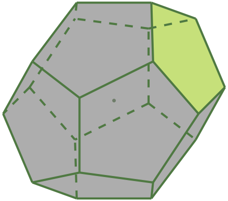
Brief Intro to Linear and Integer Programming



Marco Chiarandini

Institut for Matematik og Datalogi (IMADA)

SYDDANSK UNIVERSITET
UNIVERSITY OF SOUTHERN DENMARK

Outline

1. Linear Programming
   - Modeling
     - Resource Allocation
     - Diet Problem
   - Solution Methods
     - Gaussian Elimination
     - Simplex Method

2. Integer Linear Programming
   - Solution Methods
   - Applications
   - Finance

Outline

1. Linear Programming
   - Modeling
     - Resource Allocation
     - Diet Problem
   - Solution Methods
     - Gaussian Elimination
     - Simplex Method

2. Integer Linear Programming
   - Solution Methods
   - Applications
   - Finance

Outline

1. Linear Programming
- Modeling
  - Resource Allocation
  - Diet Problem
- Solution Methods
  - Gaussian Elimination
  - Simplex Method

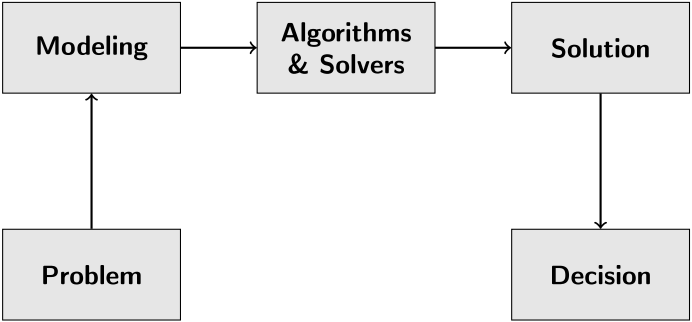
2. Integer Linear Programming
- Solution Methods
- Applications
- Finance

Operations Research

Operation Research (aka, Management Science, Analytics): is the discipline that uses a **scientific approach to decision making**. It seeks to determine how best to design and operate a system, usually under conditions requiring the allocation of scarce resources, by means of **mathematics** and **computer science**. **Quantitative methods for planning and analysis.**

Operations Research

Operation Research (aka, Management Science, Analytics): is the discipline that uses a **scientific approach to decision making**. It seeks to determine how best to design and operate a system, usually under conditions requiring the allocation of scarce resources, by means of **mathematics** and **computer science**. **Quantitative methods for planning and analysis.**



Basic Idea: Build a mathematical model describing exactly what one wants, and what the "rules of the game" are. However, **what is a mathematical model and how?**

Mathematical Modeling

- Find out exactly what the decision makes 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.

Resource Allocation

In manufacturing industry, factory planning: find the best product mix.

Resource Allocation

In manufacturing industry, factory planning: find the best product mix.

Example

A factory makes two products standard and deluxe.

A unit of standard gives a profit of 6 k Dkk.
A unit of deluxe gives a profit of 8 k Dkk.

Resource Allocation

In manufacturing industry, factory planning: find the best product mix.

Example

A factory makes two products standard and deluxe.

A unit of standard gives a profit of 6 k Dkk.
A unit of deluxe gives a profit of 8 k Dkk.

The grinding and polishing times in terms of hours per week for a unit of each type of product are given below:

	Standard	Deluxe
Grinding	5	10
Polishing	4	4

Grinding capacity: 60 hours per week
Polishing capacity: 40 hours per week

Resource Allocation

In manufacturing industry, factory planning: find the best product mix.

Example

A factory makes two products standard and deluxe.

A unit of standard gives a profit of 6 k Dkk.
A unit of deluxe gives a profit of 8 k Dkk.

The grinding and polishing times in terms of hours per week for a unit of each type of product are given below:

	Standard	Deluxe
Grinding	5	10
Polishing	4	4

Grinding capacity: 60 hours per week
Polishing capacity: 40 hours per week
How much of each product, standard and deluxe, should we produce to maximize the profit?

Mathematical Model

Decision Variables

$x_1 \geq 0$ units of product standard
$x_2 \geq 0$ units of product deluxe

Object Function

$\max\ 6x_1 + 8x_2$ maximize profit

Constraints

$$
\begin{array}{lllll}
5x_1 & + & 10x_2 & \leq & 60 \text{ Grinding capacity} \\
4x_1 & + & 4x_2 & \leq & 40 \text{ Polishing capacity}
\end{array}
$$

Mathematical Model

Machines/Materials A and B
Products 1 and 2

$$
\begin{array}{rrcrcr}
\max & 6x_1 & + & 8x_2 & & \\
 & 5x_1 & + & 10x_2 & \leq & 60 \\
 & 4x_1 & + & 4x_2 & \leq & 40 \\
 & & & x_1 & \geq & 0 \\
 & & & x_2 & \geq & 0
\end{array}
$$

a_{ij}	1	2	b_i
A	5	10	60
B	4	4	40
c_j	6	8	

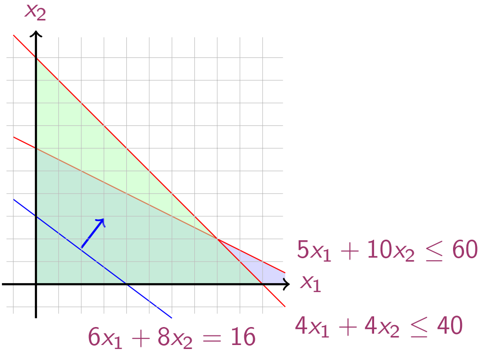
Mathematical Model

Machines/Materials A and B
Products 1 and 2

$$
\begin{array}{lrcrcr}
\max & 6x_1 & + & 8x_2 & & \\
 & 5x_1 & + & 10x_2 & \leq & 60 \\
 & 4x_1 & + & 4x_2 & \leq & 40 \\
 & & & x_1 & \geq & 0 \\
 & & & x_2 & \geq & 0
\end{array}
$$

a_{ij}	1	2	b_i
A	5	10	60
B	4	4	40
c_j	6	8	

Graphical Representation:



Resource Allocation - General Model

Managing a production facility

$1, 2, \ldots, n$	products
$1, 2, \ldots, m$	materials
b_i	units of raw material at disposal
a_{ij}	units of raw material i to produce one unit of product j
$c_j = \sigma_j - \sum_{i=1}^{n} \rho_i a_{ij}$	profit per unit of product j
σ_j	market price of unit of jth product
ρ_i	prevailing market value for material i
x_j	amount of product j to produce

Resource Allocation - General Model

Managing a production facility

$1, 2, \ldots, n$	products
$1, 2, \ldots, m$	materials
b_i	units of raw material at disposal
a_{ij}	units of raw material i to produce one unit of product j
$c_j = \sigma_j - \sum_{i=1}^{n} \rho_i a_{ij}$	profit per unit of product j
σ_j	market price of unit of jth product
ρ_i	prevailing market value for material i
x_j	amount of product j to produce

$$
\begin{array}{rcccccccccl}
\max & c_1x_1 & + & c_2x_2 & + & c_3x_3 & + \ldots + & c_nx_n & = & z \\
\text{subject to} & a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & + \ldots + & a_{1n}x_n & \leq & b_1 \\
 & a_{21}x_1 & + & a_{22}x_2 & + & a_{23}x_3 & + \ldots + & a_{2n}x_n & \leq & b_2 \\
 & \ldots & & & & & & & & \\
 & a_{m1}x_1 & + & a_{m2}x_2 & + & a_{m3}x_3 & + \ldots + & a_{mn}x_n & \leq & b_m \\
 & & & & & & & x_1, x_2, \ldots, x_n & \geq & 0
\end{array}
$$

