Lecture 12 Formulating Mixed-Integer Linear Programming Models

Can Li

ChE 597: Computational Optimization
Purdue University

Outline of this lecture

- Overview of the branch and cut algorithm.
  - One needs to have a basic understanding of the algorithm to understand what a good model means.
- How to formulate "good" MILP models?

MILP

$$\min Z = a^T x + b^T y$$
$$\text{s.t.} \quad Ax + By \leq d$$
$$x \in \mathbb{R}^{n_1} \quad y \in \mathbb{Z}^{n_2}$$

- y represent the integer variables. In most problems, y are binary variables $y \in \{0, 1\}^{n_2}$ (represent logic).
- A naive way to solve MILP is to fix the binary variables at 0 or 1 and solve the LPs.
- Pick the binary combination that yields the smallest objective.
- The caveat is that we need to solve 2^{n_2} LPs to enumerate all the binary combinations.
- The "branch-and-bound" algorithm is a smart way of enumeration.

Linear Program (LP) relaxation
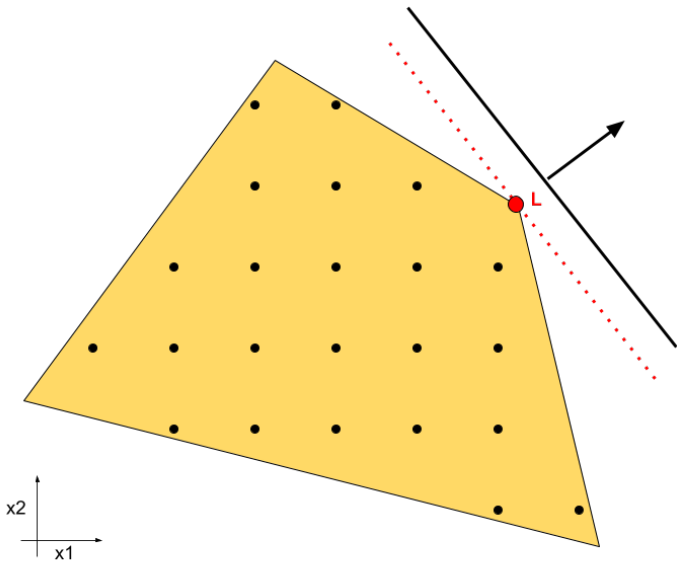
Branch-and-Bound

Split the LP recursively over a non-integral variable, i.e. $\exists i \leq p \mid x_i^\star \notin \mathbb{Z}$

