CE 205A
Transportation Logistics

Lecture 3

Integer Programming

Lecture Outline

1. Integer Programs
2. Strong Formulations

Lecture Outline

Integer Programs

Integer Programs

Introduction

An Integer Program (IP) additionally restricts the decision variables to take integer values

$$\begin{aligned} \min\ & \mathbf{c}^\mathsf{T}\mathbf{x} \\ \text{s.t. } & \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & \mathbf{x} \geq 0 \\ & \mathbf{x} \in \mathbb{Z}_+^n \end{aligned}$$

Usually, non-negativity constraints are redundant since the **x** variables are restricted to non-negative integers.

$$\begin{aligned} \min\ & \mathbf{c}^\mathsf{T}\mathbf{x} \\ \text{s.t. } & \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & \mathbf{x} \in \mathbb{Z}_+^n \end{aligned}$$

Integer Programs

Variants

In many problems, the decision variables are 0-1 integer that indicates true-false type situations.

$$\begin{aligned} \min\ & \mathbf{c}^{\mathsf{T}}\mathbf{x} \\ \text{s.t. } & \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & \mathbf{x} \in \{0,1\}^n \end{aligned}$$

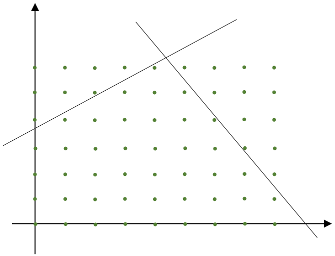
It is also common to have both continuous and discrete variables in a single formulation. These are called Mixed Integer Programs (MIP).

$$\begin{aligned} \min\ & \mathbf{c}^{\mathsf{T}}\mathbf{x} + \hat{\mathbf{c}}^{\mathsf{T}}\mathbf{y} \\ \text{s.t. } & \mathbf{A}\mathbf{x} + \hat{\mathbf{A}}\mathbf{y} \geq \mathbf{b} \\ & \mathbf{x} \geq 0, \mathbf{y} \in \mathbb{Z}_+^n \end{aligned}$$

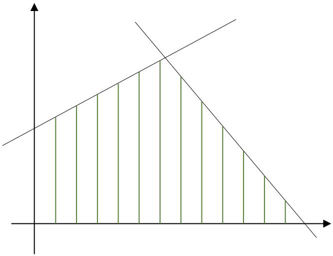
Integer Programs

Geometry

Integer Program



Mixed Integer Program



Are 'corner points' solutions optimal? How does the convex hull of the MIP problem look like?

Suppose we have a travel bag of capacity b kgs. Assume there are n items that can be included in the bag. For item j, the utility is c_j and the weight is a_j respectively. Formulate an optimization problem to maximize the utility.

$$
\begin{aligned}
\max \; & \sum_{j=1}^{n} c_j x_j \\
\text{s.t.} \; & \sum_{j=1}^{n} a_j x_j \leq b \\
& x_j \in \{0,1\} \; \forall \, j = 1, \ldots, n
\end{aligned}
$$

Integer Programs

Example 2: Generalized Assignment Problem

Suppose there are m agents and n jobs. Each agent i can perform job j in t_{ij} units of time. The amount of time available for agent i is s_i.

Assigning agent i to job j yields a profit of p_{ij}. Formulate an IP to allocate agents to jobs to maximize profits while ensuring that all jobs are served.

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} x_{ij} & \\
\text{s.t.} \quad & \sum_{i=1}^{m} x_{ij} = 1 & \forall j = 1, \ldots, n \\
& \sum_{j=1}^{n} t_{ij} x_{ij} \leq s_i & \forall i = 1, \ldots, m \\
& x_{ij} \in \{0, 1\} & \forall\, i = 1, \ldots, m, j = 1, \ldots, n
\end{aligned}
$$

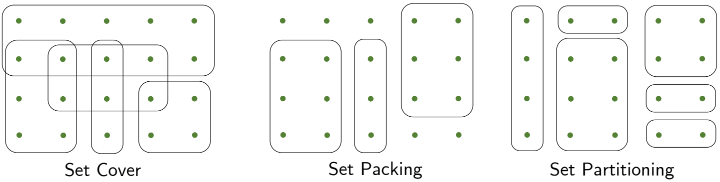
Integer Programs

Example 3: Set Cover, Packing, and Partitioning Problems

Consider a set $S = \{1, \ldots, m\}$. Let $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ be a collection of subsets of S. These subsets could for instance satisfy some property.