Notation

$$\begin{array}{rcccccccccccc}
\max & c_1x_1 & + & c_2x_2 & + & c_3x_3 & + & \ldots & + & c_nx_n & = & z \\
\text{s.t.} & a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & + & \ldots & + & a_{1n}x_n & \leq & b_1 \\
& a_{21}x_1 & + & a_{22}x_2 & + & a_{23}x_3 & + & \ldots & + & a_{2n}x_n & \leq & b_2 \\
& \ldots & & & & & & & & & & \\
& a_{m1}x_1 & + & a_{m2}x_2 & + & a_{m3}x_3 & + & \ldots & + & a_{mn}x_n & \leq & b_m \\
& & & & & & & & & x_1, x_2, \ldots, x_n & \geq & 0
\end{array}$$

$$\begin{array}{ll}
\max & \sum_{j=1}^{n} c_j x_j \\
& \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = 1, \ldots, m \\
& \quad x_j \geq 0, \quad j = 1, \ldots, n
\end{array}$$

In Matrix Form

$$
\begin{array}{rcccccccccl}
\max & c_1x_1 & + & c_2x_2 & + & c_3x_3 & + & \ldots & + & c_nx_n & = z \\
\text{s.t.} & a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & + & \ldots & + & a_{1n}x_n & \leq b_1 \\
& a_{21}x_1 & + & a_{22}x_2 & + & a_{23}x_3 & + & \ldots & + & a_{2n}x_n & \leq b_2 \\
& \ldots & & & & & & & & & \\
& a_{m1}x_1 & + & a_{m2}x_2 & + & a_{m3}x_3 & + & \ldots & + & a_{mn}x_n & \leq b_m \\
& & & & & & & & & x_1, x_2, \ldots, x_n & \geq 0
\end{array}
$$

$$c^T = \begin{bmatrix} c_1 & c_2 & \ldots & c_n \end{bmatrix}$$

$$
\begin{array}{rrcl}
\max & z & = & c^Tx \\
& Ax & = & b \\
& x & \geq & 0
\end{array}
$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & & & \\ a_{31} & a_{32} & \ldots & a_{mn} \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Our Numerical Example

$$\begin{aligned} \max \ & \sum_{j=1}^{n} c_j x_j \\ & \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \ \ i = 1, \ldots, m \\ & x_j \geq 0, \ \ j = 1, \ldots, n \end{aligned}$$

$$\begin{aligned} \max \ 6x_1 &+ 8x_2 & & \\ 5x_1 &+ 10x_2 &\leq& \ 60 \\ 4x_1 &+ 4x_2 &\leq& \ 40 \\ & x_1, x_2 &\geq& \ 0 \end{aligned}$$

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

$$x \in \mathbb{R}^n, c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$

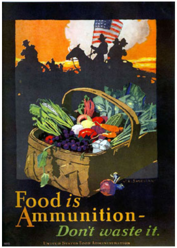
$$\max \ \begin{bmatrix} 6 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 10 \\ 4 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 60 \\ 40 \end{bmatrix}$$

$$x_1, x_2 \geq 0$$

The Diet Problem (Blending Problems)

- Select a set of foods that will satisfy a set of daily nutritional requirement at minimum cost.
- Motivated in the 1930s and 1940s by US army.
- Formulated as a linear programming problem by George Stigler
- First linear program
- (programming intended as planning not computer code)



min   cost/weight
subject to nutrition requirements:
    eat enough but not too much of Vitamin A
    eat enough but not too much of Sodium
    eat enough but not too much of Calories
    ...

The Diet Problem

Suppose there are:

- 3 foods available, corn, milk, and bread, and
- there are restrictions on the number of calories (between 2000 and 2250) and the amount of Vitamin A (between 5000 and 50,000)

Food	Cost per serving	Vitamin A	Calories
Corn	$0.18	107	72
2% Milk	$0.23	500	121
Wheat Bread	$0.05	0	65

The Mathematical Model

Parameters (given data)

F = set of foods

N = set of nutrients

a_{ij} = amount of nutrient j in food i, $\forall i \in F$, $\forall j \in N$

c_i = cost per serving of food $i, \forall i \in F$

F_{mini} = minimum number of required servings of food $i, \forall i \in F$

F_{maxi} = maximum allowable number of servings of food $i, \forall i \in F$

N_{minj} = minimum required level of nutrient $j, \forall j \in N$

N_{maxj} = maximum allowable level of nutrient $j, \forall j \in N$

The Mathematical Model

Parameters (given data)

F = set of foods

N = set of nutrients

a_{ij} = amount of nutrient j in food i, $\forall i \in F$, $\forall j \in N$

c_i = cost per serving of food $i, \forall i \in F$

F_{mini} = minimum number of required servings of food $i, \forall i \in F$

F_{maxi} = maximum allowable number of servings of food $i, \forall i \in F$

N_{minj} = minimum required level of nutrient $j, \forall j \in N$

N_{maxj} = maximum allowable level of nutrient $j, \forall j \in N$

Decision Variables

x_i = number of servings of food i to purchase/consume, $\forall i \in F$

The Mathematical Model

Objective Function: Minimize the total cost of the food

$$\text{Minimize} \sum_{i \in F} c_i x_i$$

The Mathematical Model

Objective Function: Minimize the total cost of the food

$$\text{Minimize} \sum_{i \in F} c_i x_i$$

Constraint Set 1: For each nutrient $j \in N$, at least meet the minimum required level

$$\sum_{i \in F} a_{ij} x_i \geq N_{minj}, \forall j \in N$$

The Mathematical Model

Objective Function: Minimize the total cost of the food

$$\text{Minimize} \sum_{i \in F} c_i x_i$$

Constraint Set 1: For each nutrient $j \in N$, at least meet the minimum required level

$$\sum_{i \in F} a_{ij} x_i \geq N_{minj}, \forall j \in N$$

Constraint Set 2: For each nutrient $j \in N$, do not exceed the maximum allowable level.

$$\sum_{i \in F} a_{ij} x_i \leq N_{maxj}, \forall j \in N$$

The Mathematical Model

Objective Function: Minimize the total cost of the food

$$\text{Minimize} \sum_{i \in F} c_i x_i$$

Constraint Set 1: For each nutrient $j \in N$, at least meet the minimum required level

$$\sum_{i \in F} a_{ij} x_i \geq N_{minj}, \forall j \in N$$

Constraint Set 2: For each nutrient $j \in N$, do not exceed the maximum allowable level.

$$\sum_{i \in F} a_{ij} x_i \leq N_{maxj}, \forall j \in N$$

Constraint Set 3: For each food $i \in F$, select at least the minimum required number of servings

$$x_i \geq F_{mini}, \forall i \in F$$

The Mathematical Model

Objective Function: Minimize the total cost of the food

$$\text{Minimize} \sum_{i \in F} c_i x_i$$

Constraint Set 1: For each nutrient $j \in N$, at least meet the minimum required level

$$\sum_{i \in F} a_{ij} x_i \geq N_{minj}, \forall j \in N$$

Constraint Set 2: For each nutrient $j \in N$, do not exceed the maximum allowable level.

$$\sum_{i \in F} a_{ij} x_i \leq N_{maxj}, \forall j \in N$$

Constraint Set 3: For each food $i \in F$, select at least the minimum required number of servings

$$x_i \geq F_{mini}, \forall i \in F$$

Constraint Set 4: For each food $i \in F$, do not exceed the maximum allowable number of servings.

$$x_i \leq F_{maxi}, \forall i \in F$$

The Mathematical Model

system of equalities and inequalities

$$\min \quad \sum_{i \in F} c_i x_i$$

$$\sum_{i \in F} a_{ij} x_i \geq N_{minj}, \qquad \forall j \in N$$

$$\sum_{i \in F} a_{ij} x_i \leq N_{maxj}, \qquad \forall j \in N$$

$$x_i \geq F_{mini}, \qquad \forall i \in F$$

$$x_i \leq F_{maxi}, \qquad \forall i \in F$$

- The linear program consisted of 9 equations in 77 variables
- Stigler, guessed an optimal solution using a heuristic method
- In 1947, the National Bureau of Standards used the newly developed simplex method to solve Stigler's model.
  It took 9 clerks using hand-operated desk calculators 120 man days to solve for the optimal solution

