Deloitte.

Deloitte Data Science Academy

Lesson 4: **Optimization**

Optimization and its use cases, Linear Programming and Integer Linear Programming, Branch and Bound algorithm

19/11/2019

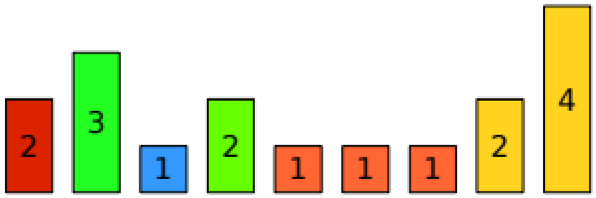
Optimization

- **Optimization** is a term for the mathematical discipline that is concerned with the minimization/maximization of some **objective function** subject to **constraints** or decision that no solution exists.
- Combinatorics is the mathematics of discretely structured problems.
- Combinatorial optimization is an optimization that deals with discrete variables.

- Typical application areas:
  - Production (production speed up, cost reduction, efficient utilization of resources…)
  - Transportation (fuel saving, reduction of delivery time…)
  - Employees scheduling (reduction of human resources…)
  - Hardware design (acceleration of computations…)
  - Communication network design (end-to-end delay reduction)

Problems

- **Bin packing problem:**
  - Objects of different sizes and containers of given capacity



- **Container loading:**
  - To store as much boxes as possible in a container
  - Constraints – size, capacity, loading process, stability, orientation,...
- **Assignments of shifts of employees, Project Scheduling:**
  - **Nurses scheduling**
    - Demand, holidays, regulation, personal preferences, fairness, ...
- **Project scheduling**
  - When should each activity of a project start, resources constraints, precedence, min-max time, etc.

Problems

- **Route planning in automated warehouse:**
  - Can be formulated as assymetric TSP
- **Computer graphics – coloring (Sýkora)**
  - Multiway cut problem
- **Supply network management:**
  - Given a prediction of customers demands, the aim is to plan production and transport of products in order to maximize the profit.
- **City planning**
  - New Fire department, EV charging station, etc..

Why bother?

Typical goals of optimization:

- automation of the design/decission process
- increase the volume of the production (shorter production-line cycle)
- cost reduction (fuel saving, less machines)
- risk reduction (error elimination due to automated creation of production schedule)
- lean manufacturing (supply and stores reduction, outgrowths reduction when delay in supply)
- increase of the flexibility (faster reaction to structure or constraint change)
- user-friendly solutions (balanced schedule for all employees)

Linear Programming (LP)

Let's state the problem of Linear Programming (LP).
In the general form of LP, we are given

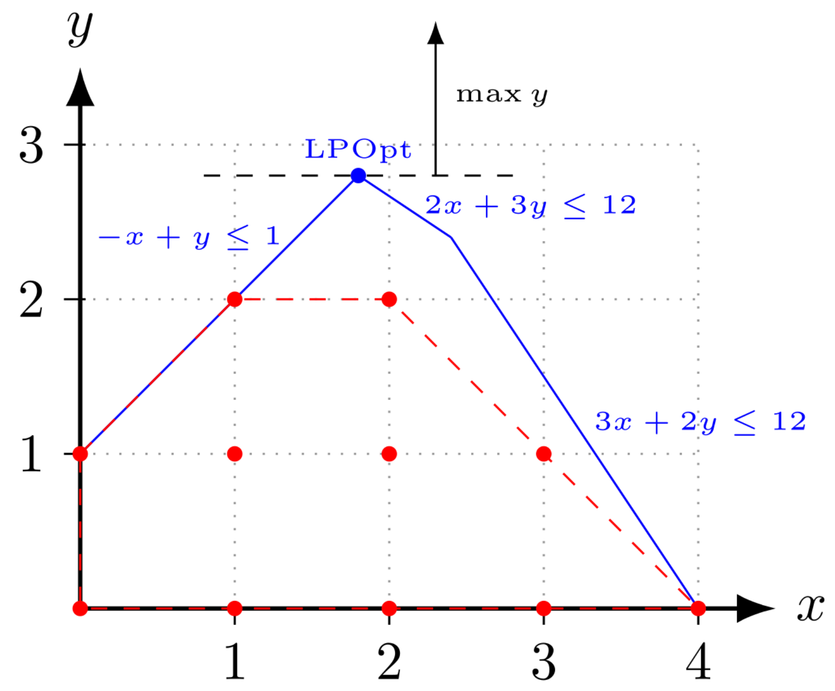
$$A \in \mathbb{R}^{m\times n}, \boldsymbol{b} \in \mathbb{R}^m, \boldsymbol{c} \in \mathbb{R}^n \text{ and } \boldsymbol{x} = [x_1, \dots, x_n] \in \mathbb{R}^n$$