$$x_i \leq \lfloor x_i^\star \rfloor \quad \vee \quad x_i \geq \lceil x_i^\star \rceil.$$
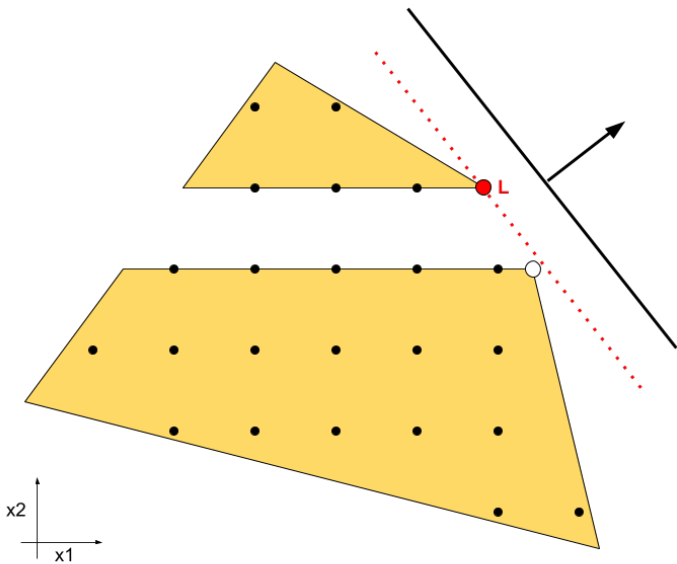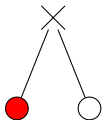
Lower bound (L): minimal among leaf nodes.
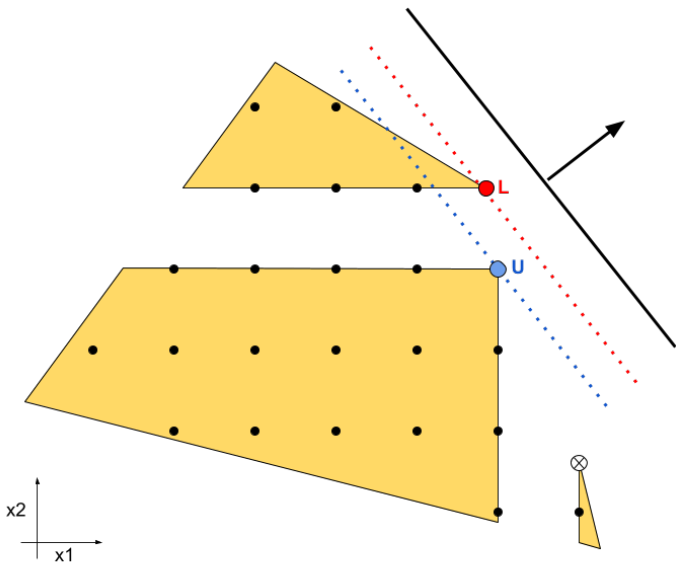Upper bound (U): minimal among leaf nodes with integral solution.
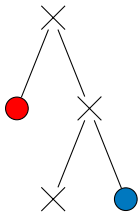
Stopping criterion:

- $L = U$ (optimality certificate)
- $L = \infty$ (infeasibility certificate)
- L - U ¡ threshold (early stopping)

Branch-and-Bound

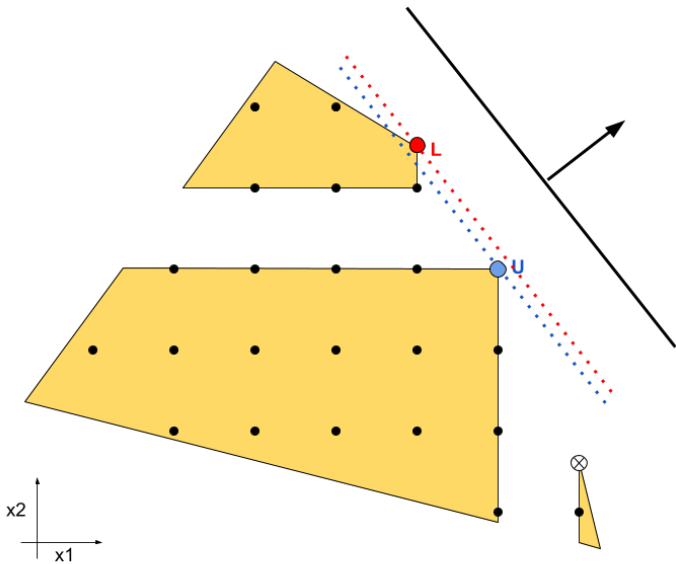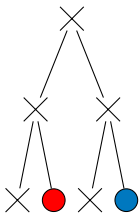# Branch-and-Bound
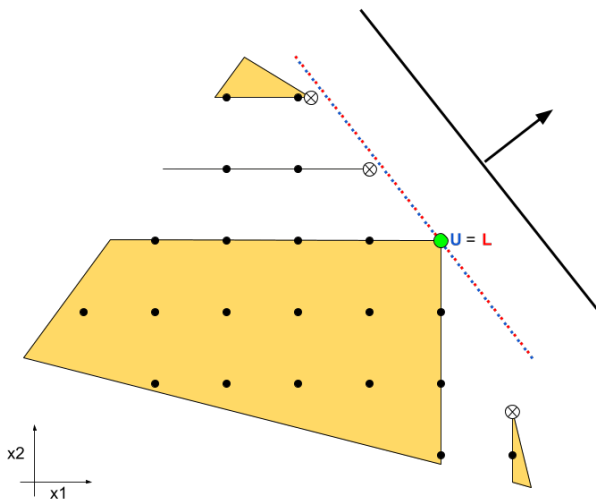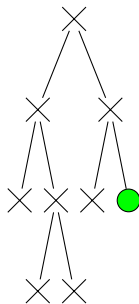
Cuts can be added to the sub-MILPs to tighten the bounds. (Branch-and-cut)

