Lecture 2

Integer Programming Formulations

MATH3220 Operations Research and Logistics

Jan. 8, 2015

Pan Li

The Chinese University of Hong Kong

Agenda

1. **Integer Programming**
2. **Logical constraints**
3. **Nonlinear Functions**

Integer Programming

Integer Programming: a linear program plus the additional constraints that some or all of the variables must be integer valued.

We also permit "$x_j \in \{0, 1\}$", or equivalently, "x_j is binary".

This is a shortcut for writing the constraints:

$$0 \leq x_j \leq 1 \text{ and } x_j \text{ is integer.}$$

Simple logical constraints

Here, we address different logical constraints that can be transformed into integer programming constraints.

- **selection of items from a subset**

Logical constraint	**IP constraint**
If item i is selected, then item j is also selected.	$x_i - x_j \leq 0$
Either item i is selected or item j is selected, but not both.	$x_i + x_j = 1$
Item i is selected or item j is selected or both.	$x_i + x_j \geq 1$
If item i is selected, then item j is not selected.	$x_i + x_j \leq 1$
If item i is not selected, then item j is not selected.	$-x_i + x_j \leq 0$
At most one of item i, j and k are selected.	$x_i + x_j + x_k \leq 1$
At most two of items i, j and k are selected.	$x_i + x_j + x_k \leq 2$
Exactly one of items i, j, and k are selected.	$x_i + x_j + x_k = 1$
At least one of items i, j and k are selected.	$x_i + x_j + x_k \geq 1$

Simple logical constraints - con't

Restricting a variable to take on one of several values.

Suppose that we want to restrict x to be one of the elements {4, 8, 13}. This is accomplished as follows.

$$\begin{aligned} x = 4w_1 + 8w_2 + 13w_3 \\ w_1 + w_2 + w_3 = 1 \\ w_i \in \{0, 1\} \text{for } i = 1 \text{ to } 3. \end{aligned}$$

Question: If we want to restrict x to be one of the elements {0, 4, 8, 13}, how should we model it?

Simple logical constraints - con't

Restricting a variable to take on one of several values.

Suppose that we want to restrict x to be one of the elements {4, 8, 13}. This is accomplished as follows.

$$\begin{aligned} x = 4w_1 + 8w_2 + 13w_3 \\ w_1 + w_2 + w_3 = 1 \\ w_i \in \{0, 1\} \text{for } i = 1 \text{ to } 3. \end{aligned}$$

Question: If we want to restrict x to be one of the elements {0, 4, 8, 13}, how should we model it?

Answer: It suffices to use the above formulation with the equality constraint changed to "$w_1 + w_2 + w_3 \leq 1$".

Simple logical constraints - con't

Restricting a variable to take on discontinuous values.

Suppose that we want to restrict x must be either 0 or between particular positive bounds. In algebraic notation:

$$x = 0 \qquad \text{or} \qquad l \leq x \leq u$$

Define:

$$y = \begin{cases} 0 & \text{for } x = 0 \\ 1 & \text{for } l \leq x \leq u \end{cases}$$

Then, the logical constraints can be modeled by:

$$\begin{aligned} ly \leq x &\leq uy \\ y &\in \{0, 1\} \end{aligned}$$

Other logical constraints, and the big M method.

Big-M method for IP formulations

- Assume that all variables are integer valued.
- Assume a bound u^* on coefficients and variables;

$$\text{e.g. } x_j \leq 1,000 \text{ for all } j.$$
$$|a_{ij}| \leq 1,000 \text{ for all } i, j$$

- Choose M really large so that for every constraint i,

$$|a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n - b_i| \leq M$$

That is, we will be able to satisfy any "$\leq$" constraint by adding M to the RHS.

And we can satisfy any "$\geq$" constraint by subtracting M from the RHS.

Logical constraint - Constraint feasibility

Binary variables that are 1 when a constraint is satisfied.

$$w = \begin{cases} 1 & \text{if } f(x_1, x_2, \ldots, x_n) \leq b \\ 0 & \text{otherwise} \end{cases}$$

Here we assume that $f(x)$ is bounded.

