Discrete Optimization Problems

- An integer programming (IP) problem is formulated as:

$$\max_{\boldsymbol{x}} z = \boldsymbol{c}^T \boldsymbol{x}, \text{ subject to } \boldsymbol{Ax} \leq \boldsymbol{b}, \ \boldsymbol{x} \in \mathbb{Z}^n, \ \boldsymbol{x} \geq \boldsymbol{0}$$

A binary integer programming (BIP) problem is formulated as:

$$\max_{\boldsymbol{x}} z = \boldsymbol{c}^T \boldsymbol{x}, \text{ subject to } \boldsymbol{Ax} \leq \boldsymbol{b}, \ \boldsymbol{x} \in \{0,1\}^n$$

- A Mixed integer programming (MIP) problem is formulated as:

$$\max_{\boldsymbol{x}} z = \boldsymbol{c}^T \boldsymbol{x}, \text{ subject to } \boldsymbol{Ax} \leq \boldsymbol{b}, \ x_i \geq 0, x_i \in \mathbb{Z}, i = 1, \dots, n_d;$$
$$x_{iL} \leq x_i \leq x_{iU}, i = n_d + 1, \dots, n$$

A general mixed variable design optimization problem is formulated as:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}),$$

Subject to $h_i(\boldsymbol{x}) = 0, \ i = 1, \dots, l; \ \ g_j(x) \leq 0, j = i, \dots, m;$

$x_i \in D_i, i = 1, \dots, n_d; \ \ x_{iL} \leq x_i \leq x_{iU}, i = n_d + 1, \dots, n$

Enumeration Method

- Discrete optimization problems may be solved by enumeration, i.e., through an ordered listing of all solutions.
- The number of solution combinations to be evaluated is given as: $N_c = \prod_{i=1}^{n_d} q_i$, where n_d is the number of design variables and q_i is the number of discrete values for the design variable x_i.
- Note, N_c increases rapidly with increase in n_d and q_i.

LP Problems with Integral Coefficients

- Consider optimization problem modeled with integral coefficients:

  $\min_{\boldsymbol{x}} \ z = \boldsymbol{c}^T \boldsymbol{x}$, Subject to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, $\boldsymbol{x} \geq \boldsymbol{0}$,

  where $\boldsymbol{A} \in \mathbb{Z}^{m \times n}, rank(\boldsymbol{A}) = m, \boldsymbol{b} \in \mathbb{Z}^m, \boldsymbol{c} \in \mathbb{Z}^n$
- Assume that $\boldsymbol{A}$ is totally unimodular, i.e., every square submatrix $\boldsymbol{C}$ of $\boldsymbol{A}$, has $\det(\boldsymbol{C}) \in \{0, \pm 1\}$. In particular, every basis matrix $\boldsymbol{B}$ has $\det(\boldsymbol{B}) = \pm 1$; hence $\boldsymbol{B}^{-1} = \pm Adj\ \boldsymbol{B}$; and $\boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b}$ are integral.
- Further, if $\boldsymbol{A}$ is totally unimodular, so is $[\boldsymbol{A}\ \boldsymbol{I}]$. Let a problem with LE constraints be represented as: $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{I}\boldsymbol{s} = \boldsymbol{b}$; then, if $\boldsymbol{A} \in \mathbb{Z}^{m \times n}$ is totally unimodular and $\boldsymbol{b} \in \mathbb{Z}^m$, all BFSs to the problem are integral.
- Total unimodularity of A is a sufficient condition for integral solution to the LP problem modeled with integral coefficients. A necessary condition is that every $m \times m$ matrix $\boldsymbol{B}$ has $\det(\boldsymbol{B}) = \pm 1$.

