Graph Coloring Facets from a Constraint Programming Formulation

David Bergman
J. N. Hooker

Carnegie Mellon University

INFORMS 2011

Motivation

- **0-1** variables often encode choices that can be represented with **finite domain** variables.
  - x_i = finite domain variable
    - Job assigned to worker *i*
    - Start time of job *i*
    - City visited after city *i*
    - Number of packages on truck *i*
  - y_{ij} = corresponding 0-1 variable
    - $y_{ij} = 1$ if $x_i = j$

Motivation

- A **constraint programming** formulation often uses **finite-domain** variables.
  - If the variables are numeric, the problem has **polyhedral structure** very different from the 0-1 problem.
  - **Finite-domain cuts** can be mapped into the 0-1 model.
  - This may yield **stronger cuts** in the 0-1 model.

Motivation

- A **constraint programming** formulation often uses **finite-domain** variables.
  - If the variables are numeric, the problem has **polyhedral structure** very different from the 0-1 problem.
  - **Finite-domain cuts** can be mapped into the 0-1 model.
  - This may yield **stronger cuts** in the 0-1 model.

- We apply this idea to **graph coloring**.
  - May apply to other problems with both 0-1 and CP formulations.

Motivation

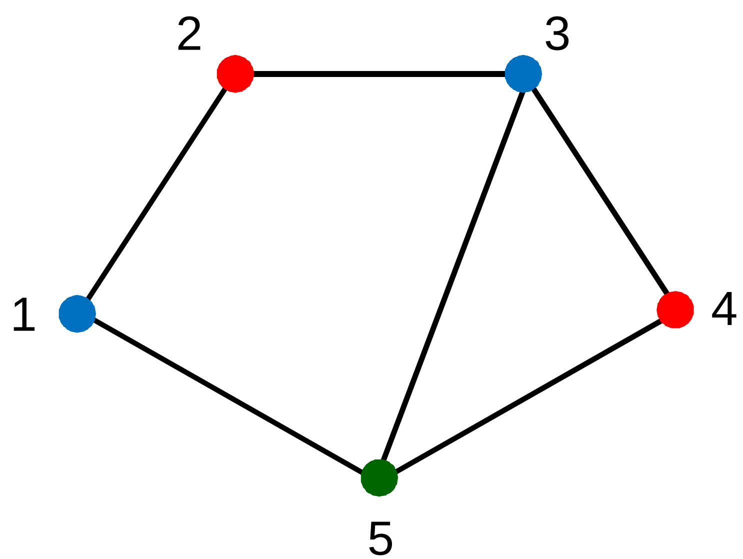
- We obtain two kinds of results:
  - If you find a structure (e.g., odd hole) that yields a known valid inequality in 0-1 space…
    - We will give you a stronger cut for **free**.
    - Use whatever separation algorithm you want.

Motivation

- We obtain two kinds of results:
  - If you find a structure (e.g., odd hole) that yields a known valid inequality in 0-1 space…
    - We will give you a stronger cut for **free**.
    - Use whatever separation algorithm you want.
  - We identify **additional** structures that yield valid inequalities.
    - They are **stronger** than **known cuts**.
    - We have separation algorithms.

Graph Coloring

- We focus on the **vertex coloring** problem.
  - Given a graph, assign colors to vertices so that no two adjacent vertices receive the same color.
  - Minimize the number of colors.



Graph Coloring

- 0-1 model

$$\min \sum_j w_j$$

= 1 if color j is used

$$\sum_j y_{ij} = 1, \quad \text{all vertices } i$$

$$y_{1j} + y_{2j} \le w_j, \quad \text{all colors } j$$

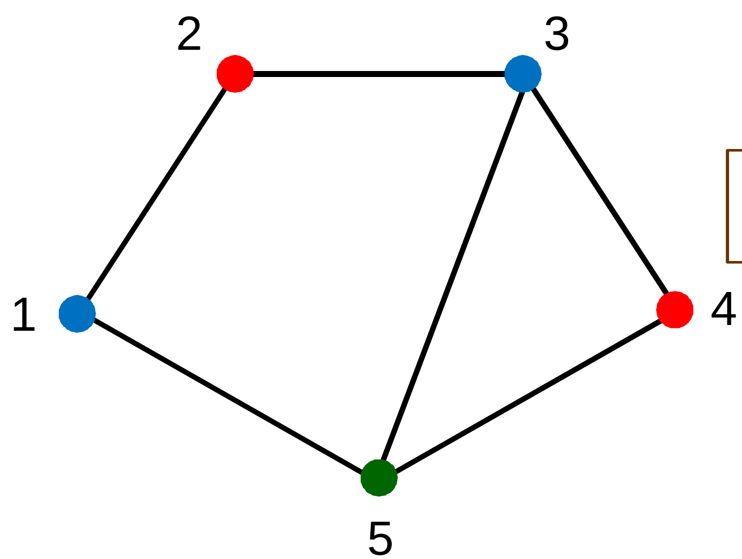
$$y_{1j} + y_{5j} \le w_j, \quad \text{all colors } j$$

$$y_{2j} + y_{3j} \le w_j, \quad \text{all colors } j$$

$$y_{3j} + y_{4j} + y_{5j} \le w_j, \quad \text{all colors } j$$

$$y_{ij} \in \{0,1\}$$

= 1 if vertex i receives color j



Graph Coloring

- General model:

= 1 if color j is used

$$\min \sum_j w_j$$

$$\sum_j y_{ij} = 1, \quad \text{all vertices } i$$

$$\sum_{i \in V_k} y_{ij} \le w_j, \quad \text{all colors } j\text{, cliques } V_k \text{ that cover vertices}$$

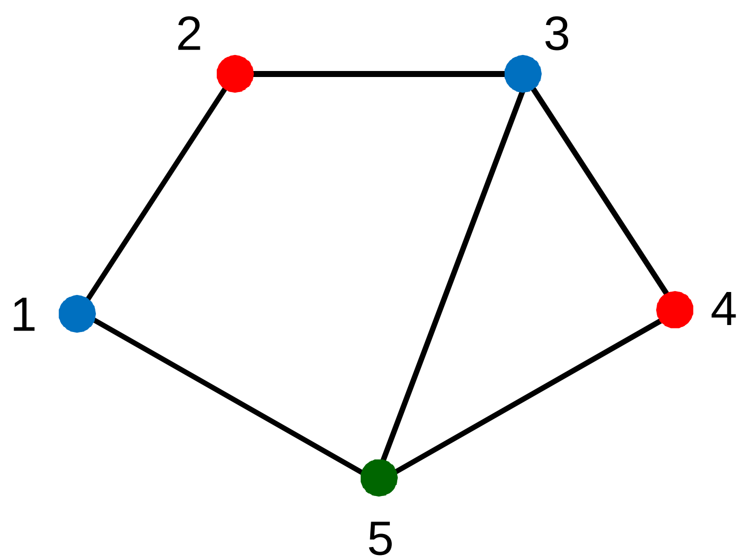
$$y_{ij} \in \{0,1\}$$

= 1 if vertex i receives color j

$O(n^2)$ variables
$O(n^3)$ constraints

Alldiff Systems

- Use an **all-different** constraint for each clique.



$$
\begin{aligned}
&\min\ z \\
&z \ge x_i, \quad \text{all vertices } i \\
&\text{alldiff}(x_1, x_2), \quad \text{all colors } j \\
&\text{alldiff}(x_1, x_5), \quad \text{all colors } j \\
&\text{alldiff}(x_2, x_3), \quad \text{all colors } j \\
&\text{alldiff}(x_3, x_4, x_5), \quad \text{all colors } j \\
&x_i \in \{1, \ldots, 5\}
\end{aligned}
$$

x_i = color assigned to vertex i

Alldiff Systems

- General model:

$$\min z$$

$$z \geq x_i, \quad \text{all vertices } i$$

$$\text{alldiff}\left(x_i \mid i \in V_k\right), \text{ all cliques } V_k$$

$$x_i \in \{1, \ldots, n\}$$

= color assigned to vertex *i*

$O(n)$ variables
$O(n^2)$ constraints

Objective reduces **symmetry**

Alldiff Systems

- Applications:
  - Scheduling, timetabling.
    - Employee scheduling.
    - Course timetabling.
  - Latin squares.
    - Alldiff for each row, column.
    - Experimental design: orthogonal Latin squares.
    - Sudoku puzzles.
  - Graph coloring.
    - Many applications.

Related Work

- Convex hull of single alldiff.
  - Hooker (2000), Williams and Yan (2001).
- Convex hull of 2 alldiffs.
  - Appa, Magos and Mourtos (2004)
- Convex hull of alldiff systems with inclusion property.
  - Appa, Magos and Mourtos (2011).
  - Same facets as individual alldiffs.
- Some facets of systems without inclusion property.
  - Magos and Mourtos (2011).

