15.093: Optimization Methods

Lecture 13: Exact Methods for IP
1 Outline

- Cutting plane methods
- Branch and bound methods

2 Cutting plane methods

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \geq 0 \\
& \quad x \text{ integer},
\end{align*}
\]

LP relaxation

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \geq 0.
\end{align*}
\]

2.1 Algorithm

- Solve the LP relaxation. Let \( x^* \) be an optimal solution.
- If \( x^* \) is integer stop; \( x^* \) is an optimal solution to IP.
- If not, add a linear inequality constraint to LP relaxation that all integer solutions satisfy, but \( x^* \) does not; go to Step 1.

2.2 Example

- Let \( x^* \) be an optimal BFS to LP relaxation with at least one fractional basic variable.
- \( N \): set of indices of the nonbasic variables.
- Is this a valid cut?

\[
\sum_{j \in N} x_j \geq 1.
\]

2.3 The Gomory cutting plane algorithm

- Let \( x^* \) be an optimal BFS and \( B \) an optimal basis.
- \( x_B + B^{-1} A_N x_N = B^{-1} b. \)
- \( \bar{u}_i = (B^{-1} A_j)_i, \bar{u}_i^0 = (B^{-1} b)_i. \)
\[ x_i + \sum_{j \in N} \bar{a}_{ij} x_j = \bar{a}_{i0}. \]

- Since \( x_j \geq 0 \) for all \( j \),
  \[ x_i + \sum_{j \in N} |\bar{a}_{ij}| x_j \leq x_i + \sum_{j \in N} \bar{a}_{ij} x_j = \bar{a}_{i0}. \]

- Since \( x_j \) integer,
  \[ x_i + \sum_{j \in N} \bar{a}_{ij} x_j \leq \lfloor \bar{a}_{i0} \rfloor. \]

- Valid cut

### 2.4 Example

\[
\begin{align*}
\text{min} & \quad x_1 - 2x_2 \\
\text{s.t.} & \quad -4x_1 + 6x_2 \leq 9 \\
& \quad x_1 + x_2 \leq 4 \\
& \quad x_1, x_2 \geq 0 \\
& \quad x_1, x_2 \text{ integer.}
\end{align*}
\]

We transform the problem in standard form

\[
\begin{align*}
\text{min} & \quad x_1 - 2x_2 \\
\text{s.t.} & \quad -4x_1 + 6x_2 + x_3 + x_4 = 9 \\
& \quad x_1 + x_2 + x_4 = 4 \\
& \quad x_1, \ldots, x_4 \geq 0 \\
& \quad x_1, \ldots, x_4 \text{ integer.}
\end{align*}
\]

LP relaxation: \( x^1 = (15/10, 25/10) \).

- \( x_2 + \frac{1}{10} x_3 + \frac{1}{10} x_4 = \frac{25}{10} \).
- Gomory cut \( x_2 \leq 2 \).
- Add constraints \( x_2 + x_3 = 2, x_3 \geq 0 \)
- New optimal \( x^2 = (3/4, 2) \).
- One of the equations in the optimal tableau is \( x_1 - \frac{1}{4} x_3 + \frac{6}{4} x_5 = \frac{3}{4} \).
- New Gomory cut \( x_1 - x_3 + x_5 \leq 0 \).
- New optimal solution is \( x^3 = (1, 2) \).
3 Branch and bound

1. **Branching**: Select an active subproblem $F_i$
2. **Pruning**: If the subproblem is infeasible, delete it.
3. **Bounding**: Otherwise, compute a lower bound $b(F_i)$ for the subproblem.
4. **Pruning**: If $b(F_i) \geq U$, the current best upperbound, delete the subproblem.
5. **Partitioning**: If $b(F_i) < U$, either obtain an optimal solution to the subproblem (stop), or break the corresponding problem into further subproblems, which are added to the list of active subproblem.

3.1 LP Based

- Compute the lower bound $b(F)$ by solving the LP relaxation of the discrete optimization problem.
- From the LP solution $x^*$, if there is a component $x^*_i$ which is fractional, we create two subproblems by adding either one of the constraints
  \[ x_i \leq \lfloor x^*_i \rfloor, \text{ or } x_i \geq \lfloor x^*_i \rfloor. \]
  Note that both constraints are violated by $x^*$.
- If there are more than 2 fractional components, we use selection rules like maximum infeasibility etc. to determine the inequalities to be added to the problem.
- Select the active subproblem using either depth-first or breadth-first search strategies.

