TACsy Symmer School, Vienna 2024

Integer Linear Programming
Modelling, MinMax, Enumeration, CRN

Daniel Merkle, Christoph Flamm

3. juli 2024

Outline

Outline

Integer Linear Programming

Linear Objective
Linear Constraints
but! integer variables

Integer Linear Programming

Linear Objective
Linear Constraints
but! integer variables

$$\begin{aligned} \max\ & \boldsymbol{c}^T \boldsymbol{x} \\ & A\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \end{aligned}$$

Linear Programming
(LP)

Integer Linear Programming

Linear Objective
Linear Constraints
but! integer variables

$$\max \boldsymbol{c}^T \boldsymbol{x} \quad A\boldsymbol{x} \leq \boldsymbol{b} \quad \boldsymbol{x} \geq 0$$

Linear Programming (LP)

$$\max \boldsymbol{c}^T \boldsymbol{x} \quad A\boldsymbol{x} \leq \boldsymbol{b} \quad \boldsymbol{x} \geq 0 \quad \boldsymbol{x} \text{ integer}$$

Integer Linear Programming (ILP)

Integer Linear Programming

Linear Objective
Linear Constraints
but! integer variables

$$\begin{aligned} \max\ & \boldsymbol{c}^T\boldsymbol{x} \\ & A\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \end{aligned}$$

Linear Programming (LP)

$$\begin{aligned} \max\ & \boldsymbol{c}^T\boldsymbol{x} \\ & A\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \\ & \boldsymbol{x}\ \text{integer} \end{aligned}$$

Integer Linear Programming (ILP)

$$\begin{aligned} \max\ & \boldsymbol{c}^T\boldsymbol{x} \\ & A\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \in \{0,1\}^n \end{aligned}$$

Binary Integer Program (BIP)
0/1 Integer Programming

Integer Linear Programming

Linear Objective
Linear Constraints
but! integer variables

$$\begin{aligned} \max\ & \boldsymbol{c}^T\boldsymbol{x} \\ & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \end{aligned}$$

Linear Programming (LP)

$$\begin{aligned} \max\ & \boldsymbol{c}^T\boldsymbol{x} \\ & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \\ & \boldsymbol{x}\ \text{integer} \end{aligned}$$

Integer Linear Programming (ILP)

$$\begin{aligned} \max\ & \boldsymbol{c}^T\boldsymbol{x} \\ & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \in \{0,1\}^n \end{aligned}$$

Binary Integer Program (BIP)
0/1 Integer Programming

$$\begin{aligned} \max\ & \boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{h}^T\boldsymbol{y} \\ & \boldsymbol{A}\boldsymbol{x} + \ \boldsymbol{G}\boldsymbol{y} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \\ & \boldsymbol{y} \geq 0 \\ & \boldsymbol{y}\ \text{integer} \end{aligned}$$

Mixed Integer Linear Programming (MILP)

Integer Linear Programming

Linear Objective
Linear Constraints
but! integer variables

$$\begin{aligned} \max\ & \boldsymbol{c}^T\boldsymbol{x} \\ & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \end{aligned}$$

Linear Programming (LP)

$$\begin{aligned} \max\ & \boldsymbol{c}^T\boldsymbol{x} \\ & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \\ & \boldsymbol{x}\ \text{integer} \end{aligned}$$

Integer Linear Programming (ILP)

$$\begin{aligned} \max\ & \boldsymbol{c}^T\boldsymbol{x} \\ & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \in \{0,1\}^n \end{aligned}$$

Binary Integer Program (BIP)
0/1 Integer Programming

$$\begin{aligned} \max\ & \boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{h}^T\boldsymbol{y} \\ & \boldsymbol{A}\boldsymbol{x} + \ \boldsymbol{G}\boldsymbol{y} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \\ & \boldsymbol{y} \geq 0 \\ & \boldsymbol{y}\ \text{integer} \end{aligned}$$

Mixed Integer Linear Programming (MILP)

$$\begin{aligned} \max\ & f(\boldsymbol{x}) \\ & g(\boldsymbol{x}) \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \end{aligned}$$