Variable Mapping

- There is a linear mapping from x_i to y_{ij}:

$$x_i = \sum_j j y_{ij}$$

- Any valid linear inequality in x_i-space maps to a valid linear inequality in y_{ij}-space.
  - Just substitute above expression for x_i.
  - Convert any finite domain cut to a 0-1 cut.

Variable Mapping

- There is a linear mapping from x_i to y_{ij}:

$$x_i = \sum_j j y_{ij}$$

- Any valid linear inequality in x_i-space maps to a valid linear inequality in y_{ij}-space.
  - Just substitute above expression for x_i.
  - Convert any finite domain cut to a 0-1 cut.
- Objective function more likely to be linear in y-space.
  - For coloring, it is linear in both x-space and y-space.

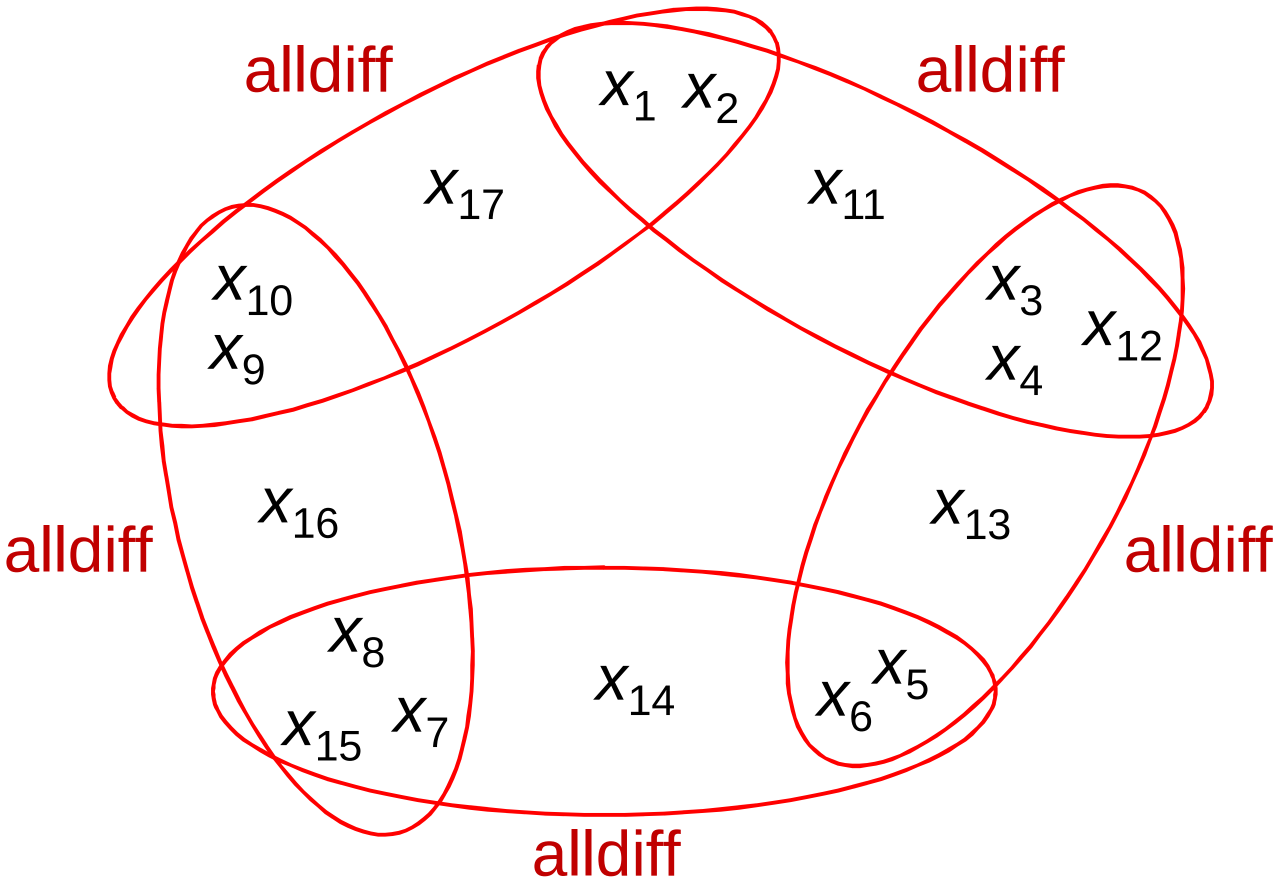
Choice of Domain

- We will assume each x_i has domain $\{0, \ldots, n - 1\}$.
  - To simplify exposition.

- Most results to follow can be generalized to an arbitrary numeric domain $\{v_1, \ldots, v_n\}$ with each $v_i \geq 0$.
  - Some results are valid for domain $D = \{0, \delta, \ldots, (n - 1)\delta\}$ with $\delta > 0$.

Odd Cycles

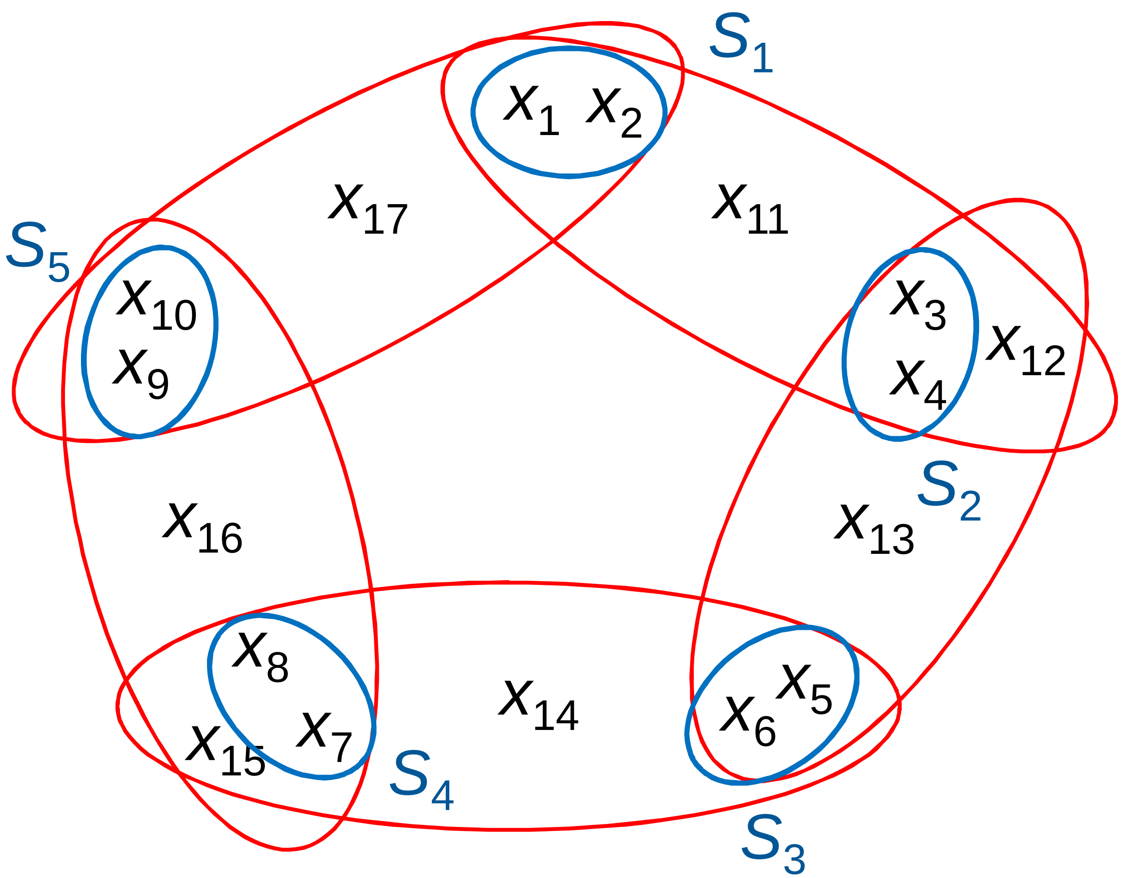
- A *q*-cycle consists of *q* alldiff constraints that look like this:



Odd Cycles

- Select any subset of s vertices in each overlap:

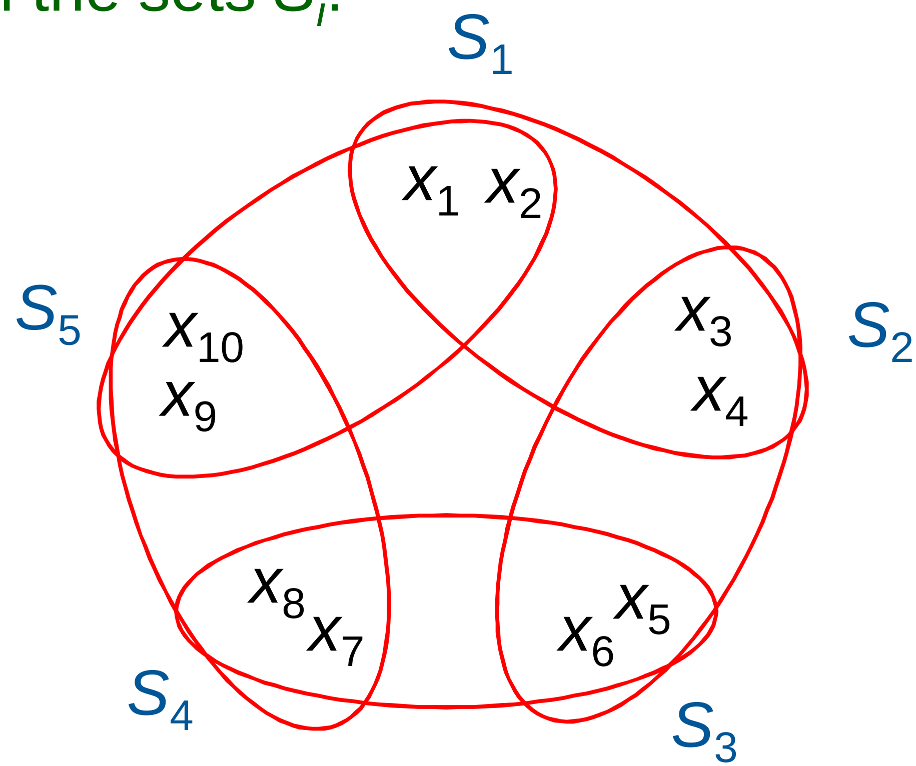
$s = 2$



Odd Cycles

- Focus on the sets S_i:

$s = 2$
$q = 5$



sq = 10 vertices
Each color can be assigned to at most $(q – 1)/2$ = 2 vertices.

We need at least $L = \left\lceil \frac{sq}{(q-1)/2} \right\rceil = 5$ colors

Odd Cycles

- Focus on the sets S_i:

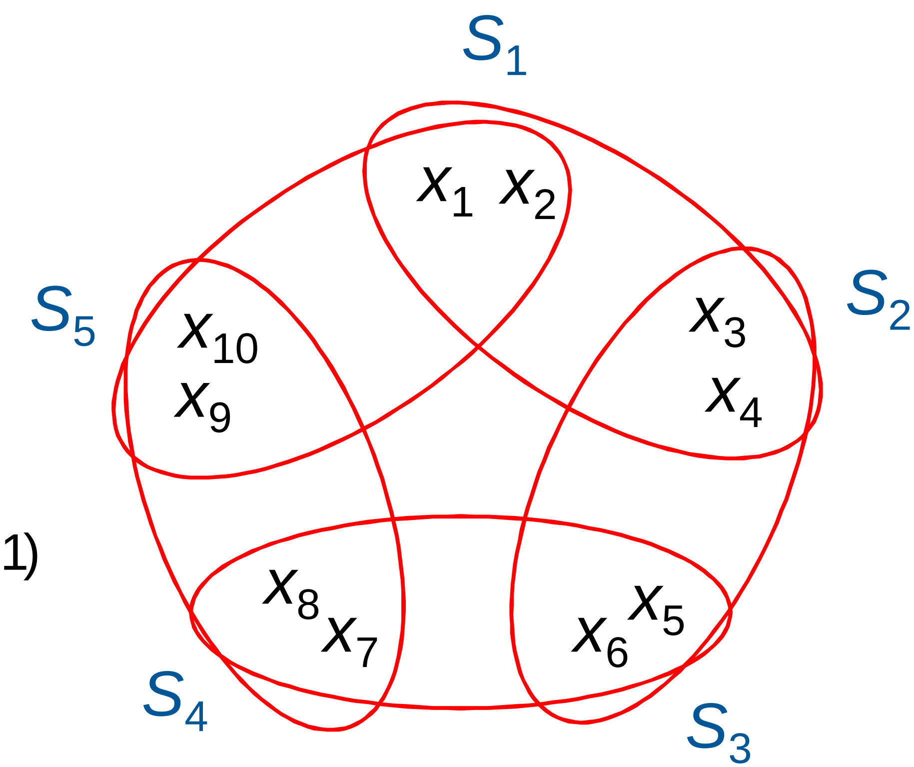
$s = 2$

$q = 5$ $\quad S = \bigcup_k S_k$

So

$$\sum_{i \in S} x_i \geq \frac{q-1}{2} \cdot 0 + \frac{q-1}{2} \cdot 1 + \cdots$$

$$\cdots + \frac{q-1}{2}(L-2) + \left(sq - \frac{q-1}{2}(L-1) \right)(L-1)$$



sq = 10 vertices
Each color can be assigned to at most $(q – 1)/2$ = 2 vertices.

We need at least $L = \left\lceil \frac{sq}{(q-1)/2} \right\rceil = 5$ colors

Odd Cycles

- Focus on the sets S_i:

$s = 2$

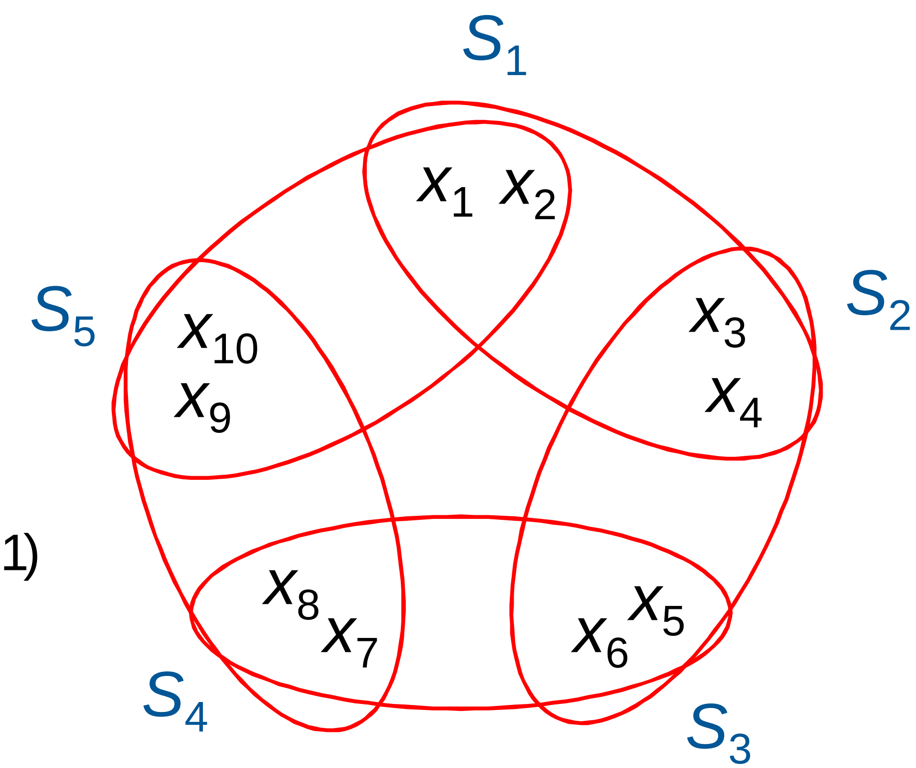
$q = 5 \qquad S = \bigcup_k S_k$

So

$$\sum_{i \in S} x_i \geq \frac{q-1}{2} \cdot 0 + \frac{q-1}{2} \cdot 1 + \cdots$$

$$\cdots + \frac{q-1}{2}(L-2) + \left(sq - \frac{q-1}{2}(L-1) \right)(L-1)$$

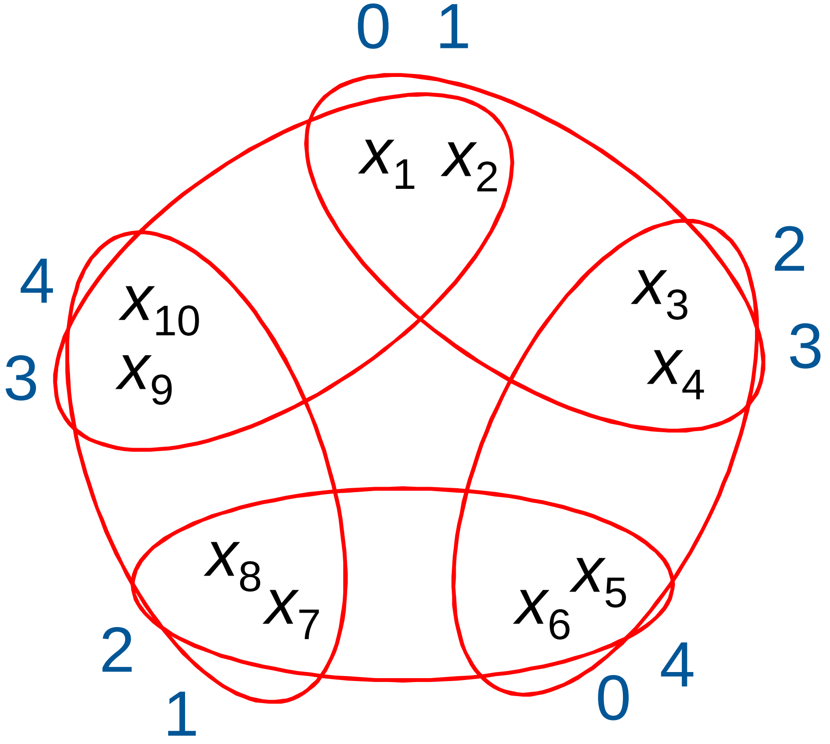
$$= \left(sq - \frac{q-1}{4} L \right)(L-1) = 20$$