Cutting Planes

Cuts can be added to the sub-MILPs to tighten the bounds.
(Branch-and-cut)

Cuts can be added to the sub-MILPs to tighten the bounds. (Branch-and-cut)

Cutting Planes

Cutting planes can be added to tighten the LP relaxation. Tighter relaxations can reduce the number of nodes in the branch and bound algorithm. An extreme case is we have the convex hull of the feasible points where we only need to solve a single LP.

How to compare two MILP formulations of the same problem?
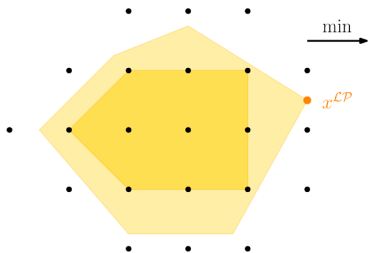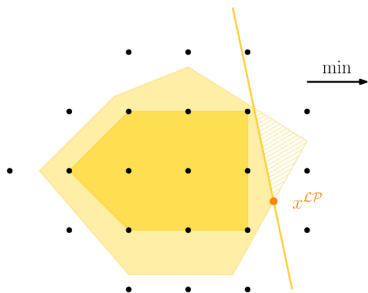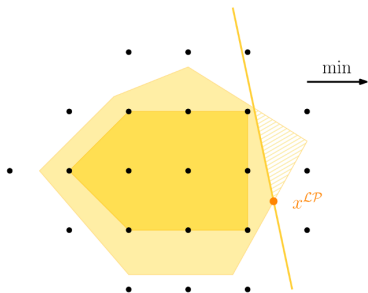
Rule of thumb Assume that $\{(x, y) : A_1 x + G_1 y \leq b_1, y \text{ integral}\}$ and $\{(x, y) : A_2 x + G_2 y \leq b_2, y \text{ integral}\}$ represent the same mixed integer set S and consider their linear relaxations $P_1 = \{(x, y) : A_1 x + G_1 y \leq b_1\}, P_2 = \{(x, y) : A_2 x + G_2 y \leq b_2\}$. If $P_1 \subset P_2$ the first representation is tighter. If $P_1 = P_2$ the two representations are equivalent and if $P_1 \backslash P_2$ and $P_2 \backslash P_1$ are both nonempty, the two representations are incomparable.

How to prove $P_1 \subseteq P_2$? Assume $P_1 \neq \emptyset$. $P_1 \subseteq P_2$ if and only if for every inequality $a_2 x + g_2 y \leq \beta_2$ in $A_2 x + G_2 y \leq b_2$ is implied by the inequalities $A_1 x + G_1 y \leq b_1$, i.e., there exists multipliers u such that $u A_1 = a_2, u G_1 = g_2, u b_1 \leq \beta_2, u \geq 0$.

Proof. (Farkas lemma. Theorem of alternative). One can view u as the Lagrangian multipliers.

