Day 14 Notes

Zach Neveu

May 29, 2019

1 Agenda

- Branch and Bound to solve ILP
- Announcements
  - Reading
  - Project due Monday
  - Quiz Tuesday

2 Branch & Bound

- Review of last class example
- LP soln is **optimistic bound** on ILP soln
- Add constraints to break up problem into subproblems
- Solve each subproblem with LP solver
- Repeat recursively until subproblem has integral soln in LP
- Don't expand subproblem if…
  - Subproblem is infeasible (No LP solns)
  - LP soln in integral
- How to explore graph?
- Best soln is to use DFS
- Z score of LP soln at parent node is upper bound on Z score of children
- This means that branches with low Z scores can be pruned!
- Suddenly problem not exponential time any more!

Fathoming Rules: don't expand a subproblem if…

1. Subproblem is infeasible
2. LP solution is integral

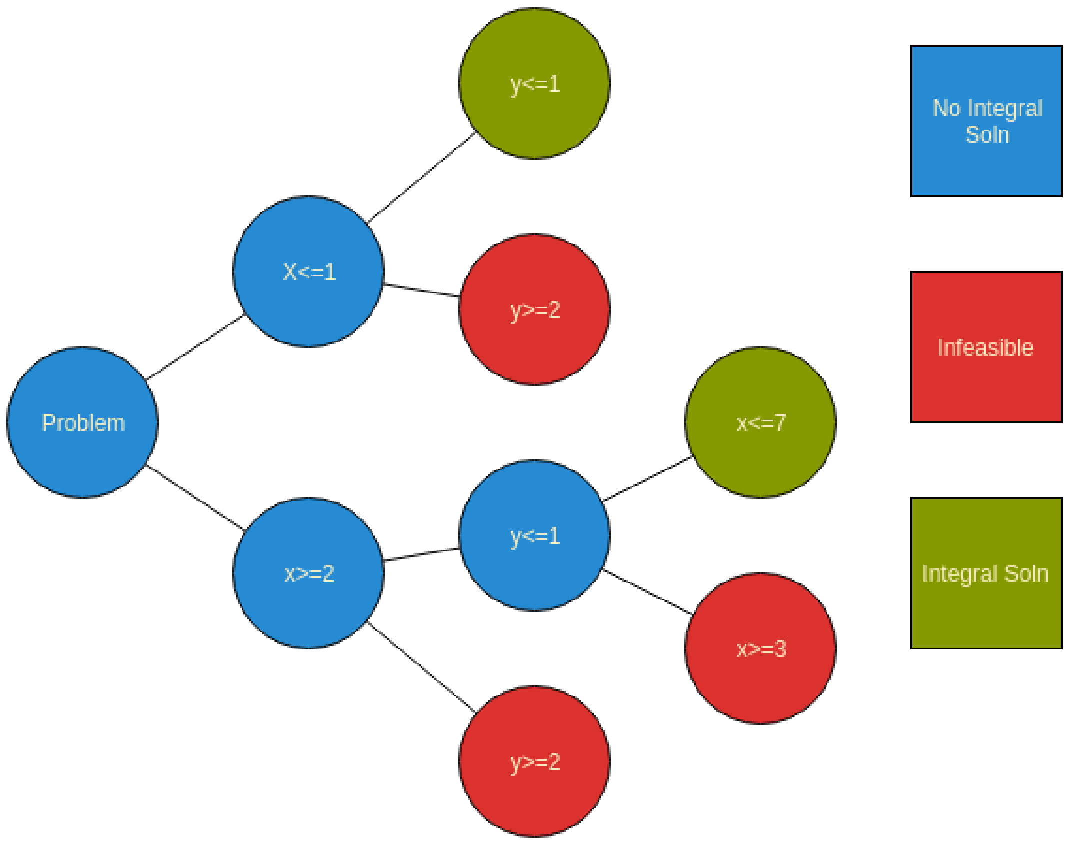

Figure 1: Subproblem Search Diagram

3. Optimistic bound is worse than the best integral solution found so far

- This only prunes if the bound is tight.
- Branches with loose bound will not be pruned until far down the tree
- LP bound is quite tight
- What order to branch variables? x first? y first?
- Crucial decision, but no fixed best algorithm - solvers offer choice
- **IDEA:** Try different sized problems with different strategies, find optimal domain for each strategy. Potentially weed out strategies that don't have a region where they are optimal.

Problem Example

$$
\begin{gathered}
Z = 9x_1 + 5x_2 + 6x_3 + 4x_4 \\
6x_1 + 3x_3 + 5x_3 + 2x_4 \leq 10 \\
x_3 + x_4 \leq 1 \\
-x_1 + x_3 \leq 0 \\
-x_3 + x_4 \leq 0 \\
x_i \in \{0, 1\}
\end{gathered}
$$

