ILP and MILP

Fall 2021

Overview

1. Integer Linear Programming
   - Review
   - Writing Logic Statements
2. Examples
   - Job Scheduling Problem
   - Plan for a move

Review

- ILP stands for Integer Linear Programming and MILP for Mixed Integer Linear Programming (notation may change depending on the reference).
- Both ILP and MILP can be seen as:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n} c_i x_i \\
\text{s.t.} \quad & \sum_{i=1}^{n} a_{ij} x_i \geq, \leq, = b_j \quad \forall j \in \{1, \dots m\} \\
& x_i \text{ integer} \quad \forall i \in \{1, \dots, n\} \quad \text{(ILP)} \\
& x_i \text{ integer or real} \quad \forall i \in \{1, \dots, n\} \quad \text{(MILP)}
\end{aligned}
$$

- A special case of integer is binary. Notice that x binary can also be written as $0 \leq x \leq 1$, integer.

Review

Integer variables, and in particular binary variables, are well suited to write logical statements. Usually we think of $x = 0$ as false, and $x = 1$ as true.

Super Important: Your reformulation should express no more and no less than what you are trying to express. For example, if you are trying to express an implication for one direction, you don't want to also obligate the implication on the other way or extra implications.

Logical Statements

For the following exercises assume all variables to be binary.

Logical Statements

For the following exercises assume all variables to be binary.

- Express: $x_1 = 0 \rightarrow x_2 = 1$.

Logical Statements

For the following exercises assume all variables to be binary.

- Express: $x_1 = 0 \rightarrow x_2 = 1$.
- Solution: $x_2 \geq 1 - x_1$.

Logical Statements

Remember that $A \rightarrow B$ also implies that $\neg B \rightarrow \neg A$. For the following exercises assume all variables to be binary.

Logical Statements

Remember that $A \rightarrow B$ also implies that $\neg B \rightarrow \neg A$. For the following exercises assume all variables to be binary.

- Express: (x_1 false, x_2 true) implies x_3 false.

Logical Statements

Remember that $A \rightarrow B$ also implies that $\neg B \rightarrow \neg A$. For the following exercises assume all variables to be binary.

- Express: (x_1 false, x_2 true) implies x_3 false.
- **Solution:** $x_3 \leq 2 - (1 - x_1) - x_2$

Logical Statements

Remember that $A \to B$ also implies that $\neg B \to \neg A$. For the following exercises assume all variables to be binary.

- Express: (x_1 false, x_2 true) implies x_3 false.
- **Solution:** $x_3 \leq 2 - (1 - x_1) - x_2$
- Express: (x_1 true, x_2 false) implies x_3 true.

Logical Statements

Remember that $A \rightarrow B$ also implies that $\neg B \rightarrow \neg A$. For the following exercises assume all variables to be binary.

- Express: (x_1 false, x_2 true) implies x_3 false.
- **Solution:** $x_3 \leq 2 - (1 - x_1) - x_2$
- Express: (x_1 true, x_2 false) implies x_3 true.
- **Solution:** $x_3 \geq -1 + x_1 + (1 - x_2)$

Logical Statements

Remember that $A \to B$ also implies that $\neg B \to \neg A$. For the following exercises assume all variables to be binary.

- Express: (x_1 false, x_2 true) implies x_3 false.
- **Solution:** $x_3 \leq 2 - (1 - x_1) - x_2$
- Express: (x_1 true, x_2 false) implies x_3 true.
- **Solution:** $x_3 \geq -1 + x_1 + (1 - x_2)$
- Express: $\sum_{i=1}^{n} a_i x_i > b$ implies y true . For this problem assume $\sum_{i=1}^{n} a_i x_i \leq M$ for any choice of $(x_1, \ldots, x_n)$ feasible. In this question we must think of a constraint that is always feasible.

Logical Statements

Remember that $A \to B$ also implies that $\neg B \to \neg A$. For the following exercises assume all variables to be binary.

- Express: (x_1 false, x_2 true) implies x_3 false.
- **Solution:** $x_3 \leq 2 - (1 - x_1) - x_2$
- Express: (x_1 true, x_2 false) implies x_3 true.
- **Solution:** $x_3 \geq -1 + x_1 + (1 - x_2)$
- Express: $\sum_{i=1}^{n} a_i x_i > b$ implies y true . For this problem assume $\sum_{i=1}^{n} a_i x_i \leq M$ for any choice of $(x_1, \ldots, x_n)$ feasible. In this question we must think of a constraint that is always feasible.
- **Solution:** $\sum_{i=1}^{n} a_i x_i \leq (1 - y)b + yM$

