Optimization with Metaheuristics

CSE480 & CSE591- Week 1 - Introduction to Optimization

Asst. Prof. Gizem Süngü Terci

October 4, 2025

YEDİTEPE UNIVERSITY

Overview

1. Motivation Example
2. Definition of Optimization
3. More Examples
4. Optimization Process
5. Exact vs. Heuristic Algorithms
6. Heuristic vs. Metaheuristic Algorithms
7. Course Plan

Transportation Planning Example

Transportation Planning

Task

Build a system that advises a logistics company on how to complete all transportation orders while minimizing costs.[a]

[a] The example is taken from: https://thomasweise.github.io/aitoa-slides/01_introduction.pdf

Transportation Planning

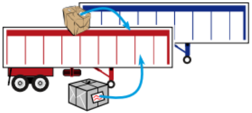
Analyze:

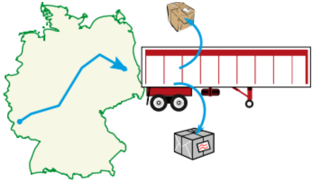
- Find routes on the map

Transportation Planning

Analyze:

- Find routes on the map and assignments of orders to containers

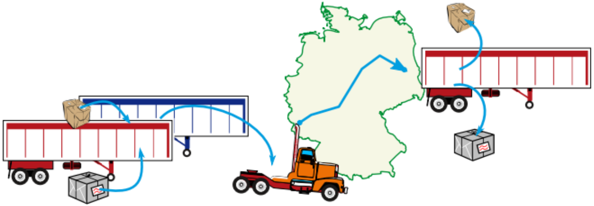




Transportation Planning

Analyze:

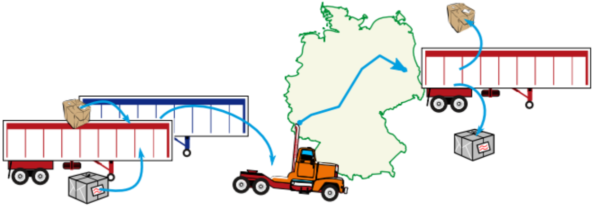
- Find routes on the map and assignments of orders to containers and containers to trucks/trains



Transportation Planning

Analyze:

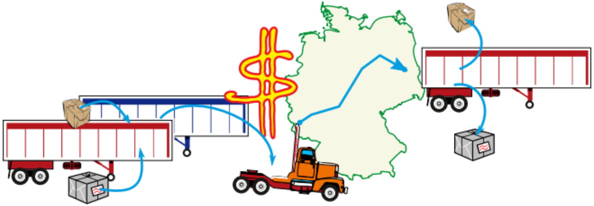
- Find routes on the map and assignments of orders to containers and containers to trucks/trains which minimize the undelivered orders



Transportation Planning

Analyze:

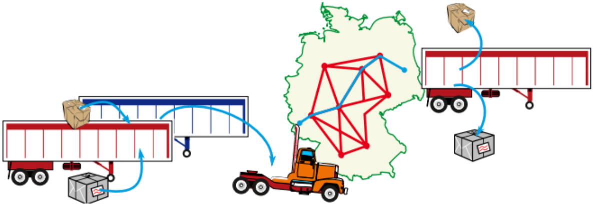
- Find routes on the map and assignments of orders to containers and containers to trucks/trains which minimize the undelivered orders and the total distance for



Transportation Planning

Analyze:

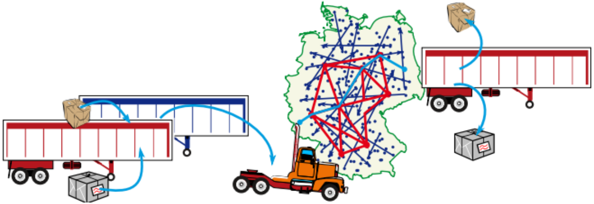
- Find routes on the map and assignments of orders to containers and containers to trucks/trains which minimize the undelivered orders and the total distance for
- multiple depots



Transportation Planning

Analyze:

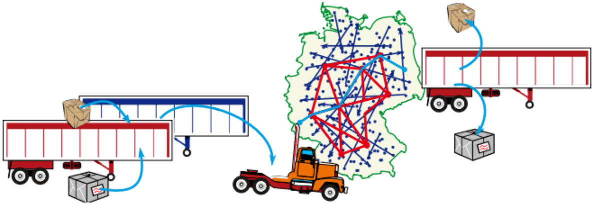
- Find routes on the map and assignments of orders to containers and containers to trucks/trains which minimize the undelivered orders and the total distance for
- multiple depots and pickup and delivery locations, while considering that



Transportation Planning

Analyze:

- Find routes on the map and assignments of orders to containers and containers to trucks/trains which minimize the undelivered orders and the total distance for
- multiple depots and pickup and delivery locations, while considering that
- vehicles (trucks and trains) have capacity limits



Transportation Planning

Analyze:

- Find routes on the map and assignments of orders to containers and containers to trucks/trains which minimize the undelivered orders and the total distance for orders and the total distance for
- multiple depots and pickup and delivery locations, while considering that
- vehicles (trucks and trains) have capacity limits
- time windows for pickup and delivery

Transportation Planning

Analyze:

- Find routes on the map and assignments of orders to containers and containers to trucks/trains which minimize the undelivered orders and the total distance for
- multiple depots and pickup and delivery locations, while considering that
- vehicles (trucks and trains) have capacity limits
- time windows for pickup and delivery
- and constraints and laws.
- Time limit to find a solution: 1 day

Transportation Planning

Solution?

Transportation Planning

Solution? No algorithm or existing solution is available. $\rightarrow$ NP-Hard

Transportation Planning

Solution?

