Implementation exercises for the course

Heuristic Optimization

Dr. J. Dubois-Lacoste, Dr. T. Stützle
Université Libre de Bruxelles — 2015

Implementation exercise sheet 1

Implement iterative improvement algorithms for the linear ordering problem (LOP). Information on the LOP is provided in the accompanying introduction to the implementation exercise. Apply the implemented algorithms to all the instances that are provided through the lecture's webpage at `http://iridia.ulb.ac.be/~stuetzle/Teaching/HO/`. On the lecture's webpage also a basic `C++` code can be downloaded that allows to (i) read an instance, (ii) compute the objective function value, and (iii) generate random permutations.

Exercise 1.1 Implementation, deadline April 10, 2015, 23:59

Implement iterative improvement algorithms with a

- first-improvement and another with a
- best-improvement

pivoting rule for each of the three neighborhoods transpose, exchange, and insert. For each of these three neighborhoods implement incremental updates of the evaluation function value.

As a starting solution for iterative improvement, consider two possibilities. The first is to generate a random permutation, that is, to use the method "Uninformed Random Picking" (see slides of lectures). The second is to use the heuristic of Chenery and Watanabe (CW) — see the slides on the implementation task for details.

1. Apply the six resulting iterative improvement algorithms (all combinations of the two pivoting rules and the three neighborhoods) once to each instance. Do these experiments once using a random initial solution and once using as initial solution the one obtained by the CW heuristic (this results in 12 different combinations of starting heuristic and iterative improvement algorithm). Compute the following statistics for each of the 12 algorithms:
   - average percentage deviation from best known solutions;
   - total computation time across all instances.
2. Determine by means of statistical tests (in this case, the Student t-test or the Wilcoxon test), whether there is a statistically significant difference between the solutions generated by the different perturbative local search algorithms.
   Note: For applying the statistical test, the `R` statistics software can be used. The system is downloadable from `http://www.r-project.org/`. A short introduction to the most important commands for executing the tests will be given in the introduction into the implementation exercise on Feb. 25.

Exercise 1.2 Implementation, deadline April 10, 2015, 23:59

Implement a variable neighborhood descent (VND) algorithm. In this algorithm, consider the two possible (reasonable) orderings of the neighborhood relations:

- transpose, exchange, insert
- transpose, insert, exchange

Implement the VND algorithms only for the iterative first-improvement algorithms.

1. Compute the following statistics for each algorithm using as initial solution the heuristic of Chenery and Watanabe.
   - average percentage deviation from the best known solutions;
   - total computation time across all instances.

2. Apply again the statistical tests to compare the solution quality reached by the two VND algorithms.

Additional information on the implementation exercise:

- Recall that the successful completion of the implementation task is a pre-condition for passing the examination.
- Every student solves the implementation exercises independently of others and sends (i) a report in pdf format that shortly explains the implementation, reports the above requested results, and gives a short interpretation of the overall results (e.g. impact of initial solution on quality of local optima, relative solution quality reached in different neighborhoods, computation times of first- vs. best-improvement, improvement incurred by VND over simple neighborhoods) results of statistical tests); (ii) the source code of the implementation; and (iii) a spreadsheet or textfile of the raw data that were used for statistical testing to `jeremie.dubois-lacoste@ulb.ac.be`. The source code is to be collected into a tar or a zip file. The source code needs to compile without errors and when executed on the instance specified in the slides to the implementation exercise produce the requested output.
- As programming language, you may use `C`, `C++`, or `Java`.
- Please take care that the code is well documented and mention the exact commands for compilation and execution on the example instance.
- The implementation for this implementation exercise sheet will be re-used for the second implementation exercise sheet.

Deadline for the implementation exercise:

- **April 10, 2015 at 23:59**