Chapter 1: Introduction to Optimization

CS-616: Optimization Algorithms

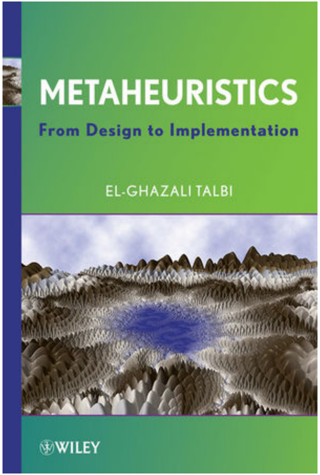
Dr. Mahdi A. Khemakhem & Dr. Esraa M. Eldesouky

Department of Computer Science
College of Computer Engineering and Science
Prince Sattam bin Abdulaziz University

AY: 2025/2026

Materials

- **Textbook: "Metaheuristics: From Design to Implementation"** by *El-Ghazali Talbi*
- ➥ Read Chapter 1 (Sections 1.1, 1.2, and 1.3) for this chapter's material.

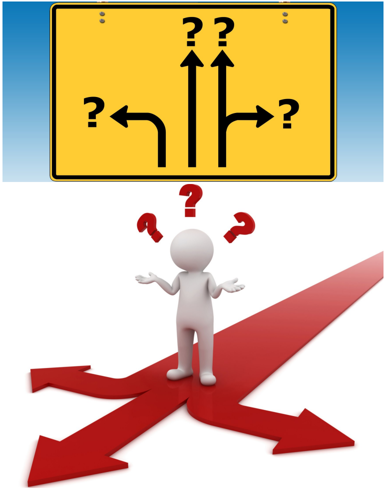


Outline

Outline

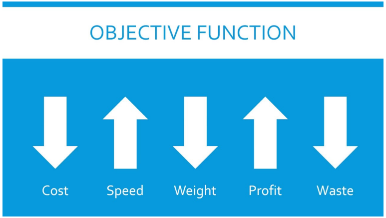
What is an Optimization Problem?

- An optimization problem (OP) is a computational problem in which the object is to find the best of all possible solutions.
- More formally, an OP is to find a solution in the feasible region which has the minimum (or maximum) value of the objective function.



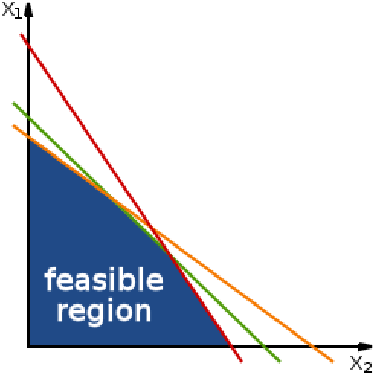
What is an Objective Function?

- An objective function (OF) is a function associated with an OP which determines how good a solution is.
- An OF should be defined using decision variables.



What is a Feasible Region?

- A feasible region is the set of all possible solutions of an OP.
- In mathematical optimization, a feasible region, feasible set, search space, or solution space is the set of all possible points (sets of values of the choice variables) of an OP that satisfy the problem's constraints.



What are Constraints?

- **Constraints are logical conditions that a solution to an OP must satisfy.**
- **Constraints reflect real-world limits on *production capacity*, *market demand*, *available funds*, and so on.**
- **Constraints should be defined using decision variables.**

What is a Decision Variable?

- A **decision variable** is an **unknown-value variable** associated to an **OP**.
- **Decision variable** has a **domain**, which is a compact representation of the **set of all possible values for the variable**.
- **Decision variable types** are references to objects whose exact nature depends on the underlying **optimizer of a model**.
- **Decision variables** are a set of quantities to be determined for solving the **OP**; i.e., the problem is solved when the best values of the **decision variables** have been identified.
- They are called **decision variables** because **the problem is to decide what value each variable should take**.

Optimization Problem Requirements

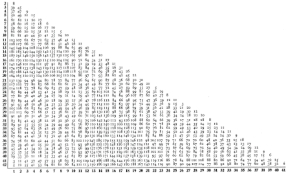
An optimization problem definition requires a prior definitions of:

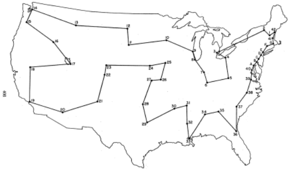
1. A set of decision variable(s).
2. An objective function using defined decision variable(s) to precise how good solution is.
3. A set of constraints(s) using defined decision variable(s) to determine the feasible region containing all OP's possible solutions.

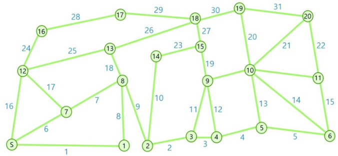
Solving an optimization problem consists to finding the better solution(s) in the feasible region according to the defined objective function and the constraint(s) set limitations.

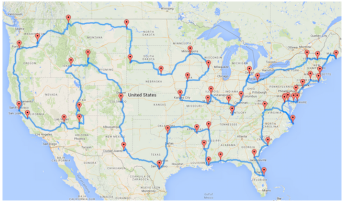
Traveling Salesman Problem (1/2)

The most popular combinatorial optimization problem is the Traveling Salesman Problem (TSP). It can be defined as follows: given n cities and a distance matrix $d_{n,n}$, where each element d_{ij} represents the distance between the cities i and j, find a tour that minimizes the total distance. A tour visits each city exactly once (Hamiltonian cycle).





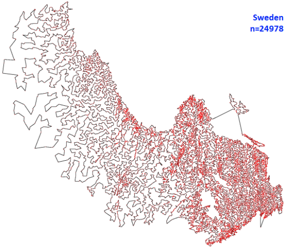




Traveling Salesman Problem (2/2)

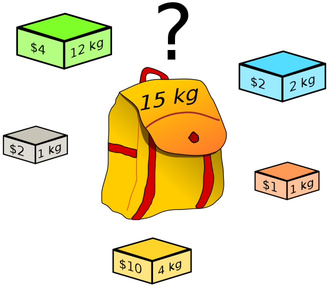
- **The size of the search space is $n!$.**
- **Table below shows the combinatorial explosion of the number of solutions regarding the number of cities.**
- **Unfortunately, enumerating exhaustively all possible solutions is impractical for moderate and large instances.**

Number of Cities n	Size of the Search Space
5	120
10	3,628,800
75	2.5×10^{109}



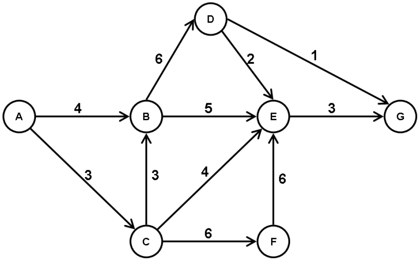
Knapsack Problem

The **Knapsack Problem (KP)** can be defined as follows. Given a set of N objects. Each object i has a specified weight w_i and a specified profit p_i. Given a capacity C, which is the maximum total weight of the knapsack.
The KP consists of finding a subset of the objects whose total weight is at most equal to the knapsack capacity and whose total profit is maximized.



