0 with or without them. However, the one finite domain cut (22) yields a bound of 4.5. Addition of the  $x$ -cut (21) strengthens the bound further, raising it to 5.0. This bound is actually sharp in the present instance, because the chromatic number is 5. Thus two finite domain cuts significantly improve the bound, while 320 odd hole cuts have no effect on the bound. Further comparisons appear below.

## 9 Computational Results

To measure the effect of various cuts on the quality of bounds, we generated a series of graph coloring instances. Each instance consists of a  $q$ -cycle formulated as a 0-1 programming model and is parameterized by  $s$  and  $q$ . The overlap sets  $S_k$  have size  $s$ , and vertex set  $V_k = S_k \cup S_{k+1}$  for  $k = 1, \dots, q - 1$  (with  $V_k = S_k \cup S_1$ ).

For each instance, we solved the linear programming relaxation that minimizes  $\sum_j w_j$  subject to (20) and various classes of cuts:

- No cuts, aside from the clique inequalities (20b), which are always present.
- Odd hole cuts (23) only. As explained in Section 8, there are  $ns^q$  such cuts.
- The  $x$ -cut (21) only, with  $\bar{S}_k = S_k$ .
- The  $z$ -cut (22) only, with  $\bar{S}_k = S_k$ .
- the  $x$ -cut and the  $z$ -cut.

We generated the instances indicated in Table 1, which shows the resulting bounds, the optimal value (chromatic number), and the number of odd hole cuts.

For  $s = 1$ , the table confirms that  $x$ -cuts are redundant of odd hole cuts. However, the combination of one  $x$ -cut and one  $z$ -cut yields a tighter bound than  $n$  odd hole cuts. It is therefore advantageous to replace any set of standard cuts generated for an odd hole with these two cuts.

For  $s > 1$ , neither odd hole cuts nor  $x$ -cuts alone have any effect on the bound when clique inequalities are present. However, a single  $z$ -cut significantly improves the bound. Combining the  $z$ -cut with the  $x$ -cut raises the bound still further, substantially reducing the integrality gap, sometimes to zero. Two finite domain cuts therefore provide a much tighter relaxation than a large set of standard clique inequalities and odd hole cuts.

**Table 1.** Lower bounds on the chromatic number in a 0-1 clique formulation of problem instances consisting of one  $q$ -cycle with overlap of  $s$ .

$q$	$s$	Without cuts	Odd hole cuts only	$x$ -cut only	$z$ -cut only	$x$ -cut & $z$ -cut	Optimal value	No. of odd hole cuts
5	1	2.00	2.50	2.00	2.30	2.60	3	5
	2	4.00	4.00	4.00	4.50	5.00	5	320
	3	6.00	6.00	6.00	6.77	7.53	8	3645
	4	8.00	8.00	8.00	9.00	10.00	10	20,480
	5	10.00	10.00	10.00	11.26	12.52	13	78,125
7	1	2.00	2.33	2.00	2.21	2.43	3	7
	2	4.00	4.00	4.00	4.36	4.71	5	1792
	3	6.00	6.00	6.00	6.50	7.00	7	45,927
	4	8.00	8.00	8.00	8.68	9.36	10	458,752
9	1	2.00	2.25	2.00	2.17	2.33	3	9
	2	4.00	4.00	4.00	4.28	4.56	5	9216
	3	6.00	6.00	6.00	6.39	6.78	7	531,441

An interesting question is whether the finite domain cuts are equally effective in the  $x$ -space, where they take their original form (5) and (18). We formulated a linear relaxation of the CP model that minimizes  $z + 1$  subject to  $z \geq x_i$  for all  $i$ , plus cuts. The results appear in Table 2. The  $x$ -cut and  $z$ -cut, in isolation, yield weaker bounds than in the 0-1 model. However, when combined, the two cuts yield exactly the same bound as in the 0-1 model.