Non-linear Programming (NLP)

Recall:

- $\mathbb{Z}$ set of integers
- $\mathbb{Z}^+$ set of positive integer
- $\mathbb{Z}_0^+$ set of nonnegative integers ($\{0\} \cup \mathbb{Z}^+$)
- $\mathbb{N}_0$ set of natural numbers, ie, nonnegative integers $\{0, 1, 2, 3, 4, ...\}$

Outline

Mathematical Programming: Modeling

- Find out exactly what the decision maker needs to know:
  - which investment?
  - which product mix?
  - which job j should a person i do?
- Define Decision Variables of suitable type (continuous, integer valued, binary) corresponding to the needs and Known Parameters corresponding to given data.
- Formulate Objective Function computing the benefit/cost
- Formulate mathematical Constraints indicating the interplay between the different variables.

How to "build" a constraint

- Formulate relationship between the variables in plain words
- Then formulate your sentences using logical connectives **and, or, not, implies**
- Finally convert the logical statement to a mathematical constraint.

Example

- "The power plant must not work in both of two neighbouring time periods"
- on/off is modelled using **binary** integer variables
- $x_i = 1$ or $x_i = 0$
- $x_i = 1$ implies $\Rightarrow x_{i+1} = 0$
- $x_i + x_{i+1} \leq 1$

Outline

The Assignment Problem

Problem

Common application: **Assignees** are being assigned to perform **tasks**.

Suppose we have n persons and n jobs
Each person has a certain proficiency at each job.

Formulate a mathematical model that can be used to find an assignment that maximizes the total proficiency.

The Assignment Problem

Model

Decision Variables:

Objective Function:

The Assignment Problem

Model

Decision Variables:

$$x_{ij} = \begin{cases} 1 \text{ if person } i \text{ is assigned job } j \\ 0 \text{ otherwise,} \end{cases} \quad \text{for } i, j = 1, 2, \ldots, n$$

Objective Function:

The Assignment Problem
Model

Decision Variables:

$$x_{ij} = \begin{cases} 1 \text{ if person } i \text{ is assigned job } j \\ 0 \text{ otherwise,} \end{cases} \quad \text{for } i, j = 1, 2, \ldots, n$$

Objective Function:

$$\max \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \rho_{ij} x_{ij}$$

where ρ_{ij} is person i's proficiency at job j

The Assignment Problem
Model

Constraints:
Each person is assigned one job:

$$\sum_{j=1}^{n} x_{ij} = 1 \text{ for all } i$$

e.g. for person 1 we get $x_{11} + x_{12} + x_{13} + \cdots + x_{1n} = 1$

Each job is assigned to one person:

$$\sum_{i=1}^{n} x_{ij} = 1 \text{ for all } j$$

e.g. for job 1 we get $x_{11} + x_{21} + x_{31} + \cdots + x_{n1} = 1$

Outline

The Knapsack Problem

Problem ..

Input: Given a set of n items, each with a value v_i and weight w_i $(i = 1, \ldots, n)$

Task: determine (the numbers of) for each item to include in a collection so that the total weight is less than a given limit, W, and the total value is as large as possible.

The Knapsack Problem

Problem ..

Input: Given a set of n items, each with a value v_i and weight w_i $(i = 1, \ldots, n)$

Task: determine (the numbers of) for each item to include in a collection so that the total weight is less than a given limit, W, and the total value is as large as possible.

The "knapsack" name derives from the problem faced by someone who is constrained by a fixed-size knapsack and must fill it with the most useful items.

The Knapsack Problem

Problem ..

Input: Given a set of n items, each with a value v_i and weight w_i $(i = 1, \ldots, n)$

Task: determine (the numbers of) for each item to include in a collection so that the total weight is less than a given limit, W, and the total value is as large as possible.

The "knapsack" name derives from the problem faced by someone who is constrained by a fixed-size knapsack and must fill it with the most useful items.

Assuming we can take at least one of any item and that $\sum_i w_i > W$, formulate a mathematical model to determine which items give the largest value.

The Knapsack Problem

Problem ..