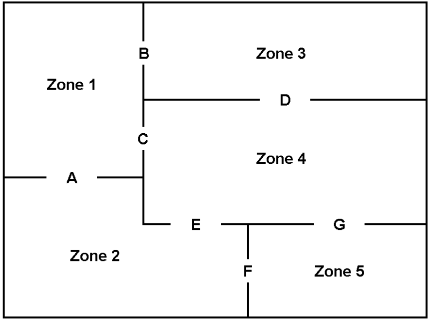
Computer Network

Given an IP network where we consider only one direction of transmission indicated by arrows. The number on each link indicates the time taken by an IP packet (in seconds) to be transferred over that link.
What is the fastest path to forward an IP packet from router A to router G?



Mobile Phone Coverage

A mobile phone company is newly installed in a country whose plan is presented in the figure beside. The transmitting antennas can be placed in sites A,B,...,G located on the common borders of the different areas of the country. An antenna placed on a given site can cover the two zones whose common borders shelter this site.
The company's goal is to ensure, at the lowest cost, the coverage of each zone with at least one antenna, while covering zone 4 with at least two antennas.



Scheduling

A hospital service director is responsible for organizing the nurses' schedules. A working day is divided into 12 time slots of two hours each. Staffing needs vary from one time slot to another. For example, few nurses are needed during the night, but the staff must be reinforced in the morning to ensure patient care. The table beside gives the staffing needs for each of the time slots.
What is the minimum number of nurses needed to cover time slot requirements? It's important to know that each nurse works 8 hours a day and has a 2-hours break after 4 hours of work.

#	Time slot	Nurses min. numb.
1	06 am - 08 am	35
2	08 am - 10 am	40
3	10 am - 12 pm	40
4	12 pm - 02 pm	35
5	02 pm - 04 pm	30
6	04 pm - 06 pm	30
7	06 pm - 08 pm	35
8	08 pm - 10 pm	30
9	10 pm - 12 am	20
10	12 am - 02 am	15
11	02 am - 04 am	15
12	04 am - 06 am	15

Military Operations

During a military operation, an officer must choose the members of a patrol of minimum size among the different soldiers who are under his command and whose skills are described in the table below:

Soldier	Radio	Camouflage	Sniper	Physical endurance	Non-commissioned officer
A	✓	✓	6	5	✓
B		✓	8	7	
C	✓	✓	7	7	
D	✓		4	9	
E		✓	7	9	✓
F	✓		8	6	

The officer must respect the following conditions:

- One and only one non-commissioned officer is needed.
- At least 1 Radio operator and at least 1 Camouflage specialist are required.
- At least two radio operators will not be part of the patrol in order to remain available for other missions.
- Patrol members must have an average sniping rate of at least 7 and an average physical endurance rate of at least 7.5.
- If solder A is part of the patrol, then neither B nor C should be part of it.

Part-time Student-job Working Hours

Mohamed is a college student who works two jobs on campus.
- He must work for at least 5 hours per week at the library and 2 hours per week as a tutor, but he is not allowed to work more than 20 hours per week total.
- Mohamed gets \$7 per hour at the library and \$10 per hour at tutoring and he want to make at last \$170 during the current week.
- He prefers working at the library though, so he wants to have at least as many library hours as tutoring hours.

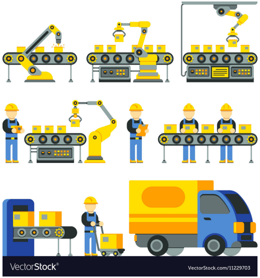
What is the minimum number of hours he can work at each job this week to meet his goals and preferences?

Factory Production

A factory produces two types of building materials.

- **Sale price: Product A** = \$140/ton, **Product B**=\$160/ton.
- During production, a **special ingredient X** is added.
- Each ton of product A or B produced requires 2 , 4 cubic meters of ingredient X, respectively. Only 28 cubic meters of ingredient X are available in production per week.
- The worker who produces the materials can work up to 50 hours/week. The machine producing the materials is able to construct a ton of product at a time, while the process lasts 5 hours.
- The finished products are stored in bins: 8 tons of product A and 6 tons of product B.

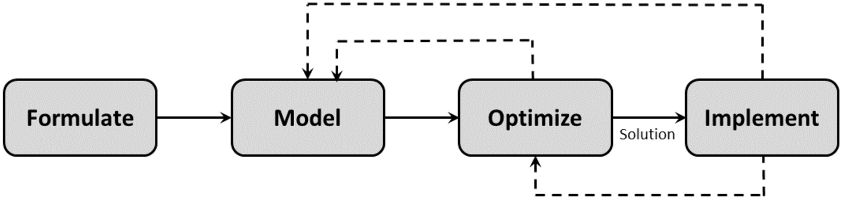
The purpose of **solving the problem** is to **determine the quantity of product A and of product B** that can be produced **every week** in order to achieve **maximization of the total weekly profit**.



Outline

Decision Making Steps

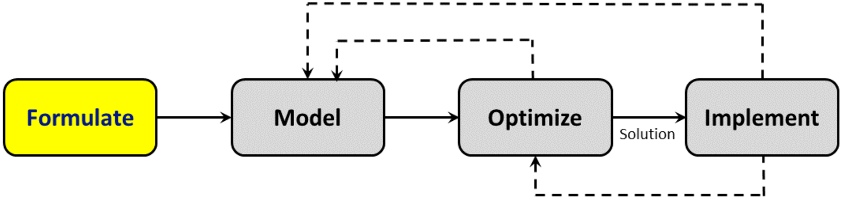
- As scientists, engineers, and managers, we always have to take decisions.
- Decision making is everywhere.
- As the world becomes more and more complex and competitive, decision making must be tackled in a rational and optimal way.
- Decision making consists in the following steps:



The classical process in decision making: formulate, model, solve, and implement.
In practice, this process may be iterated to improve the optimization model or algorithm until an acceptable solution is found.

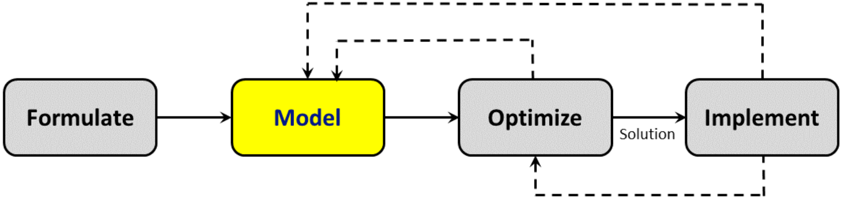
Formulate the Problem

1. The first step consists to formulate the problem:
   - A decision problem is identified.
   - Then, an initial statement of the problem is made.
   - This formulation may be imprecise.
   - The internal and external factors, and the objective(s) of the problem are outlined.
   - Many decision makers may be involved in formulating the problem.