- A collection $X \subseteq \mathcal{S}$ is a *cover* of S if $\cup_{S_i \in X} S_i = S$
- X is a *packing* if $S_i \cap S_j = \emptyset \, \forall \, S_i, S_j \in X$
- X is a *partition* if it is both a cover and a packing.

E.g., Let $S = \{1, 2, 3, 4, 5\}$. What are the members of $\mathcal{S}$ if every element of it has a cardinality of at least 2?



Construct an example of a cover, packing, partition for the above example?

Imagine that each subset S_i has a weight w_i. For the set covering problem, we wish to find a cover that minimizes the sum of the weights of the chosen subsets. Formulate this as an optimization problem.

Define an incidence matrix $\mathbf{A}$ whose rows are the elements of S, i.e., $1, \ldots, m$ and columns $1, \ldots, n$ represent set membership in $\mathcal{S}$. That is, a_{ij} is 1 if the ith element is present in the jth subset.

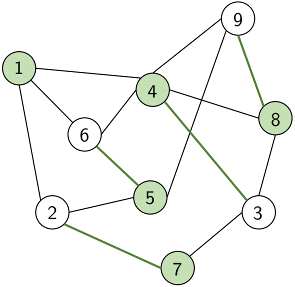
$$
\begin{aligned}
\min \ & \sum_{j=1}^{n} w_j x_j & \\
\text{s.t.} \ & \sum_{j=1}^{n} a_{ij} x_j \geq 1 & \forall i = 1, \ldots, m \\
& x_j \in \{0, 1\} & \forall j = 1, \ldots, n
\end{aligned}
$$

Can you solve a similar problem for the packing version? Does a minimization objective make sense?

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{n} w_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \leq 1 && \forall i = 1, \ldots, m \\
& x_j \in \{0, 1\} && \forall j = 1, \ldots, n
\end{aligned}
$$

What about the partitioning problem? Can be formulated as a minimization or maximization problem with equality constraints.

Consider an undirected graph $G = (V, E)$. A *matching* $M \subseteq E$ is a set of disjoint edges (edges that do not have a node in common). A *node cover* is a set $N \subseteq V$ such that every edge has at least one end point in N.



Formulate the maximum cardinality matching and minimum cardinality cover problems using the set cover/packing/partitioning framework.

Integer Programs

Example 4: Bin Packing Problem

Bin packing problem is another generalization of the knapsack problem in which there are n items and n bins. Item j has a weight w_j and each bin has a capacity c.

The goal is to minimize the number of bins used without exceeding the capacity limits. Formulate this as an optimization problem.

Integer Programs

Example 4: Bin Packing Problem

Let y_i be 1 if bin i is used and is 0 otherwise. Let x_{ij} take a value 1 if item j is assigned to bin i and is 0 otherwise.

$$
\begin{aligned}
\max \ & \sum_{i=1}^{n} y_i \\
\text{s.t.} \ & \sum_{j=1}^{n} w_j x_{ij} \leq c y_i && \forall i = 1, \dots, n \\
& \sum_{i=1}^{n} x_{ij} = 1 && \forall j = 1, \dots, n \\
& y_i \in \{0, 1\} && \forall i = 1, \dots, n \\
& x_{ij} \in \{0, 1\} && \forall i = 1, \dots, n, j = 1, \dots, n
\end{aligned}
$$

The first constraint is an example of *forcing constraints* of the type $x \leq My$, where one of the variables is allowed to take a non-negative quantity only if the other is active.

Consider a multi-period production planning problem in which producing an item in period t involves a fixed cost f_t. The unit production cost is p_t and h_t is the per-period inventory cost. Let the demand in period t be d_t.

Determine how much supply must be produced in each period (if any) to minimize total costs.