Logical Statements

- $\sum_{i=1}^{n} a_i x_i \neq b$ implies z true. Here you can assume

$$-M \leq \sum_{i=1}^{n} a_i x_i \leq M.$$

Logical Statements

- $\sum_{i=1}^{n} a_i x_i \neq b$ implies z true. Here you can assume

$$-M \leq \sum_{i=1}^{n} a_i x_i \leq M.$$

- **Solution:**

$$\begin{aligned}
\sum_{i=1}^{n} a_i x_i &\leq (1-y_1)b + y_1 M & \qquad \left(\sum_{i=1}^{n} a_i x_i > b \rightarrow y_1 = 1\right) \\
\sum_{i=1}^{n} a_i x_i &\geq (1-y_2)b - y_2 M & \qquad \left(\sum_{i=1}^{n} a_i x_i < b \rightarrow y_2 = 1\right) \\
z &\geq \frac{y_1 + y_2}{2} &
\end{aligned}$$

Logical Statements

Let's say we have two constraints $\sum_{i=1}^{n} a_{i1}x_i \geq b_1$ and $\sum_{i=1}^{n} a_{i2}x_i \geq b_2$ where the x's and a'are all ≥ 0. Write a constraint or set of constraints to enforce that at least one of the two inequalities is enforced at all times.

Solution: Let's introduce "y" binary variable. A set of constraints that solves the problem is:

$$\begin{aligned} \sum_{i=1}^{n} a_{i1}x_i &\geq y \cdot b_1 \\ \sum_{i=1}^{n} a_{i2}x_i &\geq (1-y)b_2 \end{aligned}$$

Because both left sides are always ≥ 0.

Logical Statements

Let x be a non-negative real variable, and assume that if $x > 0$ then we always have that $x > \epsilon$. Let z be a binary variable and assume that $x \leq M$.

Logical Statements

Let x be a non-negative real variable, and assume that if $x > 0$ then we always have that $x > \epsilon$. Let z be a binary variable and assume that $x \leq M$.

- $x = 0$ if and only if z false.

Logical Statements

Let x be a non-negative real variable, and assume that if $x > 0$ then we always have that $x > \epsilon$. Let z be a binary variable and assume that $x \leq M$.

- $x = 0$ if and only if z false.
- **Solution:** The two next inequalities do the job.

$$\begin{aligned} x &\leq z \cdot M \quad (x > 0 \rightarrow z = 1) \\ z &\leq \frac{x}{\epsilon} \quad (x = 0 \rightarrow z = 0) \end{aligned}$$

Logical Statements

For the typical production problem, where z could represent the decision of activating a machine and $K > 0$ could be the cost of activating it, we only need to write:

$$\begin{aligned} \min \quad & (\dots) + K \cdot z \\ \text{s.t.} \quad & \text{production} \leq M \cdot z \\ & (\dots), \; z \text{ binary.} \end{aligned}$$

Because the part $z = 0 \rightarrow \textit{production} = 0$ is implied directly by the constraint, and $\textit{production} = 0 \rightarrow z = 0$ is obtained by the fact that we are minimizing.

Job Scheduling Problem

(Exercise 10.7 from Introduction to Linear Programming, Bertsimas & Tsitklis) We consider the production of a single product over T periods. If we decide to produce at period t, a setup cost of c_t is incurred. For $t = 1, \ldots, T$ let d_t be the demand for this product in period t, and let p_t, h_t be the unit production and storage cost resp. for period t.

1. Formulate a MILP in order to minimize the total cost of production, storage, and setup.

Job Scheduling Problem

Variables:

Job Scheduling Problem

Variables:

- z_t : binary variable that indicates if we produce in month t.

Job Scheduling Problem

Variables:

- z_t : binary variable that indicates if we produce in month t.
- x_t : production on month t.

Job Scheduling Problem

Variables:

- z_t : binary variable that indicates if we produce in month t.
- x_t : production on month t.
- I_t : inventory on month t (Also I_0 is included).

Job Scheduling Problem

Variables:

- z_t : binary variable that indicates if we produce in month t.
- x_t : production on month t.
- I_t : inventory on month t (Also I_0 is included).

Formulation:

Job Scheduling Problem

Variables:

- z_t : binary variable that indicates if we produce in month t.
- x_t : production on month t.
- I_t : inventory on month t (Also I_0 is included).

Formulation:

$$
\begin{aligned}
\min \quad & \sum_{t=1}^{T} (z_t c_t + x_t p_t + I_t h_t) \\
\text{s.t.} \quad & x_t \leq z_t \left(\sum_{i=1}^{T} d_i \right), \\
& I_t = I_{t-1} + x_t - d_t, \quad \forall t \in \{1, \ldots, T\} \\
& I_0 = 0, \ z_t \text{ binary}, \quad \forall t \in \{1, \ldots, T\} \\
& x_t \geq 0, \quad \forall t \in \{1, \ldots, T\} \\
& I_t \geq 0, \quad \forall t \in \{1, \ldots, T\}
\end{aligned}
$$

Job Scheduling Problem

2. Suppose we allow demand to be lost in every period except for period T, at a cost of b_t per unit lost of demand. Show how to modify the model to handle this option.

