Introduction to Integer Programming

Pramesh Kumar

IIT Delhi

April 9, 2024

Linear Integer Program

For $\mathbf{c}, \mathbf{x} \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, and $\mathbf{b} \in \mathbb{R}^m$

$$
\begin{aligned}
Z = \underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{c}^T \mathbf{x} && \text{(1a)} \\
\text{subject to} \quad & A\mathbf{x} = \mathbf{b} && \text{(1b)} \\
& x_i \in \mathbb{Z}_+, i = 1, ..., p && \text{(1c)} \\
& x_i \in \mathbb{R}_+, i = p+1, ..., n && \text{(1d)}
\end{aligned}
$$

- p decision variables are integers
- $n - p$ decision variables are continuous

Binary Integer Program

$$
\begin{aligned}
Z = \underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{c}^T \mathbf{x} && \text{(2a)} \\
\text{subject to} \quad & A\mathbf{x} = \mathbf{b} && \text{(2b)} \\
& x_i \in \{0, 1\}, \forall i = 1, ..., n && \text{(2c)}
\end{aligned}
$$

- All decision variables can either take value 1 or 0.

Example: Knapsack Problem

Given a set of items N, each with a weight w_i and a value a_i, determine which items to include in the collection so that the total weight is less than or equal to a given limit W and the total value is as large as possible.

Let $x_i = \begin{cases} 1 & \text{if } i \text{ is picked} \\ 0 & \text{otherwise} \end{cases}$

$$Z = \underset{\mathbf{x}}{\text{maximize}} \quad \sum_{i=1}^{n} a_i x_i \tag{3a}$$

$$\text{subject to} \quad \sum_{i=1}^{n} w_i x_i \leq W \tag{3b}$$

$$x_i \in \{0, 1\}, \forall i = 1, ..., n \tag{3c}$$

Example: Uncapacitated Facility Location

Given a set of potential depots $N = \{1, \ldots, n\}$ and a set $M = \{1, \ldots, m\}$ of clients, suppose there is a fixed cost f_j associated with the opening of depot j, and a transportation cost c_{ij} if all of client i's order is delivered from depot j. The problem is to decide which depots to open and which depot serves each client so as to minimize the sum of the fixed and transportation costs.

Let $y_j = \begin{cases} 1 & \text{if depot } j \text{ is opened} \\ 0 & \text{otherwise} \end{cases}$ $x_{ij} = \begin{cases} 1 & \text{if } i\text{'s order are served from depot } j \\ 0 & \text{otherwise} \end{cases}$

$$\underset{\mathbf{x},\mathbf{y}}{\text{minimize}} \quad \sum_{j \in N} \sum_{i \in N} c_{ij} x_{ij} + \sum_{j \in n} f_j y_j \tag{4}$$

$$\text{subject to} \quad \sum_{j \in N} x_{ij} = 1, \forall i \in M \tag{5}$$

$$\sum_{i \in M} x_{ij} \leq m y_j, \forall j \in N \tag{6}$$

$$x_{ij} \geq 0, \forall i \in M, \forall j \in N \tag{7}$$

$$y_j \in \{0, 1\}, \forall j \in N \tag{8}$$

where, m is a large positive integer.

Example: Matching in bipartite graph

Let $x_{ij} = \begin{cases} 1 & \text{if } i \in L \text{ is matched to } j \in R \\ 0 & \text{otherwise} \end{cases}$

$$\underset{\mathbf{x}}{\text{minimize}} \quad \sum_{j \in R} \sum_{i \in L} w_{ij} x_{ij} \tag{9}$$

$$\text{subject to} \quad \sum_{j \in R} x_{ij} = 1, \forall i \in L \tag{10}$$

$$\sum_{i \in L} x_{ij} = 1, \forall i \in R \tag{11}$$

$$x_{ij} \in \{0, 1\}, \forall (i, j) \in A \tag{12}$$

$$\tag{13}$$

Example: Traveling salesman problem (TSP)

Given a set of cities N and cost of traveling from city i to city j denoted as c_{ij}, find a tour that visits all the cities in minimum total travel cost.

