Integer Programming
ISE 418

Lecture 22

Dr. Ted Ralphs

Reading for This Lecture

- "Symmetry in Integer Programming," Margot
- This is a modified version of slides authored by Jim Ostrowski

Motivating Example

$$\min_{x \in \{0,1\}^{n+1}} \{x_{n+1} \mid 2x_1 + 2x_2 + \ldots + 2x_n + x_{n+1} = 2k+1\}$$

- Yes, this problem is very easy!
- Let's try to solve it using branch and bound.
- Source: Bertsimas and Tsitsiklis's *Introduction to Linear Programming*

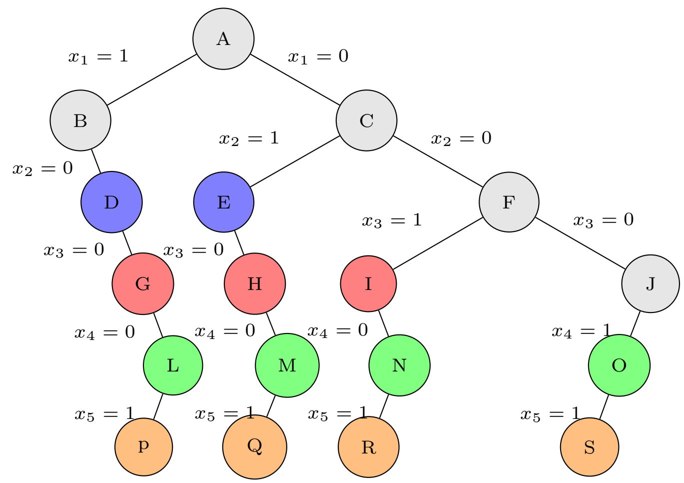
Solving with Branch and Bound

n	k	Time (seconds)	Nodes
20	5	3.24	54,262
20	6	6.97	116,278
20	7	12.24	203,400
20	8	17.83	293,928
20	9	21.68	352,714
20	10	21.74	352,714
25	5	13.86	23,228
25	6	38.96	657,798
25	7	92.71	1,562,273
30	5	42.7	736,284
30	6	160.15	2,629,573

What's going on!?!

The Branch-and-Bound Tree (N=4 K=1)

$$\min_{x \in \{0,1\}^5} \{x_5 \mid 2x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 = 3\}$$



- Subproblem at node D can be written as:

$$\min_{x \in \{0,1\}^5} \{x_5 \mid 2x_3 + 2x_4 + x_5 = 1\}$$

- Subproblem at node E can be written as:

$$\min_{x \in \{0,1\}^5} \{x_5 \mid 2x_3 + 2x_4 + x_5 = 1\}$$

Preliminaries

- $\Pi^n \stackrel{\text{def}}{=}$ the set of all permutations of $I^n = \{1, \ldots, n\}$.
- Given $\pi \in \Pi^n$, $\sigma \in \Pi^m$, let $A(\pi, \sigma) \stackrel{\text{def}}{=}$ the matrix obtained by permuting the columns of A by π and the rows of A by σ, i.e. $A(\pi, \sigma) = P_\sigma A P_\pi$,
- The *symmetry group* $\mathcal{G}$ of the matrix A is the set of permutations of the columns such that there is a corresponding permutation of the rows that when applied yields the original matrix

$$\mathcal{G}(A) \stackrel{\text{def}}{=} \{\pi \in \Pi^n \mid \exists \sigma \in \Pi^m \text{ such that } A(\pi, \sigma) = A\}$$