AMPL Model

```
# diet.mod
set NUTR;
set FOOD;
#
param cost {FOOD} > 0;
param f_min {FOOD} >= 0;
param f_max { i in FOOD} >= f_min[i];
param n_min { NUTR } >= 0;
param n_max {j in NUTR } >= n_min[j];
param amt {NUTR,FOOD} >= 0;
#
var Buy { i in FOOD} >= f_min[i], <= f_max[i]
#
minimize total_cost: sum { i in FOOD } cost [i] * Buy[i];
subject to diet { j in NUTR }:
      n_min[j] <= sum {i in FOOD} amt[i,j] * Buy[i] <= n_max[i];
```

AMPL Model

```
# diet.dat
data;

set NUTR := A B1 B2 C ;
set FOOD := BEEF CHK FISH HAM MCH MTL SPG
     TUR;

param: cost f_min f_max :=
  BEEF 3.19 0 100
  CHK 2.59 0 100
  FISH 2.29 0 100
  HAM 2.89 0 100
  MCH 1.89 0 100
  MTL 1.99 0 100
  SPG 1.99 0 100
  TUR 2.49 0 100 ;

param: n_min n_max :=
   A 700 10000
   C 700 10000
   B1 700 10000
   B2 700 10000 ;
# %
```

```
param amt (tr):
          A C B1 B2 :=
   BEEF 60 20 10 15
   CHK 8 0 20 20
   FISH 8 10 15 10
   HAM 40 40 35 10
   MCH 15 35 15 15
   MTL 70 30 15 15
   SPG 25 50 25 15
   TUR 60 20 15 10 ;
```

Duality

Resource Valuation problem: Determine the value of the raw materials on hand such that: The company must be willing to sell the raw materials should an outside firm offer to buy them at a price consistent with the market

Duality

Resource Valuation problem: Determine the value of the raw materials on hand such that: The company must be willing to sell the raw materials should an outside firm offer to buy them at a price consistent with the market

z_i	value of a unit of raw material i
$\sum_{i=1}^{m} b_i z_i$	opportunity cost (cost of having instead of selling)
ρ_i	prevailing unit market value of material i
σ_j	prevailing unit product price

Duality

Resource Valuation problem: Determine the value of the raw materials on hand such that: The company must be willing to sell the raw materials should an outside firm offer to buy them at a price consistent with the market

z_i	value of a unit of raw material i
$\sum_{i=1}^{m} b_i z_i$	opportunity cost (cost of having instead of selling)
ρ_i	prevailing unit market value of material i
σ_j	prevailing unit product price

Goal is to minimize the lost opportunity cost

$$\min \sum_{i=1}^{m} b_i z_i \tag{1}$$

$$z_i \geq \rho_i, \quad i = 1 \ldots m \tag{2}$$

$$\sum_{i=1}^{m} z_i a_{ij} \geq \sigma_j, \quad j = 1 \ldots n \tag{3}$$

(1) and (2) otherwise contradicting market

Let

$$y_i = z_i - \rho_i$$

markup that the company would make by reselling the raw material instead of producing.

$$\min \sum_{i=1}^{m} y_i b_i + \sum_i \rho_i b_i$$

$$\sum_{i=1}^{m} y_i a_{ij} \geq c_j, \quad j = 1 \ldots n$$

$$y_i \geq 0, \quad i = 1 \ldots m$$

Outline

1. Linear Programming
   - Modeling
     - Resource Allocation
     - Diet Problem
   - Solution Methods
     - Gaussian Elimination
     - Simplex Method

2. Integer Linear Programming
   - Solution Methods
   - Applications
   - Finance

Notions of Computer Science

Algorithm: a finite, well-defined sequence of operations to perform a calculation

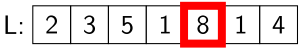
Notions of Computer Science

Algorithm: a finite, well-defined sequence of operations to perform a calculation

Algorithm: LargestNumber

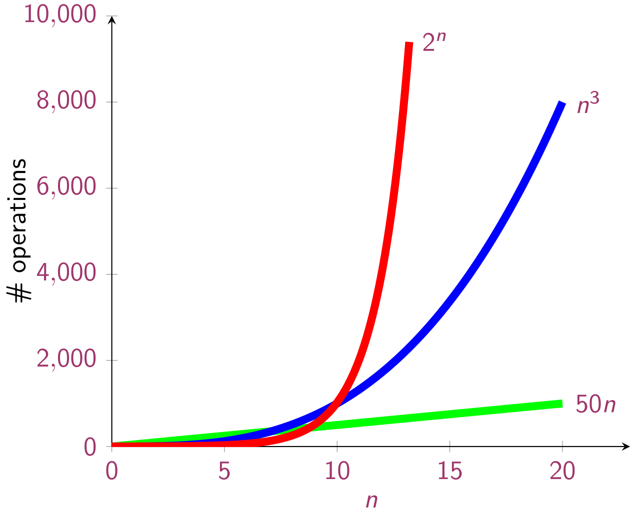
Input: A non-empty list of numbers L
Output: The largest number in the list L

```
largest ← L[0]
foreach each item in the list L do
    if the item > largest then
        largest ← the item
return largest
```



Running time: proportional to number of operations

Growth Functions



NP-hard problems: bad if we have to solve them, good for cryptology

History of Linear Programming (LP)

- Origins date back to Newton, Leibnitz, Lagrange, etc.

History of Linear Programming (LP)

- Origins date back to Newton, Leibnitz, Lagrange, etc.
- In 1827, Fourier described a variable elimination method for systems of linear inequalities, today often called Fourier-Moutzkin elimination (Motzkin, 1937). It can be turned into an LP solver but inefficient.

History of Linear Programming (LP)

- Origins date back to Newton, Leibnitz, Lagrange, etc.
- In 1827, Fourier described a variable elimination method for systems of linear inequalities, today often called Fourier-Moutzkin elimination (Motzkin, 1937). It can be turned into an LP solver but inefficient.
- In 1932, Leontief (1905-1999) Input-Output model to represent interdependencies between branches of a national economy (1976 Nobel prize)

History of Linear Programming (LP)

- Origins date back to Newton, Leibnitz, Lagrange, etc.
- In 1827, Fourier described a variable elimination method for systems of linear inequalities, today often called Fourier-Moutzkin elimination (Motzkin, 1937). It can be turned into an LP solver but inefficient.
- In 1932, Leontief (1905-1999) Input-Output model to represent interdependencies between branches of a national economy (1976 Nobel prize)
- In 1939, Kantorovich (1912-1986): Foundations of linear programming (Nobel prize with Koopmans on LP, 1975)

History of Linear Programming (LP)

- Origins date back to Newton, Leibnitz, Lagrange, etc.
- In 1827, Fourier described a variable elimination method for systems of linear inequalities, today often called Fourier-Moutzkin elimination (Motzkin, 1937). It can be turned into an LP solver but inefficient.
- In 1932, Leontief (1905-1999) Input-Output model to represent interdependencies between branches of a national economy (1976 Nobel prize)
- In 1939, Kantorovich (1912-1986): Foundations of linear programming (Nobel prize with Koopmans on LP, 1975)
- The math subfield of Linear Programming was created by George Dantzig, John von Neumann (Princeton), and Leonid Kantorovich in the 1940s.