One might obtain a fairer comparison if clique inequalities are added to the CP model, because they appear in the 0-1 model. In the CP model, clique inequalities correspond to the individual alldiff constraints. We know from [5, 14] that for domain  $D_1$ , the following is facet-defining for  $\text{alldiff}(X_k)$ :

$$\sum_{i \in V_k} x_i \geq \frac{1}{2}|V_k|(|V_k| - 1)$$

In the test instances,  $|V_k| = 2s$ . We therefore added the following cuts:

$$\sum_{i \in V_k} x_i \geq s(2s - 1), \quad k = 1, \dots, q$$

Using Theorem 2, we also added the cuts:

$$z \geq \frac{1}{qs} \sum_{i \in V_k} x_i + \frac{2s - 1}{q}, \quad k = 1, \dots, q$$

The results appear in the last four columns of Table 2. The  $x$ -cut performs as before, but now the  $z$ -cut provides the same bound as in the 0-1 model. When combined, the  $x$ -cut and  $y$ -cut again deliver the same bound as in the 0-1 model.

It is remarkable that two odd cycle cuts yield the same bound in the very small finite domain relaxation (even without clique inequalities) as in the much

**Table 2.** Lower bounds on the chromatic number in the CP model of problem instances consisting of one  $q$ -cycle with overlap of  $s$  and color set  $\{0, 1, \dots, n-1\}$ .

$q$	$s$	No cuts	$x$ -cut only	$z$ -cut only	$x$ -cut & $z$ -cut	Clique cuts	Plus $x$ -cut	Plus $z$ -cut	Plus $x$ -cut & $z$ -cut
5	1	1.00	1.80	1.80	2.60	1.50	1.80	2.30	2.60
	2	1.00	3.00	3.00	5.00	2.50	3.00	4.50	5.00
	3	1.00	4.27	4.27	7.53	3.50	4.27	6.77	7.53
	4	1.00	5.50	5.50	10.00	4.50	5.50	9.00	10.00
	5	1.00	6.76	6.76	12.52	5.50	6.76	11.26	12.52
7	1	1.00	1.71	1.71	2.43	1.50	1.71	2.21	2.43
	2	1.00	2.86	2.86	4.71	2.50	2.86	4.36	4.71
	3	1.00	4.00	4.00	7.00	3.50	4.00	6.50	7.00
	4	1.00	5.18	5.18	9.36	4.50	5.18	8.68	9.36
9	1	1.00	1.67	1.67	2.33	1.50	1.67	2.17	2.33
	2	1.00	2.78	2.78	4.56	2.50	2.78	4.28	4.56
	3	1.00	3.89	3.89	6.78	3.50	3.89	6.39	6.78

larger 0-1 relaxation. The finite domain relaxation contains  $n$  variables  $x_i$  and  $n+2$  constraints, while the 0-1 relaxation contains  $n^2+n$  variables  $y_{ij}, w_j$  and  $n^2+n+2$  constraints (dropping odd hole cuts). If finite domain counterparts can be identified for other families of 0-1 cuts, it may be advantageous to obtain bounds from a finite domain model rather than a 0-1 model.

## 10 Separation Algorithm

Separating cuts can be identified in either the  $x$ -space or the  $y$ -space. When a continuous relaxation of the 0-1 model is solved, the resulting values of the  $y_{ij}$ s can be used to identify a separating cut directly in 0-1 space. Alternatively, these values can be mapped to values of the  $x_j$ s using the transformation (3), and a separation algorithm applied in  $x$ -space.

In practice, a solver may apply existing algorithms to identify separating odd hole cuts. The odd holes that give rise to these cuts can trigger the generation of an  $x$ -cut and a  $z$ -cut. These superior cuts can then replace the odd hole cuts.

If odd cycle cuts for  $s > 1$  are desired, a separation algorithm can be applied to the  $x_i$ -values by heuristically seeking a cycle that gives rise to separating cuts. We show here that a simple polynomial-time algorithm identifies a separating  $x$ -cut and a separating  $z$ -cut for a given cycle if such cuts exist.