Facts About Symmetry

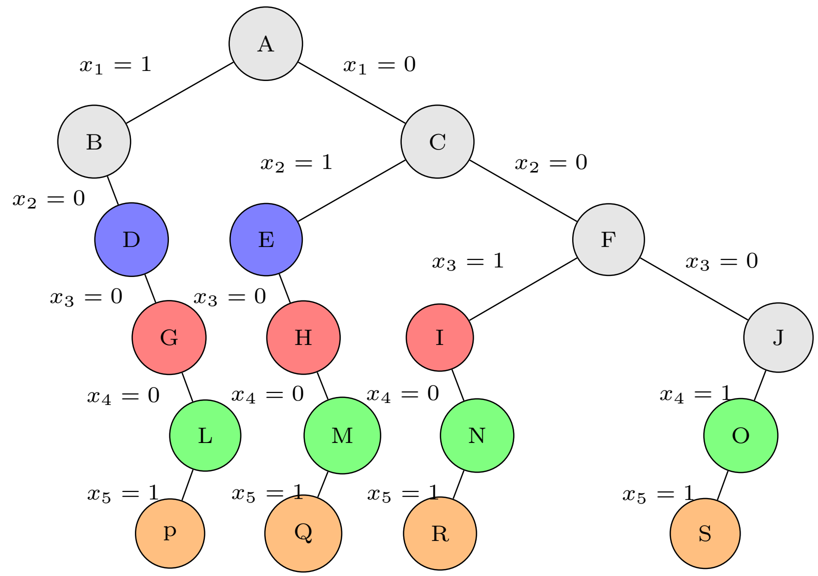
- $\pi(x) \stackrel{\text{def}}{=} (x_{\pi(1)}, x_{\pi(2)}, \ldots x_{\pi(n)})$ permutes the coordinates of x
- $\pi \in \mathcal{G}(A) \Rightarrow (x$ feasible $\Leftrightarrow \pi(x)$ feasible .)
- $e^T x = e^T \pi(x)$

Symmetry in the Real World

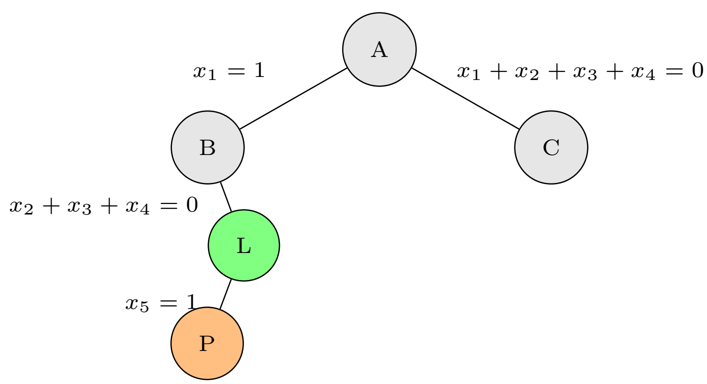
- Symmetry appears in
  - graph coloring problems,
  - cutting stock problems,
  - scheduling problems
  - and more!
- Liberti has shown that many commonly used test problems for integer programming contain symmetry.
- CPLEX has recently implemented symmetry handling techniques.
- Gurobi implemented techniques based on a technique known as *orbital branching*.

General Idea

Turn this:



Into this:



More Preliminaries

- For a set $S \subseteq I^n$, the *orbit* of S with respect to $\mathcal{G}(A)$ is the set of all subsets of I^n to which S can be sent by permutations in $\mathcal{G}(A)$:

$$orb(S) \stackrel{\text{def}}{=} \{S' \subseteq I^n \ \mid \exists \pi \in \mathcal{G}(A) \text{ such that } S' = \pi(S)\}.$$

Orbital Branching

- Let $O \in \Gamma^a$ be an orbit of the symmetry group of subproblem a.
- Surely we can branch as

$$\sum_{i \in O} x_i \geq 1 \quad \textit{or} \quad \sum_{i \in O} x_i \leq 0.$$

- If at least one variable $i \in O$ is going to be one, and they are all "equivalent", then you might as well pick (i^*) one arbitrarily.

