Implementation exercises for the course

Heuristic Optimization

Dr. F. Mascia, Dr. T. Stützle
Université Libre de Bruxelles — 2014

Implementation exercise sheet 1

Implement iterative improvement algorithms for the permutation flow-shop scheduling problem (PFSP) with weighted tardiness objective. Information on the PFSP is provided in the accompanying introduction to the implementation exercise. Apply the implemented algorithms to all instances of 50, 60, 70, 80, 90, and 100 jobs available from the lecture's webpage at `http://iridia.ulb.ac.be/~stuetzle/Teaching/HO/`

Exercise 1.1 Implementation, deadline April 11, 2014, 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.

As a starting solution for iterative improvement, consider two possibilities. The first is to generate a random job permutation, that is, to use the method "Uninformed Random Picking" (see slides of lectures). The second is to use an insertion heuristic, in particular the slack heuristic (see the introduction to 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 on each of the instances. Do these experiments once using random initial solutions and once using as initial solution the one obtained by the slack heuristic (resulting in 12 different combinations of starting heuristic and iterative improvement algorithm). Compute the following statistics for each combination of algorithms and initial candidate solutions and for each number of machines.
   - average percentage deviation from the best known solutions;
   - average computation time for each number of machines.
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. 26.

Exercise 1.2 Implementation, deadline April 11, 2014, 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 combination of algorithms and generation of initial candidate solution and for each number of machines.
   - average percentage deviation from the best known solutions;
   - average computation time for each number of machines.
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 document in pdf format that shortly explains the implementation and reports the above mentioned tasks (averages, standard deviations, and results of statistical tests) and (ii) the source code of the implementation to `fmascia@ulb.ac.be`. The source is to be collected into a tar or a zip file. It needs to compile without errors.
- 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.
- The implementation for this implementation exercise sheet will be re-used for the second implementation exercise sheet.

Deadline for the implementation exercise:

- **April 11, 2014 at 23:59**