Optimización Avanzada

Formulación de problemas de programación lineal entera

Juan Antonio Díaz García

Outline

1. Formulations
    - Integer optimization problems
    - Combinatorial optimization problems
    - Modeling using binary variables
        - Rules for implications
        - Implications
        - Generalized implications
        - Linking continuous and binary variables
        - Disjunctive constraints
        - Product of binary variables
2. Examples
    - Knapsack
    - Assignment
    - Set covering
    - TSP

What is an integer program?[1]

Suppose that we have a linear program

$$\max\{\mathbf{cx} : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$$

where $\mathbf{A}$ is an $m \times n$ matrix, $\mathbf{c}$ an n row vector, $\mathbf{b}$ and m column vector, and $\mathbf{x}$ an n column vector of variables or unknowns. Now we add constraints to ensure that certain variables must take integer values:

- **Mixed Integer Program**: if some but not all variables are integer.

$$\begin{array}{llrcl} & \max & \mathbf{cx} + \mathbf{hy} & & \\ (MIP) & & \mathbf{Ax} + \mathbf{Gy} & \leq & \mathbf{b} \\ & & \mathbf{x} & \geq & \mathbf{0} \\ & & \mathbf{y} & \in & \mathbb{Z}_+^p. \end{array}$$

  where $\mathbf{A}$ is a $m \times n$ matrix, $\mathbf{G}$ is an m by p matrix, $\mathbf{h}$ is a p row vector, and $\mathbf{y}$ is a p column vector of non negative integer variables:

[1] Wolsey, L.A., Integer Programming, Wiley Interscience Series in Mathematics and Optimization, 1998.

What is an integer program?

- **Integer Program**: if all variables are integer:

$$(IP) \qquad \begin{array}{rcl} \max & \mathbf{cx} & \\ & \mathbf{Ax} & \leq & \mathbf{b} \\ & \mathbf{x} & \in & \mathbb{Z}^n_+. \end{array}$$

- **Binary Integer Program**: if all variables are restricted to 0-1 values:

$$(BIP) \qquad \begin{array}{rcl} \max & \mathbf{cx} & \\ & \mathbf{Ax} & \leq & \mathbf{b} \\ & \mathbf{x} & \in & \{0,1\}^n. \end{array}$$

Combinatorial Optimization Problem [2]

Given a finite set $N = \{1, \ldots, n\}$, weights c_j for each $j \in N$, and a set $\mathscr{F}$ of feasible subsets of N, a combinatorial optimization problem can be defined in the following way

$$(COP) \qquad \min_{S \subseteq N} \left\{ \sum_{j \in S} c_j : S \in \mathscr{F} \right\}.$$

[2] Wolsey, L.A., Integer Programming, Wiley Interscience Series in Mathematics and Optimization, 1998.

Practical rule for implications

Let $P = \{P_1, P_2, \dots, P_n\}$ be a set of logical propositions and $A, B \subseteq P$. Also, let $x_i, i = 1, \dots, n$ be a 0-1 variable which takes the value of 1 if proposition P_i is true, 0 otherwise

Practical Rule 1: Some from set A in the left-hand side of implications.

If the left-hand side of an implication is: some propositions from set A are true, break up the statement into $|A|$ statements, with the right-hand side of the implication unchanged. For example $(P_1 \vee P_2 \vee P_3) \rightarrow (P_5 \vee P_6)$ decomposes in the following way:

$$\begin{aligned} P_1 &\rightarrow (P_5 \vee P_6) \\ P_2 &\rightarrow (P_5 \vee P_6) \\ P_3 &\rightarrow (P_5 \vee P_6) \end{aligned}$$

Practical rule for implications

Let $P = \{P_1, P_2, \ldots, P_n\}$ be a set of logical propositions and $A, B \subseteq P$. Also, let $x_i, i = 1, \ldots, n$ be a 0-1 variable which takes the value of 1 if proposition P_i is true, 0 otherwise

Practical Rule 2: All from set B in the right-hand side of implications.

