

# DM63 - Heuristics for Combinatorial Optimization Problems – Weekly Notes

## Lecture 1, Fall 2006

---

### Lecture September 4

We gave introductory notes to the organization of course. The reference web-site is:  
<http://www.imada.sdu.dk/marco/DM63/>

Due to the absence of the lecturer the next Lecture will be Thursday 21 September at 10.15.

We introduced combinatorial optimization and the related terminology, which will be adopted during the course. We then recalled the concepts of computational complexity and moved to view the methods for the solution of combinatorial optimization problems.

The second part of the lecture was dedicated to Construction heuristics for the Traveling Salesman Problem. The first Task of the Competition has been launched (see below).

Finally, we review few rules of Software Development which may be useful in the implementation of heuristic programs and suggested the use of a framework for the design of Local Search algorithms.

### Bibliographical Notes

An introduction to heuristics is given in the chapter “Informed Search and Exploration” (through page 110) of the course Notes.

The construction heuristics for TSP are described in the article “Fast Algorithms for Geometric Traveling Salesman” (pages 387–399) of the course Notes.

A rapid re-reading of Chapters 3, 16, 35 from “Introduction to Algorithms” by Cormen et al. from the course DM02 is recommended.

More on the Local Search framework is given in the article “EasyLocal++: An object-oriented framework for flexible design of local search algorithms” available from the Course Section Literature.

Web links referred during the lecture have been added in the Course Section Literature.

### Competition – Task 1

Implement basic versions of three construction heuristics for TSP.

One heuristic at choice from the category *Heuristics that Grow Fragments*

- Nearest neighborhood heuristics (NN)
- Double-Ended Nearest Neighbor heuristic (DNN)
- Multiple Fragment heuristic (*greedy*)

One heuristic at choice from the category *Heuristics that Grow Tours*

- Nearest Addition (NA)

- 
- Farthest Addition (FA)
  - Random Addition (RA)
  - Nearest Insertion (NI)
  - Farthest Insertion (FI)
  - Random Insertion (RI)
  - Clarke-Wright savings heuristic (`savings`)

One heuristic at choice from the category *Heuristics based on Trees*

- Minimum span tree heuristic (`MST`)
- Christofides' heuristics (`chris`)
- Fast recursive partitioning heuristic (`part`)

Resolve any tie decision in the construction process by random choices. For details on basic algorithms for graphs use the book "Introduction to Algorithms" by Cormen et al.

The test instances and the code implementing a framework for construction heuristics and local search algorithms are available from the Course Section Resources. The code contains the functions for reading the instances.

Read the documentation under the Course Section Competition for the modalities of submission. The program must run with an additional option: `-c NAME_HEURISTIC`. The `NAME_HEURISTIC` for each construction heuristic is given between parentheses above. Write in the README file which heuristics have been implemented.

The deadline is at 23.59 of Wednesday 20 September.