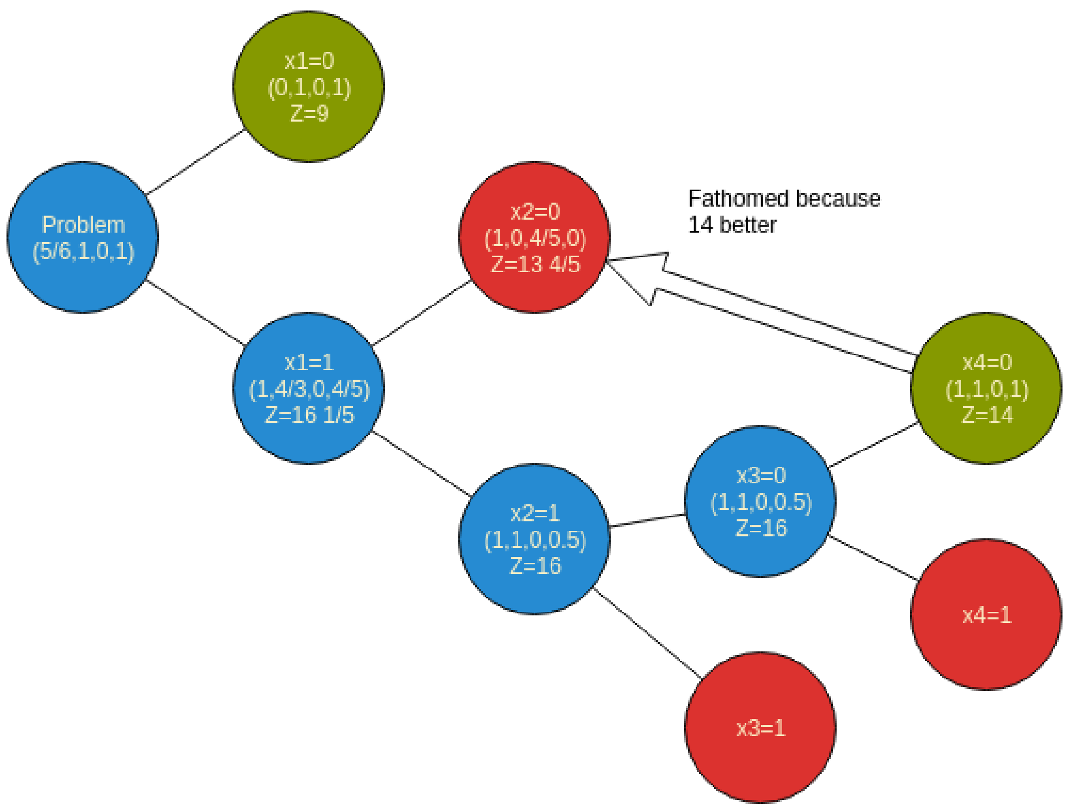

Figure 2: Example Problem Search

3 Project Help

Graph Coloring ILP

- Goal: Minimize Conflicts
- Constraints: # colors, edges
- Graph given as list of edges -> constant set param in ampl

- Convert into adjacency matrix -> Boolean matrix (numNodes x numNodes)
- Possible variables:
  - `Colors[numNodes] = {0..numColors}` -> hard to create linear objective
  - `Colors[numNodes,numColors] = {0,1}` -> Conflicts easy! Just sum of columns!

Table 1: caption

	c1	c2	c3	c4
node 1	0	1	0	0
node 2	0	1	0	0
node 3	0	0	1	0
conflicts	0	1	0	0

```
param edge {i in 0..numNodes-1, j in 0..numNodes-1} :=
    (if(i,j) in edgelist or (j,i) in edgelist then 1 else 0);
```

4 Advanced ILP Techniques

Fix Variables

- Consider constraint $3x_1 \leq 2$. x_1 must be 0.
- $3x_1 + x_2 \leq 2$: x_1 must be 0.
- $5x_1 + x_2 - 2x_3 \leq 2$: x_1 must be 0.
- In general, constants bigger than constraint must be set to 0.

Eliminate Redundant Constraints

- If constraint depends entirely upon value of different constraint, eliminate it.

$$
\begin{gathered}
3x_3 - x_5 + x_7 \leq 1 \\
x_2 + x_4 + x_0 \leq 1 \\
x_1 - 2x_5 + 2x_6 \geq 2 \\
x_1 + x_2 - x_4 \leq 0
\end{gathered}
$$

- Fix variables
  - $x_3 = 0$
  - $x_6 = 1$

- $x_5 = 0$
- $x_2 = 0$
- $x_1 = 0$
- x_7 is only real free variable!

Cutting Planes

- Consider problem in fig. 3
- Top right corner points are not integral solutions
- Can eliminate these infeasible points before sending to solver
- Do this elimination by adding a redundant constraint that all integral feasible solutions satisfy
- Consider $6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10$
- If $x_1, x_2, x_4 = 1$, then this doesn't work
- Add $x_1 + x_2 + x_4 \leq 2$ to speed up ILP solver.
- Without redundant constraint, solver produces $(\frac{5}{6}, 1, 0, 1)$, but the redundant constraint does not allow for this, allowing faster fathoming.

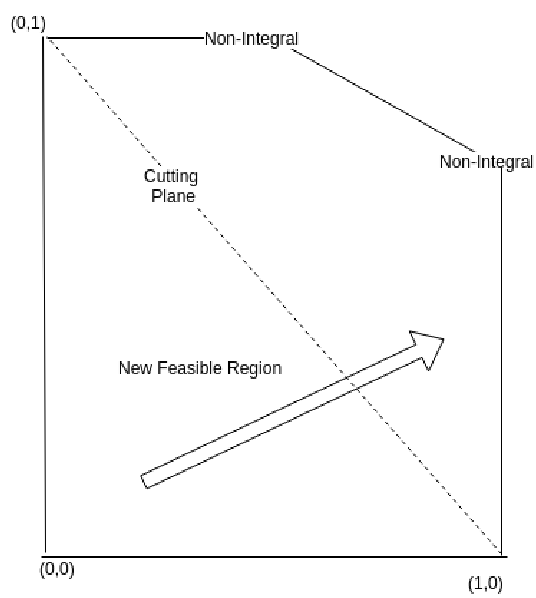

Figure 3: Cutting Plane Example