If the right-hand side of an implication is: All propositions from set B are true, break up the statement into $|B|$ statements with the left-hand side of the implication unchanged. For example $P_1 \rightarrow (P_2 \wedge P_3)$ decomposes in the following way:

$$
\begin{aligned}
P_1 &\rightarrow P_2 \\
P_1 &\rightarrow P_3
\end{aligned}
$$

Practical rule for implications

Let $P = \{P_1, P_2, \ldots, P_n\}$ be a set of logical propositions and $A, B \subseteq P$. Also, let $x_i, i = 1, \ldots, n$ be a 0-1 variable which takes the value of 1 if proposition P_i is true, 0 otherwise

Practical Rule 3: All from set A implies some from set B

A situation when all the propositions of a set A are true implies that some propositions of the set B are true, can be modeled in the following way:

$$\sum_{i \in A} x_i \leq \sum_{i \in B} x_i + |A| - 1$$

Suppose that in some practical application problem one or more projects must be selected from a set of n available projects. Define n 0-1 variables,

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots, n.$$

Suppose that in some practical application problem one or more projects must be selected from a set of n available projects. Define n 0-1 variables,

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots, n.$$

Implications:

If project 1 is selected then project 2 must be selected:

Suppose that in some practical application problem one or more projects must be selected from a set of n available projects. Define n 0-1 variables,

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots, n.$$

Implications:

If project 1 is selected then project 2 must be selected:

$$x_1 \leq x_2$$

Suppose that in some practical application problem one or more projects must be selected from a set of n available projects. Define n 0-1 variables,

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots, n.$$

Implications:

If project 1 is selected then project 3 must not be selected:

Suppose that in some practical application problem one or more projects must be selected from a set of n available projects. Define n 0-1 variables,

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots, n.$$

Implications:

If project 1 is selected then project 3 must not be selected:

$$x_1 + x_3 \leq 1$$

Suppose that in some practical application problem one or more projects must be selected from a set of n available projects. Define n 0-1 variables,

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots, n.$$

Implications:

If project 1 is not selected then project 4 must be selected:

Suppose that in some practical application problem one or more projects must be selected from a set of n available projects. Define n 0-1 variables,

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots, n.$$

Implications:

If project 1 is not selected then project 4 must be selected:

$$x_1 + x_4 \geq 1$$

Suppose that in some practical application problem one or more projects must be selected from a set of n available projects. Define n 0-1 variables,

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots, n.$$

Implications:

If project 1 is selected then project 5 must be selected, and if project 5 is selected then project 1 must be selected:

Suppose that in some practical application problem one or more projects must be selected from a set of n available projects. Define n 0-1 variables,

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected}, \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots, n.$$

Implications:

If project 1 is selected then project 5 must be selected, and if project 5 is selected then project 1 must be selected:

$$x_1 = x_5$$

Suppose that in some practical application problem one or more projects must be selected from a set of n available projects. Define n 0-1 variables,

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots, n.$$

Implications with three 0-1 variables

If project 1 is selected then projects 3 and 5 must be selected:

Suppose that in some practical application problem one or more projects must be selected from a set of n available projects. Define n 0-1 variables,

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \dots, n.$$

Implications with three 0-1 variables

If project 1 is selected then projects 3 and 5 must be selected:

$$\begin{aligned} x_1 &\leq x_3 \\ x_1 &\leq x_5 \end{aligned}$$

Suppose that in some practical application problem one or more projects must be selected from a set of n available projects. Define n 0-1 variables,

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots, n.$$

Implications with three 0-1 variables

If project 1 is selected then project 2 or project 4 must be selected:

Suppose that in some practical application problem one or more projects must be selected from a set of n available projects. Define n 0-1 variables,