Equivalent constraints:

$$\begin{aligned} f(x_1, x_2, \ldots, x_n) \leq b + M(1 - w). \\ f(x_1, x_2, \ldots, x_n) \geq b - Mw. \\ w \in \{0, 1\} \end{aligned}$$

where the constant M is chosen to be large enough so that the constraint is always satisfied if $w = 0$. Whenever $w = 1$ gives a feasible solution to IP constraint, the logical constraint must be satisfied.

Logical constraint - Alternative constraint

We formulate the logical constraint, "$x \leq 2$ or $x \geq 6$" as follows.

Choose a binary variable w so that

- if $w = 1$, then $x \leq 2$
- if $w = 0$, then $x \geq 6$

$$\begin{aligned} x &\leq 2 + M(1 - w) \\ x &\geq 6 - Mw \\ w &\in \{0, 1\} \end{aligned}$$

To validate the formulation one needs to show:
The logical constraints are equivalent to the IP constraints.

The logical constraint - Alternative constraint (con't)

Logical constraint	IP constraints
	$x \leq 2 + M(1 - w)$
$x \leq 2$ or $x \geq 6$	$x \geq 6 - Mw$
	$w \in \{0, 1\}$

Suppose (x, w) is feasible for the IP,

$$\text{if } w = 1, \text{ then} x \leq 2.$$
$$\text{if } w = 0, \text{ then} x \geq 6.$$

In both cases, the logical constraints are satisfied.

Suppose that x satisfies the logical constraints.

If $x \leq 2$, then let $w = 1 \quad \Rightarrow \quad x \leq 2 \quad$ and $\quad x \geq 6 - M$

If $x \geq 6$, then let $w = 0 \quad \Rightarrow \quad x \leq 2 + M \quad$ and $\quad x \geq 6$

In both cases, the IP constraints are satisfied.

The logical constraint - Alternative constraint (con't)

Logical constraint	IP constraints
$2x_1 + 3x_2 \geq 14$ or	$2x_1 + 3x_2 \geq 14 - M(1 - w)$
$5x_2 - 7x_3 \leq 3$	$5x_2 - 7x_3 \leq 3 + Mw$
	$w \in \{0, 1\}$
Suppose x_i is bounded for all i.	Suppose that M is very large.

To show:
The logical constraints are equivalent to the IP constraints.

Suppose (x, w) is feasible for the IP,

$$\text{if } w = 1, \text{ then } 2x_1 + 3x_2 \geq 14.$$
$$\text{if } w = 0, \text{ then } 5x_2 - 7x_3 \leq 3.$$

Therefore, the logical constraints are satisfied.

The logical constraint - Alternative constraint (con't)

Logical constraint	IP constraints
$2x_1 + 3x_2 \geq 14$ or	$2x_1 + 3x_2 \geq 14 - M(1 - w)$
$5x_2 - 7x_3 \leq 3$	$5x_2 - 7x_3 \leq 3 + Mw$
	$w \in \{0, 1\}$

To show:
The logical constraints are equivalent to the IP constraints.

Suppose that x satisfies the logical constraints.

If $2x_1 + 3x_2 \geq 14$, then let $w = 1 \quad \Rightarrow \quad 2x_1 + 3x_2 \geq 14$ and $5x_2 - 7x_3 \leq 3 + M$

If $5x_2 - 7x_3 \leq 3$, then let $w = 0 \quad \Rightarrow \quad 2x_1 + 3x_2 \geq 14 - M$ and $5x_2 - 7x_3 \leq 3$

In both cases, the IP constraints are satisfied.

