Implementation exercises for the course

Heuristic Optimization

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

Implementation exercise sheet 2

Implement two stochastic local search algorithms for the permutation flow-shop problem (PFSP) with weighted tardiness objective (PFSP-WT), building on top of the perturbative local search methods from the first implementation exercise. Apply your algorithms to the same instances as in the first implementation exercise.

The SLS algorithms can be based on either simple, hybrid, or population-based methods chosen among the ones described in the lectures. The two SLS methods chosen must belong to two different classes. For example, the first one could be a tabu search (simple SLS) and the second one could be an ACO algorithm (population-based). To get inspiration for ways how to generate solutions, operators etc. you may consider algorithms that have been proposed in the literature for the same or similar problems. In fact, the PFSP-WT has not been very frequently treated in the literature. For inspiration you may also check papers on the related PFSP with total tardiness objective or the objective of minimizing total flowtime. Some references are given below.

The two algorithms implemented should be evaluated on all instances provided and be compared using run-time distributions and statistical tests. The implementation should make use of the iterative improvement algorithms from the first implementation exercise.

Please note that the experimental comparison should be among implementations that are coded in the same programming language, the algorithms should be compiled using the same compiler with same compiler flags, and the experiments should be run on the same computer under same conditions (e.g. no other compute or memory intensive programs are running on the same machine). In other words, the observed differences should be attributable to differences in the algorithms and not to differences in the experimental conditions.

Exercise 2.1 Implementation, deadline May 16, 2014

1. Run each algorithm 5 times with different random seed on each instance. Instances are available from `http://iridia.ulb.ac.be/~stuetzle/Teaching/HO/`. As termination criterion, for each instance, use the maximum computation time it takes to run a full VND (implemented in the previous exercise) on the same instance and then multiply this time by 100 (to allow for long enough runs of the SLS algorithms).

2. Compute for each of the two SLS algorithms and each instance the mean relative percentage deviation from the best known solutions.

3. Produce correlation plots of the mean relative percentage deviation for the two SLS algorithms (see lectures).

4. Determine, using statistical tests (in this case, the Wilcoxon test), whether there is a statistically significant difference between the mean relative percentage deviations reached by the two algorithms. Note: For applying the statistical test, the R statistics software can be used. The software can be download from `http://www.r-project.org/`.

5. Measure, for each of the two implemented SLS algorithms on 6 instances, qualified run-time distributions to reach sufficiently high quality solutions (e.g. the best-known solutions available or some solution value close to the best-known one such as 0.5%, 1%, or 2% above the best-known solution). Measure the run-time distributions across 25 repetitions using a cut-off time of 10 times the termination criterion above. As the instances take in each case the first instance of size $50, 60, \ldots, 100$.

6. Produce a written report on the implementation exercise:

   - Please make sure that each implemented SLS algorithm is appropriately described and that the computational results are carefully interpreted. Justify also the choice of the parameter settings and the choice of the iterative improvement algorithm for the hybrid SLS algorithm.
   - Present the results as in the previous implementation exercise (tables, statistical tests).
   - Present graphically the results of the analysis of the run-time distributions.
   - Interpret appropriately the results and make conclusions on the relative performance of the algorithms across all the benchmark instances studied.

Some recommended starting points for literature search:

The articles below could be a starting point to explore the available literature for examples of how the various existing SLS methods can be adapted to the PFSP-WT.

1. Eva Vallada, Rubén Ruiz, Gerardo Minella. Minimising total tardiness in the m-machine flowshop problem: A review and evaluation of heuristics and metaheuristics. *Computers & OR*, 35(4):1350–1373, 2008.
2. Rubén Ruiz and Thomas Stützle. An Iterated Greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. European Journal of Operational Research 187(3):1143–1159, 2008.
3. Eva Vallada and Rubén Ruiz. Genetic algorithms with path relinking for the minimum tardiness permutation flowshop problem. *OMEGA*, 38(1–2):57–67, 2010.

Additional information on the implementation exercise:

- Recall that the completion of the implementation task is a pre-condition for the examination.
- Every student sends (i) the above mentioned report in pdf format and (ii) the source code of the implementation to `fmascia@ulb.ac.be`. Please send one single archive (zip or tar.gz) with all implementations. A `README` file should explain how to compile and run the code. As programming language, you may use `C`, `C++`, or `Java`. Please take care that the programs are reasonably documented and mention the exact commands for compilation in GNU/Linux (for example Ubuntu Linux).
- For each of the implemented SLS algorithms, provide a concise description of at least one page of the main features of the algorithm (e.g. choice of initial solution, iterative improvement algorithm, crossover operator in case of MA, perturbation operators, etc.)
- Please take care that the programs are reasonably documented.
- The articles mentioned above are available from the lecture's webpage in pdf format.

Deadline for the implementation exercises:

- **May 16, 2014**