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots, n.$$

Implications with three 0-1 variables

If project 1 is selected then project 2 or project 4 must be selected:

$$x_1 \leq x_2 + x_4$$

Modeling with binary variables

Implications

Suppose that in some practical application problem one or more projects must be selected from a set of n available projects. Define n 0-1 variables,

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots, n.$$

Implications with three 0-1 variables

If project 1 or project 5 are selected then project 6 must be selected:

Suppose that in some practical application problem one or more projects must be selected from a set of n available projects. Define n 0-1 variables,

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots, n.$$

Implications with three 0-1 variables

If project 1 or project 5 are selected then project 6 must be selected:

$$\begin{aligned} x_1 &\leq x_6 \\ x_5 &\leq x_6 \end{aligned}$$

Suppose that in some practical application problem one or more projects must be selected from a set of n available projects. Define n 0-1 variables,

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots, n.$$

Implications with three 0-1 variables

If project 2 and project 3 are selected then project 7 must be selected:

Suppose that in some practical application problem one or more projects must be selected from a set of n available projects. Define n 0-1 variables,

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots, n.$$

Implications with three 0-1 variables

If project 2 and project 3 are selected then project 7 must be selected:

$$x_2 + x_3 - 1 \quad \leq \quad x_7$$

Generalized implications

Generalized implications examples

If two of more projects among projects 2, 4, 6, 8, and 10 are selected, then project 1 must be selected:

Generalized implications

Generalized implications examples

If two of more projects among projects 2, 4, 6, 8, and 10 are selected, then project 1 must be selected:

$$\frac{1}{4}\left(x_2 + x_4 + x_6 + x_8 + x_{10} - 1\right) \quad \leq \quad x_1$$

Generalized implications

Generalized implications examples

If two of more projects among projects 2, 4, 6, 8, and 10 are selected, then project 1 must be selected:

If m or more projects among a subset of projects $S \subset \{1, 2, \ldots, n\}$ are selected (with $m < |S|$), then project 2 must be selected:

Generalized implications

Generalized implications examples

If two of more projects among projects 2, 4, 6, 8, and 10 are selected, then project 1 must be selected:

If m or more projects among a subset of projects $S \subset \{1, 2, \ldots, n\}$ are selected (with $m < |S|$), then project 2 must be selected:

$$\frac{1}{|S| - m + 1}\left(\sum_{j \in S} x_j - m + 1\right) \leq x_2$$

Linking continuous and binary variables[3]

Suppose that x is a non negative continuous variable. We may wish to use an indicator variable $\delta \in \{0, 1\}$ to distinguish between the state where $x = 0$ and the state where $x > 0$. By introducing the following constraint, we can force δ to take the value 1 when $x > 0$:

$$x - M\delta \leq 0$$

where M is a constant coefficient representing a known upper bound for x.
Logically, we have achieved the condition:

$$x > 0 \rightarrow \delta = 1 \tag{1}$$

[3]Williams, H.P., Model building in mathematical programming, Wiley, 2003

Linking continuous and binary variables

There are applications where we also wish to impose the condition

$$x = 0 \quad \rightarrow \quad \delta = 0 \tag{2}$$

Condition (2) is another way of saying

$$\delta = 1 \quad \rightarrow \quad x > 0 \tag{3}$$

Together, conditions (1) and (2) (or (3)) impose the condition

$$\delta = 1 \quad \leftrightarrow \quad x > 0 \tag{4}$$

Linking continuous and binary variables

Is not possible to represent conditions (2) (or (3)) by a constraint. Condition (3) states that if $\delta = 1$ the quantity represented by x must be greater than zero. In practical applications, therefore, we wish to distinguish a threshold level m below which we regard the quantity represented by x to be equal to 0. Condition (3) can be rewritten as

$$\delta = 1 \quad \rightarrow \quad x \geq m$$

This condition can be modeled with the inequality

$$x - m\delta \geq 0$$

Linking continuous and binary variables

Modeling fixed costs[4]

We want to model a situation where x represents the quantity of a product to be manufactured at a marginal cost of C_1 per unit. In addition, if the product is manufactured at all there is a set-up cost of C_2. Then, it can be summarized as follows:

$$f(x) = \begin{cases} 0, & \text{if } x = 0, \\ C_1 x + C_2, & \text{if } x > 0. \end{cases}$$

Let δ be a 0-1 variable which takes the value of 1 if any of the product is manufactured and 0 otherwise.