Example: Integer BFS

- Consider the LP problem with integral coefficients

  $\max_{\boldsymbol{x}} \; z = 2x_1 + 3x_2$

  Subject to: $x_1 \leq 3, x_2 \leq 5, x_1 + x_2 \leq 7, \boldsymbol{x} \in \mathbb{Z}^2, \boldsymbol{x} \geq \boldsymbol{0}$
- Following the introduction of slack variables, the constraint matrix and the right hand side are given as: $\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} 3 \\ 5 \\ 7 \end{bmatrix}$
- Note, $\boldsymbol{A}$ is totally unimodular and $\boldsymbol{b} \in \mathbb{Z}^3$.
- Then, using the simplex method, the optimal integral solution is obtained as: $\boldsymbol{x}^T = [2,5,1,0,0]$, with $z^* = 19$.

Example: Transportation Problem

- Consider a transportation problem with $m = 2$ supply and $n = 3$ demand nodes

$$\min_{x_{ij}} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}\,, \; i = 1,2; j = 1,2,3$$

Subject to: $\sum_{i=1}^{m} x_{ij} = a_i,\; \sum_{j=1}^{n} x_{ij} = b_j$

Where

x_{11}	x_{12}	x_{13}	10
x_{21}	x_{22}	x_{23}	10
5	9	6	

- Every BFS to the problem has $m + n - 1 = 4$ nonzero components
- For example, possible BFS are:

0	4	6	10
5	5	0	10
5	9	6	

1	9	0	10
4	0	6	10
5	9	6	

5	0	5	10
0	9	1	10
5	9	6	

Binary Integer Programming Problems

Consider the BIP problem:

$$\min_{\boldsymbol{x}} \; z = \boldsymbol{c}^T\boldsymbol{x}, \text{ Subject to } \boldsymbol{Ax} \geq \boldsymbol{b}, \; x_i \in \{0,1\}, \; i = 1, \dots, n$$

- Most LP problems can be reformulated in the BIP framework. If the number of variables is small, and the bounds $x_{min} < x_i < x_{max}$ on the design variables are known, then each x_i can be represented as a binary number using k bits, where $2^{k+1} \geq x_{max} - x_{min}$. The resulting problem involves selection of bits and in a BIP problem.

Implicit Enumeration

- BIP problems can be solved via implicit enumeration. In this process, obviously infeasible solutions are eliminated and the remaining ones are evaluated (i.e., enumerated) to find the optimum. The search starts from $x_i = 0,\ i = 1, \dots, n$, which is optimal. If this is not feasible, then we systematically adjust individual variable values till feasibility is attained.

Implicit Enumeration Algorithm

Implicit Enumeration Algorithm:

1. Initialize: set $x_i = 0,\ i = 1, \dots, n$; quit, if this solution is feasible.
2. For some i, set $x_i = 1$. If the resulting solution is feasible, then record it if it is the first feasible solution; also record it, if it improves upon a previously recorded feasible solution.
3. Backtrack (set $x_i = 0$) if a feasible solution was reached in the previous step, or if feasibility appears impossible in this branch.
4. Choose another i and return to 2.

- The progress of the algorithm is recorded in a decision-tree using nodes and arcs, with node 0 representing the initial solution ($x_i = 0,\ i = 1, \dots, n$), and node i representing a change in the value of variable x_i. From node k, if we choose to raise variable x_i to one, then we draw an arc from node k to node i.

Implicit Enumeration Algorithm

- At node i the following possibilities exist:
  - The resulting solution is feasible, hence no further improvement in this branch is possible.
  - Feasibility is impossible from this branch.
  - The resulting solution is not feasible, but feasibility or further improvements are possible.
- In the first two cases, the branch is said to have been fathomed. If that happens, we then backtrack to node k, where variable x_i is returned to zero. We next seek another variable to be raised to one.
- The algorithm continues till all branches have been fathomed, and returns an optimum 0-1 solution.