Odd Cycles

- So we have a valid inequality:

$$\sum_{i \in S} x_i \geq \left(sq - \frac{q-1}{4} L \right)(L-1) = 20$$

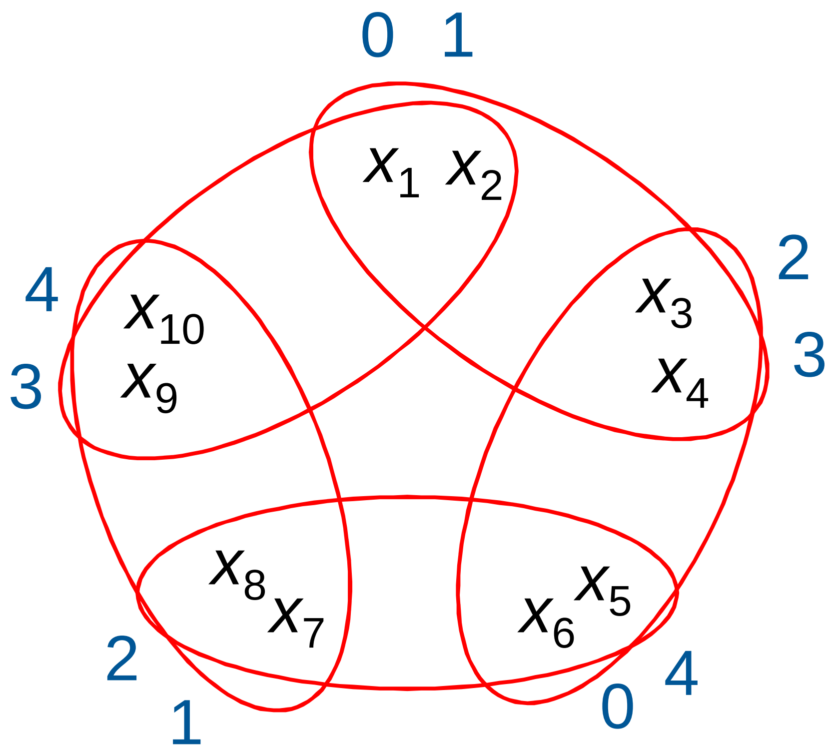


Odd Cycles

- So we have a valid inequality:

$$\sum_{i \in S} x_i \geq \left(sq - \frac{q-1}{4} L \right)(L-1) = 20$$

- The inequality is **facet-defining** if *q* is odd.
  - and if the *q*-cycle is the subgraph induced by vertices in the cycle.



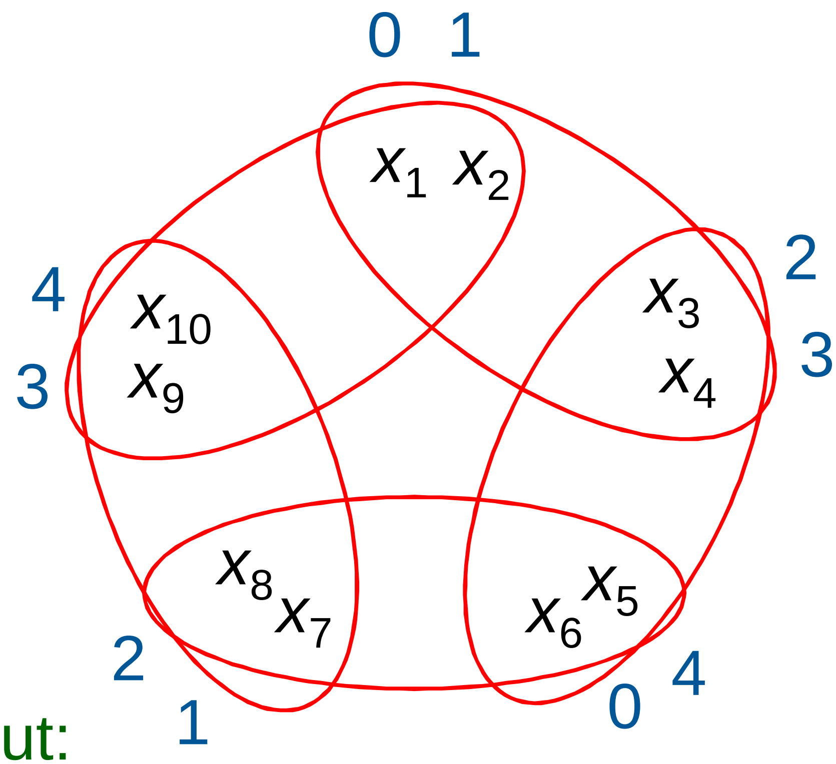
Odd Cycles

- So we have a valid inequality:

$$\sum_{i \in S} x_i \geq \left(sq - \frac{q-1}{4} L \right)(L-1) = 20$$

- The inequality is **facet-defining** if *q* is odd.
  - and if the *q*-cycle is the subgraph induced by vertices in the cycle.
- For *s* = 1, we have odd hole cut:

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



Odd Cycles

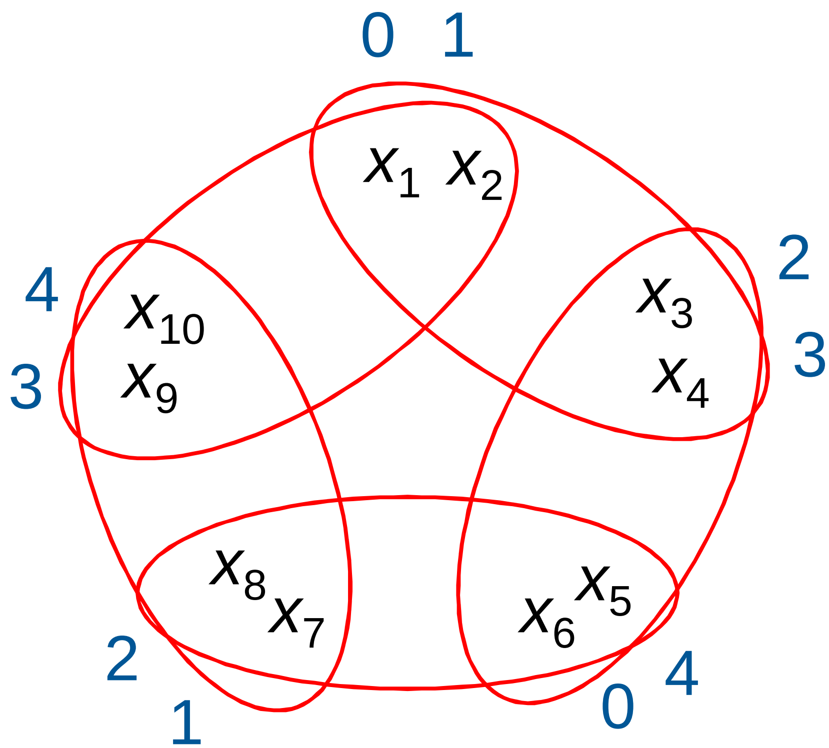
- So we have a valid inequality:

$$\sum_{i\in S} x_i \geq \left(sq - \frac{q-1}{4} L \right)(L-1) = 20$$

- We can obtain a valid bound on number of colors z by substituting $z - x_i$ for x_i:

$$z \geq \frac{1}{qs} \sum_{i\in S} x_i + \left(1 - \frac{q-1}{4qs} L \right)(L-1)$$

$$= \frac{1}{10} \sum_{i\in S} x_i + 2$$



This is **facet defining** for domain *D*.

z-cuts in general

- In fact, facet-defining **x**-cuts for a graph coloring problem always give rise to facet-defining **z**-cuts:

  – **Theorem:** if $ax \geq b$ is facet defining for a coloring problem with domain $D = \{0, \delta, 2\delta, \ldots, (n-1)\delta\}$ for $\delta > 0$, then $aez \geq ax + b$ is also facet defining, where $e = (1, \ldots, 1)$.