With an **optimization** algorithm, we can get reasonable (feasible) solutions.

Optimization

Optimization

What is the cheapest way to get from İzmir to İstanbul?

Optimization

How can I package these products using the fewest boxes?

Optimization

How do I arrange the components on a circuit board so I need the shortest electrical cable length?

Optimization

Definition (Economical View)

An optimization problem is a situation which requires deciding for one choice from a set of possible alternatives in order to reach a predefined/required benefit at minimal costs.

Optimization

Definition (General Mathematical Formulation)

Finding the best choice among a set of options subject to a set of constraints

- **Decision variables:** $x \in \mathbb{R}^n$ represent the choices to be made.
- **Objective function:** $f(x) : \mathbb{R}^n \to \mathbb{R}$ assigns a cost or benefit to each choice.
- **Constraints:** A feasible set $\mathcal{X} = \{x \in \mathbb{R}^n \mid g_i(x) \leq 0,\ h_j(x) = 0\}$ restricts the admissible values of x.

The optimization task is then:

$$\begin{aligned} &\min_{x \in \mathcal{X}} \ f(x) \\ &\text{subject to:} \quad g_i(x) \leq 0,\ i = 1, \ldots, m, \\ &\qquad\qquad\quad\ \ h_j(x) = 0,\ j = 1, \ldots, p. \end{aligned}$$

More Examples

More Examples

Many questions in the real world are optimization problems.

More Examples

Problem 1: An animal feed company must produce 200 kg of a mixture consisting of ingredients X_1 and X_2 daily. X_1 costs Rs. 3 per kg and X_2 costs Rs. 8 per kg. Not more than 80 kg of X_1 can be used, and at least 60 kg of X_2 must be used. Find how much of each ingredient should be used if the company wants to maximize cost.

More Examples

Problem 1: An animal feed company must produce 200 kg of a mixture consisting of ingredients X_1 and X_2 daily. X_1 costs Rs. 3 per kg and X_2 costs Rs. 8 per kg. Not more than 80 kg of X_1 can be used, and at least 60 kg of X_2 must be used. Find how much of each ingredient should be used if the company wants to maximize cost.

Solution: Firstly, define the components of the optimization

- Decision variables:
- Objective function:
- Consraints:

More Examples

Problem 2: A company makes two kinds of leather belts. Belt A is a high quality belt, and belt B is of lower quality. The respective profits are Re. 0.40 and Re. 0.30 per belt. The supply of leather is sufficient for only 800 belts per day (both A and B combined). Belt A requires a fancy buckle, and only 400 per day are available. There are only 700 buckles a day available for belt B.
What should be the daily production of each type of belt? Formulate the problem model.

More Examples

Homework: A factory produces two types of smartphone cases: **Model A** (premium) and **Model B** (standard).

- Profit per unit: A = 24 TL, B = 16 TL.
- Molding time: A = 6 min, B = 3 min; molding line capacity = 1,200 min/day.
- Finishing time: A = 5 min, B = 4 min; finishing station capacity = 1,000 min/day.
- Material limit: at most 260 kg/day of polymer; A uses 0.6 kg/unit, B uses 0.4 kg/unit.
- Packaging kits: at most 180/day for A and 260/day for B.
- Demand requirement: at least 80 units/day of Model B must be produced.

Tasks:

1. Define the decision variables.
2. Formulate the linear programming model (objective and constraints, including non-negativity).
3. (Graphical) Sketch the feasible region and determine the optimal production plan.

More Examples

Graph Coloring Problem: For a graph $G = (V, E)$, a function

$$c : V \to \{1, 2, \dots, k\}$$

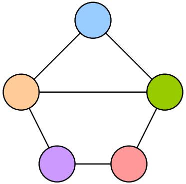
that assigns a color to each vertex $v \in V$ must be found such that for every edge $(u, v) \in E$, the condition $c(u) \neq c(v)$ holds.
The smallest value of k for which this condition is satisfied is called the chromatic number of the graph, and it is denoted by $\chi(G)$.

More Examples

Graph Coloring Problem: For a graph $G = (V, E)$, a function

$$c : V \to \{1, 2, \ldots, k\}$$

that assigns a color to each vertex $v \in V$ must be found such that for every edge $(u, v) \in E$, the condition $c(u) \neq c(v)$ holds.
The smallest value of k for which this condition is satisfied is called the chromatic number of the graph, and it is denoted by $\chi(G)$.

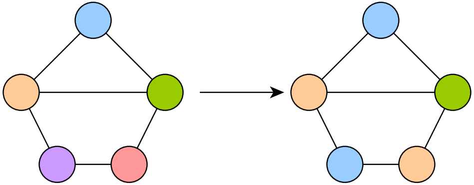


More Examples

Graph Coloring Problem: For a graph $G = (V, E)$, a function

$$c : V \to \{1, 2, \ldots, k\}$$

that assigns a color to each vertex $v \in V$ must be found such that for every edge $(u, v) \in E$, the condition $c(u) \neq c(v)$ holds.
The smallest value of k for which this condition is satisfied is called the chromatic number of the graph, and it is denoted by $\chi(G)$.