Let x_t be the demand produced in t and let y_t indicate if production happens in period t. Assume that s_t is the stock at the end of period t.

$$
\begin{aligned}
\max \ & \sum_{t=1}^{n} (p_t x_t + h_t s_t + f_t y_t) & \\
\text{s.t. } & s_{t-1} + x_t = d_t + s_t & \forall t = 1, \ldots, n \\
& x_t \leq M y_t & \forall t = 1, \ldots, n \\
& s_0 = 0, s_t, x_t \geq 0 & \forall t = 1, \ldots, n \\
& y_t \in \{0, 1\} & \forall t = 1, \ldots, n
\end{aligned}
$$

Note that the fixed cost applies only if $x_t > 0$. What is a good M?

Integer Programs

Example 6: Facility Location Problem

Suppose you run a logistics company. You can open your offices at any of n potential locations in the city and there is a fixed cost of opening a branch at node j, which is denoted by f_j.

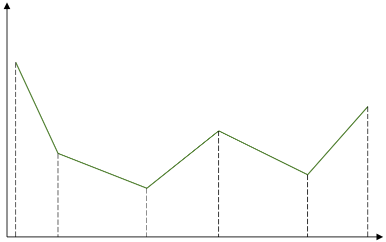
You can serve customer demand at m locations from any of the branches. The cost of serving customer at i from branch j is c_{ij}. Where should you open branches and how do you pair customers and branches?

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{n} f_j y_j + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} & \\
\text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} = 1 & \forall\, i = 1, \ldots, m \\
& x_{ij} \leq y_j & \forall\, i = 1, \ldots, m, j = 1, \ldots, n \\
& x_{ij} \in \{0, 1\} & \forall\, i = 1, \ldots, m, j = 1, \ldots, n \\
& y_j \in \{0, 1\} & \forall\, j = 1, \ldots, n
\end{aligned}
$$

Consider a piece-wise linear objective function f with $a_1, \ldots, a_k$ break points as shown in the figure.

Any feasible x can be written as a convex combination of the a values, i.e., $\sum_{i=1}^{k} \lambda_i a_i$. Are the λ values unique?

Suppose y_i is 1 if $x \in [a_i, a_{i+1}]$, and zero otherwise. If $y_i = 1$, we can write $x = \lambda_i a_i + \lambda_{i+1} a_{i+1}$.



Assume that λs are also decision variables. To make them unique, we can let λs take non-negative values depending on ys. Formulate the problem of minimizing the function as a MIP.

If y_i is 1, only λ_i and λ_{i+1} can take non-negative values. Suppose the objective is separable with respect to the decision variables.

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{k} \lambda_i f_i(a_i) & \\
\text{s.t.} \quad & \sum_{i=1}^{k} \lambda_i = 1 & \\
& \sum_{i=1}^{k-1} y_i = 1 & \\
& \lambda_1 \leq y_1 & \\
& \lambda_i \leq y_{i-1} + y_i & \forall i = 2, \ldots, k-1 \\
& \lambda_k \leq y_{k-1} & \\
& \lambda_i \geq 0 & \forall i = 1, \ldots, k \\
& y_i \in \{0, 1\} & \forall i = 1, \ldots, k-1
\end{aligned}
$$

Suppose we have two either-or-or type constraints $\sum_{j=1}^{n} a_{1j}x_j \leq b_1$ and $\sum_{j=1}^{n} a_{2j}x_j \leq b_2$, where at least one of them must be satisfied.

$$\sum_{j=1}^{n} a_{1j}x_j \leq b_1 + M_1(1 - y_1)$$
$$\sum_{j=1}^{n} a_{2j}x_j \leq b_2 + M_2(1 - y_2)$$
$$y_1 + y_2 = 1$$
$$y_1, y_2 \in \{0, 1\}$$

- What is a good choice of M_1 and M_2? How do you model the problem if the first constraint was of the $\geq$ type?
- What if we had m constraints of which at least k constraints must be satisfied?

Disjunctive constraints are common in many problems such as scheduling. Consider two jobs with processing times p_1 and p_2. Depending on the order in which they are carried out, the start times must satisfy either one of the two constraints $t_2 \geq t_1 + p_1$ and $t_1 \geq t_2 + p_2$ must hold.