History of Linear Programming (LP)

- Origins date back to Newton, Leibnitz, Lagrange, etc.
- In 1827, Fourier described a variable elimination method for systems of linear inequalities, today often called Fourier-Moutzkin elimination (Motzkin, 1937). It can be turned into an LP solver but inefficient.
- In 1932, Leontief (1905-1999) Input-Output model to represent interdependencies between branches of a national economy (1976 Nobel prize)
- In 1939, Kantorovich (1912-1986): Foundations of linear programming (Nobel prize with Koopmans on LP, 1975)
- The math subfield of Linear Programming was created by George Dantzig, John von Neumann (Princeton), and Leonid Kantorovich in the 1940s.
- In 1947, Dantzig (1914-2005) invented the (primal) simplex algorithm working for the US Air Force at the Pentagon. (program=plan)

History of LP (cntd)

- In 1958, Integer Programming was born with cutting planes by Gomory and branch and bound

History of LP (cntd)

- In 1958, Integer Programming was born with cutting planes by Gomory and branch and bound
- In 1979, L. Khachain found a new efficient algorithm for linear programming. It was terribly slow. (Ellipsoid method)

History of LP (cntd)

- In 1958, Integer Programming was born with cutting planes by Gomory and branch and bound
- In 1979, L. Khachain found a new efficient algorithm for linear programming. It was terribly slow. (Ellipsoid method)
- In 1984, Karmarkar discovered yet another new efficient algorithm for linear programming. It proved to be a strong competitor for the simplex method. (Interior point method)

Linear Programming

$$
\begin{array}{rrcll}
\text{objective func.} & \max / \min c^T \cdot x & & & c \in \mathbb{R}^n \\
\text{constraints} & A \cdot x & \gtreqless & b & A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m \\
 & x & \geq & 0 & x \in \mathbb{R}^n, 0 \in \mathbb{R}^n
\end{array}
$$

Essential features of a Linear program:

1. continuity (later, integrality)
2. linearity $\rightsquigarrow$ proportionality + additivity
3. certainty of parameters

Definition

- $\mathbb{N}$ natural numbers, $\mathbb{Z}$ integer numbers, $\mathbb{Q}$ rational numbers, $\mathbb{R}$ real numbers
- column vector and matrices
  scalar product: $y^T x = \sum_{i=1}^{n} y_i x_i$
- linear combination

$$\begin{array}{r} x \in \mathbb{R}^k \\ x_1 \in \mathbb{R}, \ldots, x_k \in \mathbb{R} \\ \lambda = (\lambda_1, \ldots, \lambda_k)^T \in \mathbb{R}^k \end{array} \qquad x = \sum_{i=1}^{k} \lambda_i x_i$$

Fundamental Theorem of LP

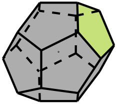
Theorem (Fundamental Theorem of Linear Programming)

Given:

$$\min\{c^T x \mid x \in P\} \text{ where } P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$$

If P is a bounded polyhedron and not empty and x^* is an optimal solution to the problem, then:

- x^* *is an extreme point (vertex) of* P, *or*
- x^* *lies on a face* $F \subset P$ *of optimal solution*



Fundamental Theorem of LP

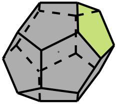
Theorem (Fundamental Theorem of Linear Programming)

Given:

$$\min\{c^T x \mid x \in P\} \text{ where } P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$$

If P is a bounded polyhedron and not empty and x^* is an optimal solution to the problem, then:

- *x^* is an extreme point (vertex) of P, or*
- *x^* lies on a face $F \subset P$ of optimal solution*



Proof:

- assume x^* not a vertex of P then $\exists$ a ball around it still in P. Show that a point in the ball has better cost
- if x^* is not a vertex then it is a convex combination of vertices. Show that all points are also optimal.

Implications:

- the optimal solution is at the intersection of hyperplanes supporting halfspaces.
- hence finitely many possibilities
- Solution method: write all inequalities as equalities and solve all $\binom{n}{m}$ systems of linear equalities
- for each point we need then to check if feasible and if best in cost.
- each system is solved by Gaussian elimination

Gaussian Elimination

1. Forward elimination
   reduces the system to triangular (row echelon) form (or degenerate)
   elementary row operations (or LU decomposition)
2. back substitution

Example:

$$\begin{array}{rcrcrcrr} 2x & + & y & - & z & = & 8 & (I) \\ -3x & - & y & + & 2z & = & -11 & (II) \\ -2x & + & y & + & 2z & = & -3 & (III) \end{array}$$

```
|-----------+---+-----+-----+---|
|           | 2 |   1 |  -1 | 8 |
| 3/2 I+II  | 0 | 1/2 | 1/2 | 1 |
| I+III     | 0 |   2 |   1 | 5 |
|-----------+---+-----+-----+---|
```

$$\begin{array}{rcrcrcrr} 2x & + & y & - & z & = & 8 & (I) \\ & + & \frac{1}{2}y & + & \frac{1}{2}z & = & 1 & (II) \\ & + & 2y & + & 1z & = & 5 & (III) \end{array}$$

```
|-----------+---+-----+-----+---|
|           | 2 |   1 |  -1 | 8 |
|           | 0 | 1/2 | 1/2 | 1 |
| -4 II+III | 0 |   0 |  -1 | 1 |
|-----------+---+-----+-----+---|
```

$$\begin{array}{rcrcrcrr} 2x & + & y & - & z & = & 8 & (I) \\ & + & \frac{1}{2}y & + & \frac{1}{2}z & = & 1 & (II) \\ & & & - & z & = & 1 & (III) \end{array}$$

```
|---+-----+-----+---|
| 2 |   1 |  -1 | 8 |
| 0 | 1/2 | 1/2 | 1 |
| 0 |   0 |  -1 | 1 |
|---+-----+-----+---|
```

$$\begin{array}{rcrcrcrr} 2x & + & y & - & z & = & 8 & (I) \\ & + & \frac{1}{2}y & + & \frac{1}{2}z & = & 1 & (II) \\ & & & - & z & = & 1 & (III) \end{array}$$

```
|---+---+---+----|
| 1 | 0 | 0 |  2 | => x=2
| 0 | 1 | 0 |  3 | => y=3
| 0 | 0 | 1 | -1 | => z=-1
|---+---+---+----|
```

$$\begin{array}{cccrr} x & & & = 2 & (I) \\ & y & & = 3 & (II) \\ & & z & = -1 & (III) \end{array}$$

A Numerical Example

$$\begin{aligned} \max \quad & \sum_{j=1}^{n} c_j x_j \\ & \sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad i = 1, \ldots, m \\ & x_j \ge 0, \quad j = 1, \ldots, n \end{aligned}$$

$$\begin{aligned} \max \quad 6x_1 &+ 8x_2 \\ 5x_1 &+ 10x_2 \le 60 \\ 4x_1 &+ 4x_2 \le 40 \\ & x_1, x_2 \ge 0 \end{aligned}$$

$$\begin{aligned} \max \quad & c^T x \\ & Ax \le b \\ & x \ge 0 \end{aligned}$$

$x \in \mathbb{R}^n, c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$

$$\max \quad \begin{bmatrix} 6 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 10 \\ 4 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \le \begin{bmatrix} 60 \\ 40 \end{bmatrix}$$

$$x_1, x_2 \ge 0$$

Standard Form

Each linear program can be converted in the form:

$$
\begin{aligned}
\max\ & c^T x \\
Ax &\le b \\
x &\in \mathbb{R}^n
\end{aligned}
$$

$c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$

Standard Form

Each linear program can be converted in the form:

$$\begin{aligned} \max \ & c^T x \\ Ax & \leq b \\ x & \in \mathbb{R}^n \end{aligned}$$

$c \in \mathbb{R}^n, A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$

- if equations, then put two constraints, $ax \leq b$ and $ax \geq b$
- if $ax \geq b$ then $-ax \leq -b$
- if $\min c^T x$ then $\max(-c^T x)$

and then be put in standard (or equational) form

$$\begin{aligned} \max \ & c^T x \\ Ax & = b \\ x & \geq 0 \end{aligned}$$

$x \in \mathbb{R}^n, c \in \mathbb{R}^n, A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$

1. "$=$" constraints
2. $x \geq 0$ nonnegativity constraints
3. $(b \geq 0)$
4. max

Simplex Method

introduce slack variables (or surplus)

$$\begin{array}{rcrcrcl} 5x_1 & + & 10x_2 & + & x_3 & = & 60 \\ 4x_1 & + & 4x_2 & + & x_4 & = & 40 \end{array}$$

Simplex Method

introduce slack variables (or surplus)

$$\begin{aligned} 5x_1 + 10x_2 + x_3 &= 60 \\ 4x_1 + 4x_2 + x_4 &= 40 \end{aligned}$$

$$\max \quad z = \begin{bmatrix} 6 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 10 & 1 & 0 \\ 4 & 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 60 \\ 40 \end{bmatrix}$$

$$x_1, x_2, x_3, x_4 \geq 0$$

Canonical std. form: one decision variable is isolated in each constraint and does not appear in the other constraints or in the obj. func.