Mapping into 0-1 Space

- The **x-cut**

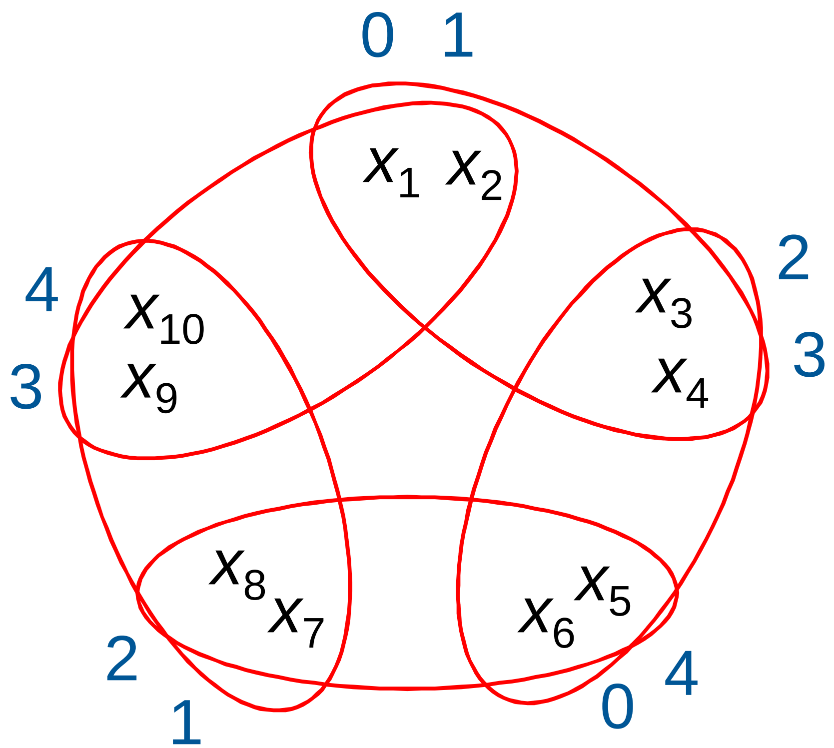
$$\sum_{i \in S} x_i \geq \left(sq - \frac{q-1}{4} L \right)(L-1) = 20$$

and the **z-cut**

$$z \geq \frac{1}{qs} \sum_{i \in S} x_i + \left(1 - \frac{q-1}{4qs} L \right)(L-1)$$

map into a 0-1 cut
by replacing x_i with $\sum_j j y_{ij}$

– How do they compare with classical odd hole cuts?

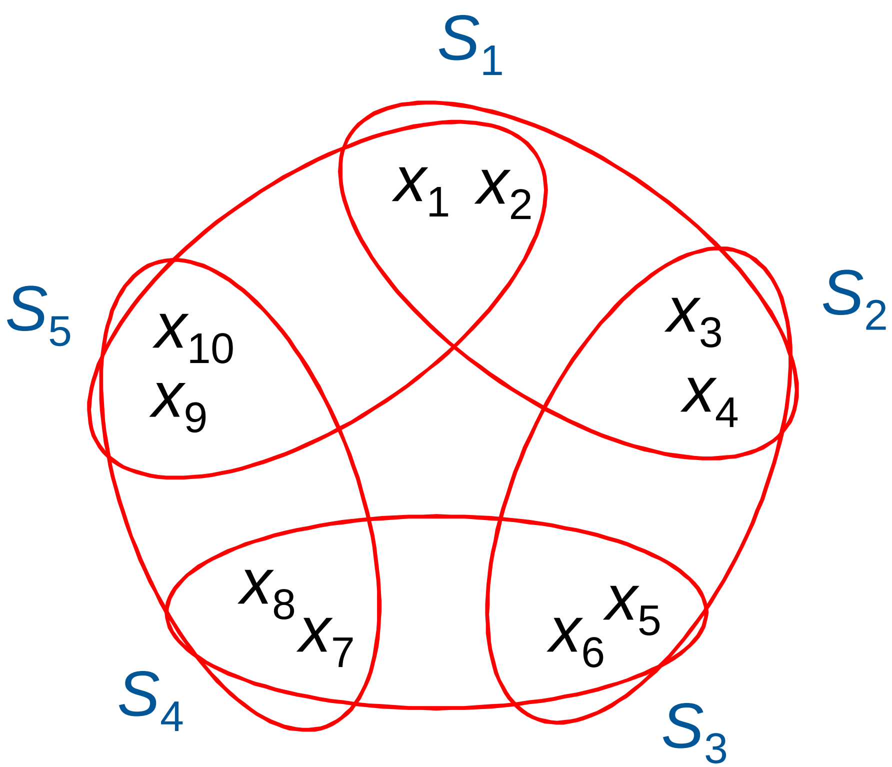


Comparison with Odd Hole Cuts

- A q-cycle defines $s^q = 32$ odd hole cuts for each color:

$$\sum_{i \in T} y_{ij} \leq \frac{q-1}{2} w_j, \text{ all } T, j$$

  – where T selects one vertex from each S_k



Comparison with Odd Hole Cuts

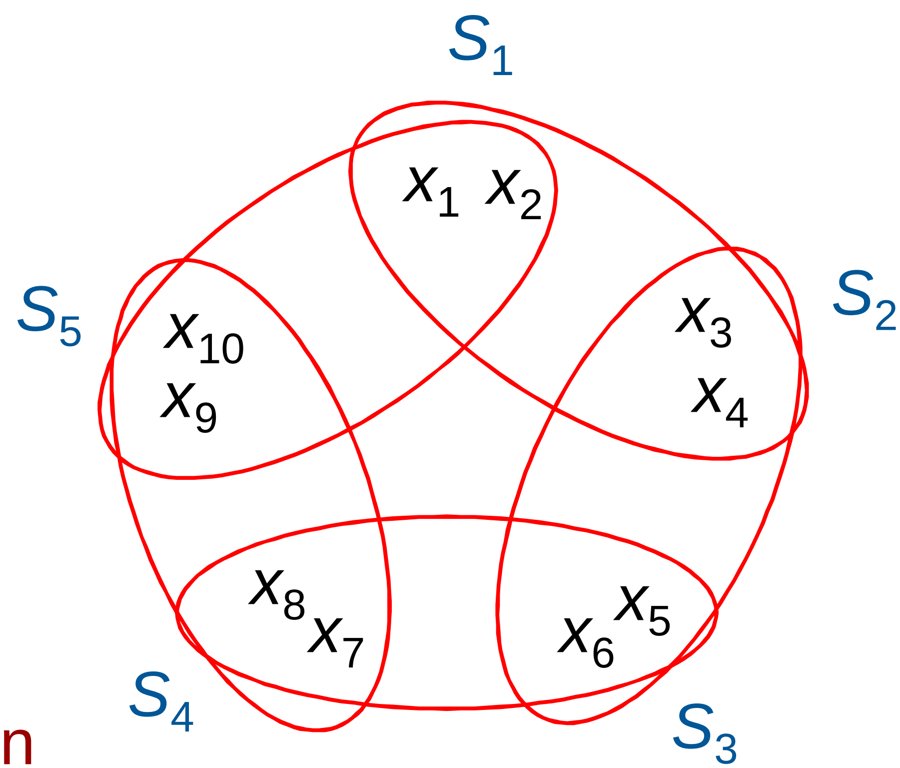
- A q-cycle defines $s^q = 32$ odd hole cuts for each color:

$$\sum_{i \in T} y_{ij} \le \frac{q-1}{2} w_j, \ \text{all } T, j$$

  – where T selects one vertex from each S_k

- For $s \ge 2$, one **x**-cut is stronger than all of these odd hole cuts.

  – For $s = 1$, the finite domain cut is redundant of odd cycle cuts and other 0-1 constraints.