In special cases, where the constraints are of the form $\sum_{j=1}^{n} a_{1j}x_j \geq b_1$ and $\sum_{j=1}^{n} a_{2j}x_j \geq b_2$ and the coefficients of the constraints are all non-negative and $\mathbf{x} \geq \mathbf{0}$, it is possible to model disjunctive constraints without using M.

$$\sum_{j=1}^{n} a_{1j}x_j \geq y_1 b_1$$

$$\sum_{j=1}^{n} a_{2j}x_j \geq y_2 b_2$$

$$y_1 + y_2 = 1$$

$$y_1, y_2 \in \{0, 1\}$$

Suppose we have a scenario where if a constraint $\sum_{j=1}^{n} a_{1j}x_j \leq b_1$ is satisfied, then $\sum_{j=1}^{n} a_{2j}x_j \leq b_2$ must hold.

This is equivalent to $\sum_{j=1}^{n} a_{1j}x_j > b_1$ or $\sum_{j=1}^{n} a_{2j}x_j \leq b_2$.

Using strict inequalities in pure integer programming models is not an issue unlike in LPs. (Why?) However, since the solution methods exploit LP methods, we use sufficiently small ϵ variables.

Thus, we can recast the problem as disjunctive constraints $\sum_{j=1}^{n} a_{1j}x_j \geq b_1 + \epsilon$ or $\sum_{j=1}^{n} a_{2j}x_j \leq b_2$.

Integer Programs

Example 10: Product Terms

In some cases it is possible to eliminate product terms of the type $x_1 x_2$ using integer variables.

Case 1: Suppose both x_1 and x_2 are binary.

$$
\begin{aligned}
& y \leq x_1 \\
& y \leq x_2 \\
& y \geq x_1 + x_2 - 1 \\
& y \in \{0, 1\}
\end{aligned}
$$

Case 2: Suppose x_1 is binary and $x_2 \in [0, u]$ is continuous.

$$
\begin{aligned}
& y \leq u x_1 \\
& y \leq x_2 \\
& y \geq u x_1 + x_2 - u \\
& y \in \{0, 1\}
\end{aligned}
$$

Integer Programs

Example 10: Product Terms

Case 3: Suppose both $x_1 \in [l_1, u_1]$ and $x_2 \in [l_2, u_2]$ are continuous. In this case, it is not possible to convert product terms to linear variables. Instead, one could approximate it as a separable objective and use a piece-wise linear approximation.

$$\begin{aligned} y_1 &= 0.5(x_1 + x_2) \\ y_2 &= 0.5(x_1 - x_2) \\ x_1 x_2 &= y_1^2 - y_2^2 \end{aligned}$$

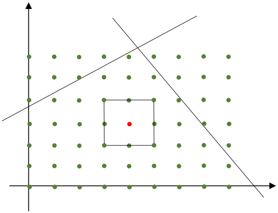
Integer Programs

Example 11: Exclusions

Suppose we wish to exclude a certain point $\mathbf{y}$ from the feasible region $X = \{\mathbf{x} \in \mathbb{Z}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$.

We can add a constraint to eliminate the boxed portion around $\mathbf{y}$.

$$\sum_{i=1}^{n} |x_i - y_i| \geq 1$$



However, this is non-linear, and hence we introduce auxiliary variables $z_i = |x_i - y_i|$. But we do not know how the xs appear in the objective to try a min-max trick. Hence, we replace the above constraint with

$$\sum_{i=1}^{n} z_i \geq 1$$

$$z_i \leq |x_i - y_i| \qquad \forall i = 1, \ldots, n$$

Integer Programs

Example 11: Exclusions

If $x_i - y_i \geq 0$, then we want the second constraint to take the form $z_i \leq x_i - y_i$, else it would imply $z_i \leq -(x_i - y_i)$. Thus, we write

$$\begin{aligned} &\sum_{i=1}^{n} z_i \geq 1 \\ &z_i \leq x_i - y_i + M_i w_i && \forall i = 1, \ldots, n \\ &z_i \leq -(x_i - y_i) + M_i(1 - w_i) && \forall i = 1, \ldots, n \\ &w_i \in \{0, 1\} && \forall i = 1, \ldots, n \end{aligned}$$