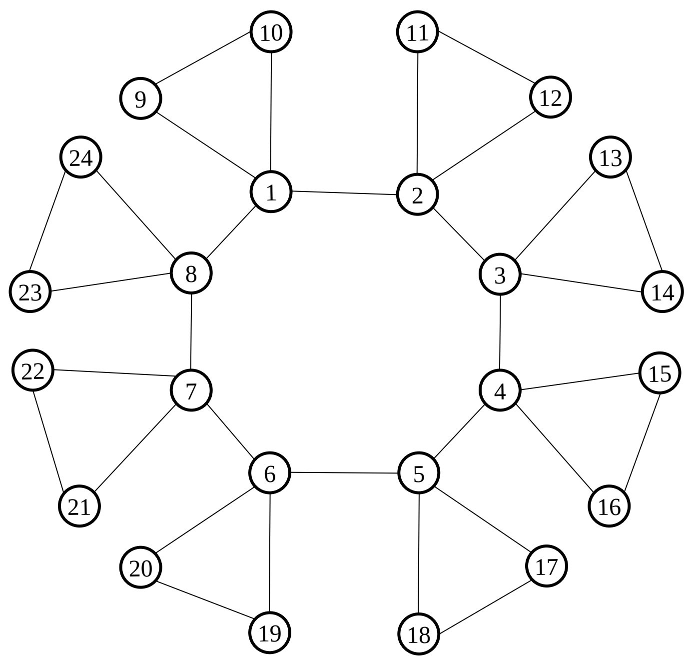
$$x_{i^*} = 1 \quad \textit{or} \quad \sum_{i \in O} x_i = 0$$

Another Way to View Orbital Branching

- Suppose that you have found that the variables x_e, x_f, x_g and x_h share an orbit at node a, $O = \{e, f, g, h\}$.
- The best solution you can find by branching on $x_f, x_g,$ and x_h will be the same as the best solution you can find by branching on x_e
- In fact, solutions will be isomorphic.
- $\Rightarrow$ Prune nodes corresponding to x_f, x_g, and x_h

Branching with Symmetry

$$\max_{x \in \{0,1\}^{|V|}} \left\{ \sum_{i \in V} x_i \mid x_i + x_j \leq 1 \ \forall (i,j) \in E \right\}.$$

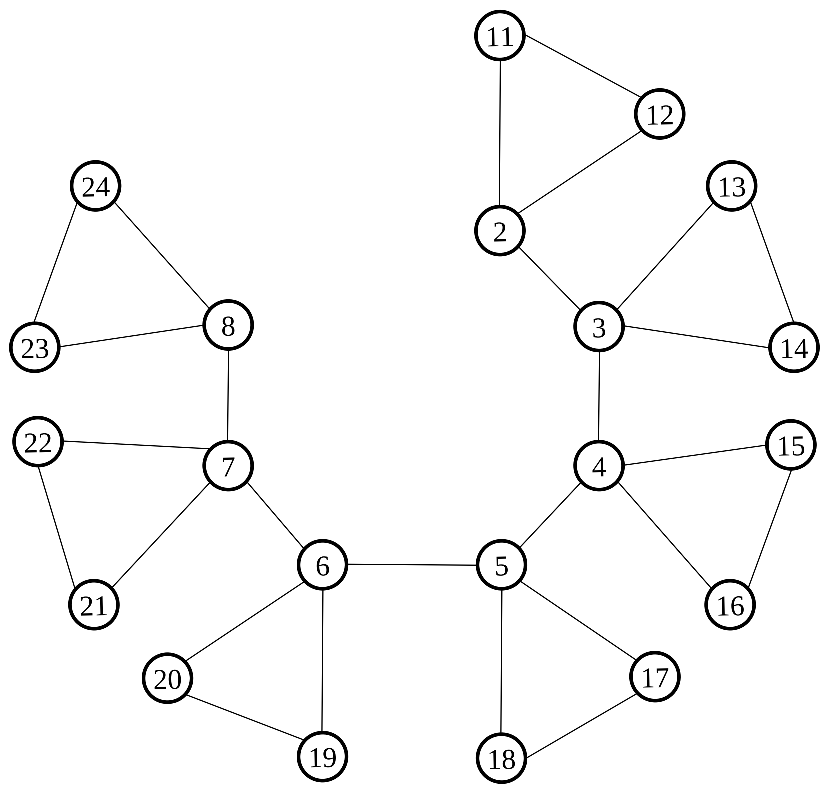


- Variables $\{x_1, x_2, \ldots x_8\}$ are "symmetric"
- Variables $\{x_9, x_{10}, \ldots x_{24}\}$ are "symmetric"