More Examples

Mathematical Formulation of Graph Coloring Problem:

Decision Variables:

$$x_{v,i} = \begin{cases} 1 & \text{if vertex } v \in V \text{ is assigned color } i, \\ 0 & \text{otherwise} \end{cases} \qquad y_i = \begin{cases} 1 & \text{if color } i \text{ is used,} \\ 0 & \text{otherwise} \end{cases}$$

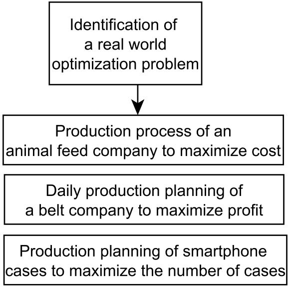
Constraints:

$$\begin{aligned} &\sum_{i=1}^{k} x_{v,i} = 1 && \forall v \in V \quad \text{(each vertex has one color)} \\ &x_{u,i} + x_{v,i} \leq 1 && \forall (u, v) \in E,\ \forall i = 1, \ldots, k \quad \text{(adjacent vertices differ)} \\ &x_{v,i} \leq y_i && \forall v \in V,\ \forall i = 1, \ldots, k \quad \text{(linking variables)} \end{aligned}$$

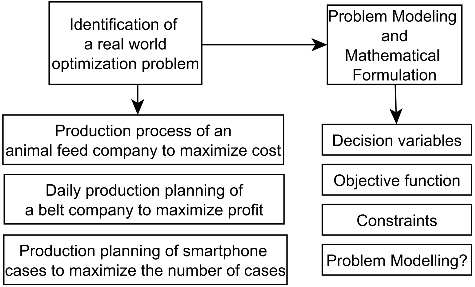
Objective Function:

$$\min \sum_{i=1}^{k} y_i \qquad \text{(minimize number of colors used)}$$

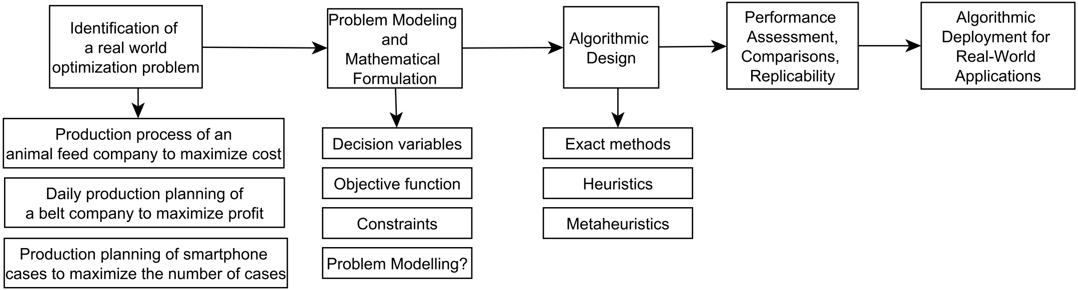
Optimization Process



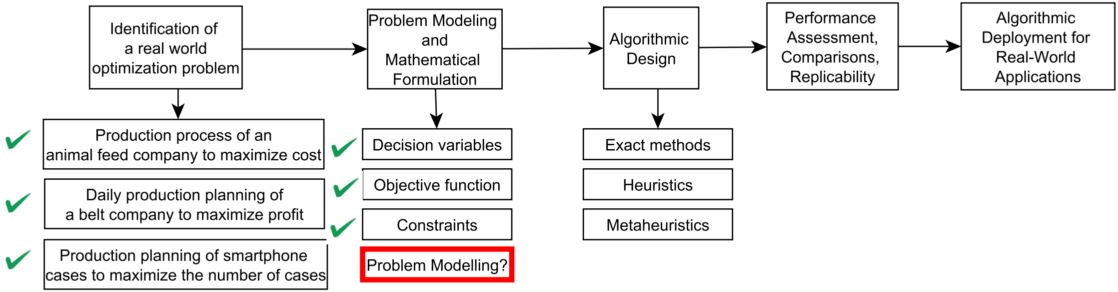
Optimization Process



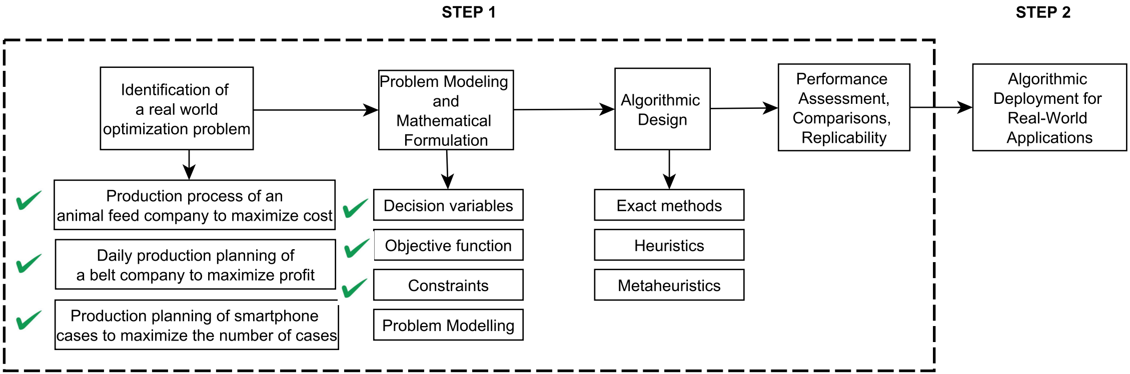
Optimization Process



Optimization Process



Optimization Process



1. Developing and implementing a good algorithm that can solve the problem at hand and
2. integrating this implementation into the existing software ecosystem.