And we have to minimize the objective function

$$\min \boldsymbol{c}^T \boldsymbol{x}$$

such that

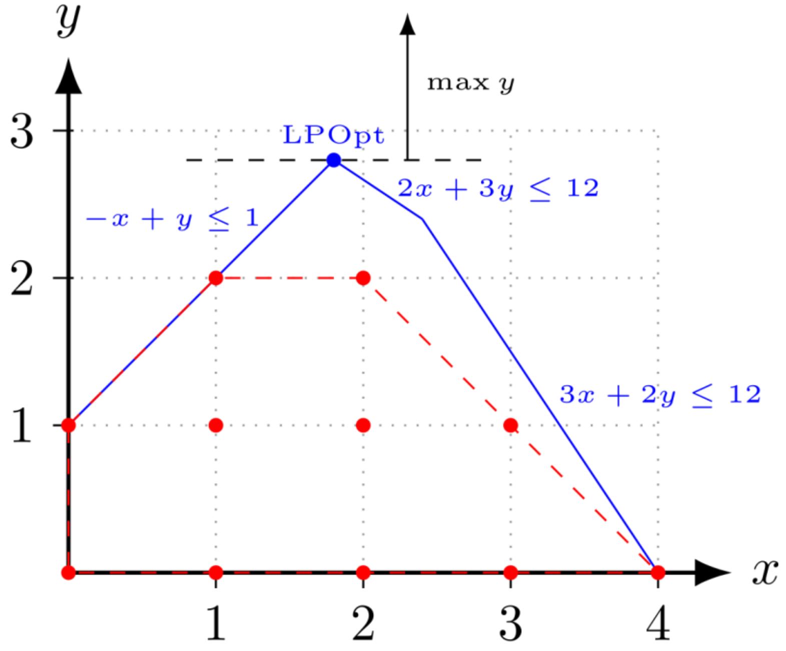
$$A\boldsymbol{x} \geq \boldsymbol{b}$$



Integer Linear Programming (ILP/MIP/MILP)

Very similar to Linear Programming – only difference is that some or all of the x values can now be integers.