Example: Implicit Enumeration

- Problem: A machine shaft is to be cut at two locations to given dimensions 1, 2, using one of the two available processes, A and B. The following information on process cost and three-sigma standard deviation is available. The problem requires that the combined maximum allowable tolerance be limited to 12mils:

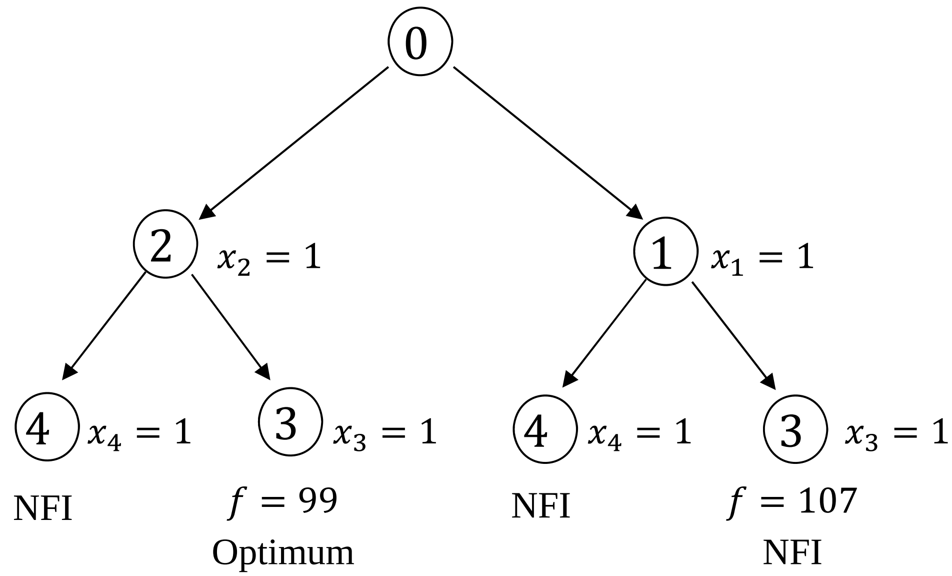
Process\Job	Job1		Job2	
	Cost	SD	Cost	SD
Process A	\$65	±4mils	\$57	±5mils
Process B	\$42	±6mils	\$20	±10mils

Example: Implicit Enumeration

- Let $x_i, i = 1, \dots, 4$ denote the available processes for both jobs; let t_i denote their tolerances. Then, the BIP problem is formulated as:

  $$\min_{\boldsymbol{x}} z = 65x_1 + 57x_2 + 42x_3 + 20x_4$$

  Subject to: $x_1 + x_2 = 1, x_3 + x_4 = 1, \sum_i t_i \leq 12, x_i \in \{0,1\}$
- The problem is solved using Implicit Enumeration algorithm



Integer Programming Problems

- Consider the IP problem:

$$\max_{\boldsymbol{x}} z = \boldsymbol{c}^T \boldsymbol{x}, \text{ Subject to } \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b}, \ \boldsymbol{x} \in \mathbb{Z}^n, \ \boldsymbol{x} \ge \boldsymbol{0}$$

- The LP relaxation of the IP problem is defined as:

$$\max_{\boldsymbol{x}} z = \boldsymbol{c}^T \boldsymbol{x}, \text{ Subject to } \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b}, \boldsymbol{x} \ge \boldsymbol{0}$$

- The LP relaxation solution serves as upper bound on the IP solution
- Two popular approaches to solve IP problems are:
  - Branch and bound method
  - Cutting plane method

Branch and Bound Method

- Start from the LP relaxation solution; successively introduce the integrality constraints on the design variables
  - Each integrality constraint on variable x_i introduces two new subprograms (branching) that can be solved via LP methods
  - Each subprogram results in one of the following:
    - There is no feasible solution.
    - The solution does not improve upon available IP solution.
    - An improved IP solution is returned and is recorded as current optimal.
- The progress of the algorithm is recorded in a decision-tree
- The algorithm ends when all branches have been fathomed

Example: Branch and Bound Method

- Consider the following IP problem:

  Max $z = 60x_1 + 90x_2 + 120x_3,$

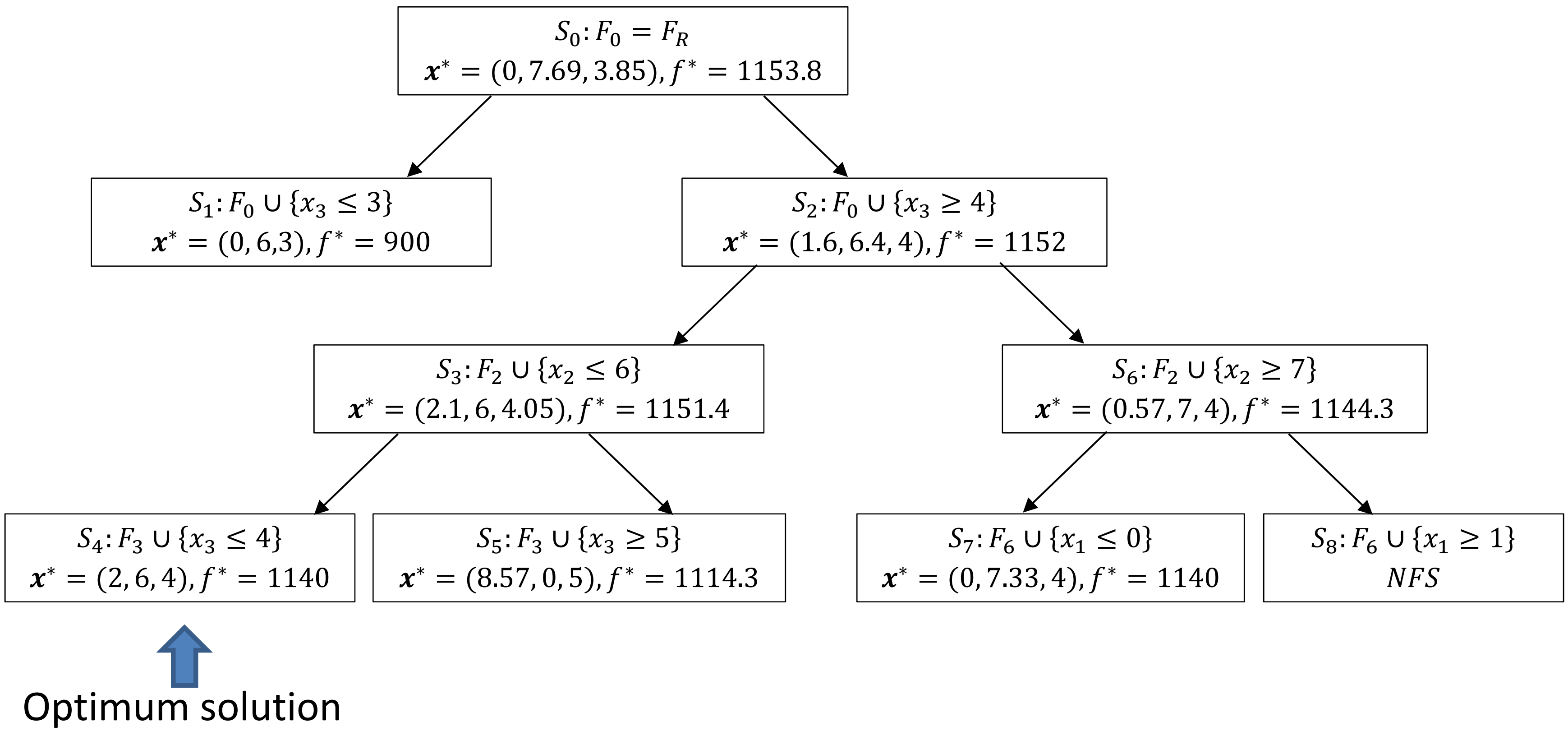
  Subject to: $35x_1 + 60x_2 + 140x_3 \leq 1000, 15x_1 + 30x_2 + 60x_3 \geq 2000,$
  $x_1 + x_2 - 2x_3 \leq 0;\ \boldsymbol{x} \in \mathbb{Z}^3,\ \boldsymbol{x} \geq \boldsymbol{0}$
- S_0: LP relaxation (F_0), $x_1^* = 0, x_2^* = 7.69, x_3^* = 3.85,\ f^* = 1153.8$; record as upper bound
- S_1: $F_0 \cup \{x_3 \leq 3\}$, integer solution: $x_1^* = 0, x_2^* = 6, x_3^* = 3,\ f^* = 900$; record as current optimum
- S_2: $F_0 \cup \{x_3 \geq 4\}$, non-integer solution: $x_1^* = 1.6, x_2^* = 6.4, x_3^* = 4,\ f^* = 1152$; further improvement possible
- S_3: $F_2 \cup \{x_2 \leq 6\}$, non-integer solution: $x_1^* = 2.1, x_2^* = 6, x_3^* = 4.05,\ f^* = 1151.4$; further improvement possible