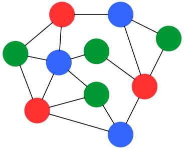
Exact vs. Heuristic Algorithms

Exact vs. Heuristic Algorithms

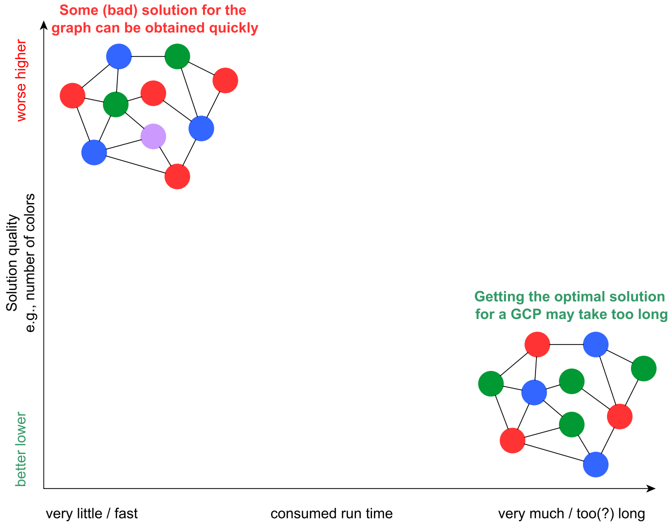
In optimization, there exist exact and heuristic algorithms.
Let's again look at **Graph Coloring Problem**.

Exact vs. Heuristic Algorithms

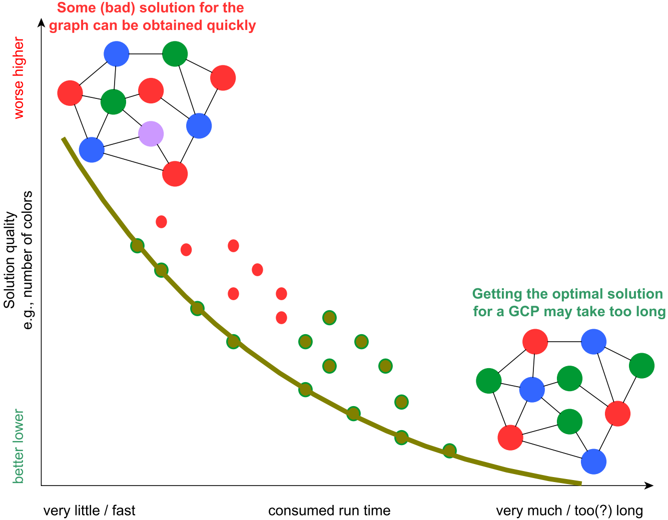
Getting the optimal solution
for a GCP may take too long