Dantzig-Fulkerson-Johnson (DFJ) formulation

$$x_{ij} = \begin{cases} 1, & \text{if they go directly from } i \text{ to } j \\ 0, & \text{otherwise} \end{cases}$$

$$\underset{x}{\text{minimize}} \quad \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij}$$

$$\text{subject to} \quad \sum_{j \in N : j \neq i} x_{ij} = 1, \forall i \in N$$

$$\sum_{i \in N : i \neq j} x_{ij} = 1, \forall j \in N$$

$$\sum_{i \in S} \sum_{j \notin S} x_{ij} \geq 1, \forall S \subset N, S \neq \phi *$$

$$x_{ij} = \{0, 1\}, \forall i \in N, \forall j \in N$$

Miller-Tucker-Zemlin (MTZ) formulation

$$x_{ij} = \begin{cases} 1, & \text{if they go directly from } i \text{ to } j \\ 0, & \text{otherwise} \end{cases}$$

$u_i =$ order in which city i is visited in the tour.

$$\underset{x,u}{\text{minimize}} \quad \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij}$$

$$\text{subject to} \quad \sum_{j \in N : j \neq i} x_{ij} = 1, \forall i \in N$$

$$\sum_{i \in N : i \neq j} x_{ij} = 1, \forall j \in N$$

$$u_1 = 1$$

$$2 \leq u_i \leq n, \forall i \neq 1$$

$$u_i - u_j + 1 \leq (|N| - 1)(1 - x_{ij}) \quad \forall i \neq 1, \forall j \neq 1$$

$$x_{ij} = \{0, 1\}, \forall i \in N, \forall j \in N$$

* One can replace these with $\sum_{i \in S, j \in S} x_{ij} \leq |S| - 1, \forall S \subset N, |S| > 1$

Example: Multicommodity capacitated fixed-charge network design

Given a directed network $G(N, A)$ and a set of commodities K to be routed according to the demand d^k from origin $O(k) \in N$ to destination $D(k)$ for each commodity k, the problem is to satisfy the demand in minimum transportation and fixed design costs without violating the capacity of links.

Let x_{ij}^k represents the flow of commodity k on link (i, j) and

$$y_{ij} = \begin{cases} 1, & \text{if } (i,j) \text{ is built} \\ 1, & \text{if } (i,j) \text{ otherwise} \end{cases}$$

$$\underset{\mathbf{x},\mathbf{y}}{\text{minimize}} \quad \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij}$$

$$\text{subject to} \quad \sum_{j \in FS(i)} x_{ij}^k - \sum_{j \in BS(i)} x_{ji}^k = \begin{cases} d^k, & \text{if } i = O(k) \\ -d^k, & \text{if } i = D(k) \\ 0, & \text{otherwise} \end{cases}, \forall i \in N, \forall k \in K$$

$$\sum_{k \in K} x_{ij}^k \leq u_{ij}, \forall (i,j) \in A$$

$$x_{ij}^k \geq 0, \forall (i,j) \in A, \forall k \in K$$

$$y_{ij} = \{0, 1\}, \forall (i,j) \in A$$

First attempt to solving an integer program

Let's relax the integral constraints and solve the linear program (which is easy) and then find an integral solution closer to the optimal solution (e.g., by rounding off) to the linear program.

- Such LP is called the LP relaxation of the integer program.
- It may work in some cases.
- In other cases, it may not even find a feasible solution, forget about the optimal.