Simplex Method

introduce slack variables (or surplus)

$$\begin{array}{rcrcrcl} 5x_1 & + & 10x_2 & + & x_3 & = & 60 \\ 4x_1 & + & 4x_2 & + & x_4 & = & 40 \end{array}$$

$$\max \quad z = \begin{bmatrix} 6 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 10 & 1 & 0 \\ 4 & 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 60 \\ 40 \end{bmatrix}$$

$$x_1, x_2, x_3, x_4 \geq 0$$

Canonical std. form: one decision variable is isolated in each constraint and does not appear in the other constraints or in the obj. func.

It gives immediately a feasible solution:

$$x_1 = 0,\ x_2 = 0,\ x_3 = 60,\ x_4 = 40$$

Is it optimal?

Simplex Method

introduce slack variables (or surplus)

$$\begin{array}{rcrcrcl} 5x_1 & + & 10x_2 & + & x_3 & = & 60 \\ 4x_1 & + & 4x_2 & + & x_4 & = & 40 \end{array}$$

$$\max \quad z = \begin{bmatrix} 6 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 10 & 1 & 0 \\ 4 & 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 60 \\ 40 \end{bmatrix}$$

$$x_1, x_2, x_3, x_4 \geq 0$$

Canonical std. form: one decision variable is isolated in each constraint and does not appear in the other constraints or in the obj. func.

It gives immediately a feasible solution:

$$x_1 = 0,\ x_2 = 0,\ x_3 = 60,\ x_4 = 40$$

Is it optimal? Look at signs in $z \rightsquigarrow$ if positive then an increase would improve.

Simplex Tableau

First simplex tableau:

	x_1	x_2	x_3	x_4	$-z$	b
x_3	5	10	1	0	0	60
x_4	4	4	0	1	0	40
	6	8	0	0	1	0

Simplex Tableau

First simplex tableau:

	x_1	x_2	x_3	x_4	$-z$	b
x_3	5	10	1	0	0	60
x_4	4	4	0	1	0	40
	6	8	0	0	1	0

we want to reach this new tableau

	x_1	x_2	x_3	x_4	$-z$	b
x_3	0	?	1	?	0	?
x_1	1	?	0	?	0	?
	0	?	0	?	1	?

Pivot operation:

1. Choose pivot:

   column: one with positive coefficient in obj. func. (to discuss later)

   row: ratio between coefficient b and pivot column: choose the one with smallest ratio:

$$\theta = \min_i \left\{ \frac{b_i}{a_{is}} : a_{is} > 0 \right\}, \qquad \theta \text{ increase value of entering var.}$$

2. elementary row operations to update the tableau

- x_4 leaves the basis, x_1 enters the basis
  - Divide row pivot by pivot
  - Send to zero the coefficient in the pivot column of the first row
  - Send to zero the coefficient of the pivot column in the third (cost) row

- x_4 leaves the basis, x_1 enters the basis
  - Divide row pivot by pivot
  - Send to zero the coefficient in the pivot column of the first row
  - Send to zero the coefficient of the pivot column in the third (cost) row

	x1	x2	x3	x4	-z	b
I'=I-5II'	0	5	1	-5/4	0	10
II'=II/4	1	1	0	1/4	0	10
III'=III-6II'	0	2	0	-6/4	1	-60

- x_4 leaves the basis, x_1 enters the basis
  - Divide row pivot by pivot
  - Send to zero the coefficient in the pivot column of the first row
  - Send to zero the coefficient of the pivot column in the third (cost) row

	x1	x2	x3	x4	-z	b
I'=I-5II'	0	5	1	-5/4	0	10
II'=II/4	1	1	0	1/4	0	10
III'=III-6II'	0	2	0	-6/4	1	-60

From the last row we read: $2x_2 - 3/2x_4 - z = -60$, that is:
$z = 60 + 2x_2 - 3/2x_4$.
Since x_2 and x_4 are nonbasic we have $z = 60$ and
$x_1 = 10, x_2 = 0, x_3 = 10, x_4 = 0$.

- Done?

- x_4 leaves the basis, x_1 enters the basis
  - Divide row pivot by pivot
  - Send to zero the coefficient in the pivot column of the first row
  - Send to zero the coefficient of the pivot column in the third (cost) row

	x1	x2	x3	x4	-z	b
I'=I-5II'	0	5	1	-5/4	0	10
II'=II/4	1	1	0	1/4	0	10
III'=III-6II'	0	2	0	-6/4	1	-60

From the last row we read: $2x_2 - 3/2x_4 - z = -60$, that is:
$z = 60 + 2x_2 - 3/2x_4$.
Since x_2 and x_4 are nonbasic we have $z = 60$ and
$x_1 = 10, x_2 = 0, x_3 = 10, x_4 = 0$.

- Done? No! Let x_2 enter the basis

- x_4 leaves the basis, x_1 enters the basis
  - Divide row pivot by pivot
  - Send to zero the coefficient in the pivot column of the first row
  - Send to zero the coefficient of the pivot column in the third (cost) row

	x1	x2	x3	x4	-z	b
I'=I-5II'	0	5	1	-5/4	0	10
II'=II/4	1	1	0	1/4	0	10
III'=III-6II'	0	2	0	-6/4	1	-60

From the last row we read: $2x_2 - 3/2x_4 - z = -60$, that is:
$z = 60 + 2x_2 - 3/2x_4$.
Since x_2 and x_4 are nonbasic we have $z = 60$ and
$x_1 = 10, x_2 = 0, x_3 = 10, x_4 = 0$.

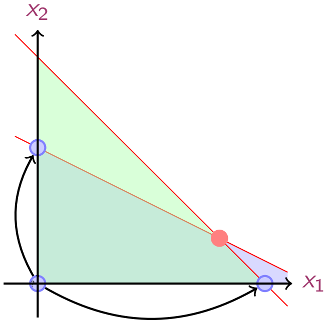
- Done? No! Let x_2 enter the basis

	x1	x2	x3	x4	-z	b
I'=I/5	0	1	1/5	-1/4	0	2
II'=II-I'	1	0	-1/5	1/2	0	8
III'=III-2I'	0	0	-2/5	-1	1	-64

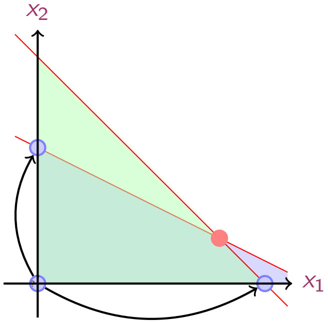
Optimality:
The basic solution is optimal when the coefficient of the nonbasic variables (reduced costs) in the corresponding simplex tableau are nonpositive, ie, such that:

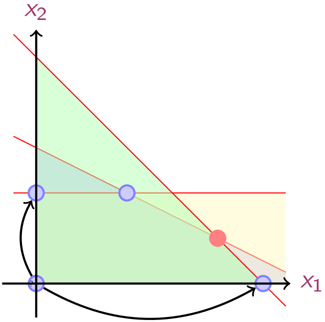
$$\bar{c}_N \leq 0$$

Graphical Representation



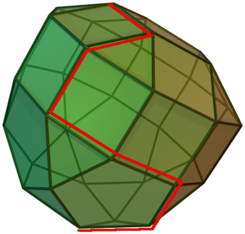
Graphical Representation





Efficiency of Simplex Method

- Trying all points is $\approx 4^m$
- In practice between $2m$ and $3m$ iterations
- Clairvoyant's rule: shortest possible sequence of steps
  Hirsh conjecture $O(n)$ but best known $n^{1+\ln n}$



Outline

1. Linear Programming
Modeling
Resource Allocation
Diet Problem
Solution Methods
Gaussian Elimination
Simplex Method

2. Integer Linear Programming
Solution Methods
Applications
Finance

Outline

1. Linear Programming
Modeling
Resource Allocation
Diet Problem
Solution Methods
Gaussian Elimination
Simplex Method

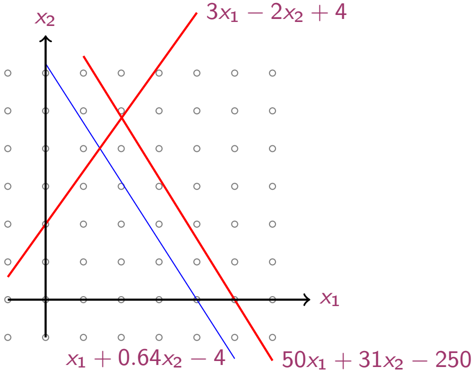
2. Integer Linear Programming
Solution Methods
Applications
Finance

Integer Linear Programming Problem

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

Integer Linear Programming Problem

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

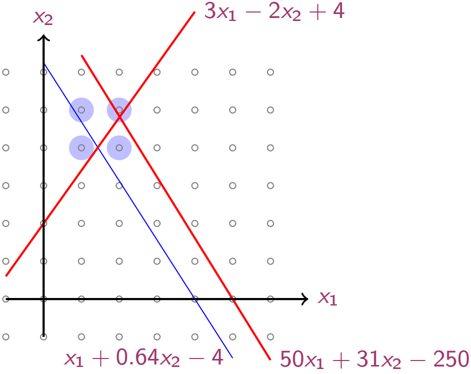


$\rightsquigarrow$ feasible region convex but not continuous: Now the optimum can be on the border (vertices) but also internal.

Integer Linear Programming Problem

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

LP optimum $(376/193, 950/193)$

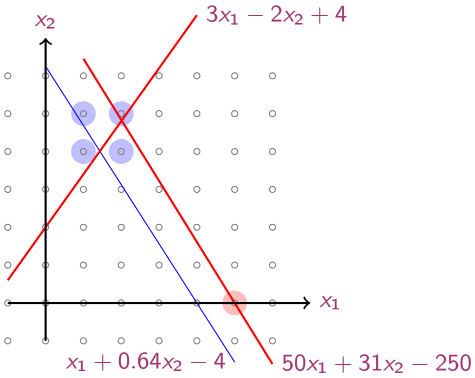


$\leadsto$ feasible region convex but not continuous: Now the optimum can be on the border (vertices) but also internal.

Integer Linear Programming Problem

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

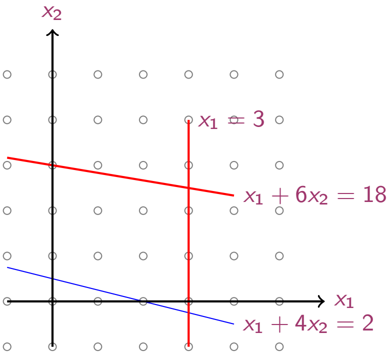
LP optimum $(376/193, 950/193)$
IP optimum $(5, 0)$



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

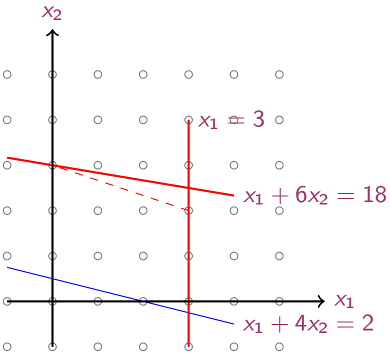
Cutting Planes

$$
\begin{array}{llll}
\max & x_1 + 4x_2 & & \\
& x_1 + 6x_2 & \leq & 18 \\
& x_1 & \leq & 3 \\
& x_1, x_2 & \geq & 0 \\
& x_1, x_2 \text{integer} & &
\end{array}
$$



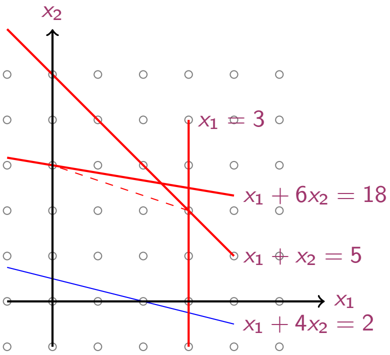
Cutting Planes

$$
\begin{array}{lrcrcl}
\max & x_1 & + & 4x_2 & & \\
 & x_1 & + & 6x_2 & \leq & 18 \\
 & x_1 & & & \leq & 3 \\
 & & & x_1, x_2 & \geq & 0 \\
 & & & x_1, x_2 & \text{integer} &
\end{array}
$$



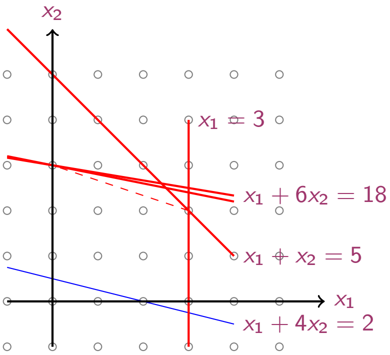
Cutting Planes

$$
\begin{array}{lrcrcl}
\max & x_1 & + & 4x_2 & & \\
 & x_1 & + & 6x_2 & \leq & 18 \\
 & x_1 & & & \leq & 3 \\
 & & & x_1, x_2 & \geq & 0 \\
 & & & x_1, x_2 & \text{integer} &
\end{array}
$$



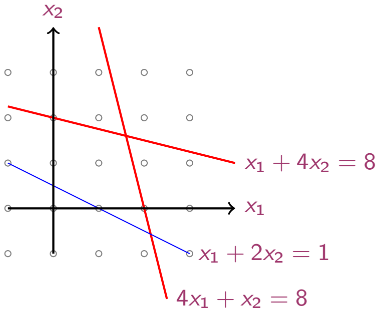
Cutting Planes

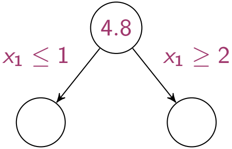
$$
\begin{array}{lrcrcl}
\max & x_1 & + & 4x_2 & & \\
 & x_1 & + & 6x_2 & \leq & 18 \\
 & x_1 & & & \leq & 3 \\
 & & & x_1, x_2 & \geq & 0 \\
 & & & x_1, x_2 & \text{integer} &
\end{array}
$$