But the ws should be connected to $x - y$. Specifically, if $x_i - y_i \geq 0$, we want w_i to be 1 and 0 otherwise. To model this, we add the following inequalities

$$\begin{aligned} &x_i - y_i \leq N_i w_i && \forall i = 1, \ldots, n \\ &-(x_i - y_i) \leq N_i(1 - w_i) && \forall i = 1, \ldots, n \end{aligned}$$

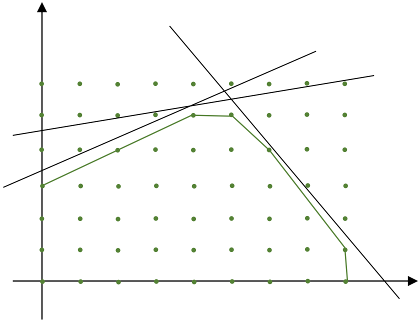
How do you set constraints that force a decision variable x to take values only in $\{a_1, \ldots, a_n\}$?

$$
\begin{aligned}
& x = \sum_{i=1}^{n} y_i a_i \\
& \sum_{i=1}^{n} y_i = 1 \\
& y_i \in \{0,1\} && \forall i = 1, \ldots, n
\end{aligned}
$$

This is an example of a special ordered set (SOS) constraint that we will revisit in subsequent lectures.

Strong Formulations

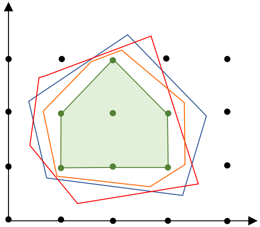
Most methods for solving integer programs rely on relaxations and LP solutions.



An ideal LP relaxation coincides with the convex hull of feasible points. (Why?)

Consider the following feasible set of points

$$X = \{(1,1),(2,1),(3,1),(1,2),(2,2),(3,2),(2,3)\}$$



There are infinitely many relaxation formulations that can lead to this feasible region.

Can you tell which of the three formulations P_1, P_2, and P_3 are strong? What about their LP relaxations?

In general, if P_1 and P_2 are two formulations of an IP, P_1 is a *stronger* formulation than P_2 if $P_1 \subset P_2$.

Suppose the feasible region of an IP problem is $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^k\}$. Then, the solution to the integer program

$$\begin{aligned} &\min\ \mathbf{c}^\mathsf{T}\mathbf{x} \\ &\text{s.t.}\mathbf{x} \in X \end{aligned}$$

is the same as the solution to the linear program

$$\begin{aligned} &\min\ \mathbf{c}^\mathsf{T}\mathbf{x} \\ &\text{s.t.}\mathbf{x} \in Conv(X) \end{aligned}$$

where $Conv(X) = \left\{ \sum_{i=1}^{k} \lambda_i \mathbf{x}^i : \sum_{i=1}^{k} \lambda_i = 1, \lambda_i \geq 0, \mathbf{x}^i \in X \right\}$

In theory, $Conv(X) = \{\mathbf{x} : \mathbf{A}^{conv}\mathbf{x} \leq \mathbf{b}^{conv}\}$, but finding the inequalities which make up this polyhedron is difficult. The above ideas extend to MIPs in the same way.

Consider three formulations for the knapsack problem.

$$P_1 = \{x \in [0,1]^4 : 83x_1 + 61x_2 + 49x_3 + 20x_4 \leq 100\}$$
$$P_2 = \{x \in [0,1]^4 : 4x_1 + 3x_2 + 2x_3 + 1x_4 \leq 4\}$$
$$P_3 = \{x \in [0,1]^4 : 4x_1 + 3x_2 + 2x_3 + 1x_4 \leq 4, x_1 + x_2 + x_3 \leq 1, x_1 + x_4 \leq 1\}$$

Do all of these formulations contain the same set of integer solutions? Can you order them on the basis of the strength of the formulations? How are their LP relaxation solutions ordered?