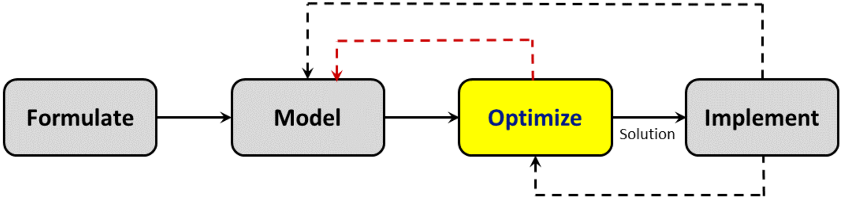
Model the Problem

2. **The second step consists to model the problem:**
   - **An abstract mathematical model is built for the problem.**
   - **The modeler can be inspired by similar models in the literature.**
   - **Usually, models we are solving are simplifications of the reality. They involve approximations and sometimes they skip processes that are complex to represent in a mathematical model.**



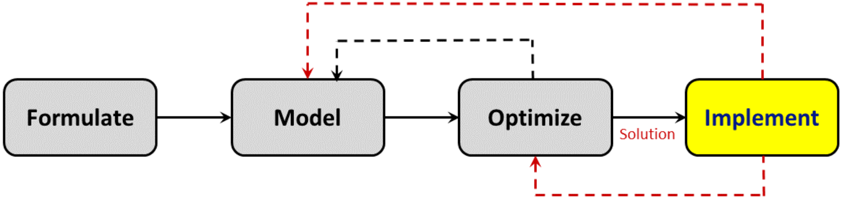
Optimize the Problem

3. The third step consists to optimize the problem:
   - The solving procedure generates a "good" solution for the problem.
   - The solution may be optimal or sub-optimal.
   - Let us notice that we are finding a solution for an abstract model of the problem and not for the originally formulated real-life problem.
   - The algorithm designer can reuse state-of-the-art algorithms on similar problems or integrate the knowledge of this specific application into the algorithm.



Implement the Solution

4. The fourth step consists to implement the solution:
   - The obtained solution is tested practically by the decision maker and is implemented if it is "acceptable".
   - Some practical knowledge may be introduced in the solution to be implemented.
   - If the solution is unacceptable, the model and/or the optimization algorithm has to be improved and the decision making process is repeated.



Classical Optimization Models

- As mentioned, optimization problems are encountered in many domains: science, engineering, management, and business.
- An optimization problem may be defined by the couple (S, f), where S represents the set of feasible solutions[1], and $f : S \rightarrow \mathbb{R}$ the objective function[2] to optimize.
- The objective function f assigns to every solution $s \in S$ of the search space a real number indicating its value.
- The objective function f allows to define a total order relation between any pair of solutions in the search space.

[1] Sometimes named: Feasible Region, or Feasible Set, or Search Space, or Solution Space
[2] Sometimes named: Cost, Utility, or Fitness Function.

Global Optimum

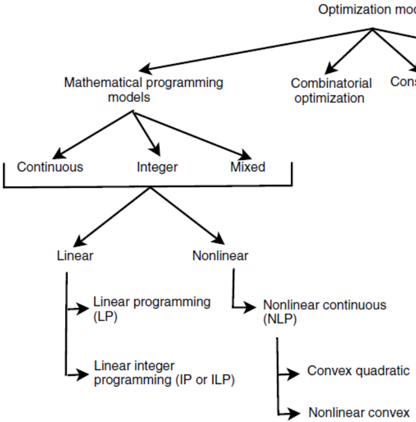
A solution $s^* \in S$ is a global optimum if it has a better objective function[3] than all solutions of the search space, that is, $\forall s \in S, f(s^*) \leq f(s)$.

- **The main goal in solving an optimization problem is to find a global optimal solution s^*.**
- **Many global optimal solutions may exist for a given problem.**
- **Hence, to get more alternatives, the problem may also be defined as finding all global optimal solutions.**

[3] We suppose without loss of generality a minimization problem. Maximizing an objective function f is equivalent to minimizing $-f$.

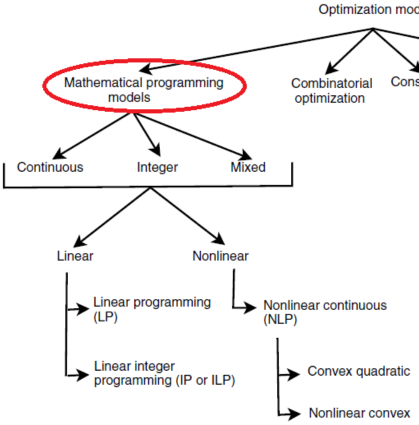
Different Families of Optimization Models

- Different families of optimization models are used in practice to formulate and solve decision-making problems.



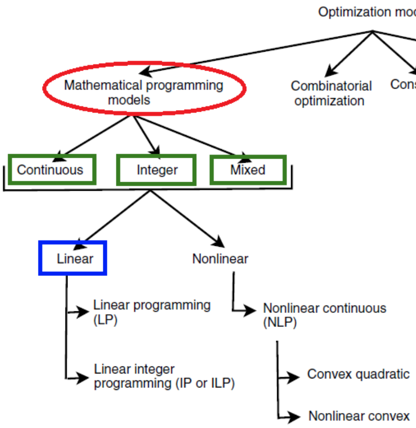
Different Families of Optimization Models

- Different families of optimization models are used in practice to formulate and solve decision-making problems.
- The most successful models are based on mathematical programming.



Different Families of Optimization Models

- Different families of optimization models are used in practice to formulate and solve decision-making problems.
- The most successful models are based on mathematical programming.
- The most common mathematical programming models are linear with continuous, or integer, or mixed decision variables.



Outline

LP Model Definition

- A commonly used model in mathematical programming is linear programming (LP).
- Linear programming (LP) model or Linear Optimisation may be defined as the problem of maximizing or minimizing a linear function that is subjected to linear constraints.
- The constraints may be equalities or inequalities.
- Linear programming problems are an important class of optimisation problems, that helps to find the feasible region and optimise the solution in order to have the highest or lowest value of the function.
- The basic components of the LP are as follows:
  - Data
  - Decision Variables
  - Constraints
  - Objective Function(s)

LP Model Characteristics (1/2)

The six characteristics of the linear programming model are as follows:

1. **Decision Variables**
   - **Unknown-value variables in a specified domain.**
   - **They are expected to be estimated as an output of the LP model solution.**
   - **For any problem, the first step is to identify the decision variables and their domains.**
2. **Objective Function**
   - **Determines the objective of the LP problem that generally aims to either maximize and/or minimize some quantitative value.**
   - **Mathematical function that evaluates the amount by which each decision variable would contribute to the net present value.**
3. **Constraints**
   - **The problem restrictions or limitations that deny to reach an infinite profit or a zero cost.**
   - **They should be expressed in the mathematical form (inequalities and/or equalities).**

LP Model Characteristics (2/2)

4. **Linearity**
   - The relationship between two or more variables of the objective function and the constraints must be linear. It means that the degree of the variable is one.
   - Products of decision variables are denied.
5. **Finiteness**
   - There should be finite and infinite input and output numbers.
   - In case, if the objective function or a constraint has infinite factors, the optimal solution is not feasible.
6. **Non-negativity**
   - Each decision variable value should be positive or zero.
   - It should not be a negative value.