Logical constraint - Conditional constraints

These constraints have the form:

$$\text{if } f_1(x_1, x_2, \ldots, x_n) > b_1, \text{ then } f_2(x_1, x_2, \ldots, x_n) \le b_2.$$

Since this implication is not satisfied *only when both* $f_1(x_1, x_2, \ldots, x_n) > b_1$ and $f_2(x_1, x_2, \ldots, x_n) > b_2$, the conditional constraint is logically equivalent to the alternate constraints

$$f_1(x_1, x_2, \ldots, x_n) \le b_1 \text{ and/or } f_2(x_1, x_2, \ldots, x_n) \le b_2$$

where at least one must be satisfied. Hence, this situation can be modeled by alternative constraints:

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &\le b_1 + B_1 y, \\ f_2(x_1, x_2, \ldots, x_n) &\le b_2 + B_2(1 - y), \\ y &\in \{0, 1\} \end{aligned}$$

At least one of three inequalities is satisfied.

$$x_1 + 4x_2 + 2x_4 \geq 7 \text{ or } 3x_1 - 5x_2 \leq 12 \text{ or } 2x_2 + x_3 \geq 6$$

Create three binary variables w_1, w_2, and w_3 and reformulate the above constraint as the following system of logical, linear and integer constraints.

If $w_1 = 1$, then $x_1 + 4x_2 + 2x_4 \geq 7$
If $w_2 = 1$, then $3x_1 - 5x_2 \leq 12$
If $w_3 = 1$, then $2x_2 + x_3 \geq 6$
$w_1 + w_2 + w_3 \geq 1$
$w_i \in \{0, 1\}$ for $i = 1$ to 3.

This above system of constraints is equivalent to the following.

$$\begin{array}{lcl} x_1 + 4x_2 + 2x_4 & \geq & 7 - M(1 - w_1) \\ 3x_1 - 5x_2 & \leq & 12 + M(1 - w_2) \\ 2x_2 + x_3 & \geq & 6 - M(1 - w_3) \\ w_1 + w_2 + w_3 & \geq & 1 \\ w_i \in \{0, 1\} \text{ for } i = 1 \text{ to } 3 & & . \end{array}$$

Logical constraint - k-Fold alternative

Suppose we must satisfy at least k of the constraints:

$$f_i(x) \leq b_i \qquad (j = 1, 2, \ldots, m)$$

Assuming that B_i are chosen so that the ignored constraints will not be binding, the general problem can be formulated as follows:

$$\begin{aligned} & f_i(x) \leq b_i + B_i(1 - y_i) \\ & \textstyle\sum_{i=1}^{m} y_i \geq k, \\ & y_i \in \{0, 1\} \quad (i = 1, 2, \ldots, m) \end{aligned}$$

Nonlinear Functions

- Nonlinear functions can be represented by integer programming formulations.

Fixed costs

In a typical production planning problem involving N products, the production cost for product j may consist of a fixed cost d_j independent of the amount produced and a variable cost c_j per unit. Thus if x_j is the production level of product j, its production cost function may be written as

$$f_j(x_j) = \begin{cases} d_j + c_j x_j & x_j > 0 \\ 0 & x_j = 0 \end{cases}$$

This is nonlinear in x_j because of the discontinuity of $f_j(x_j)$ at the origin. Consequently, the following minimum cost problem is also nonlinear:

$$\begin{array}{ll} \text{Min} & z = \sum_{j=1}^{N} f_j(x_j) \\ \text{s.t.} & Ax = b, x \geq 0. \end{array}$$

Fixed costs (con't)

If it is known that $0 \leq x_j \leq u_j$ and $d_j > 0$ then we can define a binary variable y_j that indicates when the fixed cost is incurred, so that

$$\begin{aligned} y_j &= 1, \text{ if } x_j > 0 \\ y_j &= 0, \text{ if } x_j = 0 \end{aligned}$$

Then the contribution to cost due to x_j may be written as

$$f_j(x_j) = d_j y_j + c_j x_j$$

with the constraints:

$$\begin{aligned} x_j &\leq u_j y_j, \\ x_j &\geq 0, \\ y &= 0 \text{ or } 1. \end{aligned}$$

Fixed costs (con't)