Knapsack problem

Description We are given a knapsack that can carry a maximum weight b and there are n types of items that we could take, where an item of type i has weight $a_i > 0$. We want to load the knapsack with items without exceeding the knapsack capacity b.

$$S := \left\{ x \in \{0,1\}^n : \sum_{i=1}^{n} a_i x_i \leq b, x \geq 0 \right\}$$

$$\max \left\{ \sum_{i=1}^{n} c_i x_i : x \in S \right\}$$

Knapsack formulations

Standard Formulation

$$K := \left\{ x \in \{0,1\}^n : \sum_{i=1}^{n} a_i x_i \leq b \right\}$$

Minimal Cover Formulation

$$K^C := \left\{ x \in \{0,1\}^n : \sum_{i \in C} x_i \leq |C| - 1 \text{ for every minimal cover } C \text{ for } K \right\}$$

A subset C of indices is a cover for K if $\sum_{i \in C} a_i > b$ and it is a **minimal cover** if $\sum_{i \in C \setminus \{j\}} a_i \leq b$ for every $j \in C$. That is, C is a cover if the knapsack cannot contain all items in C, and it is minimal if every proper subset of C can be loaded.

Theorem The sets $K = K^C$.

Proof. It suffices to show that (i) if C is a minimal cover of K, the inequality $\sum_{i \in C} x_i \leq |C| - 1$ is valid for K. This is true by definition of the minimal cover.

(ii) the inequality $\sum_{i=1}^{n} a_i x_i \leq b$ is valid for K^C. Let $\bar{x}$ be a vector in K^C and let $J := \{j : \bar{x}_j = 1\}$. Suppose $\sum_{i=1}^{n} a_i \bar{x}_i > b$ or equivalently $\sum_{i \in J} a_i > b$. Let C be a minimal subset of J such that $\sum_{i \in C} a_i > b$. Then obviously C is a minimal cover and $\sum_{i \in C} \bar{x}_i = |C|$. This contradicts the assumption $\bar{x} \in K^C$ and the second statement is proved.

Consider an illustrative example

$$K := \left\{ x \in \{0,1\}^3 : 3x_1 + 3x_2 + 3x_3 \leq 5 \right\}.$$

Its minimal cover formulation is

$$K^C := \left\{ x \in \{0,1\}^3 : \begin{array}{rrrr} x_1 & +x_2 & & \leq 1 \\ x_1 & & +x_3 & \leq 1 \\ & x_2 & +x_3 & \leq 1 \end{array} \right\}$$

Clearly, we can put at most one item in the knapsack.

What about their LP relaxations?

$P := \left\{ x \in [0,1]^3 : 3x_1 + 3x_2 + 3x_3 \leq 5 \right\}$, and

$$P^C := \left\{ x \in [0,1]^3 : \begin{array}{rrrl} x_1 & +x_2 & & \leq 1 \\ x_1 & & +x_3 & \leq 1 \\ & x_2 & +x_3 & \leq 1 \end{array} \right\}$$

By summing up the three inequalities in P^C we get

$$2x_1 + 2x_2 + 2x_3 \leq 3$$

which implies $3x_1 + 3x_2 + 3x_3 \leq 5$. Thus $P^C \subseteq P$. The inclusion is strict since, for instance $\left(1, \frac{2}{3}, 0\right) \in P \backslash P^C$. In other words, $P^C \subsetneq P$.

When is a formulation "perfect"?

Definition A perfect formulation of a set $S \subseteq \mathbb{R}^n$ is a linear system of inequalities $Ax \leq b$ such that $\text{conv}(S) = \{x \in \mathbb{R}^n : Ax \leq b\}$. When a perfect formulation is available for a mixed integer linear set, the corresponding integer program can be solved as a linear program.

Definition A convex set $P \subseteq \mathbb{R}^n$ is **integral** if $P = \text{conv}\,(P \cap \mathbb{Z}^n)$. If P is a polyhedron, P is called an **integral polyhedron**.

For pure integer linear sets, a classical case of perfect formulation is when the constraint matrix is **totally unimodular**.