LP Expanded Form

A Linear Programming (LP) model, with n decision variables and m constraints, can be formulated as follows:

$$\textbf{Min (or Max) } Z = \sum_{i=1}^{n} c_i x_i$$

subject to **(s.t.)**:

$$\sum_{i=1}^{n} a_{1i} x_i \leq (\text{or } \geq \text{ or } =) b_1$$

$$\sum_{i=1}^{n} a_{2i} x_i \leq (\text{or } \geq \text{ or } =) b_2$$

$$\vdots$$

$$\sum_{i=1}^{n} a_{ji} x_i \leq (\text{or } \geq \text{ or } =) b_j$$

$$\vdots$$

$$\sum_{i=1}^{n} a_{mi} x_i \leq (\text{or } \geq \text{ or } =) b_m$$

$$\forall i \in \{1, 2, \ldots, n\}, x_i \in \mathbb{R}^+ (\text{or } \mathbb{N} \text{ or } \{0, 1\})$$

LP Compact/Matrix Form

A Linear Programming (LP) model, with n decision variables and m constraints, can be compactly formulated as follows:

$$\begin{aligned} &\textbf{Min (or Max) } Z = c.x \\ &\text{subject to } \textbf{(s.t.)}: \\ &\qquad A.x \begin{array}{c} \leq \\ \geq \\ = \end{array} b \\ &\qquad \forall x_i \in x, x_i \in \mathbb{R}^+ (\text{or } \mathbb{N} \text{ or } \{0,1\}) \end{aligned}$$

With:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \; c = [c_1, c_2, \ldots, c_n], \; b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, \; A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix}$$

where x is a vector of n continuous (or integer, or mixed) decision variables, and c and b (resp. A) are constant vectors (resp. matrix) of coefficients.

First Simple LP Model (1/3)

A given company synthesizes two products $Prod_1$ and $Prod_2$ based on two kinds of raw materials M_1 and M_2. The below table presents the daily available raw materials for M_1 and M_2, and for each product $Prod_i$ the used amount of raw materials and the profit. The objective consists in finding the most profitable amounts of product mix $Prod_1$ and $Prod_2$.

	Usage for $Prod_1$	Usage for $Prod_2$	Material Availability
M_1	6	4	**24**
M_2	1	2	**6**
Profit	**\$5**	**\$4**	

First Simple LP Model (2/3)

- **Decision variables:** Two unknown positive real variables X_1 and X_2 that determine, respectively, the amounts of $Prod_1$ and $Prod_2$. A negative output of a product is not possible, therefore variables X_1 and X_2 must be positive. $X_1, X_2 \in \mathbb{R}^+$
- **Objective function:** Maximize the total profit obtained by X_1 of $Prod_1$ and X_2 of $Prod_2$. $MaxZ = 5X_1 + 4X_2$
- **Constraints:**
  1. **Material M_1 availability:** Each unit of product $Prod_1$ produced requires 6 units of material M_1. Each unit of product $Prod_2$ produced requires 4 units of material M_1. Only 24 units of material M_1 are available. $6X_1 + 4X_2 \leq 24$
  2. **Material M_2 availability:** Each unit of product $Prod_1$ produced requires 1 unit of material M_2. Each unit of product $Prod_2$ produced requires 2 units of material M_2. Only 6 units of material M_2 are available. $1X_1 + 2X_2 \leq 6$
- **Feasible region:** The set of all solutions satisfying all constraints simultaneously.
- **Optimal Solution:** further than satisfying all constraints simultaneously, it should provide the maximum value Z of the objective function.

First Simple LP Model (3/3)

The model of this problem may be formulated as an LP mathematical program:

$$MaxZ = 5X_1 + 4X_2$$

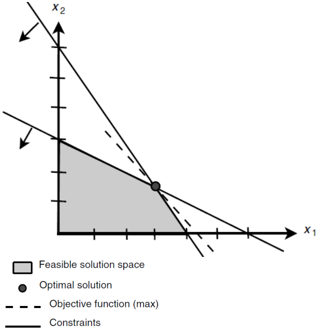
(s.t.):

$$6X_1 + 4X_2 \leq 24$$
$$1X_1 + 2X_2 \leq 6$$
$$X_1, X_2 \in \mathbb{R}^+$$

$$x = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}, c = [5, 4],$$

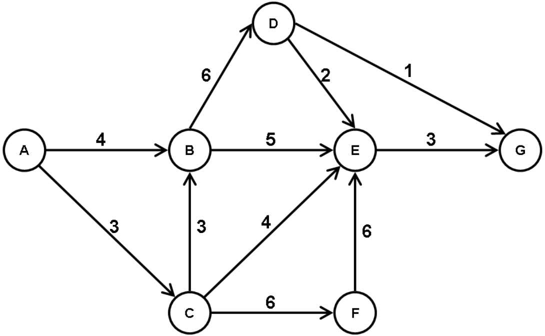
$$b = \begin{bmatrix} 24 \\ 6 \end{bmatrix}, A = \begin{bmatrix} 6 & 4 \\ 1 & 2 \end{bmatrix}$$

- The figure on the right illustrates the graphical interpretation of the model.
- Each constraint can be represented by a line.
- The objective function is an infinity of parallel lines.
- The optimum solution will always lie at an extreme point.
- The optimal solution is $(X_1 = 3, X_2 = 1.5)$ with a maximum profit of $Z = 21$.



Computer Network (1/2)

Given an IP network where we consider only one direction of transmission indicated by arrows. The number on each link indicates the time taken by an IP packet (in seconds) to be transferred over that link. What is the fastest path to forward an IP packet from router A to router G?