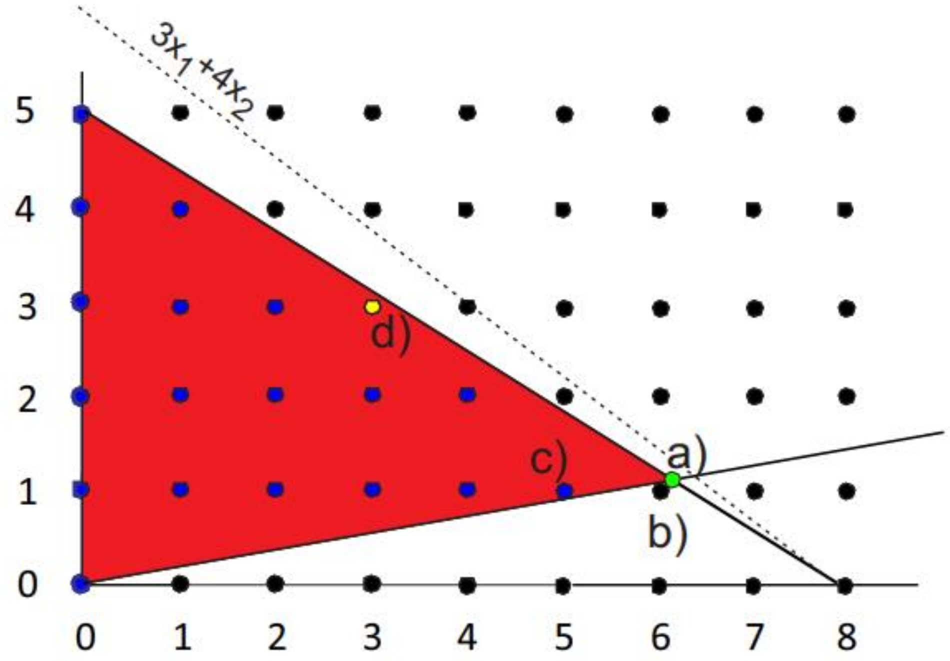
- It is not possible to just solve LP and round solution – we can arrive at suboptimal solution or not feasible one.
- ILP is not a convex set
- NP hard



ILP vs LP caveats

$$\max z = 3x_1 + 4x_2$$
$$s.t. 5x_1 + 8x_2 \leq 40$$
$$x_1 - 5x_3 \leq 0$$

a) LP solution z = 23.03 for
   x1 = 6.06, x2 = 1.21
b) Rounding leads to infeasible solution
   x1 = 6, x2 = 1
c) Nearest feasible integer is not optimal
   z = 19 for x1 = 5, x2 = 1
d) Optimal solution is z = 21 for x1 = 3, x2 = 3



Knapsack problem

Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.

$$\max \sum_i x_i \cdot v_i$$

$$s.t. \sum_i x_i w_i \leq W$$

$$x_i \in \{0,1\}, v_i \in R, w_i \in R$$

Shortest path in directed graph

Find shortest path from s to t or decide that t is unreachable from s.