Intuition

From linear programming theory, we know that basic feasible solutions take the form: $x = (x_B, x_N) = (B^{-1}b, 0)$ where B is an $m \times m$ nonsingular submatrix of (A, I) and I is an $m \times m$ identity matrix.

Observation (Sufficient Condition) If the optimal basis B has integral entries and has $\det(B) = \pm 1$, then the linear programming relaxation solves IP.

Proof. From Cramer's rule, $B^{-1} = B^*/\det(B)$, where B^* is the adjoint matrix. The entries of B^* are all products of terms of B. Thus, B^* is an integral matrix, and as $\det(B) = \pm 1$, B^{-1} is also integral. Thus, $B^{-1}b$ is integral for all integral b.

Total Unimodularity

A matrix A is totally unimodular if every square submatrix has determinant $0, \pm 1$. It follows from the definition that a totally unimodular matrix has all entries equal to $0, \pm 1$.
For example, the matrix

$$\begin{pmatrix} -1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

is totally unimodular because its determinant is 0 and all its proper square submatrices are triangular after permutation of rows and columns, and thus they have determinant equal to $0, \pm 1$.

Theorem (Hoffman and Kruskal). Let A be an $m \times n$ integral matrix. The polyhedron $\{x : Ax \leq b, x \geq 0\}$ is integral for every $b \in \mathbb{Z}^m$ if and only if A is totally unimodular.

Theorem Let P be a rational polyhedron. The following conditions are equivalent.

1. P is an integral polyhedron.
2. $\max\{cx : x \in P\}$ is attained by an integral vector x for each $c \in \mathbb{R}^n$ for which the maximum is finite.

Sufficient condition of total unimodularity

Theorem A matrix A is TU if
(i) $a_{ij} \in \{+1, -1, 0\}$ for all i, j.
(ii) Each column contains at most two nonzero coefficients ($\sum_{i=1}^{m} |a_{ij}| \leq 2$).
(iii) There exists a partition (M_1, M_2) of the set M of rows such that each column j containing two nonzero coefficients satisfies $\sum_{i \in M_1} a_{ij} - \sum_{i \in M_2} a_{ij} = 0$.

Sufficient condition of total unimodularity

Theorem A matrix A is TU if
(i) $a_{ij} \in \{+1, -1, 0\}$ for all i, j.
(ii) Each column contains at most two nonzero coefficients
($\sum_{i=1}^{m} |a_{ij}| \leq 2$).
(iii) There exists a partition (M_1, M_2) of the set M of rows such that each column j containing two nonzero coefficients satisfies $\sum_{i \in M_1} a_{ij} - \sum_{i \in M_2} a_{ij} = 0$.
Proof. Assume that A is not TU, and let B be the smallest square submatrix of A for which $\det(B) \notin \{0, 1, -1\}$. B cannot contain a column with a single nonzero entry, as otherwise B would not be minimal. So B contains two nonzero entries in each column. Now by condition (iii), adding the rows in M_1 and subtracting the rows in M_2 gives the zero vector, and so $\det(B) = 0$, and we have a contradiction.
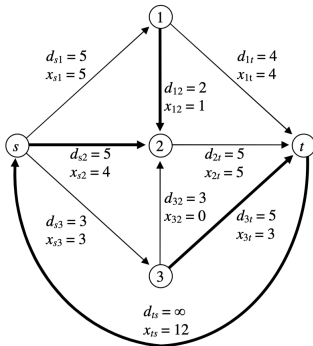
Max flow

max-flow

$$\max x_{ts}$$

$$\sum_{k \in V^+(i)} x_{ik} - \sum_{k \in V^-(i)} x_{ki} = 0 \quad \text{for } i \in V$$

$$d_{ij} - x_{ij} \geq 0 \quad \text{for } (i,j) \in A, \ (i,j) \neq (t,s) \qquad (w_{ij} \geq 0)$$

$$x_{ij} \geq 0 \quad \text{for } (i,j) \in A.$$

Taking the dual of max flow

max-flow

$$\max x_{ts}$$

$$\sum_{k \in V^-(i)} x_{ki} - \sum_{k \in V^+(i)} x_{ik} = 0 \text{ for } i \in V \qquad (u_i)$$

$$d_{ij} - x_{ij} \geq 0 \text{ for } (i,j) \in A, \ (i,j) \neq (t,s) \qquad (w_{ij} \geq 0)$$

