Implementation exercises for the course

Heuristic Optimization

Federico Pagnozzi, Dr. T. Stützle
Université Libre de Bruxelles — 2021

Implementation exercise sheet 1

Implement iterative improvement algorithms for the permutation flow-shop scheduling problem (PFSP) with the weighted sum of completion times objective. Information on the PFSP is provided in the accompanying introduction to the implementation exercise. Apply the implemented algorithms to all instances of 50 and 100 jobs available on teams. As variance reduction technique for the experiments make sure that the algorithms on a same instance start from the same initial solution. This can be ensured by using for each of the executions on a same instance the same random number seed.

Exercise 1.1 Implementation, deadline April 14, 2021, 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 *simplified RZ* 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 simplified RZ 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 jobs.
    - average percentage deviation from the best known solutions;
    - average computation time for each number of jobs.
2. Determine by means of statistical tests (in this case, the paired 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.
   With the results of this experimental comparison answer the questions (i) which initial solution is preferable, (ii) which pivoting rule generates better quality solutions and which is faster, (iii) which neighborhood generates better quality solution and what computation time is required to reach local optima.
   Justify your answers concisely. To answer these questions, do not simply do all possible comparisons, which would be 66, but pair the comparisons appropriately targeting the specific question. For example, for the first question one would do six comparisons, one for each where the algorithms differ between the initial solutions but all other features remain the same in a comparison. More concretely, this means to compare an algorithm using uniform random picking, first-improvement and the insert neighborhood to an algorithm that uses simplified RZ, first-improvement and the insert neighbourhood.
   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 was given in the introduction to the implementation exercise.

Exercise 1.2 Implementation, deadline April 14, 2021, 23:59

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

- transpose, exchange, insert

- transpose, insert, exchange

Implement the VND algorithms only for the iterative first-improvement algorithms. Consider as starting solutions the ones obtained by the simplified RZ heuristic.

1. Compute the following statistics for each combination of algorithms and generation of initial candidate solution and for each number of jobs.
   - average percentage deviation from the best known solutions;
   - average computation time for each number of jobs.
   - percentage improvement over the usage of a single neighborhood, in particular, the exchange and the insert one.
2. Apply again the statistical tests to compare the solution quality reached by the two VND algorithms.
3. With the results of the experiments quantify the improvement of the VND over a local search in a single neighborhood and examine which ordering of the neighborhoods in the VND is preferable.
   Justify your answers concisely.

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 returns (i) a document in pdf format that shortly explains the implementation, reports the above mentioned tasks (averages, standard deviations, and results of statistical tests), interprets concisely the observed results, and answers the above posed questions and (ii) the source code of the implementation. The source code needs to compile without errors. The document and the source are to be collected into a tar or a zip file.
- As programming language, you may use `C`, `C++`, `python` or `Java`. The sample routines are available only in `C++`.
- 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 14, 2021 at 23:59**