Example: Orbital Branching Subproblems

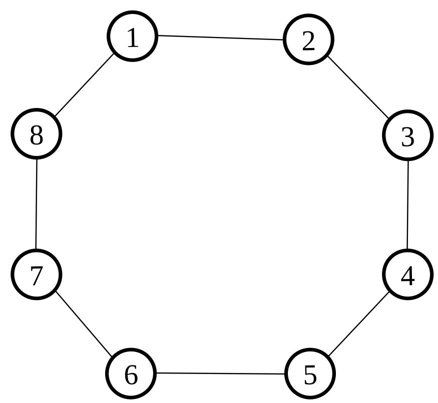
- Branching on orbit $\{9, 10, \ldots, 24\}$, gives subproblems:

$$x_9 = 1$$



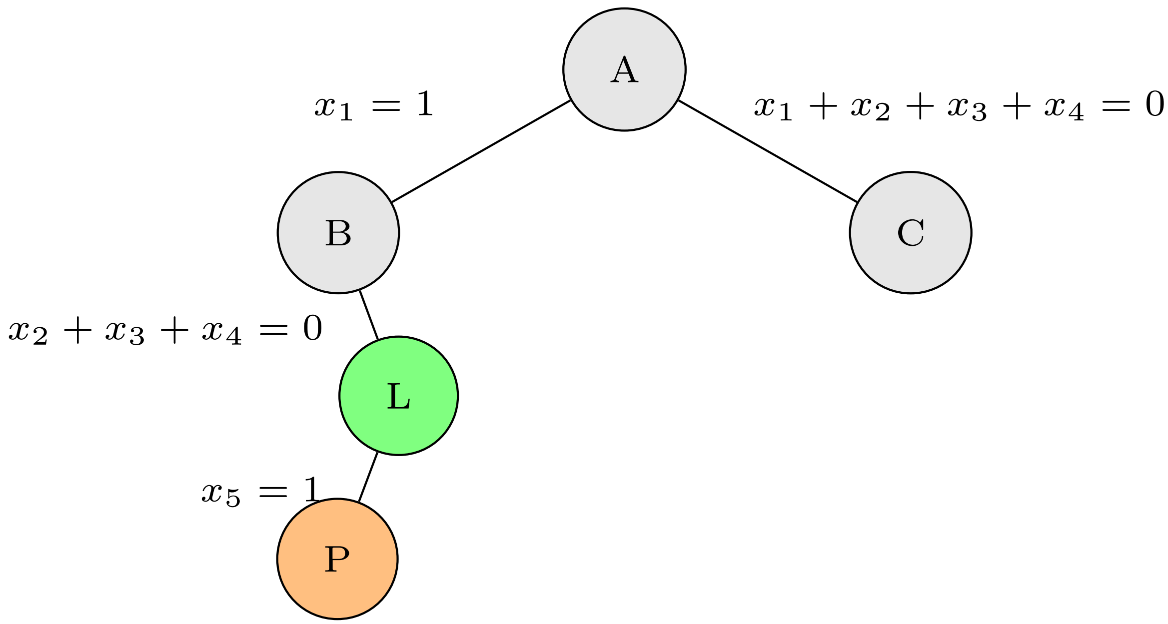
Example: Orbital Branching Subproblems

$$\sum_{j=9}^{24} x_j = 0$$



Orbital Branching and Jeroslow

$$\min_{x \in \{0,1\}^5} \{x_5 \mid 2x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 = 3\}$$



Example: Symmetry in Scheduling Problems

- Scheduling problems can have a great deal of symmetry.
  - Identical machines.
  - Identical jobs.
- This symmetry is more structured than typical problems, allowing us to better exploit the symmetry.