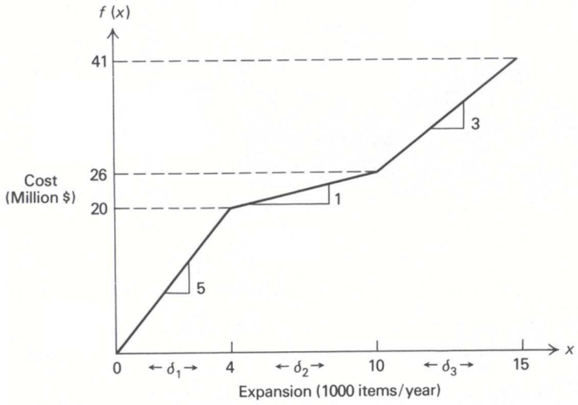
In a typical production planning problem involving N products, the production cost for product j may consist of a fixed cost d_j independent of the amount produced and a variable cost c_j per unit. Thus if x_j is the production level of product j, its production cost function may be written as

$$f_j(x_j) = \begin{cases} d_j + c_j x_j & x_j > 0 \\ 0 & x_j = 0 \end{cases}$$

This is nonlinear in x_j because of the discontinuity of $f_j(x_j)$ at the origin. Consequently, the following minimum cost problem is also nonlinear:

$$\begin{array}{ll} \text{Min} & z = \sum_{j=1}^{N} (c_j x_j + d_j y_j) \\ \text{s.t.} & Ax = b \\ & 0 \leq x_j \leq u_j y_j, \quad j = 1, 2, \ldots, N \\ & y_j \in \{0, 1\}, \quad j = 1, 2, \ldots, N \end{array}$$

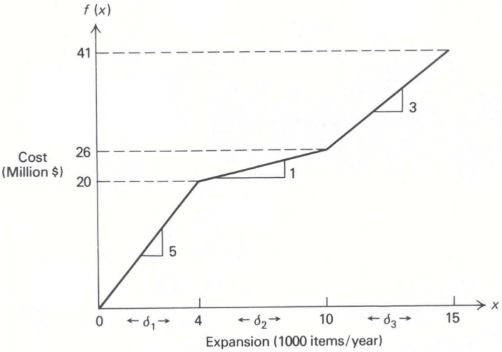
Piecewise linear representation



Define integral variables δ_1, δ_2 and δ_3 so that:

- δ_1 corresponds to the amount by which x exceeds 0, but is less than or equal to 4;
- δ_2 is the amount by which x exceeds 4, but is less than or equal to 10; and
- δ_3 is the amount by which x exceeds 10, but is less than or equal to 15.

Piecewise linear representation



Hence,

$$x = \delta_1 + \delta_2 + \delta_3,$$

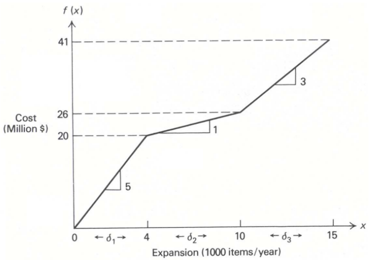
where

$$0 \leq \delta_1 \leq 4, \quad 0 \leq \delta_2 \leq 6, \quad 0 \leq \delta_3 \leq 5, \tag{1}$$

and the total variable cost is given by:

$$\text{Cost} = 5\delta_1 + \delta_2 + 3\delta_3.$$

Piecewise linear representation



However, we should have the following conditional constraints:

- $\delta_1 = 4$, if $\delta_2 > 0$
- $\delta_2 = 6$, if $\delta_3 > 0$

Hence,

$$x = \delta_1 + \delta_2 + \delta_3,$$

where

$$0 \leq \delta_1 \leq 4, \quad 0 \leq \delta_2 \leq 6, \quad 0 \leq \delta_3 \leq 5, \tag{2}$$

and the total variable cost is given by:

$$\text{Cost} = 5\delta_1 + \delta_2 + 3\delta_3.$$

Piecewise linear representation (I)