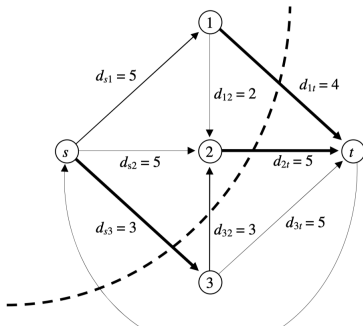$$x_{ij} \geq 0 \quad \text{for } (i,j) \in A.$$

The Lagrangian is

$$g(u,w) = \max_{x \geq 0} x_{ts} + \sum_{(i,j) \in A} (u_j - u_i)x_{ij} +$$

$$\sum_{(i,j) \in A, \ (i,j) \neq (t,s)} w_{ij}(d_{ij} - x_{ij})$$

$$= (1 + u_s - u_t)x_{ts} +$$

$$\sum_{(i,j) \in A, \ (i,j) \neq (t,s)} w_{ij}d_{ij} + (u_j - u_i - w_{ij})x_{ij}$$

The min-cut problem has integral solution

The dual is the min-cut problem: minimize the edges weights connecting the nodes on the source side to the sink side.

$$\min \sum_{(i,j)\in A,\ (i,j)\neq(t,s)} d_{ij}w_{ij}$$

$$u_i - u_j + w_{ij} \geq 0 \quad \text{for } (i,j)\in A,\ (i,j)\neq(t,s)$$

$$u_t - u_s \geq 1$$

$$w_{ij} \geq 0 \quad \text{for } (i,j)\in A,\ (i,j)\neq(t,s)$$

Proof

As the dual is unchanged if we replace u_j by $u_j + \alpha$ for all $j \in V$, we can set $u_s = 0$. From total unimodularity, an optimal solution is an integer. Given such an integer solution, let
$X = \{j \in V : u_j \leq 0\}$ and $\bar{X} = V \backslash X = \{j \in V : u_j \geq 1\}$. Now,

$$\sum_{(i,j) \in A} d_{ij} w_{ij} \geq \sum_{(i,j) \in A, i \in X, j \in \bar{X}} d_{ij} w_{ij} \geq \sum_{(i,j) \in A, i \in X, j \in \bar{X}} d_{ij},$$

where the first inequality follows from $d_{ij} \geq 0, w \geq 0$ and the second as $w_{ij} \geq u_j - u_i \geq 1$ for $(i,j) \in A$ with $i \in X$ and $j \in \bar{X}$. However, this lower bound $\sum_{(i,j) \in A, i \in X, j \in \bar{X}} d_{ij}$ is attained by the solution $u_j = 0$ for $j \in X, u_j = 1$ for $j \in \bar{X}, w_{ij} = 1$ for $(i,j) \in A$ with $i \in X$ and $j \in \bar{X}$, and $w_{ij} = 0$ otherwise. So there is an optimal 0-1 solution.

We see that $s \in X, t \in \bar{X}, \{(i,j) : w_{ij} = 1\}$ is the set of arcs of the $s - t$ cut $(X, V \backslash X)$, and we obtain the standard result that the maximum value of an $s - t$ flow equals the minimum capacity of an $s - t$ cut.

References

- Grossmann, I. E. (2021). Advanced optimization for process systems engineering. Cambridge University Press.
- Conforti, M., Cornuéjols, G., Zambelli, G (2014). Integer programming. Graduate Texts in Mathematics
- Wolsey, L. A. (2020). Integer programming. John Wiley & Sons.