Example

Integer Program

$$\begin{aligned} \underset{x_1,x_2}{\text{maximize}} \quad & 9x_1 + 17x_2 \\ \text{subject to} \quad & 3x_1 + 2x_2 \leq 11 \\ & 3x_2 \leq 11 \\ & \boxed{x_1, x_2 \in \mathbb{Z}_+} \end{aligned}$$

LP relaxation

$$\begin{aligned} \underset{x_1,x_2}{\text{maximize}} \quad & 9x_1 + 17x_2 \\ \text{subject to} \quad & 3x_1 + 2x_2 \leq 11 \\ & 3x_2 \leq 11 \\ & \boxed{x_1 \geq 0, x_2 \geq 0} \end{aligned}$$

Example

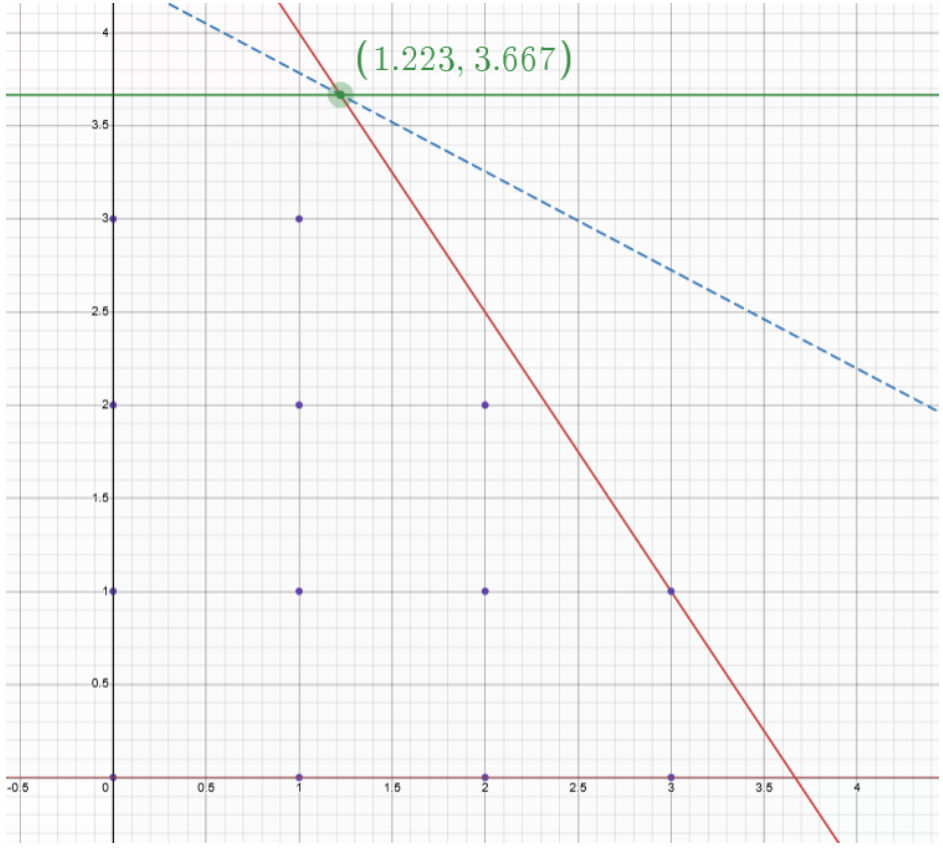

Figure: The feasible region is shown using points and the objective function is shown using dashed line

- The optimal solution of LP relaxation is $(1.223, 3.667)$ with objective value $= 73.346$.
- If we try to round it, we get $(1, 4)$ which is not even a feasible solution.

LP relaxation

Definition (Relaxation). Let $(P)\ z^* = \sup\{f(\mathbf{x}) \ : \ x \in S\}$ be an optimization problem. We say that another problem (R) $z^{'} = \sup\{g(\mathbf{x}) \ : \ x \in T\}$ is a relaxation of (P) if

1. $S \subseteq T$
2. $f(\mathbf{x}) \leq g(\mathbf{x}), \forall x \in S$

Proposition

LP relaxation is a relaxation.

Remark.

1. The optimal value of LP relaxation of a maximization IP problem provides you an UB on the optimal objective value of IP. The integrality gap is given by $z^{LP} - z^{IP}$
2. The optimal value of LP relaxation of a minimization IP problem provides you an LB on the optimal objective value of IP. The integrality gap is given by $z^{IP} - z^{LP}$.

Good news

Theorem

Is the optimal solution to the LP relaxation is feasible to IP, then it must be optimal to the IP.

