15.081J/6.251J Introduction to Mathematical Programming

Lecture 25: Exact Methods
for Discrete Optimization

1 Outline

SLIDE 1

- Cutting plane methods
- Branch and bound methods

2 Cutting plane methods

SLIDE 2

$$\begin{array}{rrcl} \min & \boldsymbol{c}'\boldsymbol{x} & & \\ \text{s.t.} & \boldsymbol{A}\boldsymbol{x} & = & \boldsymbol{b} \\ & \boldsymbol{x} & \geq & \boldsymbol{0} \\ & \boldsymbol{x} & \text{integer,} & \end{array}$$

LP relaxation

$$\begin{array}{rrcl} \min & \boldsymbol{c}'\boldsymbol{x} & & \\ \text{s.t.} & \boldsymbol{A}\boldsymbol{x} & = & \boldsymbol{b} \\ & \boldsymbol{x} & \geq & \boldsymbol{0}. \end{array}$$

2.1 Algorithm

SLIDE 3

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

2.2 Example

SLIDE 4

- Let $\boldsymbol{x}^*$ be an optimal BFS to LP ralxation 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

SLIDE 5

- Let $\boldsymbol{x}^*$ be an optimal BFS and $\boldsymbol{B}$ an optimal basis.
-

$$\boldsymbol{x}_B + \boldsymbol{B}^{-1}\boldsymbol{A}_N\boldsymbol{x}_N = \boldsymbol{B}^{-1}\boldsymbol{b}.$$

- $\overline{a}_{ij} = \left(\boldsymbol{B}^{-1}\boldsymbol{A}_j\right)_i$, $\overline{a}_{i0} = \left(\boldsymbol{B}^{-1}\boldsymbol{b}\right)_i$.

- $$x_i + \sum_{j \in N} \overline{a}_{ij} x_j = \overline{a}_{i0}.$$
- Since $x_j \geq 0$ for all j,
$$x_i + \sum_{j \in N} \lfloor \overline{a}_{ij} \rfloor x_j \leq x_i + \sum_{j \in N} \overline{a}_{ij} x_j = \overline{a}_{i0}.$$
- Since x_j integer,
$$x_i + \sum_{j \in N} \lfloor \overline{a}_{ij} \rfloor x_j \leq \lfloor \overline{a}_{i0} \rfloor.$$
- Valid cut

2.4 Example

SLIDE 6

$$\begin{array}{lrcrcl} \min & x_1 & - & 2x_2 & & \\ \text{s.t.} & -4x_1 & + & 6x_2 & \leq & 9 \\ & x_1 & + & x_2 & \leq & 4 \\ & \multicolumn{5}{l}{x_1, x_2 \geq 0} \\ & \multicolumn{5}{l}{x_1, x_2 \text{ integer.}} \end{array}$$

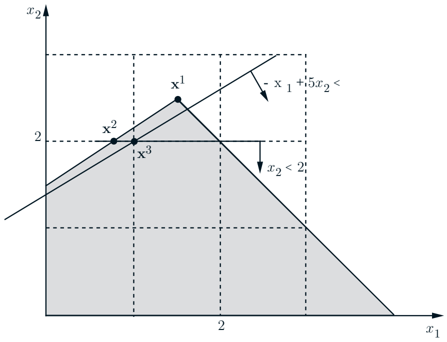
We transform the problem in standard form

$$\begin{array}{lrcrcrcrcl} \min & x_1 & - & 2x_2 & & & & & & \\ \text{s.t.} & -4x_1 & + & 6x_2 & + & x_3 & & & = & 9 \\ & x_1 & + & x_2 & & & + & x_4 & = & 4 \\ & \multicolumn{9}{l}{x_1, \ldots, x_4 \geq 0} \\ & \multicolumn{9}{l}{x_1, \ldots, x_4 \text{ integer.}} \end{array}$$

LP relaxation: $\boldsymbol{x}^1 = (15/10, 25/10)$. SLIDE 7

- $$x_2 + \frac{1}{10} x_3 + \frac{1}{10} x_4 = \frac{25}{10}.$$
- Gomory cut
$$x_2 \leq 2.$$
- Add constraints $x_2 + x_5 = 2$, $x_5 \geq 0$
- New optimal $\boldsymbol{x}^2 = (3/4, 2)$.
- 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 $\boldsymbol{x}^3 = (1, 2)$.

SLIDE 8



3 Branch and bound

SLIDE 9

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

SLIDE 10

- Compute the lower bound $b(F)$ by solving the LP relaxation of the discrete optimization problem.
- From the LP solution $\boldsymbol{x}^*$, if there is a component x_i^* which is fractional, we create two subproblems by adding either one of the constraints

$$x_i \leq \lfloor x_i^* \rfloor, \text{ or } x_i \geq \lceil x_i^* \rceil.$$

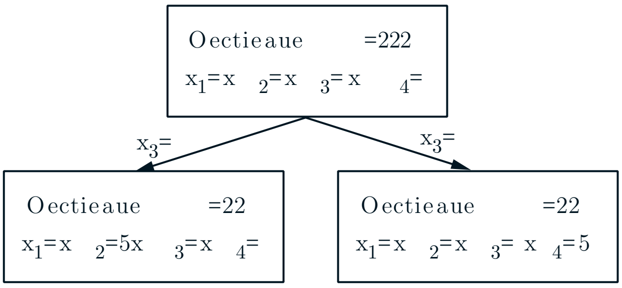
  Note that both constraints are violated by $\boldsymbol{x}^*$.

