Introduction:
Combinatorial Problems

Combinatorial Problem Solving (CPS)

Enric Rodríguez-Carbonell

February 13, 2024

Combinatorial Problems

- A combinatorial problem consists in finding, among a finite set of objects, one that satisfies a set of constraints
- Several variations:
  - Find one solution
  - Find all solutions
  - Find best solution according to an objective function

Examples (I): Prop. Satisfiability

- Given a formula F in propositional logic, is F satisfiable?

  ($=$ is there any assignment of Boolean values to variables that evaluates F to “true”?)

Examples (I): Prop. Satisfiability

- Given a formula F in propositional logic, is F satisfiable?

  ($=$ is there any assignment of Boolean values to variables that evaluates F to "true"?)

- Is $(p \vee q) \ \wedge \ (p \vee \neg q) \ \wedge \ (\neg p \vee q)$ satisfiable?

Examples (I): Prop. Satisfiability

- Given a formula F in propositional logic, is F satisfiable?

  ($=$ is there any assignment of Boolean values to variables that evaluates F to "true"?)

- Is $(p \vee q) \ \wedge \ (p \vee \neg q) \ \wedge \ (\neg p \vee q)$ satisfiable?

  Yes: set p, q to true

Examples (I): Prop. Satisfiability

- Given a formula F in propositional logic, is F satisfiable?

  ($=$ is there any assignment of Boolean values to variables that evaluates F to "true"?)

- Is $(p \vee q) \ \wedge \ (p \vee \neg q) \ \wedge \ (\neg p \vee q)$ satisfiable?

  Yes: set p, q to true

- Is $(p \vee q) \ \wedge \ (p \vee \neg q) \ \wedge \ (\neg p \vee q) \ \wedge \ (\neg p \vee \neg q)$ satisfiable?

Examples (I): Prop. Satisfiability

- Given a formula F in propositional logic, is F satisfiable?

  ($=$ is there any assignment of Boolean values to variables that evaluates F to "true"?)

- Is $(p \vee q) \ \wedge \ (p \vee \neg q) \ \wedge \ (\neg p \vee q)$ satisfiable?

  Yes: set p, q to true

- Is $(p \vee q) \ \wedge \ (p \vee \neg q) \ \wedge \ (\neg p \vee q) \ \wedge \ (\neg p \vee \neg q)$ satisfiable?

  No

Examples (I): Prop. Satisfiability

- Given a formula F in propositional logic, is F satisfiable?

  (= is there any assignment of Boolean values to variables that evaluates F to "true"?)

- Is $(p \vee q) \;\wedge\; (p \vee \neg q) \;\wedge\; (\neg p \vee q)$ satisfiable?

  Yes: set p, q to true

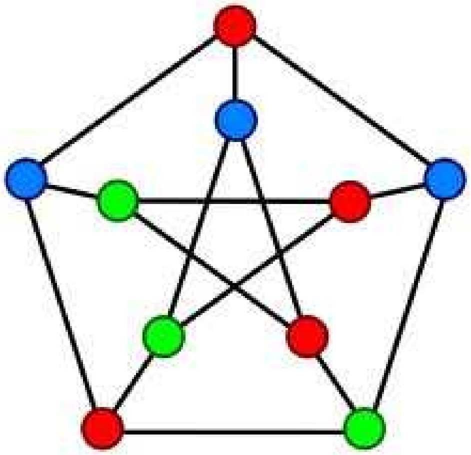
- Is $(p \vee q) \;\wedge\; (p \vee \neg q) \;\wedge\; (\neg p \vee q) \;\wedge\; (\neg p \vee \neg q)$ satisfiable?

  No

- Arises in:
  - Hardware verification
  - Circuit optimization
  - ...

Examples (II): Graph Coloring

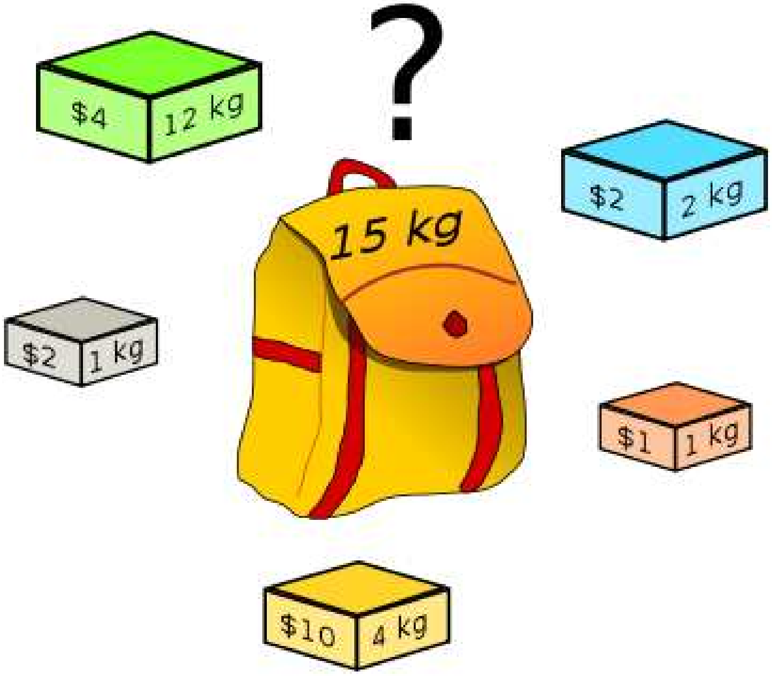
- Given a graph and a number of colors, can vertices be painted so that neighbors have different colors?