Logical constraints:

$$0 \le \delta_1 \le 4, \quad 0 \le \delta_2 \le 6, \quad 0 \le \delta_3 \le 5,$$

$$\delta_1 = 4, \text{ if } \delta_2 > 0$$

$$\delta_2 = 6, \text{ if } \delta_3 > 0$$

Define

$$w_1 = \begin{cases} 1 & \text{if } \delta_1 \text{ is at its upper bound} \\ 0 & \text{otherwise} \end{cases}$$

$$w_2 = \begin{cases} 1 & \text{if } \delta_2 \text{ is at its upper bound} \\ 0 & \text{otherwise} \end{cases}$$

Piecewise linear representation (I) (con't)

Logical constraints:

$$0 \le \delta_1 \le 4, \quad 0 \le \delta_2 \le 6, \quad 0 \le \delta_3 \le 5,$$
$$\delta_1 = 4, \text{ if } \delta_2 > 0$$
$$\delta_2 = 6, \text{ if } \delta_3 > 0$$

Then the constraints above can be replaced by:

$$\begin{array}{ccccc} 4w_1 & \le & \delta_1 & \le & 4, \\ 6w_2 & \le & \delta_2 & \le & 6w_1, \\ 0 & \le & \delta_3 & \le & 5w_2 \end{array}$$
$$w_1 \text{ and } w_2 \text{ binary.}$$

Piecewise linear representation (I) (con't)

Logical constraints:

$$0 \le \delta_1 \le 4, \quad 0 \le \delta_2 \le 6, \quad 0 \le \delta_3 \le 5,$$

$$\delta_1 = 4, \text{ if } \delta_2 > 0$$

$$\delta_2 = 6, \text{ if } \delta_3 > 0$$

Then the constraints above can be replaced by:

$$\begin{array}{ccccc} 4w_1 & \le & \delta_1 & \le & 4, \\ 6w_2 & \le & \delta_2 & \le & 6w_1, \\ 0 & \le & \delta_3 & \le & 5w_2 \end{array}$$

$$w_1 \text{ and } w_2 \text{ binary.}$$

There are three feasible combinations for the values of w_1 and w_2:

- $w_1 = 0, w_2 = 0$ corresponding to $0 \le x \le 4$
- $w_1 = 1, w_2 = 0$ corresponding to $4 \le x \le 10$
- $w_1 = 1, w_2 = 1$ corresponding to $10 \le x \le 15$

Piecewise linear representation (I) (con't)

The same general technique can be applied to piecewise linear curves with any number of segments.

The general constraint imposed upon the variable δ_j for the jth segment will be:

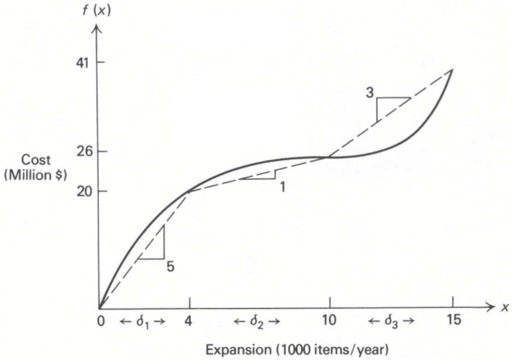
$$L_j w_j \leq \delta_j \leq L_j w_{j-1}$$

where L_j is the length of the segment.

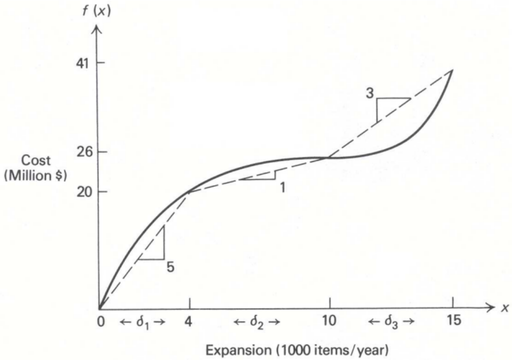
Approximation of Nonlinear Functions

One of the most useful applications of the piecewise linear representation is for approximating nonlinear function.

Suppose the expansion cost in our previous example is given by the heavy curve in the figure below.



