DM811 (5 ECTS - 1st Quarter)
Heuristics for Combinatorial Optimization

[Heuristikker og lokalsøgningsalgoritmer for kombinatorisk optimering]

Marco Chiarandini
lektor, IMADA
www.imada.sdu.dk/~marco/DM811

Combinatorial Optimization

Combinatorial optimization problem

Given • a finite set $N = \{1, 2, \ldots, n\}$ of objects • weights $w_i \; \forall i \in N$ • constraints that define feasible subsets of the objects

Find a minimum weight feasible subset

Combinatorial Optimization

Combinatorial optimization problem

Given • a finite set $N = \{1, 2, \ldots, n\}$ of objects • weights $w_i \; \forall i \in N$ • constraints that define feasible subsets of the objects

Find a minimum weight feasible subset

Examples:

- Shortest path
- Minimum spanning tree
- Matching
- Max-flow

Combinatorial Optimization

Combinatorial optimization problem

Given • a finite set $N = \{1, 2, \ldots, n\}$ of objects • weights $w_i \; \forall i \in N$ • constraints that define feasible subsets of the objects

Find a minimum weight feasible subset

Examples:

- Shortest path
- Minimum spanning tree
- Matching
- Max-flow

Others are NP-hard:

- finding shortest/cheapest tours (traveling salesman, TSP)
- finding models of propositional formulae (SAT)
- finding variable assignments satisfying constraints (CSP)
- partitioning graphs or digraphs
- coloring graphs
- ...