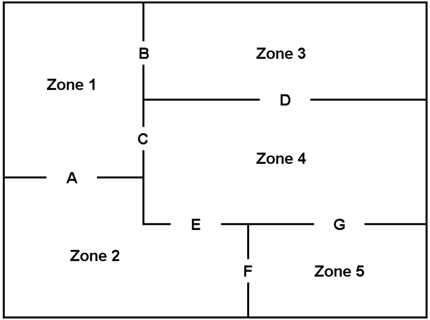


Computer Network (2/2)

- **Decision variables:** Unknown binary variables that determine for each link if it is taken or not by the packet when forwording it from A to G.
  $X_{AB}, X_{AC}, X_{BD}, X_{BE}, X_{CB}, X_{CE}, X_{CF}, X_{DE}, X_{DG}, X_{EG}, X_{FE} \in \{0, 1\}$
- **Objective function:** Minimize the total time of the path from A to G according to the taken links and their required times. $MinZ = 4X_{AB} + 3X_{AC} + 6X_{BD} + 5X_{BE} + 3X_{CB} + 4X_{CE} + 6X_{CF} + 2X_{DE} + X_{DG} + 3X_{EG} + 6X_{FE}$
- **Constraints:**
  1. **Starting point:** IP packet must start from router A. $X_{AB} + X_{AC} = 1$
  2. **Ending point:** IP packet must arrive to router G. $X_{DG} + X_{EG} = 1$
  3. **Flow conservation:** If IP packet enters to B, it must leave it. $X_{AB} + X_{CB} = X_{BD} + X_{BE}$. Same thing for router C, D, E and F. $X_{AC} = X_{CB} + X_{CE} + X_{CF}$, $X_{BD} = X_{DE} + X_{DG}$, $X_{BE} + X_{CE} + X_{DE} + X_{FE} = X_{EG}$, $X_{CF} = X_{FE}$
- **Feasible region:** The set of all solutions satisfying all constraints simultaneously.
- **Optimal Solution:** further than satisfying all constraints simultaneously, it should provide the minimum value Z of the objective function.

Mobile Phone Coverage (1/2)

A mobile phone company is newly installed in a country whose plan is presented in the figure beside. The transmitting antennas can be placed in sites A,B,...,G located on the common borders of the different areas of the country. An antenna placed on a given site can cover the two zones whose common borders shelter this site.
The company's goal is to ensure, at the lowest cost, the coverage of each zone with at least one antenna, while covering zone 4 with at least two antennas.



Mobile Phone Coverage (2/2)

- **Decision variables:** Unknown binary variables that determine for each site (from A,B,...,G) if it will contain an antenna or note. $X_A, X_B, X_C, X_D, X_E, X_F, X_G \in \{0,1\}$
- **Objective function:** Minimize the total number of placed antennas in all sites. $MinZ = X_A + X_B + X_C + X_D + X_E + X_F + X_G$
- **Constraints:**
  1. **Zone 1 requirement:** Zone 1 must be covered by at least 1 antenna placed on one of its border. $X_A + X_B + X_C \geq 1$
  2. **Zone 2 requirement:** Same thing for Zone 2. $X_A + X_E + X_F \geq 1$
  3. **Zone 3 requirement:** Same thing for Zone 3. $X_B + X_D \geq 1$
  4. **Zone 5 requirement:** Same thing for Zone 5. $X_F + X_G \geq 1$
  5. **Zone 4 requirement:** Zone 4 must be covered by at least 2 antennas placed on one of its border. $X_C + X_D + X_E + X_G \geq 2$
- **Feasible region:** The set of all solutions satisfying all constraints simultaneously.
- **Optimal Solution:** further than satisfying all constraints simultaneously, it should provide the minimum value Z of the objective function.

Scheduling (1/2)

A hospital service director is responsible for organizing the nurses' schedules. A working day is divided into 12 time slots of two hours each. Staffing needs vary from one time slot to another. For example, few nurses are needed during the night, but the staff must be reinforced in the morning to ensure patient care. The table beside gives the staffing needs for each of the time slots.

What is the minimum number of nurses needed to cover time slot requirements? It's important to know that each nurse works 8 hours a day and has a 2-hours break after 4 hours of work.

#	Time slot	Nurses min. numb.
1	06 am - 08 am	35
2	08 am - 10 am	40
3	10 am - 12 pm	40
4	12 pm - 02 pm	35
5	02 pm - 04 pm	30
6	04 pm - 06 pm	30
7	06 pm - 08 pm	35
8	08 pm - 10 pm	30
9	10 pm - 12 am	20
10	12 am - 02 am	15
11	02 am - 04 am	15
12	04 am - 06 am	15

Scheduling (2/2)

- **Decision variables:** Unknown variables that determine the number of nurses starting their working shift at the beginning of each time slot.
  $X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9, X_{10}, X_{11}, X_{12} \in \mathbb{N}$
- **Objective function:** Minimize the total number of nurses.
  $MinZ = X_1 + X_2 + X_3 + X_4 + X_5 + X_6 + X_7 + X_8 + X_9 + X_{10} + X_{11} + X_{12}$
- **Constraints:**
  1. **Time slot #1 requirement:** Consider nurses that start their shift at 6am, at 4am, at 12 am, and 10pm. Nurses that start their shift at 2am are not considered because their are in break during period #1. $X_1 + X_{12} + X_{10} + X_9 \geq 35$
  2. **Remain time slots #2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and 12 requirements:**
     $X_2 + X_1 + X_{11} + X_{10} \geq 40$, $X_3 + X_2 + X_{12} + X_{11} \geq 40$, $X_4 + X_3 + X_1 + X_{12} \geq 35$, $X_5 + X_4 + X_2 + X_1 \geq 30$, $X_6 + X_5 + X_3 + X_2 \geq 30$, $X_7 + X_6 + X_4 + X_3 \geq 35$, $X_8 + X_7 + X_5 + X_4 \geq 30$, $X_9 + X_8 + X_6 + X_5 \geq 20$, $X_{10} + X_9 + X_7 + X_6 \geq 15$, $X_{11} + X_{10} + X_8 + X_7 \geq 15$, $X_{12} + X_{11} + X_9 + X_8 \geq 15$
- **Feasible region:** The set of all solutions satisfying all constraints simultaneously.
- **Optimal Solution:** further than satisfying all constraints simultaneously, it should provide the minimum value Z of the objective function.

Military Operations (1/2)

During a military operation, an officer must choose the members of a patrol of minimum size among the different soldiers who are under his command and whose skills are described in the table below:

Soldier	Radio	Camouflage	Sniper	Physical endurance	Non-commissioned officer
A	✓	✓	6	5	✓
B		✓	8	7	
C	✓	✓	7	7	
D	✓		4	9	
E		✓	7	9	✓
F	✓		8	6	

The officer must respect the following conditions:

- One and only one non-commissioned officer is needed.
- At least 1 Radio operator and at least 1 Camouflage specialist are required.
- At least two radio operators will not be part of the patrol in order to remain available for other missions.
- Patrol members must have an average sniping rate of at least 7 and an average physical endurance rate of at least 7.5.
- If solder A is part of the patrol, then neither B nor C should be part of it.

