Department of Mathematics and Computer Science
University of Southern Denmark, Odense

May 22, 2019
Marco Chiarandini

DM865 - Heuristics and Approximation Algorithms

Obligatory Assignment – Part 2, Spring 2019

Deadline: Code: Thursday 23rd of May at midnight;
Report: Monday 27th of May at midnight.

- This is the *second obligatory assignment* of Heuristics and Approximation Algorithms. It will contribute to the final assessment.
- The assignment has to be carried out in pairs or individually. Groups of larger sizes are not allowed. Individual participation is discouraged.
- The submission is electronic via `http://valkyrien.imada.sdu.dk/DOApp/`.
- You have to hand in:
  - the source code of your implementation of a metaheuristic solver. Submit all your files in a `.tgz` archive. Your files must comply with the requirements listed in this document.
  - a report that describes the work you have done and presents the results obtained. The document should not exceed 10 pages and must be in PDF format. You cannot list source code, in case use pseudocode. You can write in Danish or in English.
- Changes to this document after its first publication on April 23 may occur. They will be emphasized in color and if they are major they will be announced via BlackBoard. It is however recommended not to download this document but to read it from the course web page.
- The project is a continuation of Part 1 and it assumes all material (project descrition and code) of that part.
- You will receive feedback on your Part 1 and it is expected that you take the feedback into account while working on Part 2.
- You are allowed to reuse code from Part 1, including, the starting package containing the instances and the code to read them is available at this link:

  `https://github.com/DM865/CVRP`

- Read all this document before you start to work.

Requirements

You are asked to carry out the following tasks:

1. Design and implement metaheuristic algorithms for the CVRP defined in Part 1. All algorithm components like construction heuristic and local search developed in the context of Part 1 can be reused in this assignment.
2. Undertake an experimental analysis where you configure and compare the algorithms from the previous point. It is a minimal requirement that you show an improvement with respect to the results that your group achieved in Part 1.
3. Describe the work done in a report of at most 10 pages. The report must at least contain a description of the best algorithm designed and the experimental analysis conducted. The level of detail must be such that it makes it possible for the reader to reproduce your work.
4. Report the results of the best algorithms on the test instances made available in a table like Table 1.
5. Submit your best algorithm in the upload page. The programs will be run on a 64-bit machine with Ubuntu Linux, equivalent to those in the terminal room. A time limit of **240 seconds** will be imposed.

Instance	Your Best Results from Part 1		Your Best Metaheuristic	
	k	cost	k	cost
CMT01				
CMT02				
CMT03				
CMT04				
CMT05				
CMT06				
...				
...				

Table 1: The table shows the median results from 5 runs per instance of the best heuristics designed. The time limit was set to 240 seconds on an Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz with 16 GB RAM running Ubuntu 16.04.

Remarks

Remark 1 The metaheuristics designed can be any among those encountered in the lectures. More specifically, you can choose one or more algorithm templates among those from the following groups treated in class:

- stochastic local search and local search based metaheuristics
- construction based metaheuristics
- population based metaheuristics

A few, well thought algorithms are better than many naive ones. However, it is expected that you provide a comparison among different attempts in your report.

Remark 2 You should briefly describe the algorithmic components you use, even if they are already described in Part 1.

Remark 3 This is a list of factors that will be taken into account in the evaluation:

- correctness in the classification of the algorithms, that is, using the right names for the metaheuristics described
- quality of the final results;
- level of detail of the study;
- complexity and originality of the approaches chosen;
- organization of experiments that guarantees reproducibility of conclusions;
- clarity of the report;
- presence of the analysis of the computational costs involved in the main operations of the local search.
- effective use of graphics in the presentation of experimental results.

Remark 4 If you search on Internet, the literature on metaheuristics for vehicle routing problems is vast but not every article is relevant. The following are relevant articles for local search design:

- Clarke, G., and J. W. Wright. "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points." Operations Research 12, no. 4 (1964): 568-81. `http://www.jstor.org/stable/167703.`
- Ann Melissa Campbell and Martin Savelsbergh, Efficient Insertion Heuristics for Vehicle Routing and Scheduling Problems, Transportation Science, 38(3), 369-378, 2004, `http://dx.doi.org/10.1287/trsc.1030.0046`,
- G. Kindevater and M. Savelsbergh, "Vehicle routing 2 - handling side constraints," tech. rep., School of Industrial and Systems Eng., Georgia Institute of Tech., 1995.

and the following for metaheuristics:

- David Pisinger, Stefan Ropke: A general heuristic for vehicle routing problems. Computers & OR 34(8): 2403-2435 (2007)
- Prodhon C., Prins C. (2016) Metaheuristics for Vehicle Routing Problems. In: Siarry P. (eds) Metaheuristics. Springer, Cham