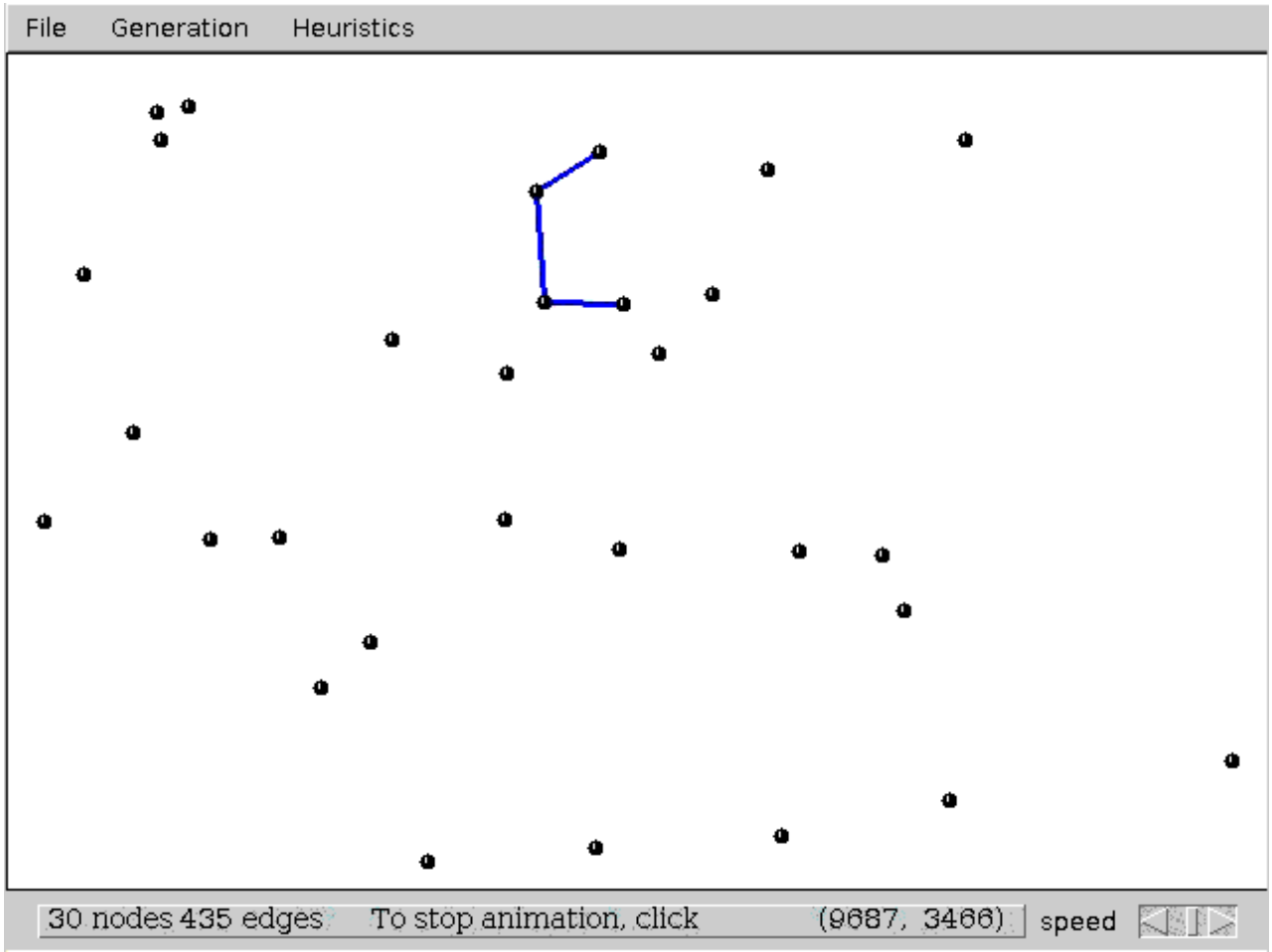
Heuristic Solutions

How can we solve NP-hard problems?

- get inspired by theories on problem-solving in human mind:
  - heuristics, common sense rules
  - trial and error
- and by apparent simplicity of processes in nature
  - simulated annealing
  - evolutionary theory

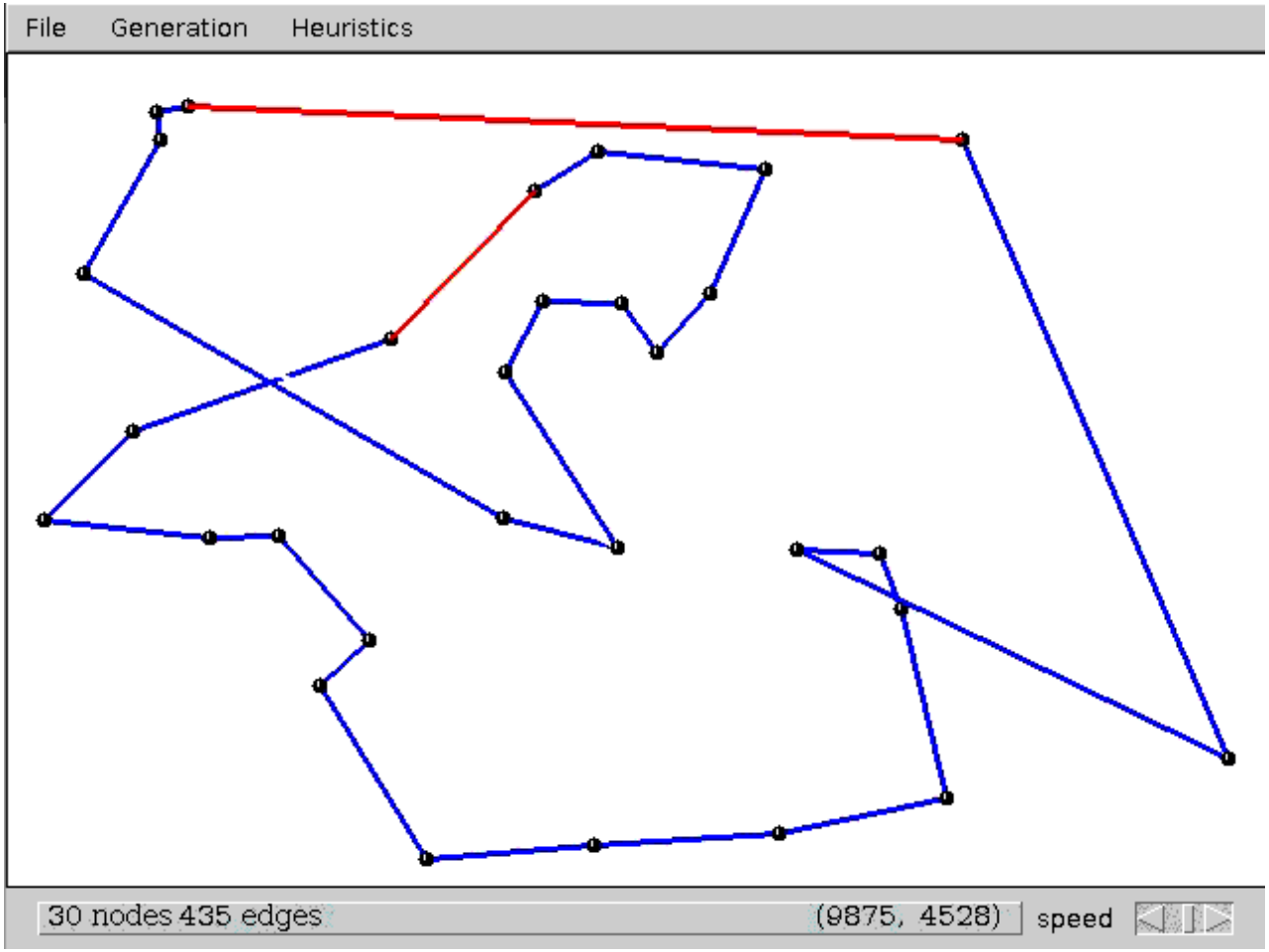
Heuristic algorithms: compute, efficiently, good solutions to a problem (without caring for theoretical guarantees on running time and approximation quality).