Military Operations (2/2)

- **Decision variables:** Unknown binary variables that determine for each soldier (from A,B,...,F) if he will be a part of the patrol or note. $X_A, X_B, X_C, X_D, X_E, X_F \in \{0, 1\}$
- **Objective function:** Minimize the total number of soldier in the patrol. $MinZ = X_A + X_B + X_C + X_D + X_E + X_F$
- **Constraints:**
  1. **NCO requirement:** One and only one NCO is needed. $X_A + X_E = 1$
  2. **Radio requirement:** At least 1 Radio operator. $X_A + X_C + X_D + X_F \geq 1$
  3. **Radio requirement:** At least two radio operators will not be part of the patrol $\rightarrow$ At most two radio operators will be a part of the patrol. $X_A + X_C + X_D + X_F \leq 2$
  4. **Camouflage requirement:** At least 1 Camouflage specialist. $X_A + X_B + X_C + X_E \geq 1$
  5. **Sniping rate requirement:** At least 7 of average sniping rate. $\frac{6X_A+8X_B+7X_C+4X_D+7X_E+8X_F}{X_A+X_B+X_C+X_D+X_E+X_F} \geq 7$
  6. **Physical endurance rate requirement:** At least 7.5 of average physical endurance rate. $\frac{5X_A+7X_B+7X_C+9X_D+9X_E+6X_F}{X_A+X_B+X_C+X_D+X_E+X_F} \geq 7.5$
  7. **Co-existence requirement :** If solder A is part of the patrol, then neither B nor C should be part of it. $2X_A + X_B + X_C \leq 2$

Part-time Student-job Working Hours (1/2)

Mohamed is a college student who works two jobs on campus.

- He must work for at least 5 hours per week at the library and 2 hours per week as a tutor, but he is not allowed to work more than 20 hours per week total.
- Mohamed gets $7 per hour at the library and $10 per hour at tutoring and he want to make at last $170 during the current week.
- He prefers working at the library though, so he wants to have at least as many library hours as tutoring hours.

What is the minimum number of hours he can work at each job this week to meet his goals and preferences?

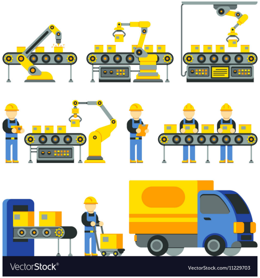
Part-time Student-job Working Hours (2/2)

- **Decision variables:** Unknown variables that determine the number of working hours at the library and the number of working hours at tutoring. $X_L, X_T \in \mathbb{N}$: number of working hours at library and at tutoring, respectively.
- **Objective function:** Minimize the total number of working hours for this week. $MinZ = X_L + X_T$
- **Constraints:**
  1. **Library requirement:** Mohamed must work at least 5 hours per week at library. $X_L \geq 5$
  2. **Tutoring requirement:** Mohamed must work at least 2 hours per week at tutoring. $X_T \geq 2$
  3. **Maximum number of working hours :** Mohamed is not allowed to work more than 20 hours per week. $X_L + X_T \leq 20$
  4. **Budget goal:** Mohamed need to make at least \$170 during the current week, while he gets \$7 per hour at the library and \$10 per hour at tutoring. $7X_L + 10X_T \geq 170$
  5. **Preferences:** Mohamed wants to have at least as many library hours as tutoring hours. $X_L \geq X_T$
- **Feasible region:** The set of all solutions satisfying all constraints simultaneously.
- **Optimal Solution:** further than satisfying all constraints simultaneously, it should provide the minimum value Z of the objective function.

Factory Production (1/2)

A factory produces two types of building materials.
- **Sale price: Product A** = \$140/ton, **Product B**=\$160/ton.
- During production, a **special ingredient X** is added. - Each ton of product A or B produced requires 2 , 4 cubic meters of ingredient X, respectively. Only 28 cubic meters of ingredient X are available in production per week.
- The worker who produces the materials can work up to 50 hours/week. The machine producing the materials is able to construct a ton of product at a time, while the process lasts 5 hours.
- The finished products are stored in bins: 8 tons of product A and 6 tons of product B.

The purpose of **solving the problem** is to **determine the quantity of product A and of product B** that can be produced **every week** in order to achieve **maximization of the total weekly profit**.



Factory Production (2/2)

- **Decision variables:** Unknown variables that determine the quantity of product A and of product B to produce each week. $X_A, X_B \in \mathbb{R}^+$: number of weekly produced tons of product A and B, respectively.
- **Objective function:** Maximize the total profit per week. $MaxZ = 140X_A + 160X_B$
- **Constraints:**
  1. **Ingredient availability:** Each ton of product A produced requires 2 cubic meters of ingredient X. Each ton of product B produced requires 4 cubic meters of ingredient X. Only 28 cubic meters of ingredient X are available per week. $2X_A + 4X_B \leq 28$
  2. **Total Production time:** The worker who produces products A and B can work up to 50 hours/week. The machine that produces products A and B is able to construct a ton of product at a time, while the process lasts 5 hours. $5X_A + 5X_B \leq 50$
  3. **Storage availability for A:** The bins stored at maximum 8 tons of product A. $X_A \leq 8$
  4. **Storage availability for B:** The bins stored at maximum 6 tons of product B. $X_B \leq 6$
- **Feasible region:** The set of all solutions satisfying all constraints simultaneously.
- **Optimal Solution:** further than satisfying all constraints simultaneously, it should provide the maximum value Z of the objective function.

Knapsack Problem

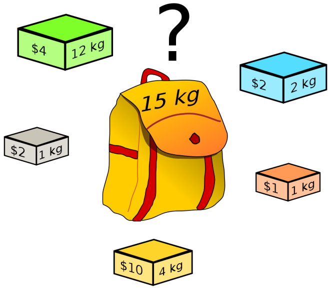
The **Knapsack Problem (KP)** can be defined as follows. Given a set of N objects. Each object i has a specified weight w_i and a specified profit p_i. Given a capacity C, which is the maximum total weight of the knapsack. The KP consists in finding a subset of the objects whose total weight is at most equal to the knapsack capacity and whose total profit is maximized

$$\textbf{Max } Z = \sum_{i=1}^{n} p_i x_i$$

subject to **(s.t.)**:

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

$$\forall i \in \{1, 2, \ldots, n\}, x_i \in \{0, 1\})$$



Outline

Complexity of Algorithms

- An algorithm needs two important resources to solve a problem: time (CPU) and space (RAM).
- The time complexity of an algorithm is the number of steps required to solve a problem of size n.
- The complexity is generally defined in terms of the worst-case analysis.
- The goal in the determination of the computational complexity of an algorithm is not to obtain an exact step count but an asymptotic bound on the step count.
- The Big-O notation makes use of asymptotic analysis. It is one of the most popular notations in the analysis of algorithms.

Definition of Big-O notation. An algorithm has a complexity $f(n) = O(g(n))$ if there exist positive constants n_0 and c such that $\forall n > n_0, f(n) \leq cg(n)$.