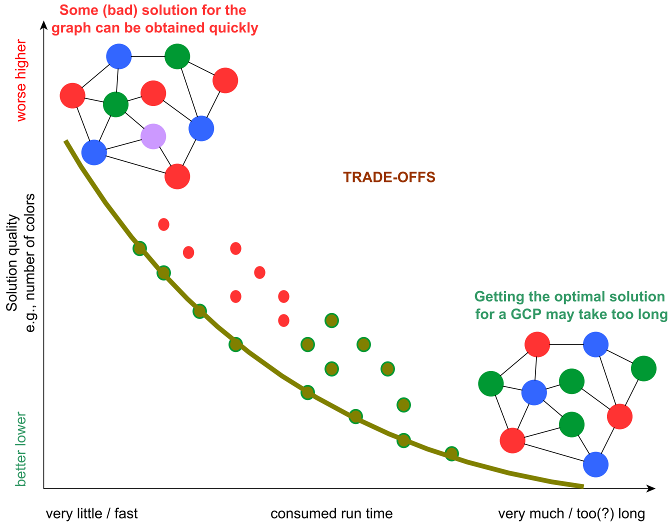
Exact vs. Heuristic Algorithms



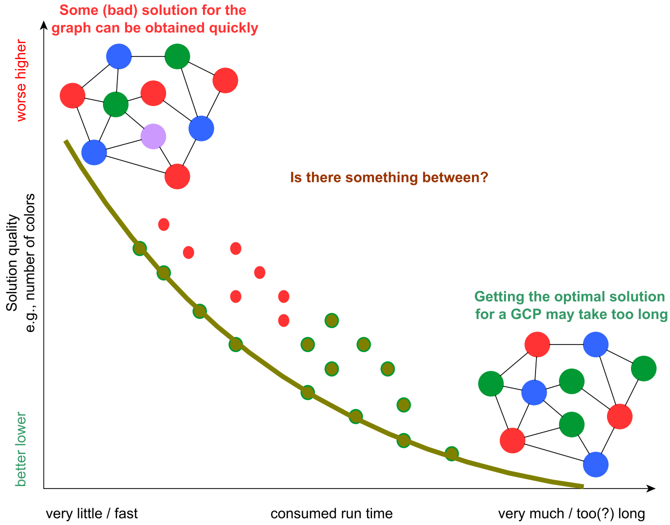
Exact vs. Heuristic Algorithms



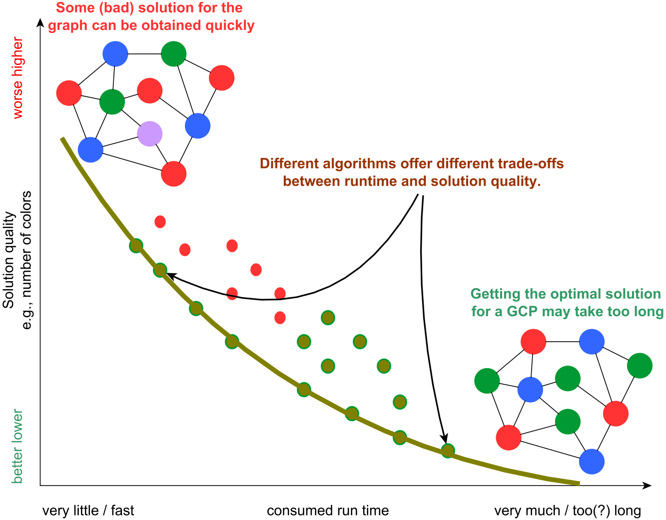
Exact vs. Heuristic Algorithms



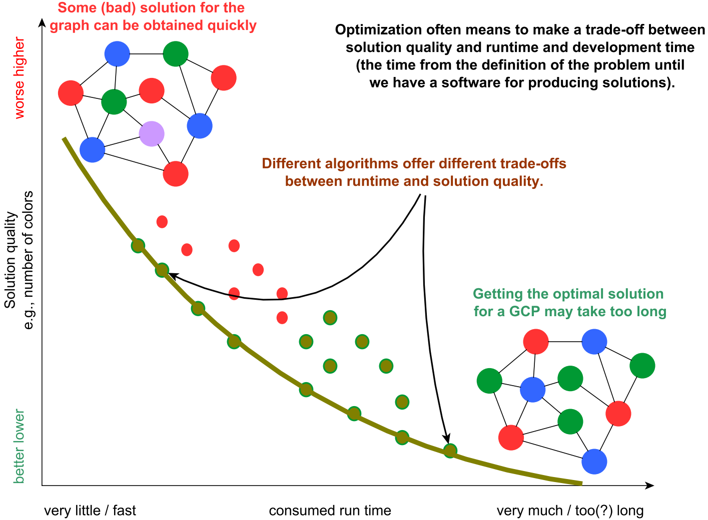
Exact vs. Heuristic Algorithms



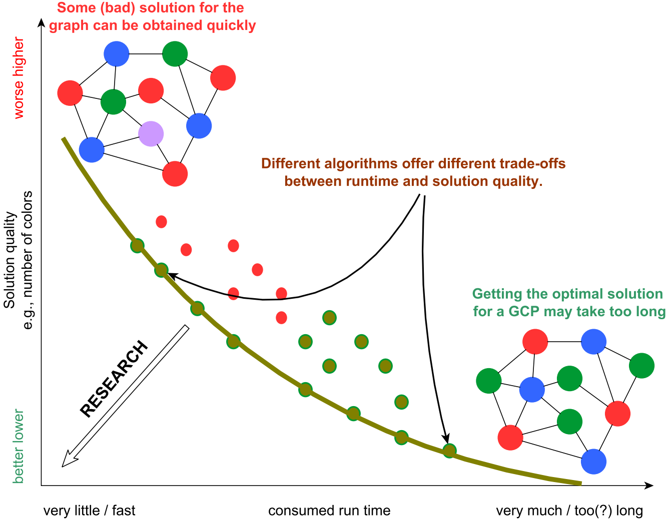
Exact vs. Heuristic Algorithms



Exact vs. Heuristic Algorithms



Exact vs. Heuristic Algorithms



Exact vs. Heuristic Algorithms

Exact Algorithms

Algorithms that always guarantee finding the optimal solution to a given problem within a finite amount of time. For instance, Integer Linear Programming (ILP) can be used to solve mathematical formulations of optimization problems with solvers such as CPLEX or Gurobi.

Exact vs. Heuristic Algorithms

Exact Algorithms

Algorithms that always guarantee finding the optimal solution to a given problem within a finite amount of time. For instance, Integer Linear Programming (ILP) can be used to solve mathematical formulations of optimization problems with solvers such as CPLEX or Gurobi.

Heuristic Algorithms

Heuristic comes from the Greek verb *heurískein*, meaning "to search" or "to discover". The word is related to the famous exclamation *Eureka!* ("I have found it!"), attributed to Archimedes.

Exact vs. Heuristic Algorithms

Exact Algorithms

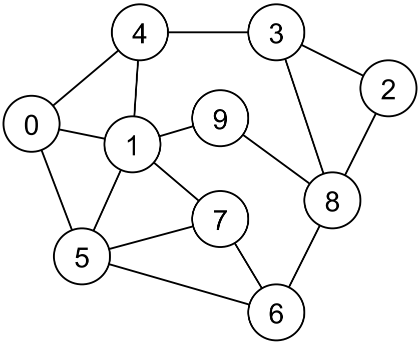
Algorithms that always guarantee finding the optimal solution to a given problem within a finite amount of time. For instance, Integer Linear Programming (ILP) can be used to solve mathematical formulations of optimization problems with solvers such as CPLEX or Gurobi.

Heuristic Algorithms

Algorithms that quickly find a feasible solution, not necessarily optimal, for a hard optimization problem. Generally a heuristic is problem-specific.

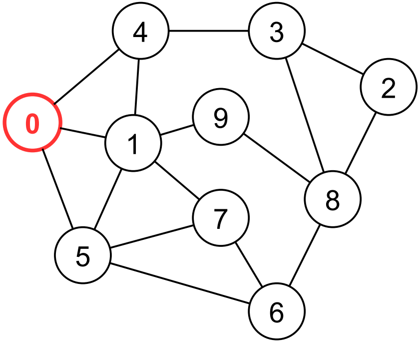
Heuristic Algorithms

Heuristic Algorithm 1 (Greedy Method)



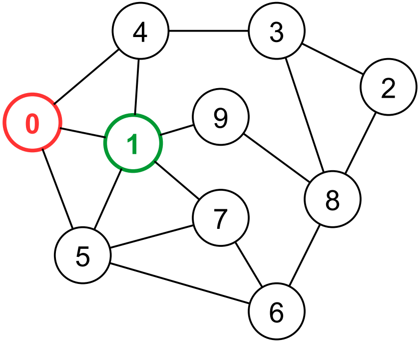
Heuristic Algorithms

Heuristic Algorithm 1 (Greedy Method)