Example: Branch and Bound Method

- S_4: $F_3 \cup \{x_3 \leq 4\}$, integer solution: $x_1^* = 2, x_2^* = 6, x_3^* = 4,\ f^* = 1140$; record as the new optimum
- S_5: $F_3 \cup \{x_3 \geq 5\}$, non-integer solution: $x_1^* = 8.57, x_2^* = 0, x_3^* = 5,\ f^* = 1114.3$, which does not improve upon the current optimum. The branch is fathomed.
- S_6: $F_2 \cup \{x_2 \geq 7\}$, non-integer solution: $x_1^* = 0.57, x_2^* = 7, x_3^* = 4,\ f^* = 1144.3$, further improvement possible
- S_7: $F_6 \cup \{x_1 \leq 0\}$, non-integer solution: $x_1^* = 0, x_2^* = 7.33, x_3^* = 4,\ f^* = 1140$, which does not improve upon the current optimum. The branch is fathomed.
- S_8: $F_6 \cup \{x_1 \geq 1\}$ has no feasible solution. The branch is fathomed.
- All branches having been fathomed, the optimal solution is: $x^* = (2,6,4), f^* = 1140$.

Example: Branch and Bound Method

- The decision tree for the problem



Cutting Plane Method

- Consider IP problem:

  $\max_{\boldsymbol{x}} z = \boldsymbol{c}^T \boldsymbol{x}$, Subject to $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$, $\boldsymbol{x} \in \mathbb{Z}^n$, $\boldsymbol{x} \geq \boldsymbol{0}$,

- Assume that the optimal BFS for the LP relaxation is represented as:

  $\boldsymbol{I}\boldsymbol{x}_B + \boldsymbol{A}_N \boldsymbol{x}_N = \boldsymbol{b}$

- Next, use the floor operator to write the *i*th constraint as:

$$x_i + \sum_{j=m+1}^{n} \left(\lfloor a_{ij} \rfloor + \alpha_{ij}\right) x_j = \lfloor b_i \rfloor + \beta_i$$

- The above constraint further resolves into:

$$x_i + \sum_{j=m+1}^{n} \lfloor a_{ij} \rfloor x_j \leq \lfloor b_i \rfloor, \qquad \sum_{j=m+1}^{n} \alpha_{ij} x_j \geq \beta_i$$

- Note that an integer feasible solution satisfies both inequalities; whereas, the optimal non-integer solution does not satisfy the 2nd inequality (known as the Gomory cut)

Cutting Plane Method

- The introduction of the Gomory cut makes the current LP solution infeasible without losing any IP solutions
- The cutting plane algorithm generates a family of polyhedra which satisfy: $\Omega \supset \Omega_1 \supset \Omega_2 \supset \cdots \supset \Omega \cap \mathbb{Z}^n$, where $\Omega = \{x \in \mathbb{R}^n : \boldsymbol{Ax} \le \boldsymbol{b}\}$ denote the polyhedral associated with the LP relaxation problem.
- The cutting plane algorithm terminates in finite steps.