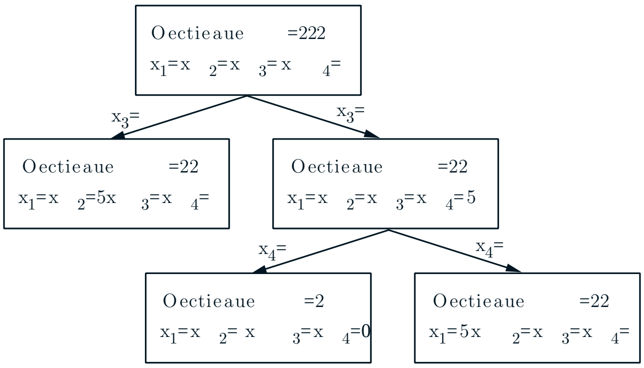
- If there are more than 2 fractional components, we use selection rules like maximum infeasibility etc. to determine the inequalities to be added to the problem
- Select the active subproblem using either depth-first or breadth-first search strategies.

3.2 Example

SLIDE 11

$$\begin{aligned} \max \quad & 12x_1 + 8x_2 + 7x_3 + 6x_4 \\ \text{s.t.} \quad & 8x_1 + 6x_2 + 5x_3 + 4x_4 \leq 15 \\ & x_1, x_2, x_3, x_4 \text{ are binary.} \end{aligned}$$





LP relaxation SLIDE 12

$$\begin{aligned} \max \quad & 12x_1 + 8x_2 + 7x_3 + 6x_4 \\ \text{s.t.} \quad & 8x_1 + 6x_2 + 5x_3 + 4x_4 \le 15 \\ & x_1 \le 1, x_2 \le 1, x_3 \le 1, x_4 \le 1 \\ & x_1, x_2, x_3, x_4 \ge 0 \end{aligned}$$

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

SLIDE 13

3.3 Pigeonhole Problem

SLIDE 14

SLIDE 15

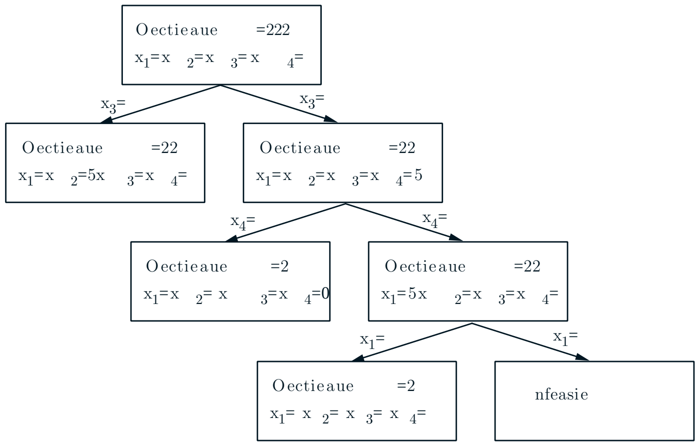
SLIDE 16

- 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.

SLIDE 17

- Formulation 1:

$$\sum_j x_{ij} = 1, \quad i = 1, \dots, n+1$$
$$x_{ij} + x_{kj} \le 1, \quad \forall j, i \ne k$$



- Formulation 2:

$$\sum_j x_{ij} = 1, \quad i = 1, \dots, n+1$$
$$\sum_{i=1}^{n+1} x_{ij} \leq 1, \quad \forall j$$

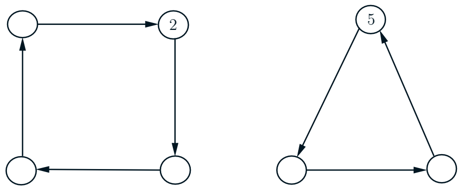
Which formulation is better for the problem? SLIDE 18

- 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.
- Mesage: Formulation of the problem is important!

3.4 Preprocessing

SLIDE 19

- 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

SLIDE 20

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.

$$
\begin{array}{ll}
\min & \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \\
\text{s.t.}: & \sum_{i=1}^{n} x_{ij} = 1, \; j = 1, \ldots, n, \\
& \sum_{j=1}^{n} x_{ij} = 1, \; i = 1, ..., n, \\
& x_{ij} \in \{0, 1\}.
\end{array}
$$

SLIDE 21

4.2 Improving BB

SLIDE 22

- 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**

6.251J / 15.081J Introduction to Mathematical Programming
Fall 2009