Search Time of Algorithms

The below table illustrates how the search time of an algorithm grows with the size of the problem using different time complexities of an optimization algorithm.

- The table shows clearly the combinatorial explosion of exponential complexities compared to polynomial ones.
- In practice, we cannot wait some centuries to solve a problem.
- The problem shown in the last line of the table needs the age of universe to solve it in an exact manner using exhaustive search.

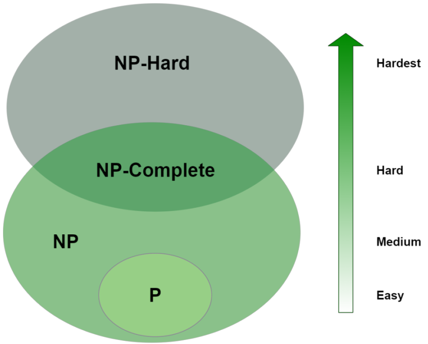
		Size				
Type of Algo.	**Complexity**	n = **10**	n = **20**	n = **30**	n = **40**	n = **50**
Logarithmic	$O(\log n)$	0.000001 s	0.000001 s	0.000001 s	0.000002 s	0.000003 s
Polynomial						
Linear	$O(n)$	0.00001 s	0.00002 s	0.00003 s	0.00004 s	0.00005 s
Quadratic	$O(n^2)$	0.0001 s	0.0004 s	0.0009 s	0.0016 s	0.0025 s
Cubic	$O(n^3)$	0.1 s	0.32 s	24.3 s	1.7 mn	5.2 mn
Exponential	$O(2^n)$	0.001 s	1.0 s	17.9 mn	12.7 *days*	35.7 *years*
	$O(3^n)$	0.059 s	58.0 mn	6.5 *years*	3855 *centuries*	2×10^8 *centuries*

Complexity of Problems

- The complexity of a problem is equivalent to the complexity of the best algorithm solving that problem.
- A problem is tractable (or easy) if there exists a polynomial-time algorithm to solve it.
- A problem is intractable (or difficult) if no polynomial-time algorithm exists to solve the problem.
- The complexity theory of problems deals with decision problems. A decision problem always has a yes or no answer.

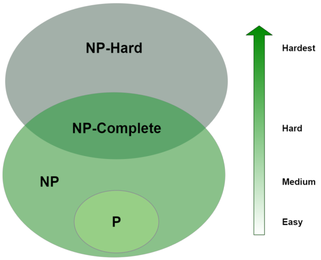
Complexity Classes of Problems (1/3)

- An important aspect of computational theory is to categorize problems into complexity classes.
- A complexity class represents the set of all problems that can be solved using a given amount of computational resources.
- There are two important classes of problems: P and NP.
- The figure on the right shows the relationship between P, NP, NP-complete, and NP-hard problems.



Complexity Classes of Problems (2/3)

- The complexity class **P** represents the family of problems easy to solve (i.e., a known polynomial-time algorithm exists to solve the problem).
- The complexity class **NP** represents the set of all decision problems that can be solved by a nondeterministic algorithm[a] in polynomial time.
- A decision problem $A \in$ **NP** is **NP-complete** if all other problems of class **NP** are reduced polynomially to the problem A (i.e., If a polynomial deterministic algorithm exists to solve an **NP-complete** problem, then all problems of class **NP** may be solved in polynomial time).

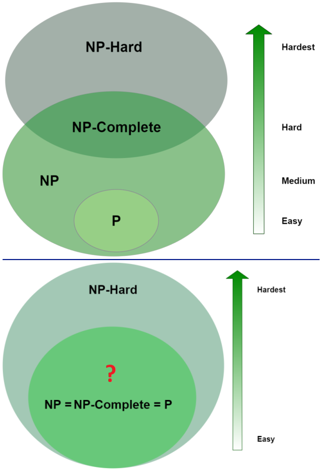


[a] In computer science, the term algorithm stands for a deterministic algorithm.

Complexity Classes of Problems (3/3)

- **NP-hard problems are optimization problems whose associated decision problems are NP-complete.**
- **Most of the real-world optimization problems are NP-hard for which provably efficient algorithms do not exist.**
- **They require exponential time (unless P = NP?[a]) to be solved in optimality.**
- **Heuristics/Metaheuristics constitute an important alternatives to solve NP-hard problems.**

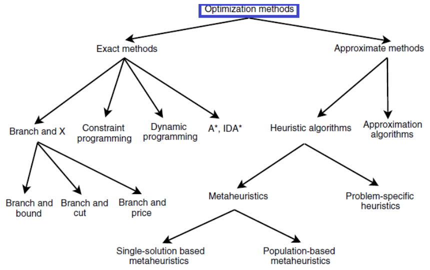
[a]This question is one of the millennium problems with a prize of US$ 1,000,000 for a first-found solution.



Outline

Optimization Methods

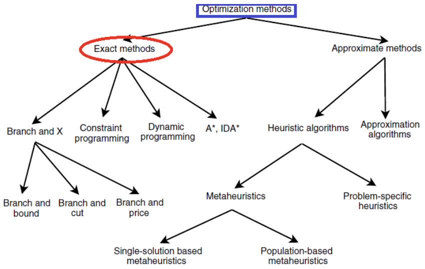
- **Following the complexity of the problem, it may be solved by an exact method or an approximate method.**



[a] In the artificial intelligence community, those algorithms are also named complete algorithms.

Optimization Methods

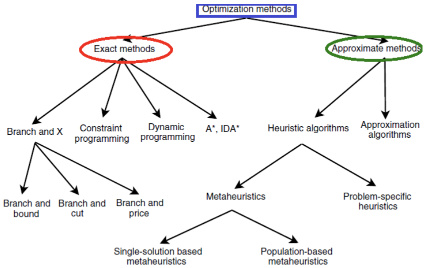
- Following the complexity of the problem, it may be solved by an exact method or an approximate method.
- Exact methods[a] obtain optimal solutions and guarantee their optimality.
- For NP-complete problems, exact algorithms are nonpolynomial-time algorithms (unless P = NP).



[a] In the artificial intelligence community, those algorithms are also named complete algorithms.

Optimization Methods

- Following the complexity of the problem, it may be solved by an exact method or an approximate method.
- Exact methods[a] obtain optimal solutions and guarantee their optimality.
- For NP-complete problems, exact algorithms are nonpolynomial-time algorithms (unless P = NP).
- Approximate (or heuristic) methods generate high-quality solutions in a reasonable time for practical use, but there is no guarantee of finding a global optimal solution.



[a] In the artificial intelligence community, those algorithms are also named complete algorithms.