- Objective function:
$$C_1 x + C_2 \delta$$
- If any quantity of the product is manufactured, then $\delta = 1$:
$$x - M\delta \leq 0$$
where M is an upper bound for x.

[4]Williams, H.P., Model building in mathematical programming, Wiley, 2003

Linking inequalities with binary variables[5]

In a similar way, it is possible to use indicator variables to show whether an inequality holds or does not hold:

$$\delta = 1 \rightarrow \sum_j a_j x_j \leq b$$

This situation can be represented with the following inequality:

$$\sum_j a_j x_j + M\delta \leq M + b$$

where M is an upper bound for the expression $\sum_j a_j x_j - b$

[5]Williams, H.P., Model building in mathematical programming, Wiley, 2003

Linking inequalities with binary variables

If we want to model the following logical condition:

$$\sum_j a_j x_j \leq b \rightarrow \delta = 1$$

it can be conveniently expressed as

$$\delta = 0 \rightarrow \sum_j a_j x_j \not\leq b,$$

or

$$\delta = 0 \rightarrow \sum_j a_j x_j > b. \tag{5}$$

In dealing with the expression $\sum_j a_j x_j > b$, we run in the same difficulty that we met with the expression $x > 0$, We must rewrite

$$\sum_j a_j x_j > b \text{ as } \sum_j a_j x_j \geq b + \epsilon$$

where ϵ is a small tolerance value beyond which we will regard the constraint $\sum_j a_j x_j \leq b$ has having been broken. Should the coefficients a_j be integers as well as the variables x_j, as often happens in this type of situation, there is not difficulty as ϵ can be taken as 1.

Expression (5) may now be rewritten as

$$\delta = 0 \rightarrow -\sum_j a_j x_j + b + \epsilon \leq 0.$$

and can be modeled as

$$\sum_j a_j x_j - (m - \epsilon)\delta \geq b + \epsilon,$$

where m is a lower bound for the expression $\sum_j a_j x_j - b$

Linking inequalities with binary variables

In a similar way, the expressions

$$\delta = 1 \rightarrow \sum_j a_j x_j \geq b$$

and

$$\sum_j a_j x_j \geq b \rightarrow \delta = 1$$

can be modeled with the expressions

$$\sum_j a_j x_j + m\delta \geq m + b$$

and

$$\sum_j a_j x_j - (M + \epsilon)\delta \leq b - \epsilon$$

where m and M are respectively, lower and upper bounds for the expression $\sum_j a_j x_j - b$.

Disjunctive constraints

Suppose that for an problem, we do not require all the constraints to hold simultaneously, but we require at least one subset of constraints to hold. This can be expressed in the following way

$$R_1 \vee R_2 \vee \ldots \vee R_N, \tag{6}$$

where R_i is the proposition "The constraints in subset i are satisfied".

We can model this by introducing N indicator 0-1 variables, $\delta_i, i = 1, \ldots N$, where

$$\delta_i = 1 \rightarrow R_i$$

and impose the condition (6) with the expression:

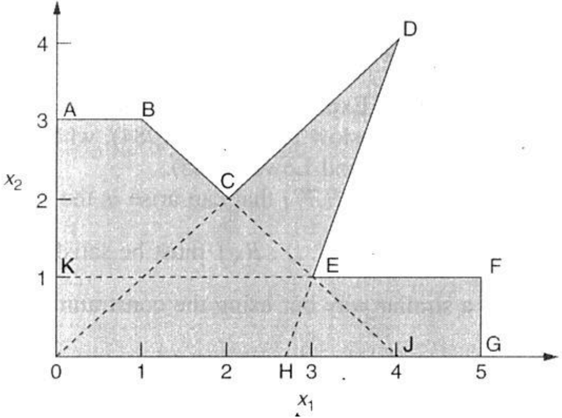
$$\delta_1 + \delta_2 + \ldots + \delta_N \geq 1$$