The algorithm is as follows. We again suppose the colors are  $0, 1, \dots, n-1$ . Let (4) be an odd  $q$ -cycle for which we wish to find a separating cut. Let  $\bar{y}, \bar{w}$  be a solution of the continuous relaxation of the 0-1 model, and let

$$\bar{x}_i = \sum_{j=1}^{n-1} j\bar{y}_{ij}, \quad i \in \bigcup_{k=1}^q V_k \quad \bar{z} = \sum_{j=0}^{n-1} \bar{w}_j - 1$$

For each  $k = 1, \dots, q$ , define the bijection  $\pi_k : \{1, \dots, |S_k|\} \rightarrow S_k$  such that  $\bar{x}_{\pi_k(\ell)} \leq \bar{x}_{\pi_k(\ell')}$  whenever  $\ell < \ell'$ . Then for  $s = 1, \dots, \min_k |S_k|$ , generate a separating  $x$ -cut

$$\sum_{k=1}^q \sum_{\ell=1}^s x_{\pi_k(\ell)} \geq \beta(q, s) \quad (25)$$

whenever

$$\sum_{k=1}^q \sum_{j=\ell}^s \bar{x}_{\pi_k(\ell)} < \beta(q, s)$$

and generate a separating  $z$ -cut

$$z \geq \frac{1}{qs} \sum_{k=1}^q \sum_{\ell=1}^s x_{\pi_k(|S_k|-\ell+1)} + \gamma(q, s) \quad (26)$$

whenever

$$\bar{z} < \frac{1}{qs} \sum_{k=1}^q \sum_{\ell=1}^s \bar{x}_{\pi_k(|S_k|-\ell+1)} + \gamma(q, s)$$

The running time of the algorithm is  $\mathcal{O}(q\bar{s} \log \bar{s})$ , where  $\bar{s} = \max_k |S_k|$  and  $\bar{s} \log \bar{s}$  is the sort time for  $\bar{s}$  values.

**Lemma 3.** *The above algorithm finds a separating  $x$ -cut for a given odd  $q$ -cycle if such a cut exists.*

*Proof.* Suppose there is a separating  $x$ -cut with  $\bar{S}_k \subset S_k$  and  $s^* = |\bar{S}_k|$  for  $k = 1, \dots, q$ . Then

$$\sum_{i \in \bar{S}} \bar{x}_i < \beta(q, s^*) \quad (27)$$

where  $\bar{S} = \bigcup_k \bar{S}_k$ . Because  $\pi_k$  orders the elements of  $S_k$  by size,

$$\sum_{\ell=1}^{s^*} \bar{x}_{\pi_k(\ell)} \leq \sum_{i \in \bar{S}_k} \bar{x}_i, \quad k = 1, \dots, q$$

Summing this over  $k = 1, \dots, q$ , we get

$$\sum_{k=1}^q \sum_{\ell=1}^{s^*} \bar{x}_{\pi_k(\ell)} \leq \sum_{i \in \bar{S}} \bar{x}_i < \beta(q, s^*)$$

where the strict inequality is due to (27). This means that the algorithm generates the separating cut for  $s = s^*$ .  $\square$

**Lemma 4.** *The above algorithm finds a separating  $z$ -cut for a given odd  $q$ -cycle if such a cut exists.*

*Proof.* Suppose there is a separating  $z$ -cut with  $\bar{S}_k \subset S_k$  and  $s^* = |\bar{S}_k|$  for  $k = 1, \dots, q$ . Then

$$\bar{z} < \sum_{i \in \bar{S}} \bar{x}_i + \gamma(q, s^*) \quad (28)$$

where  $\bar{S} = \bigcup_k \bar{S}_k$ . Because  $\pi_k$  orders the elements of  $S_k$  by size,

$$\sum_{\ell=1}^{s^*} \bar{x}_{\pi_k(|S_k|-\ell+1)} \geq \sum_{i \in \bar{S}_k} \bar{x}_i, \quad k = 1, \dots, q$$

Summing this over  $k = 1, \dots, q$ , we get

$$\sum_{k=1}^q \sum_{\ell=1}^{s^*} \bar{x}_{\pi_k(\ell)} \leq \sum_{i \in \bar{S}} \bar{x}_i$$