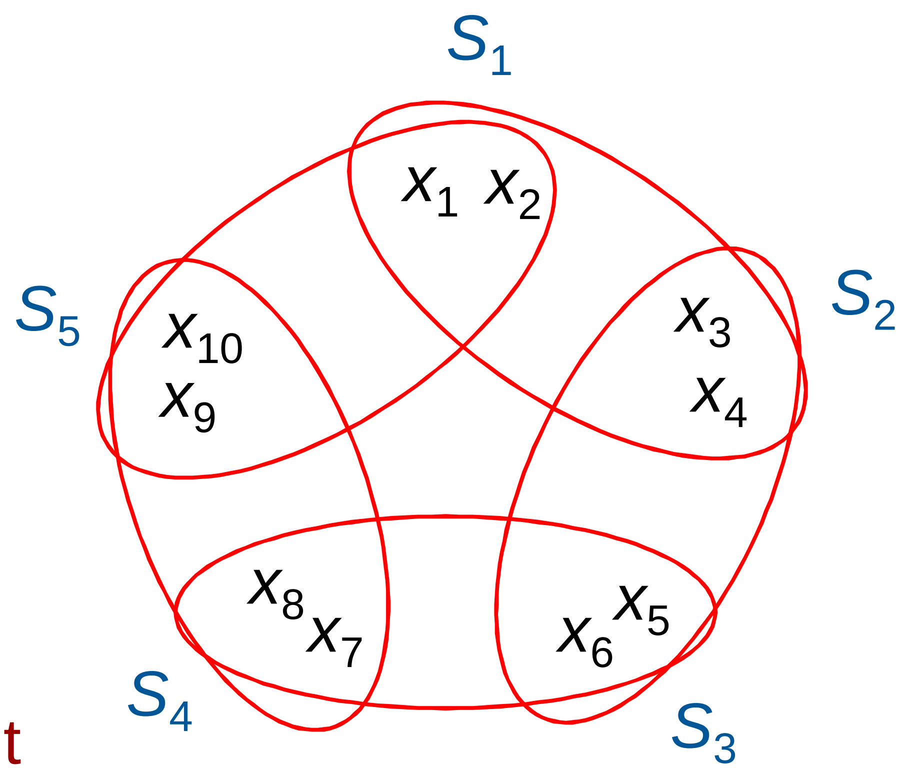


Comparison with Odd Hole Cuts

- A q-cycle defines $s^q = 32$ odd hole cuts for each color:

$$\sum_{i \in T} y_{ij} \le \frac{q-1}{2} w_j, \text{ all } T, j$$

  - where T selects one vertex from each S_k
- For $s \ge 2$, one **x**-cut is stronger than all of these odd hole cuts.
  - Adding a **z**-cut to the **x-**cut tightens the bound further.

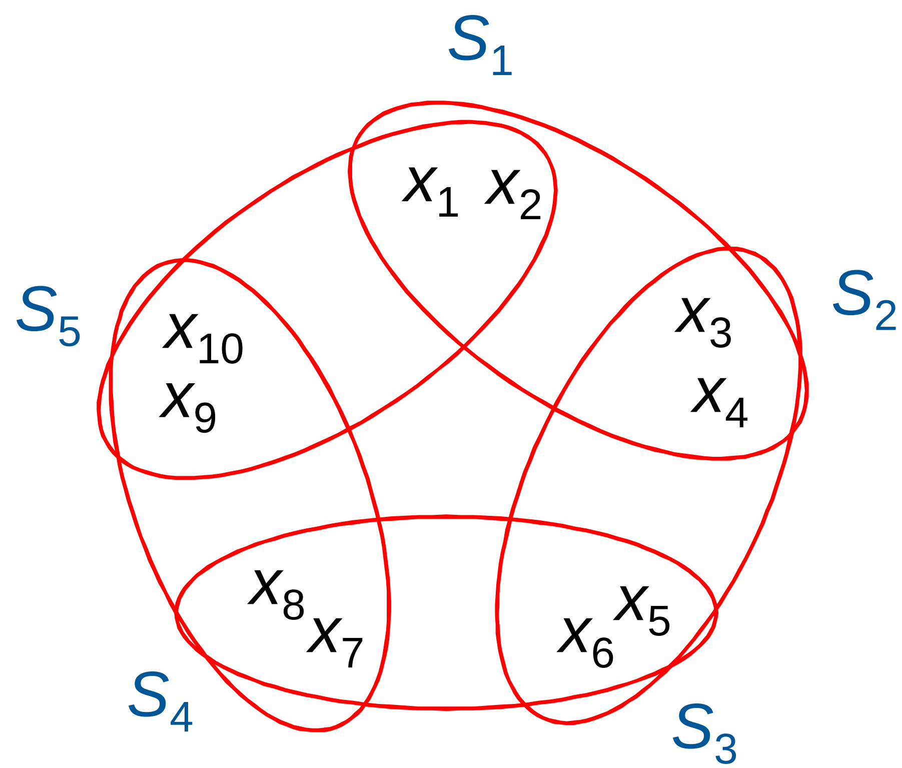


Comparison with Odd Hole Cuts

- A q-cycle defines $s^q = 32$ odd hole cuts for each color:

$$\sum_{i \in T} y_{ij} \le \frac{q-1}{2} w_j, \text{ all } T, j$$

  - where T selects one vertex from each S_k
- For **any** s (including $s = 1$), one **x**-cut and one **z**-cut are stronger than all of these odd hole cuts.

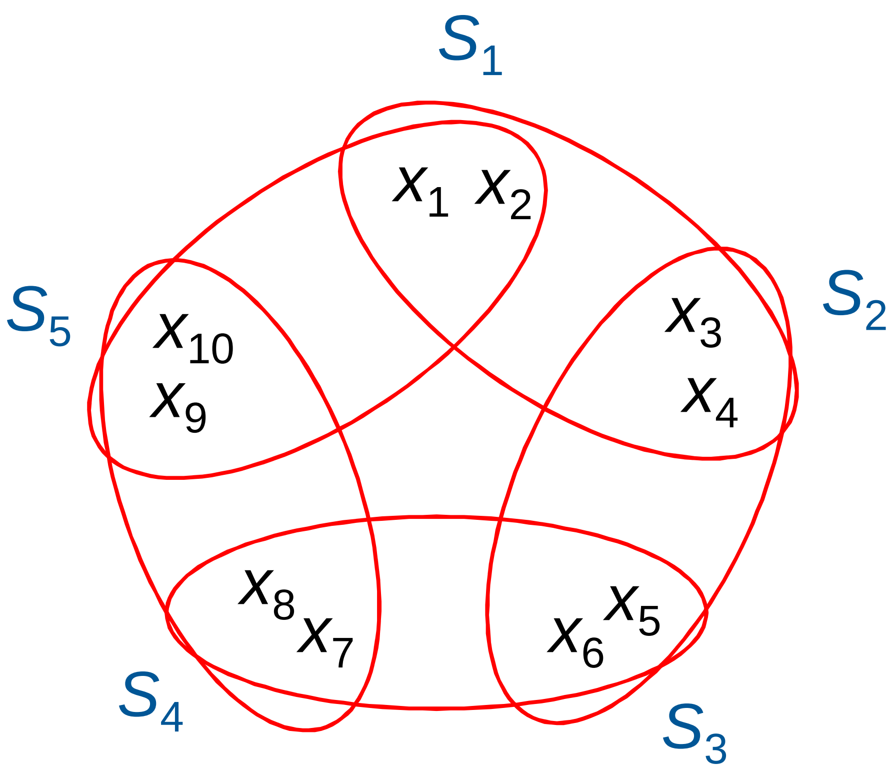


Comparison with Odd Hole Cuts

- A q-cycle defines $s^q = 32$ odd hole cuts for each color:

$$\sum_{i \in T} y_{ij} \leq \frac{q-1}{2} w_j, \ \text{all } T, j$$

  – where T selects one vertex from each S_k



- For **any** s (including $s = 1$), one **x**-cut and one **z**-cut are stronger than all of these odd hole cuts.
  – For any separating odd hole cut, replace it with **x**-cut and **z**-cut for $s = 1$ to get a stronger cut.