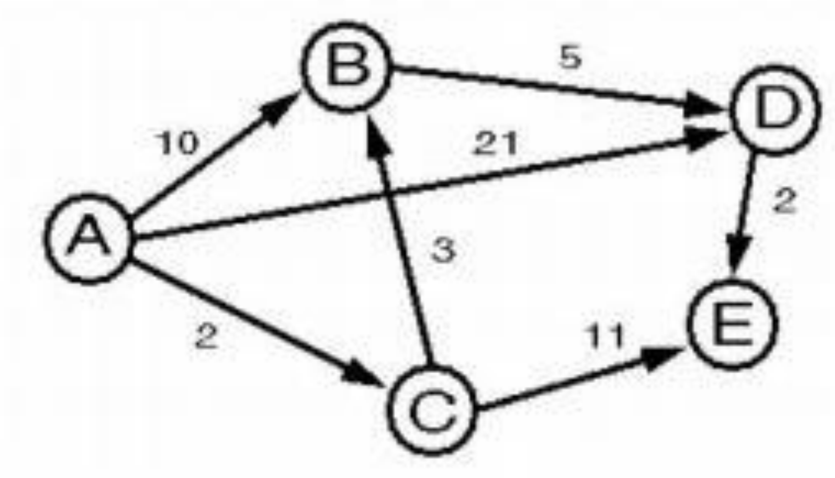
$$\max l_t$$

subject to

$l_s = 0$

$l_j \leq l_i + c_{i,j} \; for \; i \in 1 \ldots n, j \in 1 \ldots n$

$l_i \in R^{+,0}, c_{i,j} \in R^{+,0}, n \in Z^{+,0}$



Traveling Salesman Problem

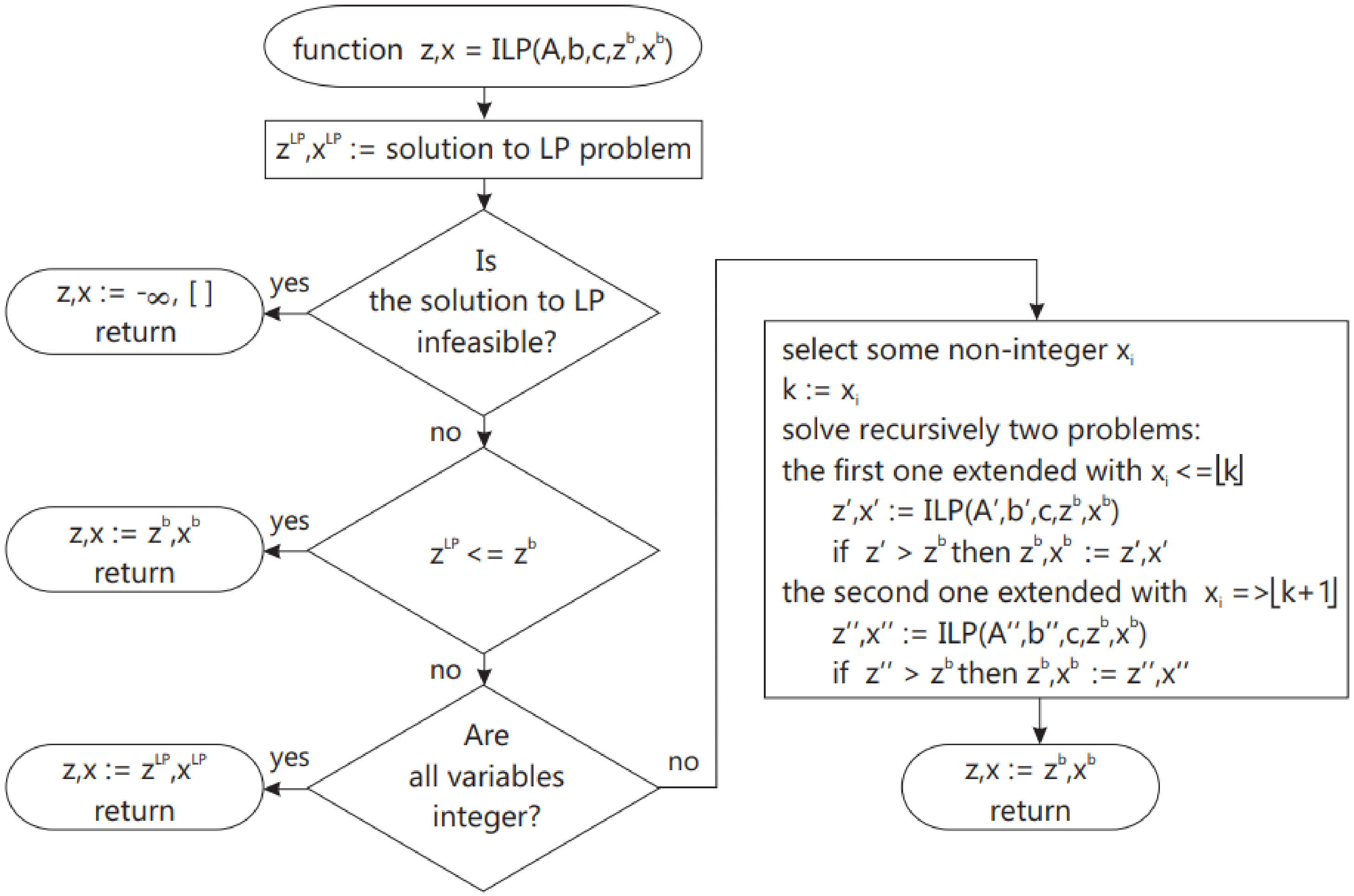
With complete directed graph and distance matrix, determine the order of visits of individual nodes such the overall price is minimized.