Question. When does solving the LP relaxation gives integral solution?

Totally unimodularity

Definition (Totally unimodular). A matrix A is totally unimodular (TU) if every square sub-matrix of A has determinant 1, -1, or 0.

Theorem (Sufficient condition for TU)

Let $A \in \mathbb{Z}^{m \times n}$. Then, A is TU if

1. *Each entry in A, i.e., a_{ij} is 0, 1, or -1.*
2. *Each column of A has at most two non-zero entries.*
3. *There exists a partition (M_1, M_2) of the set of rows $M = \{1, 2, \cdots, m\}$ such that each column j containing two non-zero entries satisfies $\sum_{i \in M_1} a_{ij} - \sum_{i \in M_2} a_{ij} = 0$.*

Example(s). The incidence matrix of a directed graph is TU.

Example

$$A = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & 1 & 0 & -1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Let $M_1 = \{1, 2, 3\}$ and $M_2 = \{4\}$. Then

$$\sum_{i \in M_1} a_{ij} - \sum_{i \in M_2} a_{ij} = 0, \forall j$$

.

Therefore, A is TU.

Theorem (Hoffman and Kruskal's Theorem)

Let $A \in \mathbb{Z}^{m \times n}$. The polyhedron $\{\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0\}$ is integral for every $b \in \mathbb{Z}^m$ if and only if A is totally unimodular.

Remark. The common misconception is that the only way you can get integral polyhedra if A is TU; not true.

Solving IP

- Generally, solving IP is hard.
- There is no polynomial time algorithm to solve an IP.
- The most commonly applied technique to exactly solve IP is branch-and-bound method proposed by Land and Doig (1960).

Branch-and-bound method

- It is a divide-and-conquer strategy.
- We divide the feasible set S into disjoint subsets $S_1, S_2, .., S_k$.
- We optimize our objective function over each subset. The motivation is that if we cannot solve the original problem directly, we can try solving the smaller subproblems.
- Dividing the original problem into subproblems is called branching. The subproblems can be used to obtain the bounds on original problem.

Proposition

Consider the problem $z^* = \max\{\mathbf{c}^T\mathbf{x} \mid \mathbf{x} \in S\}$.
Let $S = S_1 \cup S_2 \cup \cdots \cup S_k$ *s.t.* $S_i \cap S_j = \phi, \forall i \neq j$ *be a decomposition of* S *into smaller subsets. Let* $z_k^* = \max\{\mathbf{c}^T\mathbf{x} \mid x \in S_k\}$. *Assume that* $z_k^* = \max\{\mathbf{c}^T\mathbf{x} \mid \mathbf{x} \in S_k\}$ *and assume that* $\underline{z}_k \leq z_k^* \leq \overline{z}_k, \forall k = 1, \cdots K$.
Then,

1. $z^* = \max z_k^*$
2. $\max_{k=1,\cdots,K} \underline{z}_k \leq z_k^* \leq \max_{k=1,\cdots,K} \overline{z}_k$

Example

Integer Program

$$\begin{aligned} \underset{x_1,x_2}{\text{maximize}} \quad & 9x_1 + 17x_2 \\ \text{subject to} \quad & 3x_1 + 2x_2 \leq 11 \\ & 3x_2 \leq 11 \\ & \boxed{x_1, x_2 \in \mathbb{Z}_+} \end{aligned}$$

$(x_1^*, x_2^*) = (1, 3)$,
$z^* = 60$

LP relaxation

$$\begin{aligned} \underset{x_1,x_2}{\text{maximize}} \quad & 9x_1 + 17x_2 \\ \text{subject to} \quad & 3x_1 + 2x_2 \leq 11 \\ & 3x_2 \leq 11 \\ & \boxed{x_1 \geq 0, x_2 \geq 0} \end{aligned}$$

$(x_1^*, x_2^*) = (1.222, 3.667)$,
$z^{LP} = 73.333$

- By solving the LP relaxation, we get $(x_1^*, x_2^*) = (1.222, 3.667)$ with objective value $z = 73.333$. Clearly, this is an UB on the optimal objective value.
- We know that either $x_1 \leq 1$ or $x_1 \geq 2$; let's use this disjunction to create two subproblems S1 and S2.