Disjunctive constraints

In a similar way, we can model the condition "At least k of $(R_1, R_2, \ldots, R_N)$ must be satisfied" with the inequality

$$\delta_1 + \delta_2 + \ldots + \delta_N \geq k$$

Disjunctive constraints



Product of binary variables

Given a situation where we have tree 0-1 binary variables, δ_1, δ_2, and δ_3, and we want to model the following condition

$$\delta_1 = \delta_2 \delta_3$$

This can be done with the following inequalities:

$$\begin{aligned} \delta_1 &\leq \delta_2 \\ \delta_1 &\leq \delta_3 \\ \delta_1 &\geq \delta_2 + \delta_3 - 1 \end{aligned}$$

IP Example[6]

Consider the following integer program

$$\begin{aligned} \max \quad & x_1 + 0.64x_2 \\ 50x_1 + 31x_2 & \leq 250 \\ 3x_1 - 2x_2 & \geq -4 \\ x_1, x_2 & \geq 0, \text{ integer} \end{aligned}$$

[6] Wolsey, L.A., Integer Programming, Wiley Interscience Series in Mathematics and Optimization, 1998.

IP Example[6]

Consider the following integer program

$$
\begin{aligned}
\max \quad & x_1 + 0.64x_2 \\
50x_1 + 31x_2 &\leq 250 \\
3x_1 - 2x_2 &\geq -4 \\
x_1, x_2 &\geq 0, \text{ integer}
\end{aligned}
$$

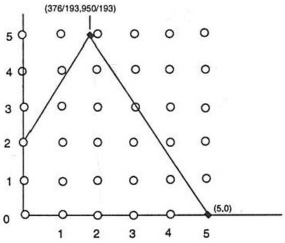


Fig. 1.1 Rounding the LP

[6] Wolsey, L.A., Integer Programming, Wiley Interscience Series in Mathematics and Optimization, 1998.

- Do / Don't do Decisions. For example, if we want to decide whether or not to invest in a certain project, we can define the following binary variable:

$$x = \begin{cases} 1, & \text{if we decide to invest in the project,} \\ 0, & \text{otherwise.} \end{cases}$$

Example: The 0-1 Knapsack Problem[7]. There is a budget b available for investment in projects during the coming year and n projects are under consideration, where a_j is the outlay for project j, and c_j is its expected return. The goal is to choose a set of projects so that the budget is not exceeded and the expected return is maximized.

[7] Wolsey, L.A., Integer Programming, Wiley Interscience Series in Mathematics and Optimization, 1998.

- Definition of the decision variables.

- Definition of the decision variables.

$$x_j = \begin{cases} 1, & \text{if project is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \dots, n.$$

- Definition of the constraints.

- Definition of the decision variables.

$$x_j = \begin{cases} 1, & \text{if project is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots, n.$$

- Definition of the constraints.
  - The budget cannot be exceeded:

- Definition of the decision variables.

$$x_j = \begin{cases} 1, & \text{if project is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots, n.$$

- Definition of the constraints.
  - The budget cannot be exceeded:

$$\sum_{j=1}^{n} a_j x_j \leq b$$

  - The variables are 0-1:

$$x_j \in \{0, 1\} \quad j = 1, \ldots, n$$

- Definition of the objective function

- Definition of the decision variables.

$$x_j = \begin{cases} 1, & \text{if project is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots, n.$$

- Definition of the constraints.
  - The budget cannot be exceeded:

$$\sum_{j=1}^{n} a_j x_j \leq b$$

  - The variables are 0-1:

$$x_j \in \{0, 1\} \quad j = 1, \ldots, n$$

- Definition of the objective function

$$\max \qquad \sum_{j=1}^{n} c_j x_j$$

- Choose among several options. Suppose that in some practical application problem, one or more projects must be selected from a set of n available projects. We can use the following 0-1 variables.