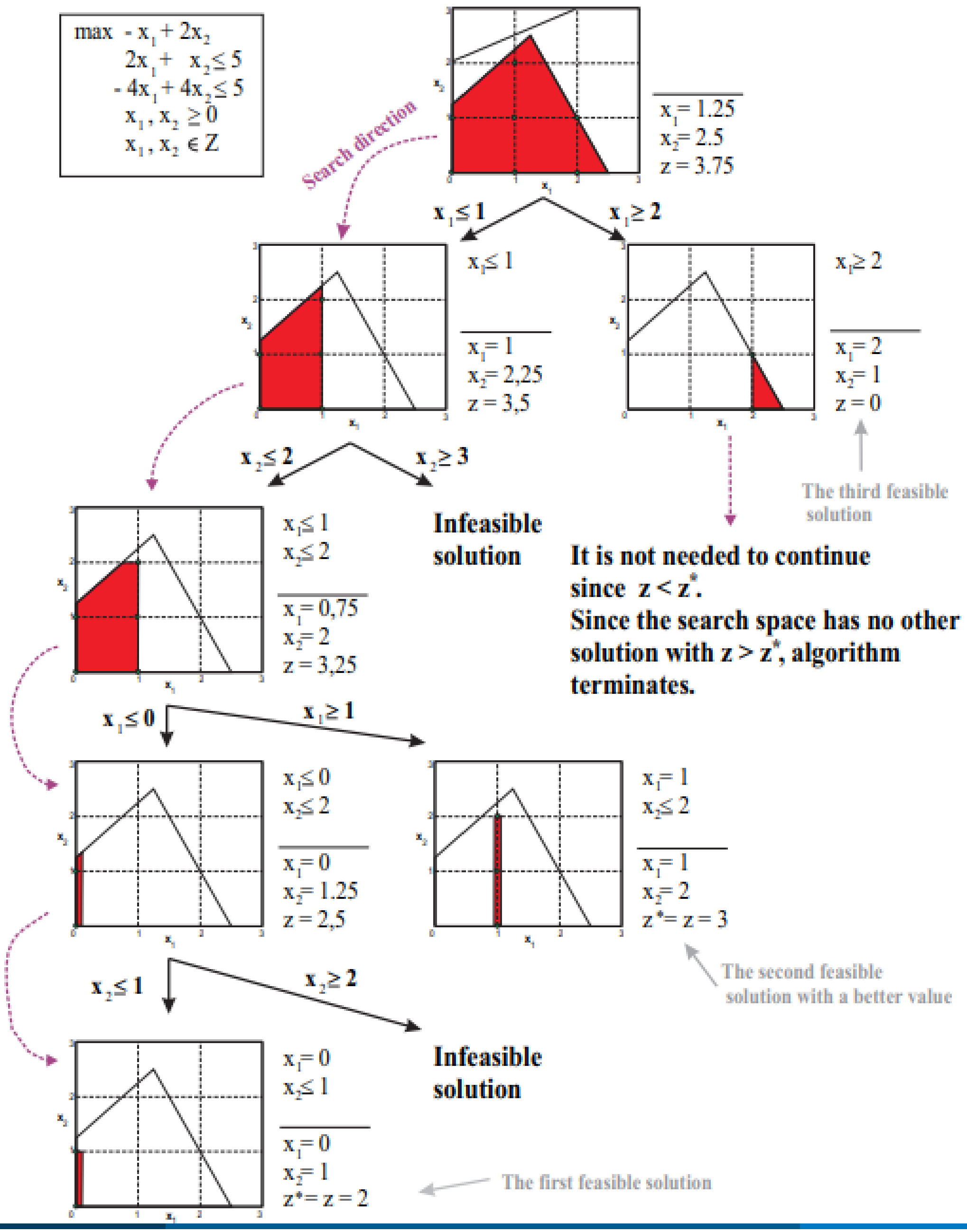
$$
\begin{array}{llll}
\text{min} & \sum_{i\in 1..n}\sum_{j\in 1..n} & c_{i,j} * x_{i,j} & \\
\text{subject to:} & & x_{i,i} = 0 \quad i \in 1..n & \text{avoid self-loop} \\
& \sum_{i\in 1..n} x_{i,j} = 1 & j \in 1..n & \text{enter once} \\
& \sum_{j\in 1..n} x_{i,j} = 1 & i \in 1..n & \text{leave once} \\
& s_i + c_{i,j} - (1 - x_{i,j}) * M \leq s_j & i \in 1..n, j \in 2..n & \text{cycle indivisibility} \\
\text{parameters:} & M \in \mathbb{Z}_0^+,\ n \in \mathbb{Z}_0^+,\ c_{i\in 1..n, j\in 1..n} \in \mathbb{Q}^+ & & \\
\text{variables:} & x_{i\in 1..n, j\in 1..n} \in \{0,1\},\ s_{i\in 1..n} \in \mathbb{R}_0^+ & &
\end{array}
$$