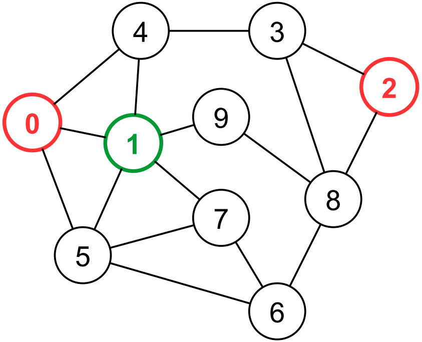
Heuristic Algorithms

Heuristic Algorithm 1 (Greedy Method)



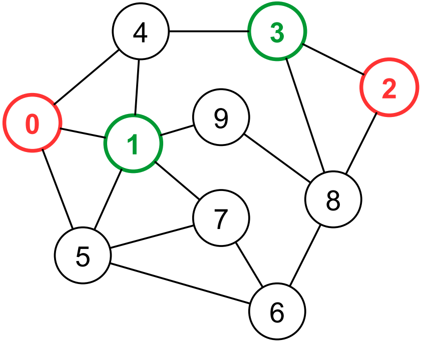
Heuristic Algorithms

Heuristic Algorithm 1 (Greedy Method)



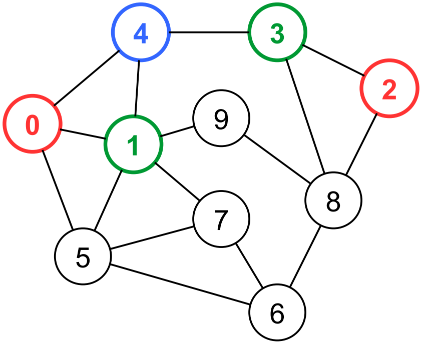
Heuristic Algorithms

Heuristic Algorithm 1 (Greedy Method)



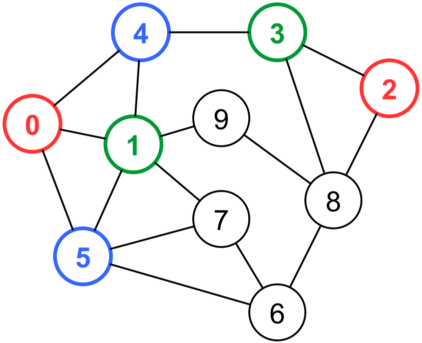
Heuristic Algorithms

Heuristic Algorithm 1 (Greedy Method)



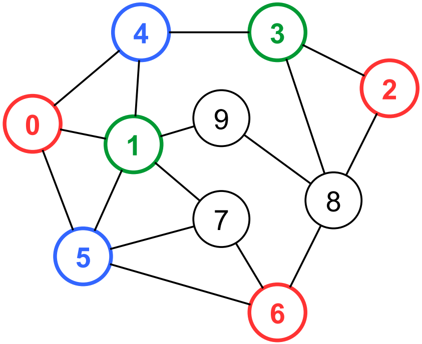
Heuristic Algorithms

Heuristic Algorithm 1 (Greedy Method)



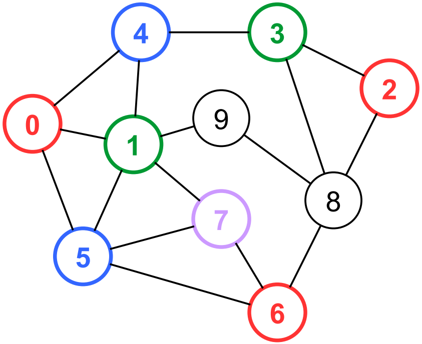
Heuristic Algorithms

Heuristic Algorithm 1 (Greedy Method)



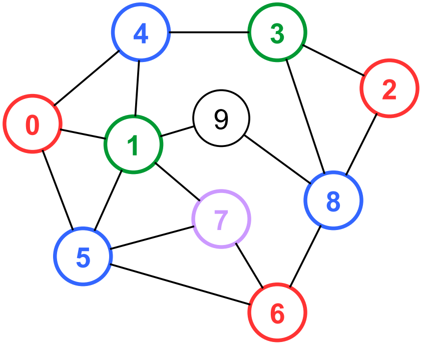
Heuristic Algorithms

Heuristic Algorithm 1 (Greedy Method)

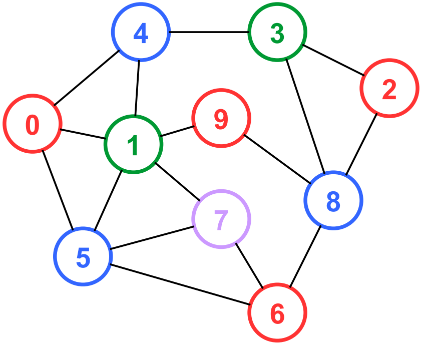


Heuristic Algorithms

Heuristic Algorithm 1 (Greedy Method)



Heuristic Algorithms



Is it possible to obtain better solution with another heuristic algorithm?

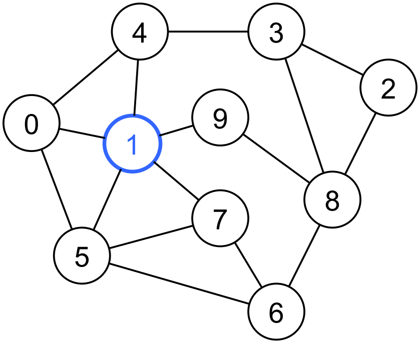
Heuristic Algorithms

DSATUR (Degree of Saturation Algorithm), Brélaz (1979) greedy algorithm: It orders the vertices using their saturation degree and determines which vertices should be colored with priority.