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots n.$$

Modeling with binary variables

- Choose among several options. Suppose that in some practical application problem, one or more projects must be selected from a set of n available projects. We can use the following 0-1 variables.

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots n.$$

Select exactly m of the n available projects:

$$\sum_{j=1}^{n} x_j = m$$

Modeling with binary variables

- Choose among several options. Suppose that in some practical application problem, one or more projects must be selected from a set of n available projects. We can use the following 0-1 variables.

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \ldots n.$$

Select at most m of the n available projects:

$$\sum_{j=1}^{n} x_j \leq m$$

- Choose among several options. Suppose that in some practical application problem, one or more projects must be selected from a set of n available projects. We can use the following 0-1 variables.

$$x_j = \begin{cases} 1, & \text{if project } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \qquad j = 1, \dots n.$$

Select at least m of the n available projects:

$$\sum_{j=1}^{n} x_j \geq m$$

There are n people available to carry out n jobs. Each person is assigned to carry out exactly one job. Some individuals are better suited to particular jobs than others, so there is an estimated cost c_{ij} if person i is assigned to job j. The problem is to find a minimum cost assignment.

- Decision variables:

Modeling with binary variables

The assignment problem

There are n people available to carry out n jobs. Each person is assigned to carry out exactly one job. Some individuals are better suited to particular jobs than others, so there is an estimated cost c_{ij} if person i is assigned to job j. The problem is to find a minimum cost assignment.

- Decision variables:

$$x_{ij} = \begin{cases} 1, & \text{if person } i \text{ does job } j, \\ 0, & \text{otherwise.} \end{cases}, \text{ para } i, j = 1, \ldots, n$$

There are n people available to carry out n jobs. Each person is assigned to carry out exactly one job. Some individuals are better suited to particular jobs than others, so there is an estimated cost c_{ij} if person i is assigned to job j. The problem is to find a minimum cost assignment.

- Decision variables:

$$x_{ij} = \begin{cases} 1, & \text{if person } i \text{ does job } j, \\ 0, & \text{otherwise.} \end{cases}, \text{ para } i, j = 1, \ldots, n$$

- Each person i does one job:

There are n people available to carry out n jobs. Each person is assigned to carry out exactly one job. Some individuals are better suited to particular jobs than others, so there is an estimated cost c_{ij} if person i is assigned to job j. The problem is to find a minimum cost assignment.

- Decision variables:

$$x_{ij} = \begin{cases} 1, & \text{if person } i \text{ does job } j, \\ 0, & \text{otherwise.} \end{cases}, \text{ para } i, j = 1, \ldots, n$$

- Each person i does one job:

$$\sum_{j=1}^{n} x_{ij} = 1 \quad i = 1, \ldots, n.$$

There are n people available to carry out n jobs. Each person is assigned to carry out exactly one job. Some individuals are better suited to particular jobs than others, so there is an estimated cost c_{ij} if person i is assigned to job j. The problem is to find a minimum cost assignment.

- Decision variables:

$$x_{ij} = \begin{cases} 1, & \text{if person } i \text{ does job } j, \\ 0, & \text{otherwise.} \end{cases}, \text{ para } i, j = 1, \dots, n$$

- Each person i does one job:

$$\sum_{j=1}^{n} x_{ij} = 1 \quad i = 1, \dots, n.$$

- Each job j is assigned to one person:

Modeling with binary variables

The assignment problem

There are n people available to carry out n jobs. Each person is assigned to carry out exactly one job. Some individuals are better suited to particular jobs than others, so there is an estimated cost c_{ij} if person i is assigned to job j. The problem is to find a minimum cost assignment.