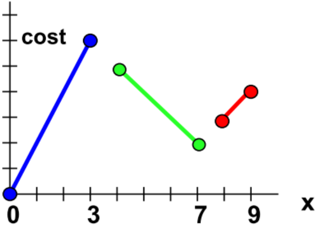
Approximation of Nonlinear Functions (con't)



If we draw linear segments joining selected points on the curve, we obtain a piecewise linear approximation, which can be used of the curve in the model. The piecewise approximation is represented by introducing integer variables as indicated before.

By using more points on the curve, we can make the approximation as close as we desire.

Piecewise linear representation (II)



$$
\begin{array}{lll}
y = & 2x & \text{if } 0 \le x \le 3 \\
y = & 9 - x & \text{if } 4 \le x \le 7 \\
y = & -5 + x & \text{if } 8 \le x \le 9
\end{array}
$$

x is integer valued.

$$w_1 = \begin{cases} 1 & \text{if } 0 \le x \le 3 \\ 0 & \text{otherwise.} \end{cases} \qquad x_1 = \begin{cases} x & \text{if } 0 \le x \le 3 \\ 0 & \text{otherwise.} \end{cases}$$

$$w_2 = \begin{cases} 1 & \text{if } 4 \le x \le 7 \\ 0 & \text{otherwise.} \end{cases} \qquad x_2 = \begin{cases} x & \text{if } 4 \le x \le 7 \\ 0 & \text{otherwise.} \end{cases}$$

$$w_3 = \begin{cases} 1 & \text{if } 8 \le x \le 9 \\ 0 & \text{otherwise.} \end{cases} \qquad x_3 = \begin{cases} x & \text{if } 8 \le x \le 9 \\ 0 & \text{otherwise.} \end{cases}$$

If the variables are defined as above, then

$$y = 2x_1 + (9w_2 - x_2) + (-5w_3 + x_3)$$

Piecewise linear representation (II) (con't)

Add constraints

Definitions of the variables.

$$w_1 = \begin{cases} 1 & \text{if } 0 \le x \le 3 \\ 0 & \text{otherwise.} \end{cases} \qquad x_1 = \begin{cases} x & \text{if } 0 \le x \le 3 \\ 0 & \text{otherwise.} \end{cases}$$

$$w_2 = \begin{cases} 1 & \text{if } 4 \le x \le 7 \\ 0 & \text{otherwise.} \end{cases} \qquad x_2 = \begin{cases} x & \text{if } 4 \le x \le 7 \\ 0 & \text{otherwise.} \end{cases}$$

$$w_3 = \begin{cases} 1 & \text{if } 8 \le x \le 9 \\ 0 & \text{otherwise.} \end{cases} \qquad x_3 = \begin{cases} x & \text{if } 8 \le x \le 9 \\ 0 & \text{otherwise.} \end{cases}$$

Constraints

$$\begin{aligned} 0 \le x_1 &\le 3w_1 \\ w_1 &\in \{0,1\} \\ 4w_2 \le x_2 &\le 7w_2 \\ w_2 &\in \{0,1\} \\ 8w_3 \le x_3 &\le 9w_3 \\ w_3 &\in \{0,1\} \end{aligned}$$

$$\begin{aligned} w_1 + w_2 + w_3 &= 1 \\ x &= x_1 + x_2 + x_3 \\ x_i &\text{ integer} \forall i \end{aligned}$$

Suppose that $0 \le x \le 9$, x integer.
If (x, w) satisfies the definitions, then it also satisfies the constraints.
If (x, w) satisfies the constraints, then it also satisfies the definitions.

Exercise

Construct integer programming formulations to represent the following piecewise linear function.

$$f(x) = \begin{cases} 0 & \text{if } x = 0 \\ 1 & \text{if } 1 \leq x \leq 2 \\ 2 & \text{if } 3 \leq x \leq 4. \end{cases}$$

Summary

- IPs can model almost any combinatorial optimization problem.
- lots of transformation techniques.
- Next lecture: how to solve integer programs.