DM554/DM545

Linear and Integer Programming

Lecture 9

Integer Linear Programming Modeling

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

Outline

1. Integer Programming

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

Discrete Optimization

- Often we need to deal with integral inseparable quantities
- Sometimes rounding can go
- Other times rounding not feasible: eg, presence of a bus on a line is 0.3...

Integer Linear Programming

Linear Objective
Linear Constraints
but! integer variables

The world is not linear: "OR is the art and science of obtaining bad answers to questions to which otherwise worse answers would be given"

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

Linear Programming (LP)

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

Integer (Linear) Programming (ILP)

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

Binary Integer Program (BIP)
0/1 Integer Programming

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

Mixed Integer (Linear) Programming (MILP)

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

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 $\{0, 1, 2, 3, 4, \ldots\}$

Combinatorial Optimization Problems

Definition (Combinatorial Optimization Problem (COP))

Input: Given a finite set $N = \{1, \ldots, n\}$ of objects,
weights $c_j \; \forall j \in N$,
a collection of feasible subsets of N, $\mathcal{F}$

Task: Find a minimum weight feasible subset, ie,

$$\min_{S \subseteq N} \left\{ \sum_{j \in S} c_j \mid S \in \mathcal{F} \right\}$$

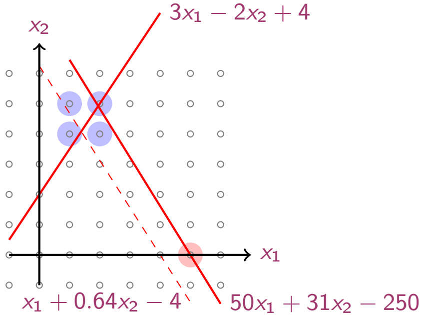
Many COP can be modelled as IP or BIP.

Typically: incidence vector of S, $\mathbf{x}^S \in \mathbb{B}^n$: $x_j^S = \begin{cases} 1 & \text{if } j \in S \\ 0 & \text{otherwise} \end{cases}$

Rounding

$$\begin{aligned} \max\ 100x_1 + 64x_2 & \\ 50x_1 + 31x_2 &\leq 250 \\ 3x_1 - \ \ 2x_2 &\geq -4 \\ x_1, x_2 &\in \mathbb{Z}^+ \end{aligned}$$

LP optimum $(376/193, 950/193)$
IP optimum $(5, 0)$



$\rightsquigarrow$ feasible region convex but not continuous: Now the optimum can be on the border (vertices) but also internal.

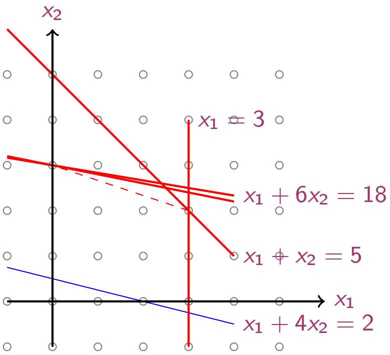
Possible way: solve the relaxed problem.

- If solution is integer, done.
- If solution is rational (never irrational) try rounding to the nearest integers (but may exit feasibility region)
  if in $\mathbb{R}^2$ then 2^2 possible roundings (up or down)
  if in $\mathbb{R}^n$ then 2^n possible roundings (up or down)

Note: rounding does not help in the example above!

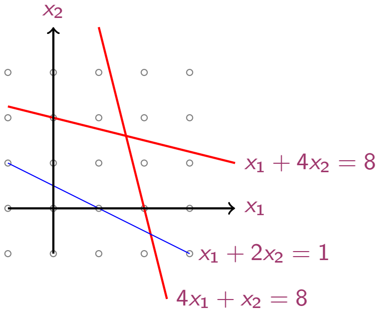
Cutting Planes

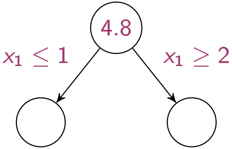
$$\begin{aligned} \max\ & x_1 + 4x_2 \\ & x_1 + 6x_2 \leq 18 \\ & x_1 \qquad\quad \leq 3 \\ & x_1, x_2 \geq 0 \\ & x_1, x_2 \text{ integers} \end{aligned}$$