The Unit Commitment Problem

- The Unit Commitment (UC) problem is a large scale MINLP that finds a low-cost generating schedule for power generators.
- These problems have quadratic objective functions, and transmission constraints can be highly nonlinear.
- These problems are typically solved as integer programs.

The Basic Problem

$$\text{Minimize} \sum_{t \in T} \sum_{j \in J} c_j(p_j(t))$$

subject to

$$\sum_{j \in J} p_j(t) \geq D(t), \quad \forall\, t \in T$$

$$p_j \in \Pi_j, \quad \forall j \in J.$$

- $c(p(t))$ gives the cost of generator j producing $p_j(t)$ units of electricity at time t.
- In every time periods, demand $D(t)$ must be met.
- Each generator must work within its physical limits (ramping constraints, minimum shut down times, etc.).

Symmetry in The Unit Commitment Problem

$$\begin{array}{c} \text{Time} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array}
\begin{array}{c} \begin{array}{cc} \text{Gen 1} & \text{Gen 2} \end{array} \\ \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \end{array}$$

On/Off status of Generators 1 & 2

- If generators 1 and 2 are identical, permuting their schedules will give an equivalent solution.
- Permutations schedules between like generators are symmetries.
- Orbital branching works, but can be tailored for this problem.

Symmetry in The Unit Commitment Problem

Equivalent Solution:

$$\begin{array}{c c}
\text{Time} & \begin{array}{cc} \text{Gen 1} & \text{Gen 2} \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} & \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \\ 1 & 0 \end{pmatrix}
\end{array}$$

Symmetry in UC

- Multiple generators of the same type can introduce symmetry into the problem.
- Suppose we had J types of generators. We can think of the on/off status of an optimal solution to UC as J many $T \times n_j$ 0/1 matrices.
- All column permutations of each of these matrices are symmetries in the UC problem.

Finding Symmetry in Subproblems is Easy

$$x^i = \begin{pmatrix} 1 & 1 & 1 & ? & ? \\ ? & ? & ? & ? & ? \\ 0 & 0 & ? & ? & ? \\ ? & ? & ? & ? & ? \end{pmatrix}$$

- Suppose x^i represented a partial solution for the on/off status of generators of type i.
- All relabellings of the columns of x^i are in the symmetry group at the root node.
- After fixings, columns 4 and 5 are still symmetric (neither column contains variables fixed by branching).
- Even though columns 1 and 2 contain fixed variables, they are still symmetric (the fixings in column 1 are identical to the fixings in column 2).

Orbital Branching on UC

$$x^i_{LP} = \begin{pmatrix} .55 & .55 & .55 & .55 \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{pmatrix}$$

- Suppose we chose to perform orbital branching, and branch on the orbit representing the first row of x^i.
- Branch is $x^i_{1,1} = 1 \vee \sum_{j=1}^{4} x^i_{1,j} = 0$.
- It is likely that the right branch is strong, but how strong is the left branch?
- The current LP solution is already suggesting that more than one machine of type i should be on.
- This is not a very useful branch (but better than non-orbital branching!).

Modified Orbital Branching in UC

- Suppose you were branching on the first row of

$$x^i_{LP} = \begin{pmatrix} .55 & .55 & .55 & .55 \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{pmatrix}.$$

- What about the disjunction "Either at least 3 generators are on or at least 2 are off"?

- Using symmetry, we can strengthen this to

$$\{x^i_{1,1} = x^i_{1,2} = x^i_{1,3} = 1\} \vee \{x^i_{1,3} = x^i_{1,4} = 0\}.$$