- Decision variables:

$$x_{ij} = \begin{cases} 1, & \text{if person } i \text{ does job } j, \\ 0, & \text{otherwise.} \end{cases}, \text{ para } i, j = 1, \ldots, n$$

- Each person i does one job:

$$\sum_{j=1}^{n} x_{ij} = 1 \quad i = 1, \ldots, n.$$

- Each job j is assigned to one person:

$$\sum_{i=1}^{n} x_{ij} = 1 \quad j = 1, \ldots, n.$$

- Variables are 0-1.

$$x_{ij} \in \{0,1\} \quad i = 1,\ldots,n, j = 1,\ldots,n.$$

- Objective function:

- Variables are 0-1.

$$x_{ij} \in \{0,1\} \quad i = 1,\ldots,n, j = 1,\ldots,n.$$

- Objective function:

$$\min \quad \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$$

Given a certain number of regions, the problem is to decide where to install a set of emergency service centers. For each possible center the cost of installing a service center, and which regions it can service are known. For instance, if the centers are fire stations, a station can service those regions for which a fire engine is guaranteed to arrive on the scene of a fire within 8 minutes. The goal is to choose a minimum cost set of service centers so that each region is covered. Define the problem as a COP and as an BIP problem.

Given a certain number of regions, the problem is to decide where to install a set of emergency service centers. For each possible center the cost of installing a service center, and which regions it can service are known. For instance, if the centers are fire stations, a station can service those regions for which a fire engine is guaranteed to arrive on the scene of a fire within 8 minutes. The goal is to choose a minimum cost set of service centers so that each region is covered. Define the problem as a COP and as an BIP problem.

Let $M = \{1, \ldots, m\}$ be the set o regions, and $N = \{1, \ldots, n\}$ the set of potential centers. Let $S_j \subseteq M$ be the regions that can be serviced by a center at $j \in N$, and c_j its installation cost. Then the problem can be formulated as a COP in the following way:

$$\min_{T \subset N} \left\{ \sum_{j \in T} c_j : \bigcup_{j \in T} S_j = M \right\}.$$

The problem can be formulated as a (BIP) in the following way:

Let $a_{ij} = \begin{cases} 1, & \text{if } i \in S_j \\ 0, & \text{otherwise.} \end{cases}$

- Decision variables:

The problem can be formulated as a (BIP) in the following way:

Let $a_{ij} = \begin{cases} 1, & \text{if } i \in S_j \\ 0, & \text{otherwise.} \end{cases}$

- Decision variables:

$$x_j = \begin{cases} 1, & \text{if potential center } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases}$$

The problem can be formulated as a (BIP) in the following way:

Let $a_{ij} = \begin{cases} 1, & \text{if } i \in S_j \\ 0, & \text{otherwise.} \end{cases}$

- Decision variables:

$$x_j = \begin{cases} 1, & \text{if potential center } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases}$$

- At least one center must service region i.

The problem can be formulated as a (BIP) in the following way:

Let $a_{ij} = \begin{cases} 1, & \text{if } i \in S_j \\ 0, & \text{otherwise.} \end{cases}$

- Decision variables:

$$x_j = \begin{cases} 1, & \text{if potential center } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases}$$

- At least one center must service region i.

$$\sum_{j \in N} a_{ij} x_j \geq 1 \quad i \in M$$

- All variables are 0-1:

$$x_j \in \{0,1\} \quad j \in N$$

- Objective function:

The problem can be formulated as a (BIP) in the following way:

Let $a_{ij} = \begin{cases} 1, & \text{if } i \in S_j \\ 0, & \text{otherwise.} \end{cases}$

- Decision variables:

$$x_j = \begin{cases} 1, & \text{if potential center } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases}$$

- At least one center must service region i.

$$\sum_{j \in N} a_{ij} x_j \geq 1 \quad i \in M$$

- All variables are 0-1:

$$x_j \in \{0, 1\} \quad j \in N$$

- Objective function:

$$\sum_{j \in N} c_j x_j$$

A salesman must visit each of n cities exactly once and then return to his starting point. The time taken to travel from city i to city j is t_{ij}. Find the order in which he should make his tour so as to finish as quickly as possible. Let $N = \{1, \ldots, n\}$ be the set of cities.

