Tutorial 2: Integer Programming

Aditya Bharti, Kritika Prakash

IIIT Hyderabad

27 Jan 2020

Overview

- Integer Programming: Introduction
- Binary Integer Programming
- Integer Program Example
- Logical Constraints
  - Constraint Feasibility
  - Alternative Constraints
  - Conditional Constraints
  - k-fold Alternatives
  - Compound Alternatives
- Representing Non-Linearities
  - Fixed Costs
  - Piece-wise Linear Functions

Integer Programming: Introduction

Integer Programs allow us to express a richer variety of optimization problems.

- Indivisible Quantities: Number of units
- Logical Constraints: Choices, if-then relationships
- Non-linearities: fixed costs, piece-wise linear functions

A formulation with integer decision variables is an Integer Programming formulation.

In **Mixed Integer Programming** formulations *some* of the decision variables are integers.
In **Pure Integer Programming** formulations *all* of the decision variables are integers.

Binary Integer Programming

Binary variables are integer variables which are restricted to $\{0, 1\}$. They are useful for representing choices and logical constraints in our programming.

In **Binary Integer Programming** formulations all decision variables are binary.

The warehousing problem illustrates most of these concepts.

Integer Programming Example

Choose which of m warehouses must be set up to meet the demand of n customers.

$$y_i = \begin{cases} 1 & \text{if warehouse } i \text{ is opened,} \\ 0 & \text{if warehouse } i \text{ is not opened;} \end{cases}$$

$$x_{ij} = \text{Amount to be sent from warehouse } i \text{ to customer } j.$$

The relevant costs are:

f_i = Fixed operating cost for warehouse i, ifopened (for example, a cost to lease the warehouse),

c_{ij} = Per-unit operating cost at warehouse i plus the transportation cost for shipping from warehouse i to customer j.

There are two types of constraints for the model:

i) the demand d_j of each customer must be filled from the warehouses; and

ii) goods can be shipped from a warehouse only if it is opened.

Logical Constraints

Frequently, problem settings impose logical constraints on the decision variables (like timing restrictions, contingencies, or conflicting alternatives), which lend themselves to integer-programming formulations.

- Constraint Feasibility
- Alternative Constraints
- Conditional Constraints
- k-fold Alternatives
- Compound Alternatives

Constraint Feasibility

Possibly the simplest logical question that can be asked in mathematical programming is whether a given choice of the decision variables satisfies a constraint. More precisely, *when* is the general constraint

$$f(x_1, x_2, \ldots, x_n) \leq b$$

satisfied?

Alternative Constraints

Consider a situation with the *alternative* constraints:

$$f_1(x_1, x_2, \ldots, x_n) \leq b_1$$

$$f_2(x_1, x_2, \ldots, x_n) \leq b_2$$

At least one, but not necessarily both of these constraints must be satisfied.

Conditional Constraints

These constraints have the form:

$f_1(x_1, x_2, \ldots, x_n) \leq b_1$ implies that $f_2(x_1, x_2, \ldots, x_n) \leq b_2$

A implies B is logically equivalent to?

k-fold Alternatives

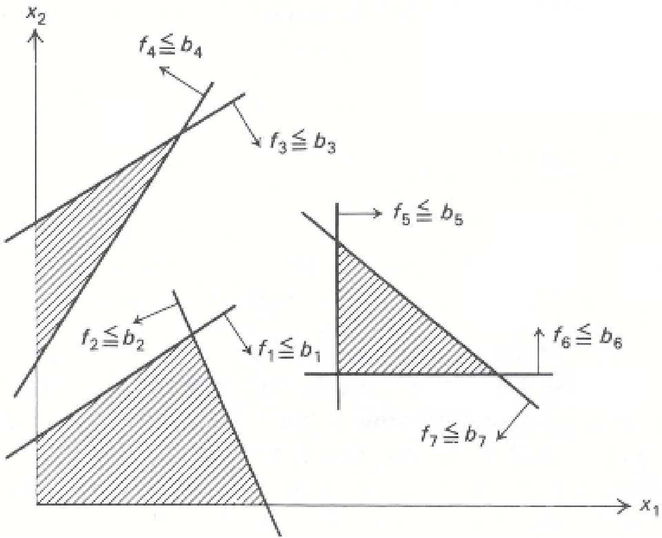
Suppose that we must satisfy at least k of the following p constraints:

$$f_j(x_1, x_2, \ldots, x_n) \leq b_j \quad (j = 1, 2, \ldots, p).$$

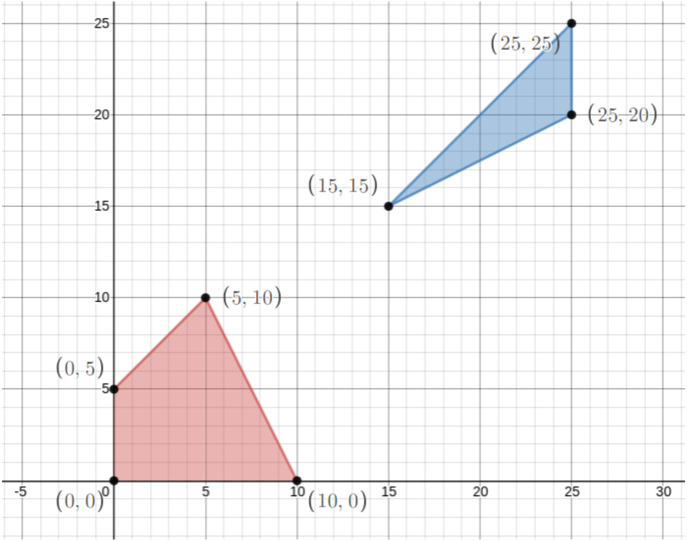
Decision variable y_i represents whether the i^{th} constraint is considered.

Compound Alternatives

Suppose we wish to express a formulation where the feasible region consists of 3 disjoint regions as follows:



Question 1: Logical Constraints



Representing Non-Linearities

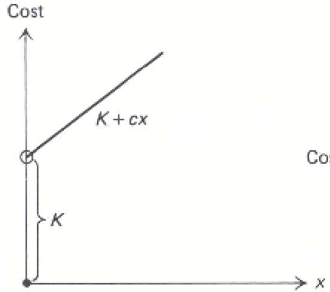
Some non-linear functions can be represented by integer-programming formulations. Let us analyze two of the most useful representations of this type.

- Fixed Costs
- Piece-wise Linear Functions

Fixed Costs

A fixed cost in a cost formulation represents an initial setup cost that is contingent on some decision.

For example, the cost of producing x units of a specific product might consist of a fixed cost of setting up the equipment and a variable cost per unit produced on the equipment.



$$\text{Cost} = \begin{cases} 0 & \text{if } x = 0 \\ K + cx & \text{if } x > 0 \end{cases}$$

Fixed Costs

Assume that the equipment has a capacity of B units. Define y to be a binary variable that indicates when the fixed cost is incurred, so that $y = 1$ when $x > 0$ and $y = 0$ when $x = 0$.

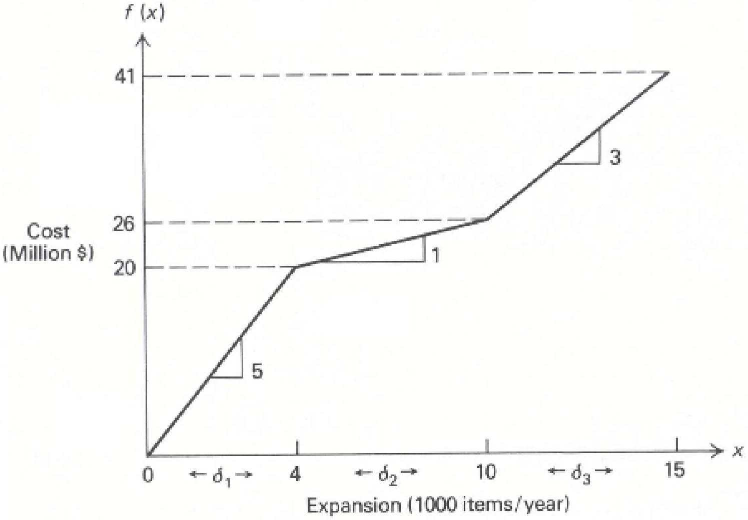
$$Ky + cx,$$

with the constraints:

$$\begin{aligned} &x \leq By, \\ &x \geq 0, \\ &y = 0 \quad \text{or} \quad 1. \end{aligned}$$

Piece-wise Linear Functions

Integer programs can also be used to represent piece-wise linear functions. Consider the following cost function:



Piece-wise Linear Functions

To model the cost curve, we express any value of x as the sum of three variables $\delta_1, \delta_2, \delta_3$, so that the cost for each of these variables is linear. Hence,

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

where

$$\begin{aligned} 0 &\leq \delta_1 \leq 4, \\ 0 &\leq \delta_2 \leq 6, \\ 0 &\leq \delta_3 \leq 5; \end{aligned} \tag{10}$$

and the total variable cost is given by:

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

- δ_1: amount by which x exceeds 0 when it is less than or equal to 4
- δ_2: amount by which x exceeds 4 when it is less than or equal to 10
- δ_3: amount by which x exceeds 10 when it is less than or equal to 15

Piece-wise Linear Functions

Enforcing bounds on δ variables:

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

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

Rewriting constraints using binary variables:

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

$$w_1 \quad \text{and} \quad w_2 \text{ binary,}$$

In general, for segments of length L_j:

$$L_j w_j \le \delta_j \le L_j w_{j-1},$$

Question 2: Piece-wise Linear Function with Fixed Cost