- Arises in:
    - Frequency assignment
    - Register allocation
    - ...

Examples (III): Knapsack

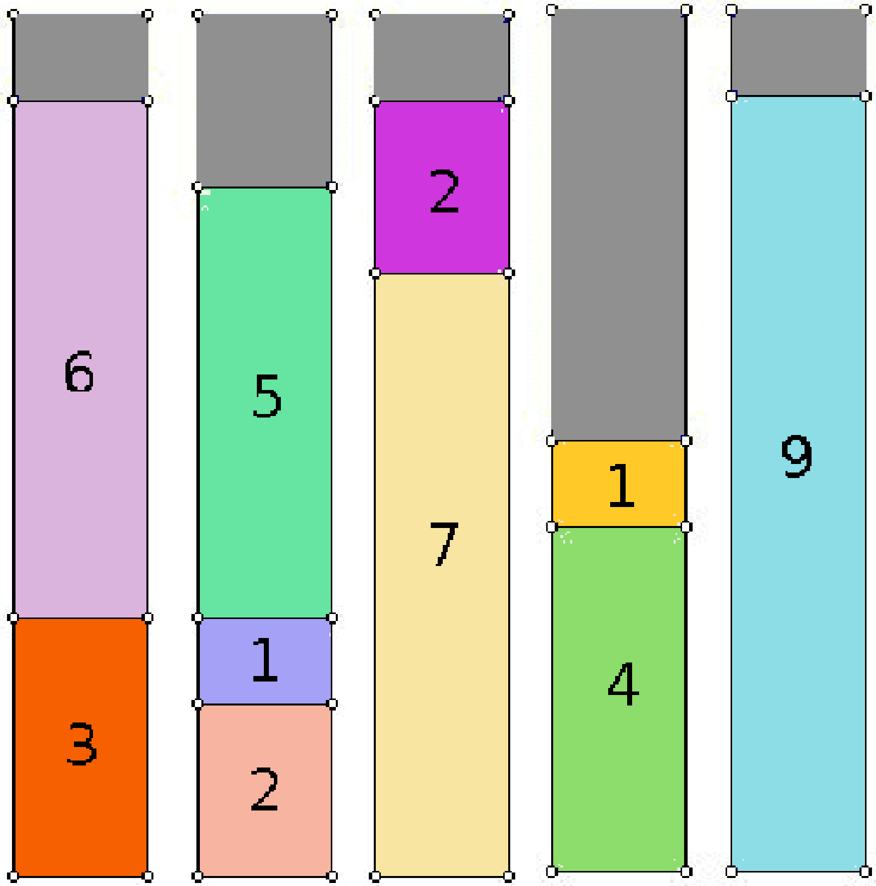
- Given n items with weights w_i and values v_i, a capacity W and a number V, is there a subset S of the items such that $\sum_{i \in S} w_i \leq W$ and $\sum_{i \in S} v_i \geq V$?



- Arises in:
  - Selection of capital investments
  - Cutting stock problems
  - ...

Examples (IV): Bin Packing

- Given n items with volumes v_i and k identical bins with capacity V, is it possible to place all items in bins?



- Arises in:
  - Logistics
  - ...

A Note on Complexity

- All previous examples are NP-complete
  - No known polynomial algorithm (likely none exists)
  - Available algorithms have worst-case exp behavior: there will be small instances that are hard to solve
  - In real-world problems there is a lot of structure, which can hopefully be exploited

A Note on Complexity

- All previous examples are NP-complete
  - No known polynomial algorithm (likely none exists)
  - Available algorithms have worst-case exp behavior: there will be small instances that are hard to solve
  - In real-world problems there is a lot of structure, which can hopefully be exploited
- Other combinatorial problems solvable in P-time, e.g.
  - Bipartite matching: given a set of boys and girls and their compatibilities, can we marry all of them?
  - Shortest paths: given a graph and two vertices, which is the shortest way to go from one to the other?

A Note on Complexity

- All previous examples are NP-complete
  - No known polynomial algorithm (likely none exists)
  - Available algorithms have worst-case exp behavior: there will be small instances that are hard to solve
  - In real-world problems there is a lot of structure, which can hopefully be exploited
- Other combinatorial problems solvable in P-time, e.g.
  - Bipartite matching: given a set of boys and girls and their compatibilities, can we marry all of them?
  - Shortest paths: given a graph and two vertices, which is the shortest way to go from one to the other?
- Our focus will be on hard (= NP-complete) problems

Approaches to Problem Solving

- Specialized algorithms
  - Costly to design, implement and extend

Approaches to Problem Solving

- Specialized algorithms
  - Costly to design, implement and extend
- Declarative methodology
  1. Choose a problem solving framework (what is my language?)
  2. Model the problem (what is a solution?)
     - Define variables
     - Define constraints
  3. Solve it (with an off-the-shelf solver)

Approaches to Problem Solving

- Specialized algorithms
  - Costly to design, implement and extend
- Declarative methodology
  1. Choose a problem solving framework (what is my language?)
  2. Model the problem (what is a solution?)
     - Define variables
     - Define constraints
  3. Solve it (with an off-the-shelf solver)
- Pros of Declarative methodology
  - Specification of the problem is all we need to solve it!
  - Fast development and easy maintenance
  - Often better performance than ad-hoc techniques

About CPS

- Problem solving frameworks
  - Constraint Programming (CP)
  - Linear Programming (LP)
  - Propositional Satisfiability (SAT)
- For each of these frameworks
  - Modeling techniques
  - Inner workings of solvers