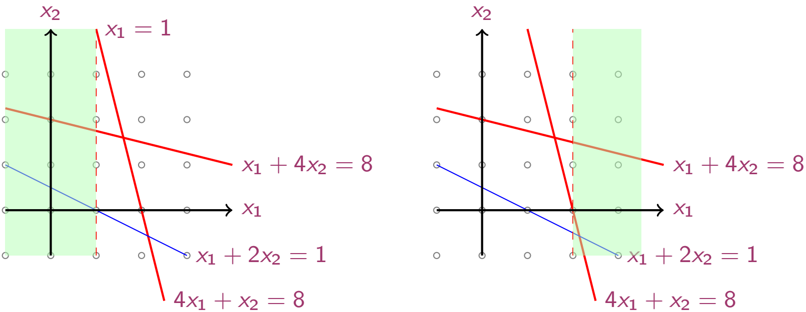


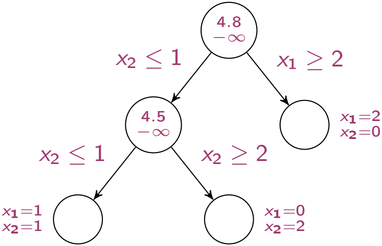
Branch and Bound

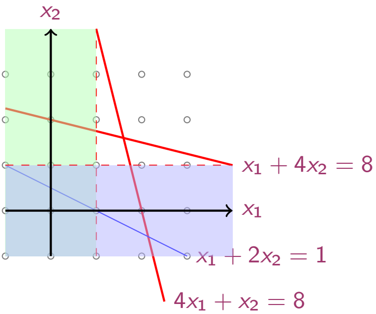
$$\begin{array}{rcrcl} \max & x_1 & + & 2x_2 & \\ & x_1 & + & 4x_2 & \leq 8 \\ & 4x_1 & + & x_2 & \leq 8 \\ & & & x_1, x_2 & \geq 0, \text{integer} \end{array}$$

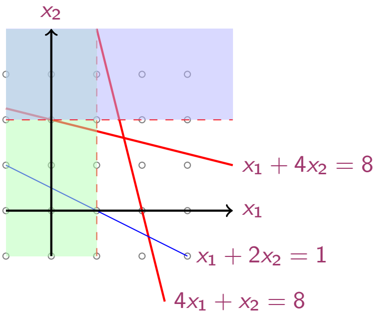


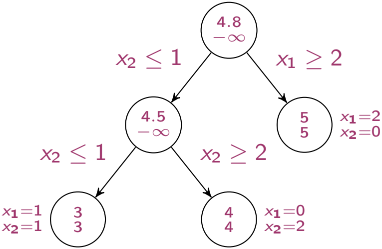

4.8
$x_1 \leq 1$
$x_1 \geq 2$

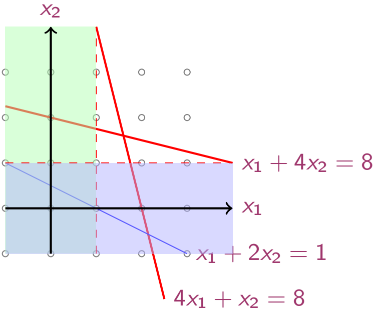

x_2
$x_1 = 1$
$x_1 + 4x_2 = 8$
x_1
$x_1 + 2x_2 = 1$
$4x_1 + x_2 = 8$
x_2
$x_1 + 4x_2 = 8$
x_1
$x_1 + 2x_2 = 1$
$4x_1 + x_2 = 8$

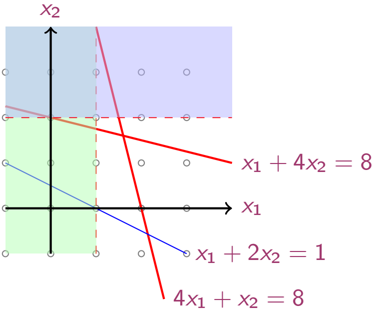

4.8
$-\infty$
$x_2 \leq 1$
$x_1 \geq 2$
4.5
$-\infty$
$x_1=2$
$x_2=0$
$x_2 \leq 1$
$x_2 \geq 2$
$x_1=1$
$x_2=1$
$x_1=0$
$x_2=2$


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


4.8
$-\infty$
$x_2 \leq 1$
$x_1 \geq 2$
4.5
$-\infty$
5
5
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

1. Linear Programming
Modeling
Resource Allocation
Diet Problem
Solution Methods
Gaussian Elimination
Simplex Method

2. Integer Linear Programming
Solution Methods
Applications
Finance

Budget Allocation

(aka, knapsack problem)
There is a budget B available for investments in projects during the coming year and n projects are under consideration, where a_j is the cost of project j and c_j its expected return.
GOAL: chose a set of project such that the budget is not exceeded and the expected return is maximized.

Variables $x_j = 1$ if project j is selected and $x_j = 0$ otherwise

Objective

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

Constraints

$$\begin{aligned} \textstyle\sum_{j=1}^{n} a_j x_j &\leq B \\ x_j &\in \{0, 1\} \forall j = 1, \ldots, n \end{aligned}$$

Facility Location

Given a certain number of regions, where to install a set of fire stations such that all regions are serviced within 8 minutes? For each station the cost of installing the station and which regions it covers are known.

Variables:
$x_j = 1$ if the center j is selected and $x_j = 0$ otherwise

Objective:

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

Constraints:

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1 \forall i = 1, \ldots, m$$
$$x_j \in \{0, 1\} \forall j = 1, \ldots, n$$

Other Applications of MILP

- Energy planning unit commitment
  (more than 1.000.000 variables of which 300.000 integer)

Other Applications of MILP

- Energy planning unit commitment (more than 1.000.000 variables of which 300.000 integer)

- Scheduling/Timetabling
  - Examination timetabling/ train timetabling
- Manpower Planning
  - Crew Rostering (airline crew, rail crew, nurses)
- Routing
  - Vehicle Routing Problem (trucks, planes, trains ...)

Outline

1. Linear Programming
   - Modeling
     - Resource Allocation
     - Diet Problem
   - Solution Methods
     - Gaussian Elimination
     - Simplex Method

2. Integer Linear Programming
   - Solution Methods
   - Applications
   - Finance

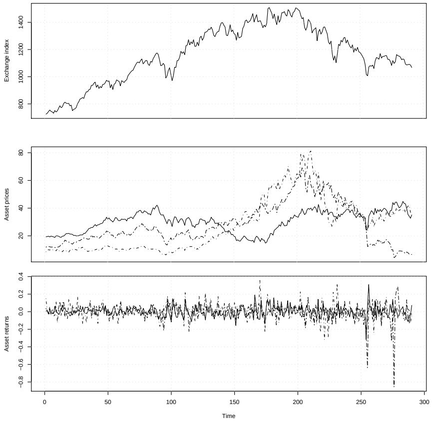
Finance

In Finance LP can be used:

- By a government to design an optimum tax package to achieve some required aim (in particular, an improvement in the balance of payments).
- In revenue management, concerned with setting prices for goods at different times in order to maximize revenue. It is particularly applicable to the hotel, catering, airline and train industries.
- In portfolio selection

Portfolio Selection

Given a sum of money to invest, how to spend it among a portfolio of shares and stocks. The objective is to maintain a certain level of risk and to maximize the expected rate of return from the investment.



c_{jt}	A	B	C
1	19.33	8.52	11.84
2	19.46	9.89	12.28
3	19.75	9.97	12.34
4	19.21	9.75	12.12
5	19.83	10.34	11.84
6	19.54	9.87	11.94
7	19.25	10.09	11.69
8	18.83	9.63	11.56
9	20.04	9.23	11.62
10	19.96	10.43	11.84
11	19.75	9.19	12.00
12	19.12	9.38	12.47
13	18.91	8.92	14.00
14	19.79	8.58	14.25
15	19.83	9.55	15.03

r_{jt}	A	B	C
1	0.01	0.15	0.04
2	0.01	0.01	0.00
3	-0.03	-0.02	-0.02
4	0.03	0.06	-0.02
5	-0.01	-0.05	0.01
6	-0.01	0.02	-0.02
7	-0.02	-0.05	-0.01
8	0.06	-0.04	0.01
9	-0.00	0.12	0.02
10	-0.01	-0.13	0.01
11	-0.03	0.02	0.04
12	-0.01	-0.05	0.12
13	0.05	-0.04	0.02
14	0.00	0.11	0.05
15	0.04	0.02	0.12

The trend of the Stock Exchange index (top), and the price (middle) and the returns (bottom) of three investments.