Construction Heuristics



Extend the path always going to the nearest neighbor

Local Search



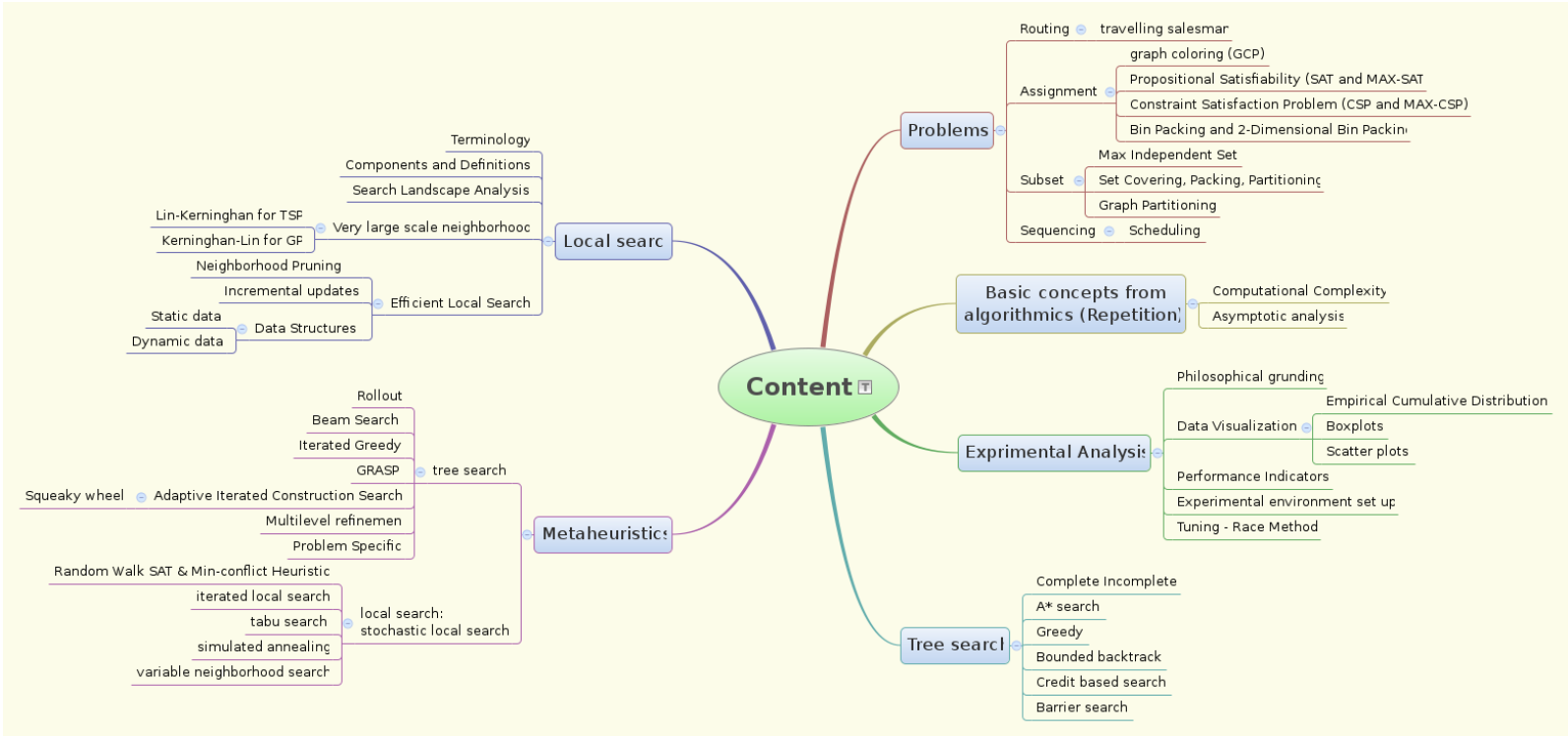
Change two edges of the tour with two new ones