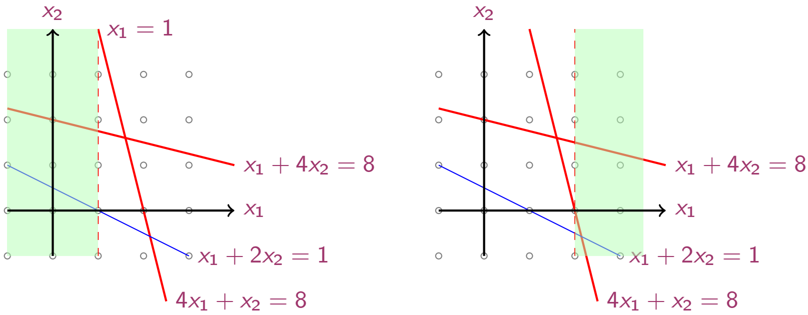


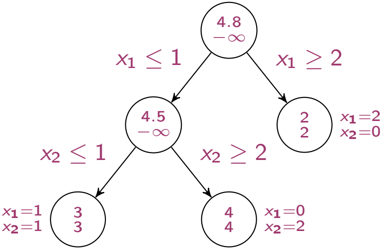
Branch and Bound

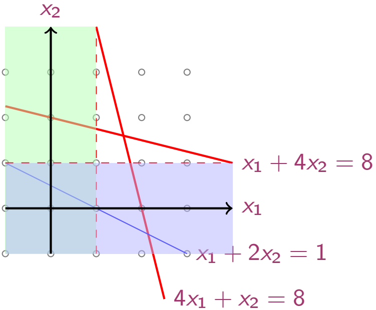
$$
\begin{aligned}
\max\ & x_1 + 2x_2 \\
& x_1 + 4x_2 \leq 8 \\
& 4x_1 + x_2 \leq 8 \\
& x_1, x_2 \geq 0, \text{integer}
\end{aligned}
$$

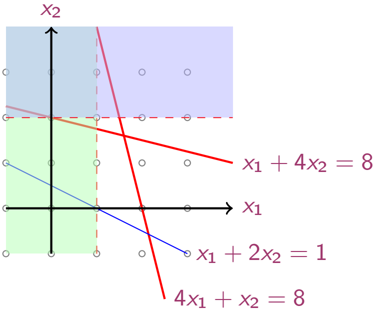



4.8
$x_1 \leq 1$
$x_1 \geq 2$


x_2
$x_1 = 1$
$x_1 + 4x_2 = 8$
x_1
$x_1 + 2x_2 = 1$
$4x_1 + x_2 = 8$
x_2
$x_1 + 4x_2 = 8$
x_1
$x_1 + 2x_2 = 1$
$4x_1 + x_2 = 8$


4.8
$-\infty$
$x_1 \leq 1$
$x_1 \geq 2$
4.5
$-\infty$
2
2
$x_1=2$
$x_2=0$
$x_2 \leq 1$
$x_2 \geq 2$
$x_1=1$
$x_2=1$
3
3
4
4
$x_1=0$
$x_2=2$


x_2
$x_1 + 4x_2 = 8$
x_1
$x_1 + 2x_2 = 1$
$4x_1 + x_2 = 8$


x_2
$x_1 + 4x_2 = 8$
x_1
$x_1 + 2x_2 = 1$
$4x_1 + x_2 = 8$

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
- 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:

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

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 number of each items 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 most 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 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:

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

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

As a COP: $M = \{1, \dots, m\}$ regions, $N = \{1, \dots, n\}$ centers, $S_j \subseteq M$ regions serviced by $j \in N$ in 8 min.

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

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

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

As a BIP:

Variables:
$\mathbf{x} \in \mathbb{B}^n$, $x_j = 1$ 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

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

Set packing

cover as many of M without overlap

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

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

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

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

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]