Portfolio Selection - Modeling

Variables: a collection of nonnegative numbers $0 \leq 0x_j \leq 1, j = 1, \ldots, N$ that divide the capital we want to invest on the stocks $j = 1, \ldots, N$.

Portfolio Selection - Modeling

Variables: a collection of nonnegative numbers $0 \leq 0x_j \leq 1, j = 1, \ldots, N$ that divide the capital we want to invest on the stocks $j = 1, \ldots, N$.

Objective:

The return (on each Krone) in the next time period that one would obtain from the investment in a portfolio is

$$R = \sum_j x_j R_j$$

and the expected return:

$$E[R] = \sum_j x_j E[R_j]$$

Portfolio Selection - Modeling

Variables: a collection of nonnegative numbers $0 \leq 0x_j \leq 1, j = 1, \ldots, N$ that divide the capital we want to invest on the stocks $j = 1, \ldots, N$.

Objective:

The return (on each Krone) in the next time period that one would obtain from the investment in a portfolio is

$$R = \sum_j x_j R_j$$

and the expected return:

$$E[R] = \sum_j x_j E[R_j]$$

We do not know $E[R_j]$ $\rightsquigarrow$ a good guess is that it is like the average from past

$$E[R_j] \approx \hat{R}_j = \frac{1}{T} \sum_{t=1}^{T} r_{jt}$$

$$E[R] \approx \hat{R} = \sum_{j=1}^{N} x_j \hat{R}_j = \sum_{j=1}^{N} x_j \frac{1}{T} \sum_{t=1}^{T} r_{jt} = \frac{1}{T} \sum_{t=1}^{T} \sum_{j=1}^{N} x_j r_{jt}$$

Portfolio Selection - Modeling

Constraints: All and only the capital must used:

$$\sum_{j=1}^{N} x_j = 1$$

Portfolio Selection - Modeling

Constraints: All and only the capital must used:

$$\sum_{j=1}^{N} x_j = 1$$

Risk: even though investments are expected to do very well in the long run, they also tend to be erratic in the short term.
Many ways to define risk.
One way is to define the risk associated with an asset as $x_j|R_j - E[R_j]|$ and then for the whole portfolio as the *mean absolute deviation* (MAD):

$$E\left[|R - E[R]|\right] = E\left[\left|\sum_j x_j(R_j - E[R_j])\right|\right] \leq \epsilon$$

Portfolio Selection - Modeling

Constraints: All and only the capital must used:

$$\sum_{j=1}^{N} x_j = 1$$

Risk: even though investments are expected to do very well in the long run, they also tend to be erratic in the short term.
Many ways to define risk.
One way is to define the risk associated with an asset as $x_j|R_j - E[R_j]|$ and then for the whole portfolio as the *mean absolute deviation* (MAD):

$$E\left[|R - E[R]|\right] = E\left[\Big|\sum_{j} x_j(R_j - E[R_j])\Big|\right] \leq \epsilon$$

Again, we do not have R_j hence, the estimates for reward $E[R]$ and risk MAD are:

$$\widehat{MAD} = \frac{1}{T}\sum_{t=1}^{T}\left[\Big|\sum_{j=1}^{N} x_j(r_{jt} - \hat{R}_j)\Big|\right] \leq \epsilon$$

Portfolio Selection - Final Model

$$\max \frac{1}{T} \sum_{t=1}^{T} \sum_{j=1}^{N} x_j r_{jt}$$

$$\text{s.t.} \sum_{j=1}^{N} x_j = 1$$

$$\sum_{j=1}^{N} x_j (r_{jt} - \hat{R}_j) \leq \epsilon \quad \forall t = 1..T$$

$$\sum_{j=1}^{N} x_j (\hat{R}_j - r_{j,t}) \leq \epsilon \quad \forall t = 1..T$$

$$0 \leq x_j \leq 1 \quad \forall j = 1..N$$

Possible Extensions (1)

Due to management costs, at least 10 different assets must be bought.

Possible Extensions (1)

Due to management costs, at least 10 different assets must be bought.
We need to introduce binary variables z_j for each $j = 1..N$ that indicates whether we are buying or not the asset and then add two constraints to the model of Task 1:

$$\max \frac{1}{T} \sum_{t=1}^{T} \sum_{j=1}^{N} x_j r_{jt}$$

$$\text{s.t.}(2) - (4)$$

$$0 \leq x_j \leq 1 \quad \forall j = 1..N$$

$$z_j \geq x_j \quad \forall j = 1..N$$

$$\sum_{j=1}^{N} z_j \leq 10$$

$$z_j \in \{0, 1\} \quad \forall j = 1..N$$

Possible Extensions (2)

Another practical issue due to management costs: the fraction of assets to allocate in one investment can be either zero **or** a value between 0.02 and 1.

$$
\begin{aligned}
& \max \frac{1}{T} \sum_{t=1}^{T} \sum_{j=1}^{N} x_j r_{jt} \\
& \text{s.t.}(2) - (4) \\
& \quad x_j \le z_j \quad \forall j = 1..N \\
& \quad x_i \ge 0.02 z_j \quad \forall j = 1..N \\
& \quad 0 \le x_j \le 1 \quad \forall j = 1..N \\
& \quad z_j \in \{0, 1\} \quad \forall j = 1..N
\end{aligned}
$$

Example

$$\begin{array}{rrcrcr} \max & 6x_1 & + & 8x_2 & & \\ & 5x_1 & + & 10x_2 & \leq & 60 \\ & 4x_1 & + & 4x_2 & \leq & 40 \\ & & & x_1, x_2 & \geq & 0 \end{array}$$

	x_1	x_2	x_3	x_4	$-z$	b
x_3	5	10	1	0	0	60
x_4	4	4	0	1	0	40
	6	8	0	0	1	0

$$\begin{array}{rclrcrcr} x_3 & = & 60 & - & 5x_1 & - & 10x_2 \\ x_4 & = & 40 & - & 4x_1 & - & 4x_2 \\ \hline z & = & & + & 6x_1 & + & 8x_2 \end{array}$$

Example

$$
\begin{array}{lrcrcr}
\max & 6x_1 & + & 8x_2 & & \\
 & 5x_1 & + & 10x_2 & \leq & 60 \\
 & 4x_1 & + & 4x_2 & \leq & 40 \\
 & & & x_1, x_2 & \geq & 0
\end{array}
$$

	x_1	x_2	x_3	x_4	$-z$	b
x_3	5	10	1	0	0	60
x_4	4	4	0	1	0	40
	6	8	0	0	1	0

$$
\begin{array}{rcrcrcr}
x_3 & = & 60 & - & 5x_1 & - & 10x_2 \\
x_4 & = & 40 & - & 4x_1 & - & 4x_2 \\
\hline
z & = & & + & 6x_1 & + & 8x_2
\end{array}
$$

...

	x_1	x_2	x_3	x_4	$-z$	b
x_2	0	1	$1/5$	$-1/4$	0	2
x_1	1	0	$-1/5$	$1/2$	0	8
	0	0	$-2/5$	-1	1	-64

$$
\begin{array}{rcrcrcr}
x_1 & = & 2 & - & 1/5x_3 & + & 1/4x_4 \\
x_2 & = & 8 & + & 1/5x_3 & - & 1/2x_4 \\
\hline
z & = & 64 & - & 2/5x_3 & - & 1x_4
\end{array}
$$

Exception Handling

1. Unboundedness
2. More than one solution
3. Degeneracies
   - benign
   - cycling
4. Infeasible starting

Summary

1. Linear Programming
   - Modeling
     - Resource Allocation
     - Diet Problem
   - Solution Methods
     - Gaussian Elimination
     - Simplex Method

2. Integer Linear Programming
   - Solution Methods
   - Applications
   - Finance

A nice talk on planning at DSB-S http://www.dr.dk/DR2/Danskernes+akademi/IT_teknik/Saet_dog_et_andet_tog_ind.htm