Example: Cutting Plane Method

- Consider the simplified manufacturing problem given as:

  $$\max_{x_1,x_2} f = 7.5x_1 + 5x_2$$

  Subject to: $10x_1 + 5x_2 \leq 60, 5x_1 + 12x_2 \leq 80; x_1, x_2 \in \mathbb{Z}$

- Using Simplex method, the final tableau is given as:

Basic	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{s_1}$	$\boldsymbol{s_2}$	Rhs
$\boldsymbol{x_1}$	1	0	0.126	-0.053	3.368
$\boldsymbol{x_2}$	0	1	-0.053	0.105	5.263
$-\mathbf{z}$	0	0	0.684	0.131	52.58

- The non-integer optimum solution is given as: $x_1^* = 3.368, x_2^* = 5.263,\ f^* = 52.58$.

Example: Cutting Plane Method

- The series of Gomory cuts that result in the optimum solution to the IP problem are given as:

No.	Cut	Optimal solution
1.	$6x_1 + 12x_2 \leq 83$	$x_1^* = 3.389, x_2^* = 5.222, f^* = 51.528$
1.	$10x_1 + 6x_2 \leq 65$	$x_1^* = 3.357, x_2^* = 5.238, f^* = 51.369$
1.	$7x_1 + 10x_2 \leq 70$	$x_1^* = 3.846, x_2^* = 4.308, f^* = 50.385$
1.	$7x_1 + 9x_2 \leq 65$	$x_1^* = 3.909, x_2^* = 4.182, f^* = 50.227$
1.	$7x_1 + 8x_2 \leq 60$	$x_1^* = 4, x_2^* = 4, f^* = 50$

Example: Cutting Plane Method

- Consider following IP problem:

Max $z = 60x_1 + 90x_2 + 120x_3,$

Subject to: $35x_1 + 60x_2 + 140x_3 \leq 1000, 15x_1 + 30x_2 + 60x_3 \geq 2000,$ $x_1 + x_2 - 2x_3 \leq 0;\ \boldsymbol{x} \in \mathbb{Z}^3,\ \boldsymbol{x} \geq \boldsymbol{0}$

LP relaxation solution: $x_1^* = 0, x_2^* = 7.69, x_3^* = 3.85,\ f^* = 1153.8$

Final Simplex tableau is:

Basic	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{x_3}$	$\boldsymbol{s_1}$	$\boldsymbol{s_2}$	$\boldsymbol{s_3}$	**Rhs**
$\boldsymbol{x_2}$	0.808	1	0	0.539	0.039	0	7.69
$\boldsymbol{x_3}$	-0.096	0	1	-0.231	0.019	0	3.85
$\mathbf{s_3}$	0.173	0	0	0.115	0.115	1	123.1
$-\mathbf{z}$	0.115	0	0	2.077	0.577	0	1153.8

Gomory Cuts for the Problem

- Consider the first constraint equation:
  $0.808x_1 + \mathrm{x}_2 + 0.539s_1 + 0.039s_2 = 7.692$
- The first cut is developed as: $0.808x_1 + 0.539s_1 + 0.039s_2 \geq 0.692$
- The following series of cuts progressively squeezes the feasible region and produces an integer solution: $\boldsymbol{x}^* = (2, 6, 4), f^* = 1140$

No.	Cut	Optimal solution
1.	$0.808x_1 + 0.539s_1 + 0.039s_2 - s_4 = 0.692$	$x_1^* = 0.857, x_2^* = 7, x_3^* = 3.929, f^* = 1152.9$
1.	$0.833s_1 + 0.024s_2 + 0.881s_4 - s_5 = 0.929$	$x_1^* = 2.162, x_2^* = 5.946, x_3^* = 4.054, f^* = 1151.3$
1.	$0.054s_1 + 0.973s_2 + 0.135s_5 - s_6 = 0.946$	$x_1^* = 2.083, x_2^* = 5.972, x_3^* = 4.028, f^* = 1145.8$
1.	$0.056s_1 + 0.139s_5 + 0.972s_6 - s_7 = 0.972$	$x_1^* = 2, x_2^* = 6, x_3^* = 4, f^* = 1140$