- 1. Initially, vertices with the highest degree are colored.
- 2. Then, the vertex with the highest saturation degree (i.e., adjacent to the largest number of differently colored vertices), and, in case of a tie, the one with the highest degree, is selected and colored.
- 3. Step 2 is repeated until all vertices are assigned a color.

Heuristic Algorithms

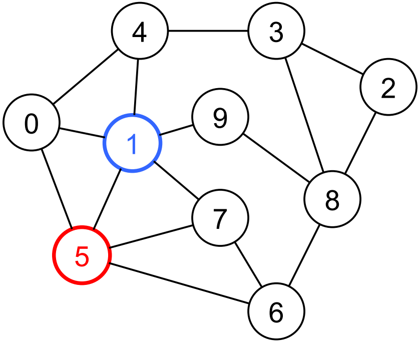
Heuristic Algorithm 2 (DSATUR):



Vertex v_1 has the highest degree with 5.

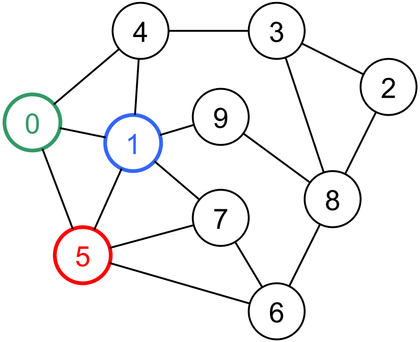
Heuristic Algorithms

Heuristic Algorithm 2 (DSATUR):



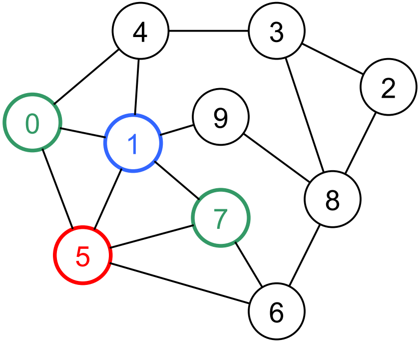
Heuristic Algorithms

Heuristic Algorithm 2 (DSATUR):



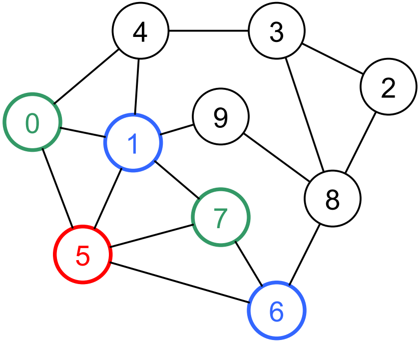
Heuristic Algorithms

Heuristic Algorithm 2 (DSATUR):



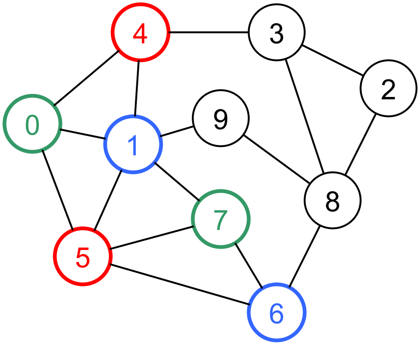
Heuristic Algorithms

Heuristic Algorithm 2 (DSATUR):



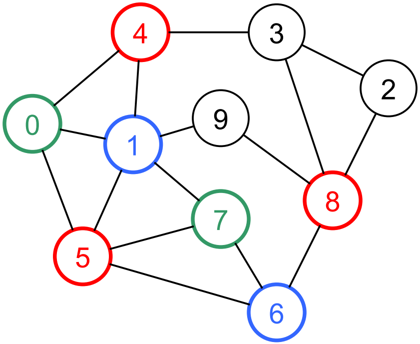
Heuristic Algorithms

Heuristic Algorithm 2 (DSATUR):



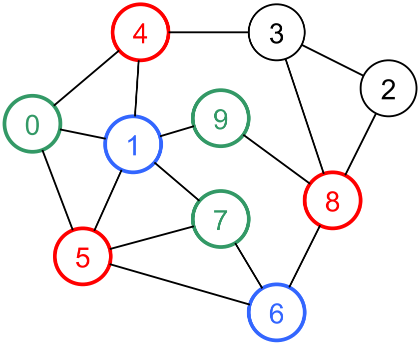
Heuristic Algorithms

Heuristic Algorithm 2 (DSATUR):



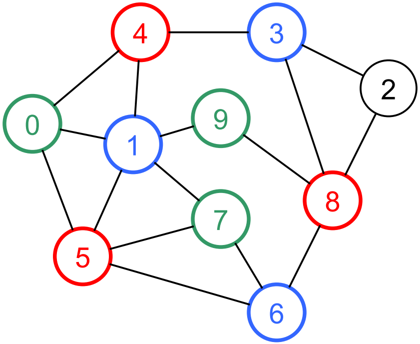
Heuristic Algorithms

Heuristic Algorithm 2 (DSATUR):



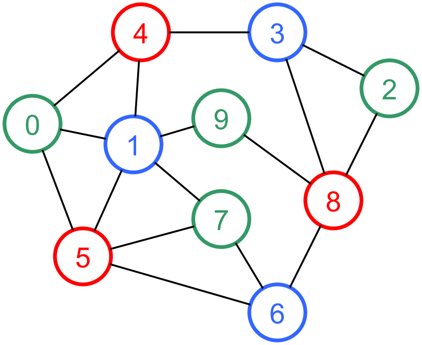
Heuristic Algorithms

Heuristic Algorithm 2 (DSATUR):