In the facility location problem (say P_1), we set $x_{ij} \leq y_j$ to indicate that a customer at i can be paired to a branch at j only if it is open, i.e., $y_j = 1$.

Alternately, we can add all such constraints for $i = 1, \ldots, m$ and write a model P_2 for which

$$\sum_{i=1}^{m} x_{ij} \leq m y_j \qquad \forall j = 1, \ldots, n$$

Which of the two formulations is better? P_2 has fewer constraints than P_1, but $P_1 \subset P_2$. Imagine $x_1 \leq 1$ and $x_2 \leq 1$ vs. $x_1 + x_2 \leq 2$. (Although the integer solutions in these two examples is not the same.)

Hence, if you solve the LP relaxations of the two problems, you may notice that $z_{P_1}^{LP} \geq z_{P_2}^{LP}$.

Note that every $(x, y) \in P_1$ is in P_2 (Why?). To show that there exists $(x, y) \in P_2$ which is not in P_1, consider the special case where $m = kn$, where $k \geq 2$. Assign each k customers to each branch. Let $y_j = k/m \, \forall \, j = 1, \ldots, n$.

$x_{ij} \leq y_j$ in formulation P_1 is violated but $\sum_{i=1}^{m} x_{ij} \leq my_j$ holds in P_2.

Caution: Pay attention to formulations involving M. Choose the tightest possible M for stronger formulations. Else, the LP relaxations, are going to be weaker.

Strong Formulations

Comparing Extended Formulations

There is more than one way to skin a cat but not all formulations can be easily compared as we just saw. One could write alternate formulations that give the same optimal solutions but the spaces in which the feasible point lie could be different.

Consider the lot sizing problem from before. Where do the feasible solutions lie? Instead, define a new variable w_{kt} which is the quantity produced in period k to satisfy demand in period t. Can you model the constraints?

$$\sum_{k=1}^{t} w_{kt} = d_t \qquad \forall t = 1, \ldots, n$$

$$w_{kt} \leq d_t y_k \qquad \forall k \leq t, t = 1, \ldots, n$$

$$x_k = \sum_{t=k}^{n} w_{kt} \qquad \forall k = 1, \ldots, n$$

$$w_{kt} \geq 0 \qquad \forall k \leq t, t = 1, \ldots, n$$

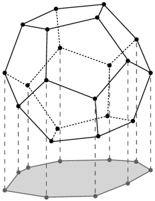
$$0 \leq y_t \leq 1 \qquad \forall t = 1, \ldots, n$$

Strong Formulations

Comparing Extended Formulations

Which of these two formulations are better? In such settings we make use of the idea of projections to compare them on even footing.

Suppose relaxation of formulation P_1 has decision variables $x \in P_1 \subseteq \mathbb{R}^n$. Consider a new formulation whose decision variables are of the type $(x, w) \in Q_2 = X \times W \subseteq \mathbb{R}^n \times \mathbb{R}^l$.

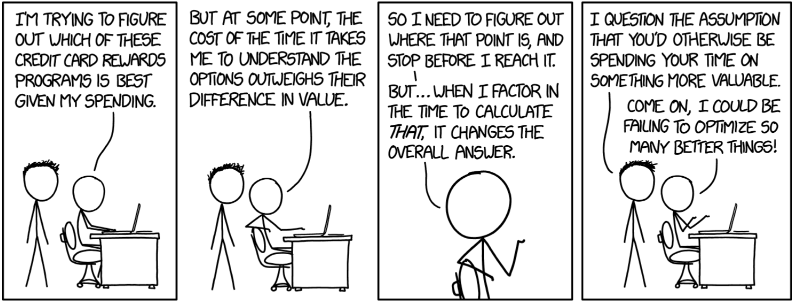


We then define a projection of Q_2 into the subspace $\mathbb{R}^n$ as follows

$$P_2 = proj_x(Q_2) = \{x \in \mathbb{R}^n : (x, w) \in Q_2 \text{ for some } w \in W\}$$

The new formulation is better only if $P_2 \subset P_1$. Using the point $x_t = d_t$ and $y_t = d_t/M$ can you comment on the strength of the two lot sizing formulations?

Your Moment of Zen



Source: xkcd