Metaheuristics

- Simulated Annealing
- Iterated Local Search
- Tabu Search
- Variable Neighborhood Search
- Guided Local Search
- Evolutionary Algorithms
- Ant Colony Optimization

Aims & Contents

- design heuristic algorithms
- implement the algorithms
- assess the programs
- describe with appropriate language
- look at different problems



Course Formalities

Prerequisites:	✓ DM507 - Algorithms and data structures ✓ DM502, DM503 - Programming I and II
Credits:	5 ECTS
Language:	English
Classes:	$2h \times 10$ intro phase $+$ $2h \times 7$ training phase
Material:	slides + pointers to literature + starting code

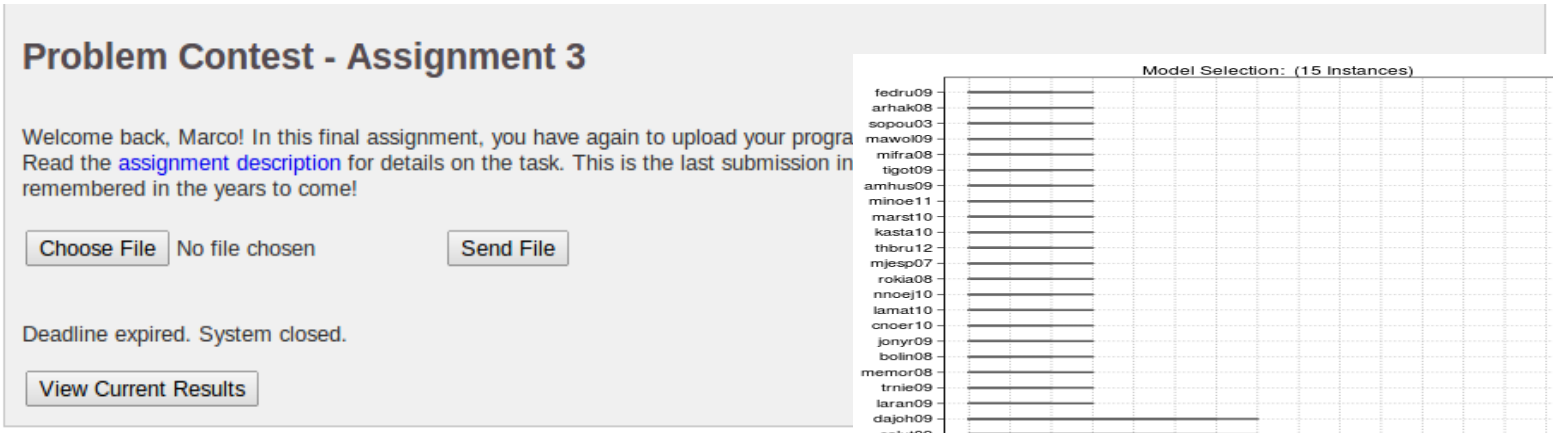
Assessment (5 ECTS)

Three obligatory assignments (individual):

Design, implement and analyse experimentally heuristics for a given problem

Deliverables: program + written report

- 1st and 2nd assignments: Pass/Fail with feedback and internal censor
- 3rd assignment (includes the work in 1st and 3rd): grade with external censor (Performance matters)



DM811 (5 ECTS - 1st Quarter)
Heuristics for Combinatorial Optimization

[Heuristikker og lokalsøgningsalgoritmer for kombinatorisk optimering]

Marco Chiarandini
lektor, IMADA
www.imada.sdu.dk/~marco/DM811