Heuristic Algorithms

Heuristic Algorithm 2 (DSATUR):



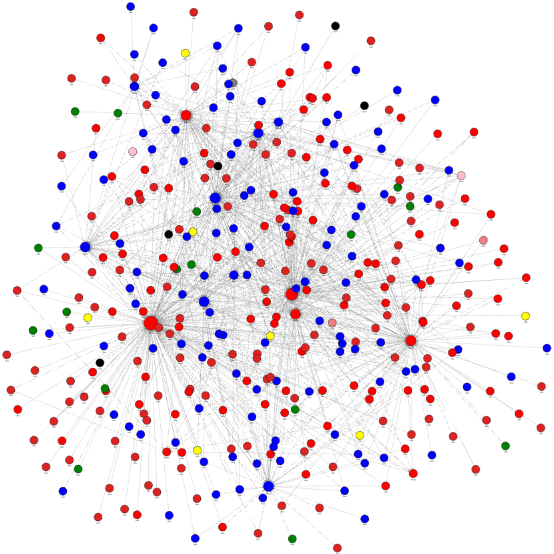
The best solution is found with 3 colors!

Heuristic Algorithms

Conclusion about Heuristic Algorithms:

- The decision mechanisms of the applied heuristics directly affect the ordering of vertex coloring.
- Different vertex coloring orders may produce different solutions for the same graph.
- To find the best solution, all ordering combinations could, in principle, be tested.
- However, is it feasible to try all combinations for a graph with millions of vertices and edges? (NP-Hard!)

Heuristic Algorithms



We need Metaheuristics!!

Heuristic vs. Metaheuristic Algorithms

Heuristic vs. Metaheuristic Algorithms

Heuristic Algorithms

Algorithms that quickly find a feasible solution, not necessarily optimal, for a hard optimization problem. Generally a heuristic is problem-specific.

Metaheuristic Algorithms

A metaheuristic is a problem-independent high-level strategy designed to guide a search in order to solve a larger spectrum of optimization problems.

Course Plan

Course Metarials

- Talbi, E.G., 2009. Metaheuristics: from design to implementation. John Wiley & Sons.
- Osaba E, Villar-Rodriguez E, Del Ser J, Nebro AJ, Molina D, LaTorre A, Suganthan PN, Coello CA, Herrera F. A tutorial on the design, experimentation and application of metaheuristic algorithms to real-world optimization problems. Swarm and Evolutionary Computation. 2021 Jul 1;64:100888.

Grading

You will be evaluated based on quizzes, a midterm exam, a term project, and a final exam. The grade distribution is as follows:

- Quizzes – 20% Five quizzes will be given throughout the semester. All quizzes will count toward your final grade, except the one with the lowest score, which will be discarded.
- Term Project – 20%
- Midterm Exam – 30%
- Final Exam – 30%

Note: Course grading will follow the university catalog system. Important: Submissions must be your own work and contain valid, original content. Material taken from the internet, textbooks, classmates' notes, or examples from lectures/labs simply to meet a submission requirement will not be accepted. Violations may result in disciplinary action by the Faculty.

Course Outline

W	Date	Topics	Quiz
1	Sep 29- Oct 3	Introduction to Optimization & Metaheuristics; Real-world Motivation	
2	Oct 6-10	Problem Formulation, Objective Functions, Constraints, Search Spaces	
3	Oct 13-17	Local Search Fundamentals; Neighborhood Structures	Quiz 1
4	Oct 20-24	Simulated Annealing	
5	Oct 27-31	Tabu Search	
6	Nov 3-7	Iterated Local Search and Variable Neighborhood Search	Quiz 2
7	**Nov 10-14**	**MIDTERM EXAM**	
8	Nov 17-21	Population-based Search – Representation & Operators	
9	Nov 24-28	Genetic Algorithms	Quiz 3
10	Dec 1-5	Evolution Strategies & Differential Evolution	
11	Dec 8-12	Particle Swarm Optimization	Quiz 4
12	Dec 15-19	Ant Colony Optimization	
13	Dec 22-26	Parameter Tuning, Performance Evaluation, Statistical Testing, Replicability	
14	Dec 29-Jan 2	Case Studies & Student Project Presentations	Quiz 5

Quizzes

- Each quiz will take 30 minutes and will be held during the last part of the lecture.
- The scope of each quiz covers topics from the previous weeks' lectures.
- The quiz schedule is provided in the syllabus.

Term Project

- Students may work individually or in groups of two.
- Details about project topics, milestones, and deliverables will be provided.
- The final presentation and report are due at the end of the semester.
- **First step:** Decide whether you will work individually or in a group of two. Define your optimization problem and related dataset. Send me an email about it to arrange a discussion.

Metaheuristics: What & Why

Intuition

A metaheuristic is a high-level *search strategy* that **guides** one or more heuristics to explore a large solution space efficiently and produce (near-)optimal solutions.

- **Not problem-specific:** Same framework, small tweaks across problems.
- **Approximate & often non-deterministic:** Aim for high-quality, not guaranteed-optimal.
- **Exploration vs. Exploitation:** Explore broadly to find promising regions; exploit to refine good solutions.
- **Use of memory/learning:** Modern methods store experience (tabu lists, pheromones, elite sets).
- **Role:** Acts like a *coach* that tells lower-level heuristics how to move in the search space.

Metaheuristics: What & Why

Intuition

A metaheuristic is a high-level *search strategy* that **guides** one or more heuristics to explore a large solution space efficiently and produce (near-)optimal solutions.

When do we use them?

When exact methods are impractical (NP-hardness, combinatorial blow-up, noisy or nonlinear objectives), metaheuristics provide robust, adaptable performance.