Input: Given a set of n items, each with a value v_i and weight w_i $(i = 1, \ldots, n)$

Task: determine (the numbers of) for each item to include in a collection so that the total weight is less than a given limit, W, and the total value is as large as possible.

The "knapsack" name derives from the problem faced by someone who is constrained by a fixed-size knapsack and must fill it with the most useful items.

Assuming we can take at least one of any item and that $\sum_i w_i > W$, formulate a mathematical model to determine which items give the largest value.

Model used, eg, in capital budgeting, project selection, etc.

The 0/1 Knapsack Problem

Decision Variables:

$$x_i = \begin{cases} 1 \text{ if item } i \text{ is taken} \\ 0 \text{ otherwise,} \end{cases} \quad \text{for } i = 1, 2 \ldots, n$$

Objective Function:

$$\max \quad \sum_{i=1}^{n} v_i x_i$$

Constraints:
Knapsack capacity restriction:

$$\sum_{i=1}^{n} w_i x_i \leq W$$

Run Gurobi/Python

```
pip install gurobipy
python3 knapsack.py
```

https://tacsy-school-2024.algochem.techfak.de/

Outline

Set Covering

Problem

Given: a set of regions, a set of possible construction locations for emergency centers, regions that can be served in less than 8 minutes, cost of installing an emergency center in each location.

Task: decide where to install a set of emergency centers such that the total cost is minimized and all regions are safely served

Set Covering

Problem

Given: a set of regions, a set of possible construction locations for emergency centers, regions that can be served in less than 8 minutes, cost of installing an emergency center in each location.

Task: decide where to install a set of emergency centers such that the total cost is minimized and all regions are safely served

As Combinatorial Optimization Problem (COP): $M = \{1, \ldots, m\}$ regions, $N = \{1, \ldots, n\}$ centers, $S_j \subseteq M$ regions serviced by $j \in N$ in 8 min.

$$\min_{T \subseteq N} \left\{ \sum_{j \in T} c_j \,\middle|\, \bigcup_{j \in T} S_j = M \right\}$$

Set Covering

Problem

Given: a set of regions, a set of possible construction locations for emergency centers, regions that can be served in less than 8 minutes, cost of installing an emergency center in each location.

Task: decide where to install a set of emergency centers such that the total cost is minimized and all regions are safely served

As Combinatorial Optimization Problem (COP): $M = \{1, \ldots, m\}$ regions, $N = \{1, \ldots, n\}$ centers, $S_j \subseteq M$ regions serviced by $j \in N$ in 8 min.

$$\min_{T \subseteq N} \left\{ \sum_{j \in T} c_j \,\middle|\, \bigcup_{j \in T} S_j = M \right\}$$

regions: $M = \{1, \ldots, 5\}$
centers: $N = \{1, \ldots, 6\}$
cost of centers: $c_j = 1 \quad \forall j = 1, \ldots, 6$
coverages: $S_1 = (1, 2), S_2 = (1, 3, 5), S_3 = (2, 4, 5), S_4 = (3), S_5 = (1), S_6 = (4, 5)$

Example

- regions: $M = \{1, \ldots, 5\}$
  centers: $N = \{1, \ldots, 6\}$
  cost of centers: $c_j = 1 \quad \forall j = 1, \ldots, 6$
  coverages: $S_1 = (1, 2), S_2 = (1, 3, 5), S_3 = (2, 4, 5), S_4 = (3), S_5 = (1), S_6 = (4, 5)$

-

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ S_1 & S_2 & S_3 & S_4 & S_5 & S_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

As a BIP:

Variables:
$\boldsymbol{x} \in \mathbb{B}^n$, $x_j = 1$ if center j is selected, 0 otherwise

Objective:

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

Constraints:

- incidence matrix: $a_{ij} = \begin{cases} 1 \\ 0 \end{cases}$
- $\sum_{j=1}^{n} a_{ij} x_j \geq 1$

Set covering

cover each of M at least once

1. min, $\geq$
2. all RHS terms are 1
3. all matrix elements are 1

$$\begin{aligned} \min\ & \boldsymbol{c}^T \boldsymbol{x} \\ & A\boldsymbol{x} \geq 1 \\ & \boldsymbol{x} \in \mathbb{B}^n \end{aligned}$$