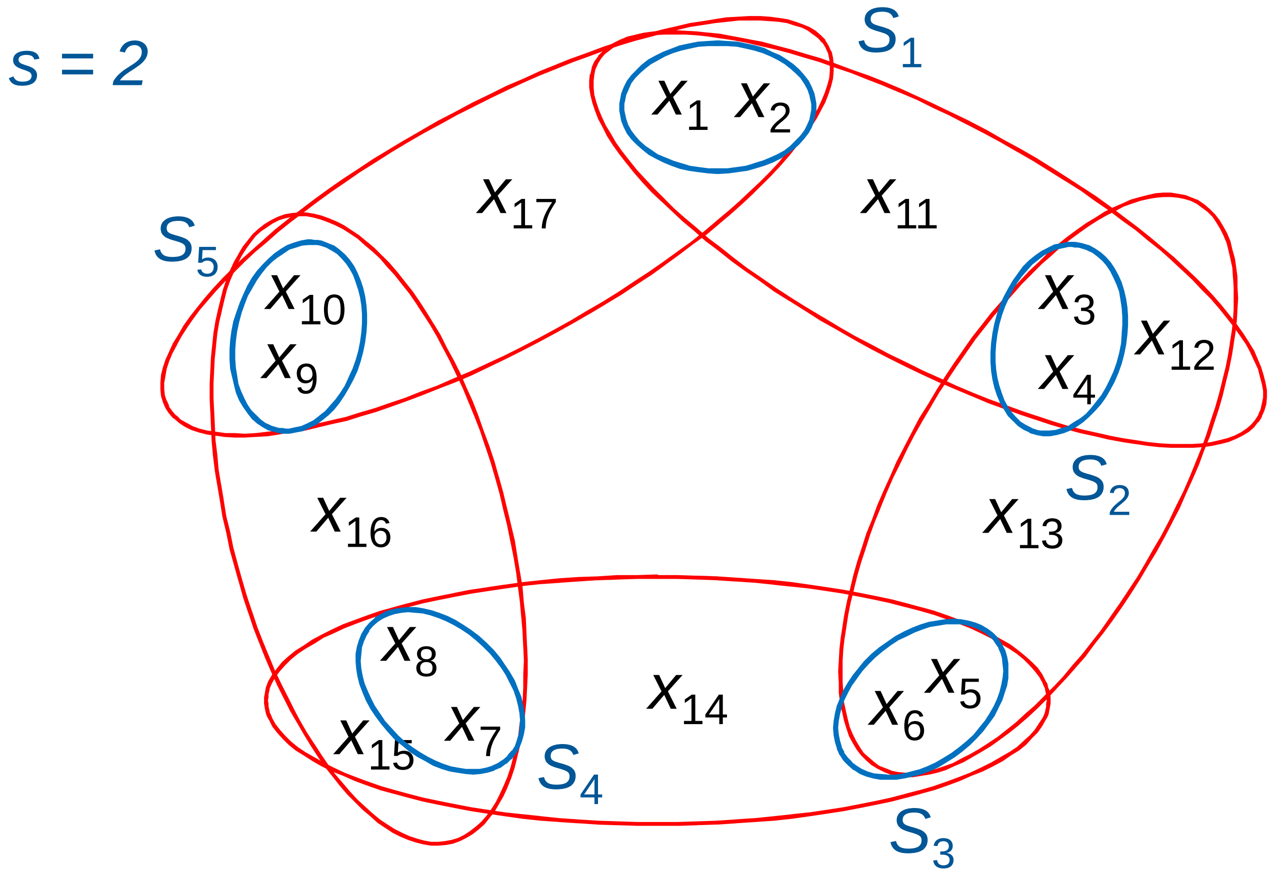
Computed Bounds

Lower bound on number of colors in
0-1 model of 5-cycle

***s* =**	**1**	**2**	**3**
All odd hole cuts	0.8	1.5	2.500
x-cut only	0.8	2.0	3.267
z-cut only	1.3	3.5	5.767
x and **z**-cut only	1.6	4.0	6.533

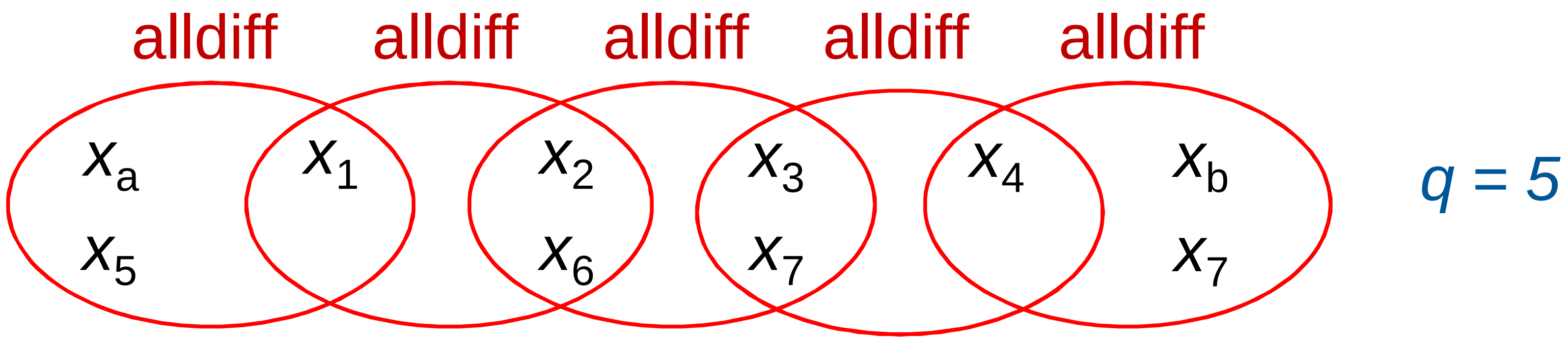
Separation Heuristic

- Select subset of s vertices in each overlap with smallest values in current relaxation:



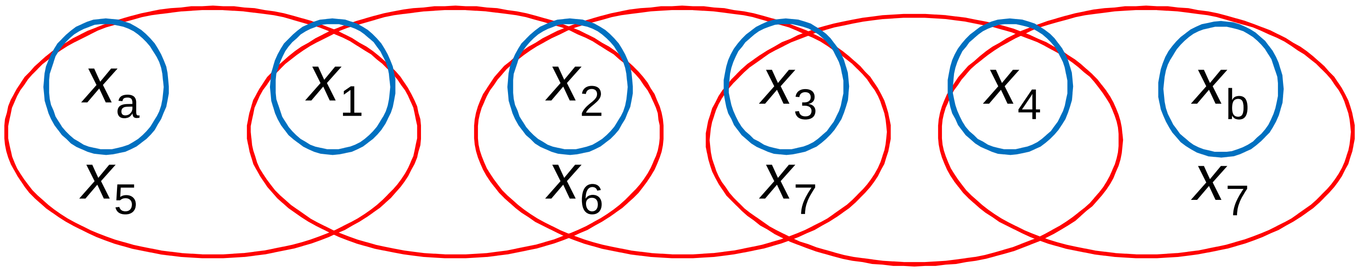
Odd Paths

- A *q*-path looks like



Odd Paths

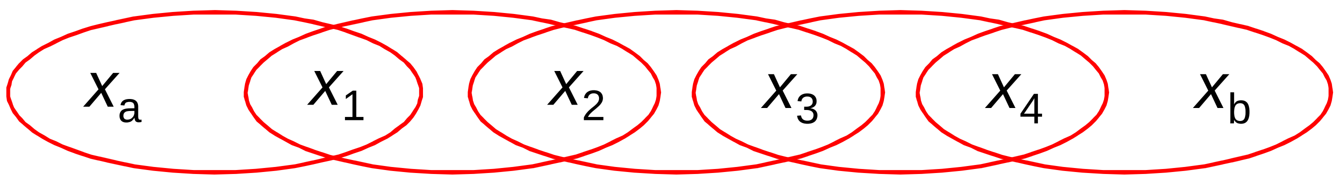
- Select $q + 1$ variables:



$q = 5$

Odd Paths

- This yields a valid inequality (**x**-cut)



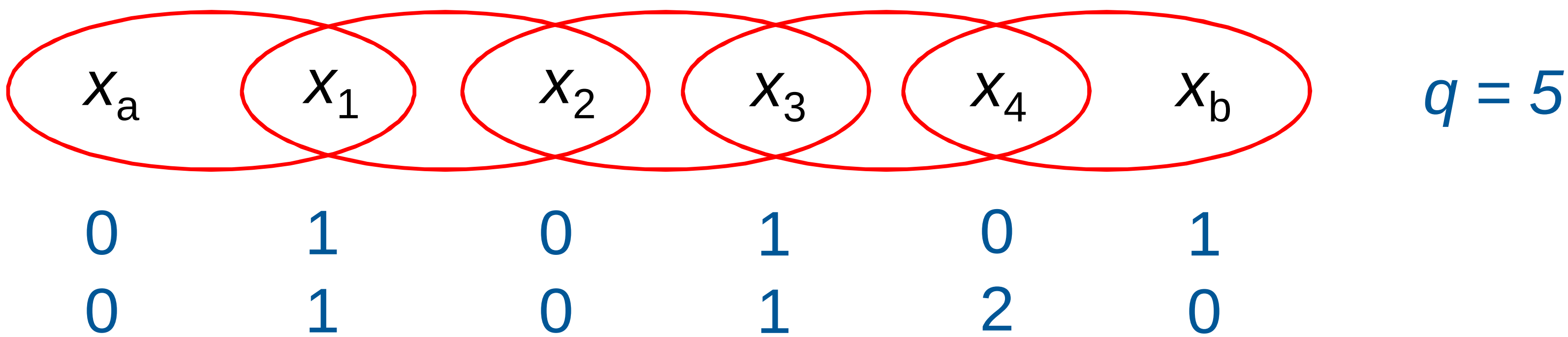
$q = 5$

0	1	0	1	0	1
0	1	0	1	2	0

$$2\left(x_a + x_b\right) + \sum_{i=1}^{q-1} x_i \geq \frac{q+3}{2} = 4$$

Odd Paths

- This yields a valid inequality (**x**-cut):

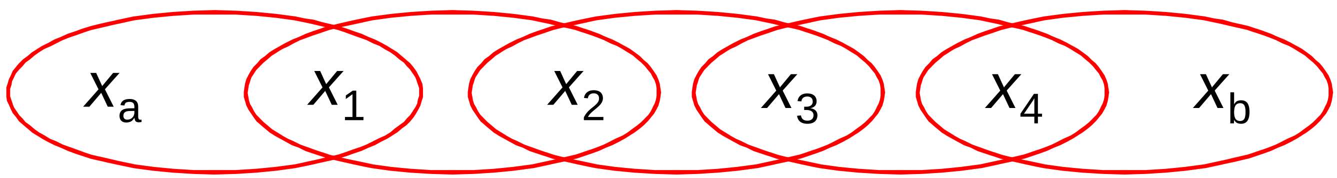


$$2\left(x_a + x_b\right) + \sum_{i=1}^{q-1} x_i \geq \frac{q+3}{2} = 4$$

- The inequality is **facet-defining** if *q* is odd.
  - and if the *q*-path is the subgraph induced by vertices in the cycle.