Example

S1

$$\begin{aligned} \underset{x_1,x_2}{\text{maximize}} \quad & 9x_1 + 17x_2 \\ \text{subject to} \quad & 3x_1 + 2x_2 \leq 11 \\ & 3x_2 \leq 11 \\ & \boxed{x_1 \leq 1} \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

$(x_1^*, x_2^*) = (1, 3.667)$,
$z^{S1} = 71.333$

S2

$$\begin{aligned} \underset{x_1,x_2}{\text{maximize}} \quad & 9x_1 + 17x_2 \\ \text{subject to} \quad & 3x_1 + 2x_2 \leq 11 \\ & 3x_2 \leq 11 \\ & \boxed{x_1 \geq 2} \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

$(x_1^*, x_2^*) = (2, 2.5)$,
$z^{S2} = 60.5$

- Solving S1 and S2 does not produce an integer solution.
- In S1, since x_2 is fractional, we know that either $x_2 \leq 3$ or $x_2 \geq 4$. Let's further divide S1 using disjunctions $x_2 \leq 3, x_2 \geq 4$ and create subproblems S3 and S4

Example

S3

$$\begin{aligned} \underset{x_1,x_2}{\text{maximize}} \quad & 9x_1 + 17x_2 \\ \text{subject to} \quad & 3x_1 + 2x_2 \leq 11 \\ & 3x_2 \leq 11 \\ & x_1 \leq 1 \\ & \boxed{x_2 \leq 3} \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

$(x_1^*, x_2^*) = (1, 3)$,
$z^{S1} = 60$

S4

$$\begin{aligned} \underset{x_1,x_2}{\text{maximize}} \quad & 9x_1 + 17x_2 \\ \text{subject to} \quad & 3x_1 + 2x_2 \leq 11 \\ & 3x_2 \leq 11 \\ & x_1 \leq 1 \\ & \boxed{x_2 \geq 4} \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

INFEASIBLE

- Solving S3 produces an integer solution. This help us setting a LB value, i.e., $z^{LB} = 60$.
- We do not explore S3 further since for this branch this is optimal since it is feasible to original LP. We call this step prunning the branch by optimality.
- S4 is infeasible and therefore we do not explore this problem further. We call this step prunning the branch by infeasibility.
- Let's further divide S2 using disjunctions $x_2 \leq 2$ or $x_2 \geq 3$ and create subproblems S5 and S6.

Example

S5

$$\begin{aligned} \underset{x_1,x_2}{\text{maximize}} \quad & 9x_1 + 17x_2 \\ \text{subject to} \quad & 3x_1 + 2x_2 \leq 11 \\ & 3x_2 \leq 11 \\ & x_1 \geq 2 \\ & \boxed{x_2 \leq 2} \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

$(x_1^*, x_2^*) = (2.333, 2)$,
$z^{S5} = 55$

S6

$$\begin{aligned} \underset{x_1,x_2}{\text{maximize}} \quad & 9x_1 + 17x_2 \\ \text{subject to} \quad & 3x_1 + 2x_2 \leq 11 \\ & 3x_2 \leq 11 \\ & x_1 \geq 2 \\ & \boxed{x_2 \geq 3} \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

INFEASIBLE

- Solving S5 does not produce an integer solution. This provides us with a bound which is even lesser than the LB. So, we do not explore this further. We call this step prunning by bound.
- S6 is infeasible, so we prune it by infeasibility.
- Since there is nothing to explore further, the current best integer solution $(x_1^*, x_2^*) = (1, 3)$ is optimal with objective value $z^* = 60$.

Final remarks

- Cutting plane algorithm is another way of solving IP.
- When combined with branch-and-bound, the method becomes branch-and-cut (which is currently implemented in many solvers).
- You need a good understanding of polyhedral theory to advance in this area.

Thank you!