- Decision Variables:

A salesman must visit each of n cities exactly once and then return to his starting point. The time taken to travel from city i to city j is t_{ij}. Find the order in which he should make his tour so as to finish as quickly as possible. Let $N = \{1, \ldots, n\}$ be the set of cities.

- Decision Variables:

$$x_{ij} = \begin{cases} 1, & \text{if salesman goes directly from city } i \text{ to city } j, \\ 0, & \text{otherwise.} \end{cases}$$

A salesman must visit each of n cities exactly once and then return to his starting point. The time taken to travel from city i to city j is t_{ij}. Find the order in which he should make his tour so as to finish as quickly as possible. Let $N = \{1, \ldots, n\}$ be the set of cities.

- Decision Variables:

$$x_{ij} = \begin{cases} 1, & \text{if salesman goes directly from city } i \text{ to city } j, \\ 0, & \text{otherwise.} \end{cases}$$

- Salesman leaves city i exactly once:

A salesman must visit each of n cities exactly once and then return to his starting point. The time taken to travel from city i to city j is t_{ij}. Find the order in which he should make his tour so as to finish as quickly as possible. Let $N = \{1, \ldots, n\}$ be the set of cities.

- Decision Variables:

$$x_{ij} = \begin{cases} 1, & \text{if salesman goes directly from city } i \text{ to city } j, \\ 0, & \text{otherwise.} \end{cases}$$

- Salesman leaves city i exactly once:

$$\sum_{j \in N : j \neq i} x_{ij} = 1 \quad i = 1, \ldots, n.$$

A salesman must visit each of n cities exactly once and then return to his starting point. The time taken to travel from city i to city j is t_{ij}. Find the order in which he should make his tour so as to finish as quickly as possible. Let $N = \{1, \ldots, n\}$ be the set of cities.

- Decision Variables:

$$x_{ij} = \begin{cases} 1, & \text{if salesman goes directly from city } i \text{ to city } j, \\ 0, & \text{otherwise.} \end{cases}$$

- Salesman leaves city i exactly once:

$$\sum_{j \in N : j \neq i} x_{ij} = 1 \quad i = 1, \ldots, n.$$

- Salesman arrives at city j exactly once:

Modeling with binary variables

Traveling salesman problem

A salesman must visit each of n cities exactly once and then return to his starting point. The time taken to travel from city i to city j is t_{ij}. Find the order in which he should make his tour so as to finish as quickly as possible. Let $N = \{1, \ldots, n\}$ be the set of cities.

- Decision Variables:

$$x_{ij} = \begin{cases} 1, & \text{if salesman goes directly from city } i \text{ to city } j, \\ 0, & \text{otherwise.} \end{cases}$$

- Salesman leaves city i exactly once:

$$\sum_{j \in N : j \neq i} x_{ij} = 1 \quad i = 1, \ldots, n.$$

- Salesman arrives at city j exactly once:

$$\sum_{i \in N : i \neq j} x_{ij} = 1 \quad j = 1, \ldots, n.$$

- No sub-tours are allowed:

- No sub-tours are allowed:

$$\sum_{i,j \in S : i \neq j} x_{ij} \leq |S| - 1 \quad S \subset N \quad 2 \leq |S| \leq n-2$$

- Objective function:

- No sub-tours are allowed:

$$\sum_{i,j \in S : i \neq j} x_{ij} \leq |S| - 1 \quad S \subset N \quad 2 \leq |S| \leq n - 2$$

- Objective function:

$$\sum_{i,j \in N : i \neq j} t_{ij} x_{ij}$$