3.2 Example

\[
\begin{align*}
\text{max} \quad & 12x_1 + 8x_2 + 7x_3 + 6x_4 \\
\text{s.t.} \quad & 8x_1 + 6x_2 + 5x_3 + 4x_4 \leq 15 \\
& x_1, x_2, x_3, x_4 \text{ are binary.}
\end{align*}
\]
LP relaxation

\[
\begin{align*}
\text{max} & \quad 12x_1 + 8x_2 + 7x_3 + 6x_4 \\
\text{s.t.} & \quad 8x_1 + 6x_2 + 5x_3 + 4x_4 \leq 15 \\
& \quad x_1 \leq 1, \; x_2 \leq 1, \; x_3 \leq 1, \; x_4 \leq 1 \\
& \quad x_1, x_2, x_3, x_4 \geq 0
\end{align*}
\]

LP solution: \( x_1 = 1, \; x_2 = 0, \; x_3 = 0.6, \; x_4 = 1 \) Profit = 22.2

3.2.1 Branch and bound tree

3.3 Pigeonhole Problem

- There are \( n + 1 \) pigeons with \( n \) holes. We want to place the pigeons in the holes in such a way that no two pigeons go into the same hole.
- Let \( x_{ij} \) = 1 if pigeon \( i \) goes into hole \( j \), 0 otherwise.
- Formulation 1:

\[
\begin{align*}
\sum_j x_{ij} &= 1, \quad i = 1, \ldots, n + 1 \\
x_{ij} + x_{kj} &\leq 1, \quad \forall j, i \neq k
\end{align*}
\]
• Formulation 2:

\[
\sum_j x_{ij} = 1, \quad i = 1, \ldots, n + 1 \\
\sum_{i=1}^{n+1} x_{ij} \leq 1, \quad \forall j
\]

Which formulation is better for the problem?

• The pigeonhole problem is infeasible.
• For Formulation 1, feasible solution \( x_{ij} = \frac{1}{n} \) for all \( i, j \). \( O(n^3) \) constraints. Nearly complete enumeration is needed for LP-based BB, since the problem remains feasible after fixing many variables.
• Formulation 2 Infeasible. \( O(n) \) constraints.
• Message: Formulation of the problem is important!

3.4 Preprocessing

• An effective way of improving integer programming formulations prior to and during branch-and-bound.
• Logical Tests
  - Removal of empty (all zeros) rows and columns;
  - Removal of rows dominated by multiples of other rows;
  - Strengthening the bounds within rows by comparing individual variables and coefficients to the right-hand-side.
  - Additional strengthening may be possible for integral variables using rounding.

• Probing: Setting temporarily a 0-1 variable to 0 or 1 and redo the logical tests. Force logical connection between variables. For example, if \( 5x + 4y + z \leq 8, x, y, z \in \{0, 1\} \), then by setting \( x = 1 \), we obtain \( y = 0 \). This leads to an inequality \( x + y \leq 1 \).
4 Application

4.1 Directed TSP

4.1.1 Assignment Lower Bound

Given a directed graph $G = (N, A)$ with $n$ nodes, and a cost $c_{ij}$ for every arc, find a tour (a directed cycle that visits all nodes) of minimum cost.

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

s.t. : $\sum_{i=1}^{n} x_{ij} = 1, \ j = 1, \ldots, n,$

$\sum_{j=1}^{n} x_{ij} = 1, \ i = 1, \ldots, n,$

$x_{ij} \in \{0,1\}.$

Branching: Set one of the arcs selected in the optimal solution to zero, i.e., add constraints of the type “$x_{ij} = 0$” to exclude the current optimal solution.

4.2 Improving BB

- Better LP solver
- Use problem structure to derive better branching strategy
- Better choice of lower bound $b(F)$ - better relaxation
- Better choice of upper bound $U$ - heuristic to get good solution
- KEY: Start pruning the search tree as early as possible