Algorithm for solving ILP - Branch & Bound

- The method is based on **splitting** the solution space into disjoint sets.
- It starts by relaxing on the integrality of the variables and **solves the LP problem**.
- If all variables xi are integers, the computation ends. Otherwise one variable $xi \notin Z$ is chosen and its value is assigned to k.
- Then the solution space is **divided into two sets** - in the first one we consider $xi \leq \lfloor k \rfloor$ and in the second one $xi \geq \lfloor k \rfloor + 1$.
- The algorithm **recursively repeats** computation for the both new sets till feasible integer solution is found.

Algorithm for solving ILP - Branch & Bound




max - x1 + 2x2
2x1 + x2 ≤ 5
- 4x1 + 4x2 ≤ 5
x1 , x2 ≥ 0
x1 , x2 ∈ Z
Search direction
x1= 1.25
x2= 2.5
z = 3.75
x1≤ 1
x1≥ 2
x1≤ 1
x1= 1
x2= 2,25
z = 3,5
x1≥ 2
x1= 2
x2= 1
z = 0
The third feasible solution
x2≤ 2
x2≥ 3
Infeasible solution
It is not needed to continue since z < z*.
Since the search space has no other solution with z > z*, algorithm terminates.
x1≤ 1
x2≤ 2
x1= 0,75
x2= 2
z = 3,25
x1≤ 0
x1≥ 1
x1≤ 0
x2≤ 2
x1= 0
x2= 1.25
z = 2,5
x1= 1
x2≤ 2
x1= 1
x2= 2
z*= z = 3
The second feasible solution with a better value
x2≤ 1
x2≥ 2
x1= 0
x2≤ 1
x1= 0
x2= 1
z*= z = 2
Infeasible solution
The first feasible solution

Example for formulation

	T-shirt	Shirt	trousers	Capacity
Labor	3	2	6	150
Material	4	3	4	160
income	6	4	7	

- Goal: maximize the income (i.e. total income minus total expenses)
- Constraints: labor capacity is 150 person-hours material capacity is 160 meters

- **TODO: Formulate the problem with ILP**

Example for formulation

	T-shirt	Shirt	trousers	Capacity
Labor	3	2	6	150
Material	4	3	4	160
income	6	4	7	

Part 2: Fixed cost has to be covered to rent the machine product. If at least 1 product is made, the rent has to be paid (will affect the objective function)

	T-shirt	Shirt	trousers
Machine Rent	200	150	100

- **TODO: Formulate the problem with ILP**

Scheduling Problem – Simplified

You have set of engagements - E, weeks - W and people for a given level – P. Assign people to individual weeks of engagements based on their availibility, experience and preference of the engagement managers.

Optimization hands-on

MIP

Deloitte.