Set packing

cover as many of M without overlap

1. max, $\leq$
2. all RHS terms are 1
3. all matrix elements are 1

$$\begin{aligned} \max\ & \boldsymbol{c}^T \boldsymbol{x} \\ & A\boldsymbol{x} \leq 1 \\ & \boldsymbol{x} \in \mathbb{B}^n \end{aligned}$$

Set partitioning

cover exactly once each element of M

1. max or min, $=$
2. all RHS terms are 1
3. all matrix elements are 1

$$\begin{aligned} \max\ & \boldsymbol{c}^T \boldsymbol{x} \\ & A\boldsymbol{x} = 1 \\ & \boldsymbol{x} \in \mathbb{B}^n \end{aligned}$$

Set covering

cover each of M at least once

1. min, $\geq$
2. all RHS terms are 1
3. all matrix elements are 1

$$\begin{aligned} \min\ & \boldsymbol{c}^T \boldsymbol{x} \\ & A\boldsymbol{x} \geq 1 \\ & \boldsymbol{x} \in \mathbb{B}^n \end{aligned}$$

Set packing

cover as many of M without overlap

1. max, $\leq$
2. all RHS terms are 1
3. all matrix elements are 1

$$\begin{aligned} \max\ & \boldsymbol{c}^T \boldsymbol{x} \\ & A\boldsymbol{x} \leq 1 \\ & \boldsymbol{x} \in \mathbb{B}^n \end{aligned}$$

Set partitioning

cover exactly once each element of M

1. max or min, $=$
2. all RHS terms are 1
3. all matrix elements are 1

$$\begin{aligned} \max\ & \boldsymbol{c}^T \boldsymbol{x} \\ & A\boldsymbol{x} = 1 \\ & \boldsymbol{x} \in \mathbb{B}^n \end{aligned}$$

Generalization: $RHS \geq 1$
Application examples:

- Aircrew scheduling: M: legs to cover, N: rosters
- Vehicle routing: M: customers, N: routes

A good written example of how to present a model:

2.1. Notation

Let N be the set of operational flight legs and K the set of aircraft types. Denote by n^k the number of available aircraft of type $k \in K$. Define Ω^k, indexed by p, as the set of feasible schedules for aircraft of type $k \in K$ and let index $p = 0$ denote the empty schedule for an aircraft. Next associate with each schedule $p \in \Omega^k$ the value c_p^k denoting the anticipated profit if this schedule is assigned to an aircraft of type $k \in K$ and a_{ip}^k a binary constant equal to 1 if this schedule covers flight leg $i \in N$ and 0 otherwise. Furthermore, let S be the set of stations and $S^k \subseteq S$ the subset having the facilities to serve aircraft of type $k \in K$. Then, define o_{sp}^k and d_{sp}^k to equal to 1 if schedule p, $p \in \Omega^k$, starts and ends respectively at station s, $s \in S^k$, and 0 otherwise.

Denote by θ_p^k, $p \in \Omega^k \backslash \{0\}$, $k \in K$, the binary decision variable which takes the value 1 if schedule p is assigned to an aircraft of type k, and 0 otherwise. Finally, let θ_0^k, $k \in K$, be a nonnegative integer variable which gives the number of unused aircraft of type k.

2.2. Formulation

Using these definitions, the DARSP can be formulated as:

$$\text{Maximize} \sum_{k \in K} \sum_{p \in \Omega^k} c_p^k \theta_p^k \tag{1}$$

subject to:

$$\sum_{k \in K} \sum_{p \in \Omega^k} a_{ip}^k \theta_p^k = 1 \quad \forall i \in N, \tag{2}$$

$$\sum_{p \in \Omega^k} (d_{sp}^k - o_{sp}^k) \theta_p^k = 0 \quad \forall k \in K, \forall s \in S^k, \tag{3}$$

$$\sum_{p \in \Omega^k} \theta_p^k = n^k \quad \forall k \in K, \tag{4}$$

$$\theta_p^k \geq 0 \quad \forall k \in K, \forall p \in \Omega^k, \tag{5}$$

$$\theta_p^k \text{ integer} \quad \forall k \in K, \forall p \in \Omega^k. \tag{6}$$

The objective function (1) states that we wish to maximize the total anticipated profit. Constraints (2) require that each operational flight leg be covered exactly once. Constraints (3) correspond to the flow conservation constraints at the beginning and the end of the day at each station and for each aircraft type. Constraints (4) limit the number of aircraft of type $k \in K$ that can be used to the number available. Finally, constraints (5) and (6) state that the decision variables are nonnegative integers. This model is a Set Partitioning problem with additional constraints.