Exact Methods (1/2)

In the class of exact methods we can find the following classical algorithms:

- **Dynamic programming is based on the recursive division of a problem into simpler subproblems.**
- **Branch and Bound and A* algorithms are based on an implicit enumeration of all solutions of the considered optimization problem.**
- **Constraint programming is a language built around concepts of tree search and logical implications.**

Exact methods can be applied to small instances of difficult problems (see the below Table).

Table: Order of Magnitude of the Maximal Size of Instances that State-of-the-Art Exact Methods can Solve to Optimality

Optimization Problems	Quadratic Assignment	Flow-Shop Scheduling	Graph Coloring	Capacitated Vehicle Routing
Size of the instances	30 objects	100 jobs 20 machines	100 nodes	60 clients

Exact Methods (2/2)

The size of the instance is not the unique indicator that describes the difficulty of a problem, but also its structure.

- For a given problem, some small instances cannot be solved by an exact algorithm while some large instances may be solved exactly by the same algorithm.
- The below table shows for some popular optimization problems (e.g., SOP[4]: Sequential Ordering Problem; QAP[5]: Quadratic Assignment Problem; GC[6]: Graph Coloring) small instances that are not solved exactly and large instances solved exactly by state-of-the-art exact optimization methods.

Optimization Problem	SOP	QAP	GC
Size of some unsolved instances	53	30	125
Size of some solved instances	70	36	561

[4] See https://arxiv.org/abs/1911.12427
[5] See https://en.wikipedia.org/wiki/Quadratic_assignment_problem
[6] See https://en.wikipedia.org/wiki/Graph_coloring

Approximate Algorithms

- In the class of approximate methods, two subclasses of algorithms may be distinguished:
  - Approximation Algorithms.
  - Heuristic Algorithms.
- Unlike heuristics, which usually find reasonably "good" solutions in a reasonable time, approximation algorithms provide provable solution quality and provable run-time bounds.

Heuristics find "good" solutions on large-size problem instances.
They allow to obtain acceptable performance at acceptable costs in a wide range of problems.
In general, heuristics do not have an approximation guarantee on the obtained solutions.

Heuristics

Heuristics find "good" solutions on large-size problem instances.

- **They allow to obtain acceptable performance at acceptable costs in a wide range of problems.**
- **In general, heuristics do not have an approximation guarantee on the obtained solutions.**
- **They may be classified into two families:**
  - **Specific Heuristics that are tailored and designed to solve a specific problem and/or instance.**
  - **Metaheuristics that are general-purpose algorithms that can be applied to solve almost any optimization problem. They may be viewed as upper level general methodologies that can be used as a guiding strategy in designing underlying heuristics to solve specific optimization problems.**

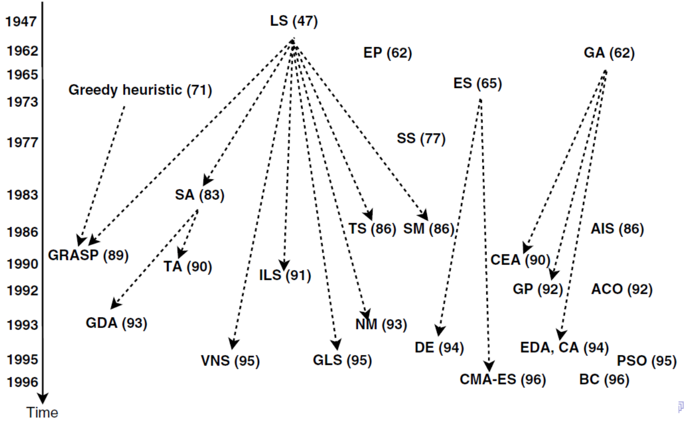
Metaheuristics (1/2)

Unlike exact methods, Metaheuristics allow to tackle large-size problem instances by delivering satisfactory solutions in a reasonable time.

- There is no guarantee to find global optimal solutions or even bounded solutions.
- Metaheuristics have received more and more popularity in the past 30 years.
- Their use in many applications shows their efficiency and effectiveness to solve large and complex problems.
- Application of Metaheuristics falls into a large number of areas; some them are:
  - Engineering design, topology optimization and structural optimization in electronics and VLSI, telecommunications, automotive, and robotics.
  - Machine learning and data mining in bioinformatics and finance.
  - System modeling, simulation and identification in chemistry, physics, and biology; control, signal, and image processing.
  - Planning in routing problems, robot planning, scheduling and production problems, logistics and transportation, supply chain management, environment, and so on.

Metaheuristics (2/2)

Optimization is everywhere; optimization problems are often complex; then Metaheuristics are everywhere. The below figure shows the genealogy of the numerous Metaheuristics in the literature.

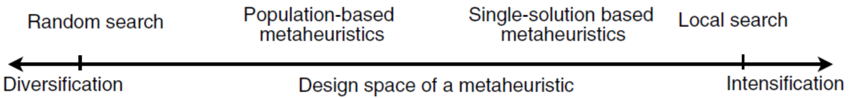


Categories of Metaheuristics (1/3)

In designing a metaheuristic, two contradictory criteria must be taken into account: exploitation of the best solutions found (intensification) and exploration of the search space (diversification).

- In intensification, the promising regions are explored more thoroughly in the hope to find better solutions.
- In diversification, nonexplored regions must be visited to be sure that all regions of the search space are evenly explored and that the search is not confined to only a reduced number of regions.

In this design space, the extreme search algorithms in terms of the exploration (resp. exploitation) are random search (resp. iterative improvement local search).



Categories of Metaheuristics (2/3)

Metaheuristics can be classified into two main categories:

- **Single-Solution Based Search Metaheuristics: intensification oriented, at each iteration one selects the best neighboring solution that improves the current solution.**
  - Local Search, Iterated Local Search, Guided Local Search.
  - Variable Neighborhood Search
  - Tabu Search
  - Simulated Annealing
- **Population-Based Search Metaheuristics: diversification oriented, at each iteration, one generates a random solution in the search space.**
  - Genetic Algorithms
  - Swarm Intelligence Algorithms
  - Scatter Search

Categories of Metaheuristics (3/3)

Many other classification criteria may be used for Metaheuristics:

- **Nature inspired versus nonnature inspired:** Many Metaheuristics are inspired by natural processes from biology (e.g., ants, bees colonies), or social science (e.g., particle swarm), or physics (e.g., simulated annealing).
- **Memory usage versus memoryless methods:** Some metaheuristic algorithms use a memory that contains some information extracted online during the search (e.g., short-term and long-term memories in tabu search).
- **Deterministic versus stochastic:** In stochastic Metaheuristics, some random rules are applied during the search where different final solutions may be obtained from the same initial solution.(e.g., simulated annealing, evolutionary algorithms).
- **Iterative versus greedy:** Greedy algorithms start from an empty solution, and at each step a decision variable of the problem is assigned until a complete solution is obtained. Most of the Metaheuristics are iterative algorithms that start with a complete solution (or population of solutions) and transform it at each iteration using some search operators.

The End