Combinatorial optimization problems:
⇒ select a “best” configuration or set of discrete parameters
to achieve some goals.

Examples:
- Shortest path
- Minimum spanning tree
- Matching
- Max-flow
Combinatorial optimization problems:
⇒ select a “best” configuration or set of discrete parameters to achieve some goals.

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

Others are NP-hard:
- finding shortest/cheapest round trips
- finding models of propositional formulae
- finding variable assignment which satisfy constraints
- partitioning graphs or digraphs
- coloring graphs
- partitioning, packing, covering sets
- ...
How to solve NP-hard problems?

- Exact methods (mixed integer programming, dynamic programming, dedicated branch and bound, constraint programming)
- Heuristic methods

**Heuristic:**

⇒ A common sense rule (or set of rules) intended to increase the probability of solving some problem.
Content of the Course

Metaheuristic

- Construction Heuristics
  - Greedy algorithms,
  - Max-regret algorithms

- Local Search
  - Trial and error by means of small changes:
    - k-exchange, k-flip, swap
Content of the Course

Construction Heuristics
Local Search
Content of the Course

Local Search
Metaheuristics

- Tabu Search

- Evolutionary Algorithms
  - Probabilistic Iterative Improvement
  - Simulated Annealing
  - Iterated Local Search
  - Guided Local Search

- Evolutionary Algorithms
  - Adaptive Greedy
  - Ant Colony Optimization
  - ...

Aims of the Course

To provide knowledge on:

- the general solution methods,
- their successful applications on special problems and
- their assessment

Let the student get acquainted with all phases of the problem solving process:

- understand the problem
- design a solution method
- implement the method
- assess the programs
- describe the algorithms
Work Load

During the Course (30 hours of lectures, 4 hours per week)

▶ Revision of the *concepts* treated in class
▶ *Active* participation at the discussions on case studies during the lectures
▶ Weekly *implementation* and *experimentation* assignments in form of competition

Final Assessment (7.5 ECTS)

▶ Individual project:
  Comparison of metaheuristics on an optimization problem
  ▶ Deliverable: a written report and a functioning program
  ▶ The project is *meant to assess the ability to apply*
Course prerequisites

- MM02 (recommended)
- DM02 (algorithms and data structures)
- Programming experience

Text book:


Alternatively, a collection of articles: “Notes for DM63”.

Further literature:

- Articles
- Lecture notes
DM63 (7.5 ECTS)
Heuristics for Combinatorial Optimization
(Heuristikker til løsning af kombinatoriske optimeringsproblemer)

Marco Chiarandini, post doc.
marco@imada.sdu.dk
http://www.imada.sdu.dk/Courses/DM63

IMADA
University of Southern Denmark