This and (28) imply

$$\bar{z} < \frac{1}{qs} \sum_{k=1}^q \sum_{\ell=1}^{s^*} \bar{x}_{\pi_k(|S_k|-\ell+1)} + \gamma(q, s^*)$$

The algorithm therefore generates the separating cut for  $s = s^*$ .  $\square$

## 11 Conclusions and Future Research

We showed that polyhedral analysis of all-different systems can give rise to a class of finite domain cuts (odd cycle cuts) that strictly dominate known 0-1 cuts for cyclic structures in graph coloring problems. Two odd cycle cuts yield a tighter bound than a large number of standard cuts. In particular, separating odd hole cuts identified by existing solvers can be replaced by stronger odd cycle cuts with no additional calculation. The alternate polyhedral perspective afforded by the finite domain formulation therefore seems beneficial, at least in the case of graph coloring.

The next step is to seek additional families of finite domain cuts for graph coloring, perhaps corresponding to odd anti-holes, paths, odd wheels, and more general structures. Once a variety of finite domain cuts have been identified, separation heuristics can be developed as needed and tested in the context of a solution method that uses polyhedral bounds.

We also discovered that odd cycle cuts provide the same tight bound in a relaxation of the finite domain model as in a relaxation of the 0-1 model. If other families of finite domain cuts follow this pattern, there could be advantage in obtaining bounds from a finite domain relaxation that is much smaller than the 0-1 model. Given that some benchmark instances result in 0-1 models that are too large even to load into a linear solver [12], this could provide a viable alternative for solving large graph coloring and related problems.

Finally, the general strategy of obtaining valid inequalities and tight bounds from finite domain formulations can be investigated for other problem classes.

## References

1. G. Appa, D. Magos, and I. Mourtos. Linear programming relaxations of multiple all-different predicates. In J. C. Régin and M. Rueher, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR 2004)*, volume 3011 of *Lecture Notes in Computer Science*, pages 364–369. Springer, 2004.
2. G. Appa, D. Magos, and I. Mourtos. On the system of two all-different predicates. *Information Processing Letters*, 94:99–105, 2004.
3. P. Coll, J. Marenco, I. Méndez-Díaz, and P. Zabala. Facets of the graph coloring polytope. *Annals of Operations Research*, 116:79–90, 2002.
4. L. Genç-Kaya and J. N. Hooker. The circuit polytope. Manuscript, Carnegie Mellon University, 2010.
5. J. N. Hooker. *Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction*. Wiley, New York, 2000.
6. J. N. Hooker. *Integrated Methods for Optimization*. Springer, 2007.
7. S. Kruk and S. Toma. On the system of the multiple all different predicates. *Congressus Numerantium*, 197:47–64, 2009.
8. S. Kruk and S. Toma. On the facets of the multiple alldifferent constraint. *Congressus Numerantium*, 204:5–32, 2010.
9. D. Magos and I. Mourtos. On the facial structure of the alldifferent system. *SIAM Journal on Discrete Mathematics*, pages 130–158, 2011.
10. D. Magos, I. Mourtos, and G. Appa. A polyhedral approach to the *alldifferent* system. *Mathematical Programming*, to appear.
11. I. Méndez-Díaz and P. Zabala. A polyhedral approach for graph coloring. *Electronic Notes in Discrete Mathematics*, 7:178–181, 2001.
12. I. Méndez-Díaz and P. Zabala. A cutting plane algorithm for graph coloring. *Discrete Applied Mathematics*, 156:159–179, 2008.
13. G. Palubeckis. On the graph coloring polytope. *Information Technology and Control*, 37:7–11, 2008.
14. H. P. Williams and H. Yan. Representations of the all\_different predicate of constraint satisfaction in integer programming. *INFORMS Journal on Computing*, 13:96–103, 2001.
15. H. Yan and J. N. Hooker. Tight representations of logical constraints as cardinality rules. *Mathematical Programming*, 85:363–377, 1995.
16. T. H. Yunes. On the sum constraint: Relaxation and applications. In P. Van Hentenryck, editor, *Principles and Practice of Constraint Programming (CP 2002)*, volume 2470 of *Lecture Notes in Computer Science*, pages 80–92. Springer, 2002.