[from G. Desaulniers, J. Desrosiers, Y. Dumas, M.M. Solomon and F. Soumis. Daily Aircraft Routing and Scheduling. Management Science, 1997, 43(6), 841-855]

Outline

Outline

Matching

Definition (Matching Theory Terminology)

Matching: set of pairwise non adjacent edges

Covered (vertex): a vertex is covered by a matching M if it is incident to an edge in M

Perfect (matching): if M covers each vertex in G

Maximal (matching): if M cannot be extended any further

Maximum (matching): if M covers as many vertices as possible

Matchable (graph): if the graph G has a perfect matching

Matching

Definition (Matching Theory Terminology)

Matching: set of pairwise non adjacent edges

Covered (vertex): a vertex is covered by a matching M if it is incident to an edge in M

Perfect (matching): if M covers each vertex in G

Maximal (matching): if M cannot be extended any further

Maximum (matching): if M covers as many vertices as possible

Matchable (graph): if the graph G has a perfect matching

$$\begin{aligned} \max \quad & \sum_{v \in V} w_e x_e \\ & \sum_{e \in E : v \in e} x_e \leq 1 \qquad \forall v \in V \\ & x_e \in \{0, 1\} \; \forall e \in E \end{aligned}$$

Matching

Definition (Matching Theory Terminology)

Matching: set of pairwise non adjacent edges

Covered (vertex): a vertex is covered by a matching M if it is incident to an edge in M

Perfect (matching): if M covers each vertex in G

Maximal (matching): if M cannot be extended any further

Maximum (matching): if M covers as many vertices as possible

Matchable (graph): if the graph G has a perfect matching

$$\begin{aligned} \max \quad & \sum_{v \in V} w_e x_e \\ & \sum_{e \in E : v \in e} x_e \leq 1 \qquad \forall v \in V \\ & x_e \in \{0, 1\} \; \forall e \in E \end{aligned}$$

Special case: bipartite matching $\equiv$ assignment problems

Vertex Cover

Select a subset $S \subseteq V$ such that each edge has at least one end vertex in S.

Vertex Cover

Select a subset $S \subseteq V$ such that each edge has at least one end vertex in S.

$$\begin{aligned} \min \quad & \sum_{v \in V} x_v \\ & x_v + x_u \geq 1 \qquad \forall u, v \in V, uv \in E \\ & x_v \in \{0,1\} \; \forall v \in V \end{aligned}$$

Traveling Salesman Problem

- Given a set of cities and the distances between each pair, the Traveling Salesman Problem (TSP) seeks to find the shortest possible route that visits each city exactly once and returns to the starting city.

- n locations, c_{ij} cost of travel

Variables:

$$x_{ij} = \begin{cases} 1 \\ 0 \end{cases}$$

Objective:

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

Constraints:

- $$\sum_{j:j\neq i} x_{ij} = 1 \qquad \forall i = 1, \ldots, n$$
$$\sum_{i:i\neq j} x_{ij} = 1 \qquad \forall j = 1, \ldots, n$$

Constraints:

- $$\sum_{j:j\neq i} x_{ij} = 1 \qquad \forall i = 1, \ldots, n$$
$$\sum_{i:i\neq j} x_{ij} = 1 \qquad \forall j = 1, \ldots, n$$

- cut set constraints

$$\sum_{i\in S} \sum_{j\notin S} x_{ij} \geq 1 \qquad \forall S \subset N, S \neq \emptyset$$

- subtour elimination constraints

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

Outline

Modeling Tricks

Objective function and/or constraints do not appear to be linear?