Job Scheduling Problem

2. Suppose we allow demand to be lost in every period except for period T, at a cost of b_t per unit lost of demand. Show how to modify the model to handle this option.
3. **Solution:** we need to add a new variable ℓ_t that is the demand lost in period t. We have to take into account the fact that it may be optimal to not satisfy demand in period t even if we could in order to use the saved storage for the next period.

Job Scheduling Problem

Model:

$$
\begin{aligned}
\min \quad & \sum_{t=1}^{T} (z_t c_t + x_t p_t + I_t h_t) + \sum_{t=1}^{T-1} \ell_t b_t \\
\text{s.t.} \quad & x_t \leq z_t \left(\sum_{i=1}^{T} d_i \right), \\
& I_t = I_{t-1} + x_t - d_t + \ell_t, \quad \forall t \in \{1, \ldots, T-1\} \\
& I_T = I_{T-1} + x_T - d_T \\
& \ell_t \leq d_t, \quad \forall t \in \{1, \ldots, T-1\} \\
& I_0 = 0,\ \ell_t \geq 0, \quad \forall t \in \{1, \ldots, T-1\} \\
& z_t \text{ binary}, \quad \forall t \in \{1, \ldots, T\} \\
& x_t \geq 0, \quad \forall t \in \{1, \ldots, T\},\ I_t \geq 0, \quad \forall t \in \{1, \ldots, T\}.
\end{aligned}
$$

Job Scheduling Problem

3. Suppose that production capacity can occur in at most five periods, but no two such periods can be consecutive. Show how to modify the model to handle this option.

Job Scheduling Problem

3. Suppose that production capacity can occur in at most five periods, but no two such periods can be consecutive. Show how to modify the model to handle this option.
4. **Solution:** We can just add the following constraints:
   - $\sum_{t=1}^{T} z_t \leq 5.$
   - $z_j + z_{j+1} \leq 1 \quad \forall j \in \{1, \ldots, T-1\}.$

Plan for a move: problem

(Exercise 10.5 from Introduction to Linear Programming, Bertsimas & Tsitklis.) Suppose you are planning to move your new house. You have n items of size a_j, $j = 1, \ldots, n$ that need to be moved. You have rented a truck that has size Q and you have bought m boxes. Box i has size b_i, $i = 1, \ldots, m$. **Formulate an integer programming problem in order to decide if the move is possible**.

Plan for a move: solution

This problem needs extra assumptions, i will assume the following two: First, Let's imagine that the truck deliverers are so good at Tetris that for any combination of boxes with total volume less than Q they are able load the truck. Second, we can put as many objects in a box as long we do not surpass its volume (but of course let's imagine we can not divide the objects).

Plan for a move: solution

Variables:

- z_i : Binary variable that represents if box $i \in \{1, \ldots, m\}$ is taken in the truck or not.

Plan for a move: solution

Variables:

- z_i : Binary variable that represents if box $i \in \{1, \ldots, m\}$ is taken in the truck or not.
- x_j^i : Binary variable that represents if item $j \in \{1, \ldots, n\}$ is stored in box i or not.

Plan for a move: solution

Formulation:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{m} z_i \text{ (Could have been anything)} \\
\text{s.t.} \quad & \sum_{i=1}^{m} x_j^i = 1, \quad \forall j \in \{1, \dots, n\} \quad \text{(Also } \geq \text{ works)} \\
& \sum_{j=1}^{n} a_j x_j^i \leq z_i b_i, \quad \forall i \in \{1, \dots, m\} \quad \text{(Box Capacity)} \\
& \sum_{j=1}^{n} z_i b_i \leq Q, \quad \forall i \in \{1, \dots, m\} \quad \text{(Truck Capacity)} \\
& z_j \text{ binary}, \quad \forall i \in \{1, \dots, m\} \\
& x_j^i \text{ binary}, \quad \forall i \in \{1, \dots, n\},\ j \in \{1, \dots, m\}
\end{aligned}
$$

AMPL

- A restriction phrase for a parameter declaration may be the word *integer* or *binary* or a comparison operator followed by an arithmetic expression.

AMPL

- A restriction phrase for a parameter declaration may be the word *integer* or *binary* or a comparison operator followed by an arithmetic expression.
- While *integer* restricts a parameter to integral (whole-number) values, *binary* restricts it to zero or one.

AMPL

- A restriction phrase for a parameter declaration may be the word *integer* or *binary* or a comparison operator followed by an arithmetic expression.
- While *integer* restricts a parameter to integral (whole-number) values, *binary* restricts it to zero or one.
- **Example:**

$$param \quad promote\{1..T, 1..T\}\ binary;$$

AMPL

- A restriction phrase for a parameter declaration may be the word *integer* or *binary* or a comparison operator followed by an arithmetic expression.
- While *integer* restricts a parameter to integral (whole-number) values, *binary* restricts it to zero or one.
- **Example:**
$$param \quad promote\{1..T, 1..T\} \ binary;$$
- **Example:**
$$param \quad promote\{1..T, 1..T\} \ integer;$$

Thank you for your attention !