Odd Paths

- We also have a **z**-cut



q = *5*

0	1	0	1	0	1
0	1	0	1	2	0

$$z \geq \frac{1}{q+3}\left(2(x_a + x_b) + \sum_{i=1}^{q-1} x_i\right) + \frac{1}{2}$$

- This is also facet defining.

Mapping into 0-1 Space

- When mapped into 0-1 space, the finite domain cuts are redundant of the 0-1 model.
  - Because the 0-1 model is already totally unimodular.

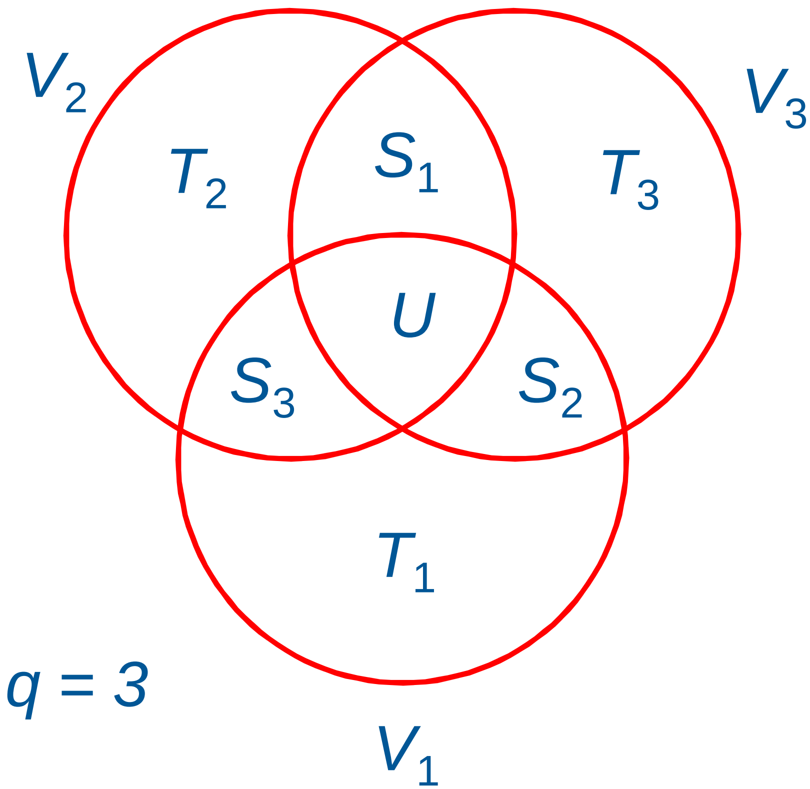
- This is also facet defining.

Mapping into 0-1 Space

- When mapped into 0-1 space, the finite domain cuts are redundant of the 0-1 model.
  – Because the 0-1 model is already totally unimodular.
- However, the finite domain cuts provide a compact relaxation.
  – For each *q*-path, **replace** *q* clique constraints with one **x**-cut and one **y**-cut.
  – Gives the same bound in a problem consisting of one path.

Clutters

- A *q*-clutter looks something like



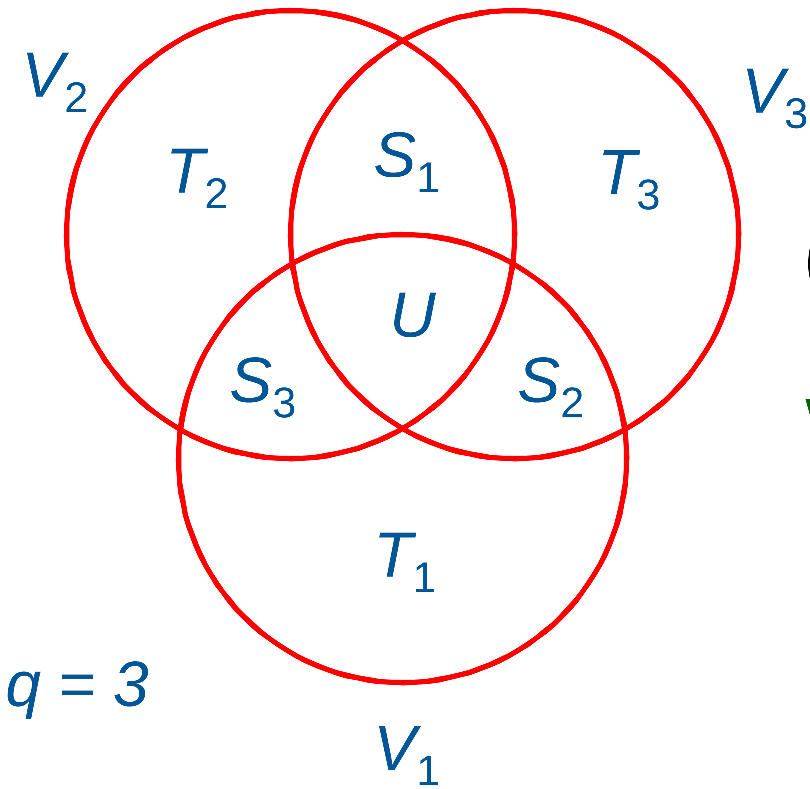
$$S_k = \bigcap_{\ell \neq k} V_\ell \setminus V_k$$

$$T_k = V_k \setminus \bigcup_{\ell \neq k} V_\ell$$

$$U = \bigcap_k V_k$$

Clutters

- Facet-defining inequality. Let $S = \bigcup_k S_k \quad T = \bigcup_k T_k \quad u = |U|$



A valid inequality is:

$$(qs+u)\sum_{i\in T} x_i + \frac{q(q-1)}{2}\sum_{i\in S\cup U} x_i \geq b$$

where

$$b = \frac{1}{2}q(q-1)(qs+u)(qs+u+1)$$

Properties of 0-1 mapping?

Another Interpretation

- Combine alternate formulations to obtain a better relaxation.

- Often done in constraint programming to obtain better propagation.
  - x_i = job assigned to worker i
  - y_j = worker assigned to job j
  - Write each constraint in one or both variable systems.
  - Add **channeling** constraints:

$$x_{y_j} = j, \quad \text{all } j$$

$$y_{x_i} = i, \quad \text{all } i$$

Another Interpretation

- We are doing the same with relaxations.

x relaxation

$$\min z$$

$$z \geq x_i, \quad \text{all } i$$

$$\sum_i x_i \geq \left(sq - \frac{q-1}{4} L \right)(L-1)$$

$$z \geq \frac{1}{qs} \sum_i x_i + \left(1 - \frac{q-1}{4qs} L \right)(L-1)$$

$$x_i = \sum_j j y_{ij}$$

"channeling" constraint

y relaxation

$$\min \sum_j w_j$$

$$\sum_j y_{ij} = 1, \quad \text{all } i$$

$$\sum_{i \in V_k} y_{ij} \leq w_j, \quad \text{all } j,\ V_k$$

$$0 \leq y_{ij} \leq 1$$

Another Interpretation

- Neither relaxation alone provides a good bound.

5-cycle problem with $s = 2$

	Bound
Odd hole cuts in *y*-relaxation	1.5
Finite domain cuts in *x*-relaxation	2.6
Combined models	4.0

Future Work

- Map other known finite-domain cuts into 0-1 models. What happens?
  - Cardinality rules. Yan and Hooker (1999).
  - Circuit constraint (TSP). Genc-Kaya and Hooker (2010).
  - Cumulative constraint.
- Polyhedral analysis for other global constraints.
  - General cardinality, nvalues, sequence, regular.

General Issues

- When do linked multiple relaxations provide better bounds than one relaxation?
- Can we say anything about the properties of different variable encodings?
- What are variables, and why do we use them?