- Absolute values
- Minimize the largest function value
- Maximize the smallest function value
- Constraints include variable division
- Constraints are either/or
- A variable must take one of several candidate values

Modeling: Absolute Values

$$\min \sum_{i=1}^{n} |f_i(\boldsymbol{x})|$$

$$\boldsymbol{x} \in \mathbb{R}^q$$

Modeling: Absolute Values

$$\min \sum_{i=1}^{n} |f_i(\boldsymbol{x})|$$

$$\boldsymbol{x} \in \mathbb{R}^q$$

$$\begin{array}{ll} \min & \sum_{i=1}^{n} z_i \\ \text{s.t.} & z_i \geq f_i(\boldsymbol{x}) \quad i = 1..n \\ & z_i \geq -f_i(\boldsymbol{x}) \; i = 1..n \\ & z_i \in \mathbb{R} \qquad i = 1..n \\ & \boldsymbol{x} \in \mathbb{R}^q \end{array}$$

n additional variables and $2n$ additional constraints.

Modeling: Absolute Values

$$\min \sum_{i=1}^{n} |f_i(\boldsymbol{x})|$$

$$\boldsymbol{x} \in \mathbb{R}^q$$

$$\begin{aligned} \min \ & \sum_{i=1}^{n} z_i & & \\ \text{s.t.} \ & z_i \geq f_i(\boldsymbol{x}) & & i = 1..n \\ & z_i \geq -f_i(\boldsymbol{x}) & & i = 1..n \\ & z_i \in \mathbb{R} & & i = 1..n \\ & \boldsymbol{x} \in \mathbb{R}^q & & \end{aligned}$$

n additional variables and $2n$ additional constraints.

$$\begin{aligned} \min \ & \sum_{i=1}^{n} (z_i^+ + z_i^-) & & \\ \text{s.t.} \ & f_i(\boldsymbol{x}) = z_i^+ - z_i^- & & i = 1..n \\ & z_i^+, z_i^- \geq 0 & & i = 1..n \\ & \boldsymbol{x} \in \mathbb{R}^q & & \end{aligned}$$

$2n$ additional variables and n additional constraints.

Modeling: Minimax

Minimize the largest of a number of function values:

$$\min \quad \max\{f_1(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})\}$$

Modeling: Minimax

Minimize the largest of a number of function values:

$$\min \quad \max\{f_1(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})\}$$

- Introduce an auxiliary variable z:

$$\begin{aligned} \min \quad & z \\ \text{s. t. } & f_1(\boldsymbol{x}) \leq z \\ & f_2(\boldsymbol{x}) \leq z \end{aligned}$$

Modeling: Divisions

Constraints include variable division:

- Constraint of the form

$$\frac{a_1x + a_2y + a_3z}{d_1x + d_2y + d_3z} \leq b$$

Modeling: Divisions

Constraints include variable division:

- Constraint of the form

$$\frac{a_1x + a_2y + a_3z}{d_1x + d_2y + d_3z} \leq b$$

  - Rearrange:

$$a_1x + a_2y + a_3z \leq b(d_1x + d_2y + d_3z)$$

    which gives:

$$(a_1 - bd_1)x + (a_2 - bd_2)y + (a_3 - bd_3)z \leq 0$$

Modeling: "Either/Or Constraints"

In conventional mathematical models, the solution must satisfy all constraints. Suppose that your constraints are "either/or":

$$a_1 x_1 + a_2 x_2 \leq b_1 \quad \text{or}$$
$$d_1 x_1 + d_2 x_2 \leq b_2$$

Modeling: "Either/Or Constraints"

In conventional mathematical models, the solution must satisfy all constraints.
Suppose that your constraints are "either/or":

$$a_1 x_1 + a_2 x_2 \leq b_1 \qquad \text{or}$$
$$d_1 x_1 + d_2 x_2 \leq b_2$$

Introduce new variable $y \in \{0, 1\}$ and a large number M:

$$a_1 x_1 + a_2 x_2 \leq b_1 + My \qquad \text{if } y = 0 \text{ then this is active}$$
$$d_1 x_1 + d_2 x_2 \leq b_2 + M(1 - y) \qquad \text{if } y = 1 \text{ then this is active}$$

Modeling: "Either/Or Constraints"

Binary integer programming allows to model alternative choices:

- Eg: 2 feasible regions, ie, disjunctive constraints, not possible in LP. introduce y auxiliary binary variable and M, a big number:

$$Ax \leq b + My \qquad \text{if } y = 0 \text{ then this is active}$$
$$A'x \leq b' + M(1 - y) \qquad \text{if } y = 1 \text{ then this is active}$$

Modeling: "Either/Or Constraints"

Generally:

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1m}x_m &\leq d_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2m}x_m &\leq d_2 \\
&\vdots \\
a_{N1}x_1 + a_{N2}x_2 + a_{N3}x_3 + \ldots + a_{Nm}x_m &\leq d_N
\end{aligned}
$$

Exactly K of the N constraints must be satisfied.

Modeling: "Either/Or Constraints"

Generally:

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1m}x_m &\leq d_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2m}x_m &\leq d_2 \\
&\vdots \\
a_{N1}x_1 + a_{N2}x_2 + a_{N3}x_3 + \ldots + a_{Nm}x_m &\leq d_N
\end{aligned}
$$

Exactly K of the N constraints must be satisfied.
Introduce binary variables $y_1, y_2, \ldots, y_N$ and a large number M

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1m}x_m &\leq d_1 + My_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2m}x_m &\leq d_2 + My_2 \\
&\vdots \\
a_{m1}x_1 + a_{N2}x_2 + a_{N3}x_3 + \ldots + a_{Nm}x_m &\leq d_N + My_N
\end{aligned}
$$

$$
y_1 + y_2 + \ldots y_N = N - K
$$

K of the y-variables are 0, so K constraints must be satisfied

Modeling: "Either/Or Constraints"

At least $K \leq N$ of $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$, $i = 1, \ldots, N$ must be satisfied

introduce y_i, $i = 1, ..., N$ auxiliary binary variables

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i + My_i, \qquad i = 1..N$$

$$\sum_{i} y_i \leq N - K$$

Modeling: "Possible Constraints Values"

A constraint must take on one of N given values:

$$
\begin{aligned}
a_1x_1 + a_2x_2 + a_3x_3 + \ldots + a_mx_m &= d_1 \text{ or} \\
a_1x_1 + a_2x_2 + a_3x_3 + \ldots + a_mx_m &= d_2 \text{ or} \\
&\vdots \\
a_1x_1 + a_2x_2 + a_3x_3 + \ldots + a_mx_m &= d_N
\end{aligned}
$$

Modeling: "Possible Constraints Values"

A constraint must take on one of N given values:

$$\begin{aligned} a_1x_1 + a_2x_2 + a_3x_3 + \ldots + a_mx_m &= d_1 \text{ or} \\ a_1x_1 + a_2x_2 + a_3x_3 + \ldots + a_mx_m &= d_2 \text{ or} \\ &\vdots \\ a_1x_1 + a_2x_2 + a_3x_3 + \ldots + a_mx_m &= d_N \end{aligned}$$

Introduce binary variables $y_1, y_2, \ldots, y_N$:

$$a_1x_1 + a_2x_2 + a_3x_3 + \ldots + a_mx_m = d_1y_1 + d_2y_2 + \ldots d_Ny_N$$

$$y_1 + y_2 + \ldots y_N = 1$$

Outline

Resume

1. Integer Programming

2. Modeling
   - Assignment Problem
   - Knapsack Problem
   - Set Problems

3. More on Modeling
   - Graph Problems
     - Matching
     - Vertex Cover
     - Traveling Salesman Problem
   - Modeling Tricks

4. ITS, Energy Barriers, PPP

References

H.P. Williams, *Model Building in Mathematical Programming*, John Wiley & Sons, Chichester, Fifth Edition, 2013.

J. Matousek and B. Gärtner, *Understanding and Using Linear Programming*, Springer Berlin Heidelberg, 2007.

Slides based on slide set from Marco Chiarandini (IMADA, SDU)