Implementation exercises for the course

Heuristic Optimization

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

Implementation exercise sheet 2

Design and implement two stochastic local search algorithms for the linear ordering problem (LOP), 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 (a simple SLS method) and the second one could be an ACO algorithm (a population-based SLS method). To get inspiration for ways how to generate solutions, operators etc. you may consider algorithms that have been proposed in the literature. In the article that is mentioned below and accessible at ULB/VUB you find a recent comprehensive review of heuristic algorithms for the LOP.

The two algorithms implemented should be evaluated on all instances of size 150 and be compared using statistical testing. In addition, on few instances the comparison should also be done by measuring run-time distributions. 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 22, 2015

1. Run each algorithm once on each instance of size 150. Instances are available from `http://iridia.ulb.ac.be/~stuetzle/Teaching/HO/`. As termination criterion, for each instance use the average computation time it takes to run a full VND (implemented in the previous exercise) on the same instance size 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 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 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 three 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 0.1%, 0.25%, or 1% worse than 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 the first three instances of size 150.

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.

Recommended starting point for literature search:

The article below is the best starting point for exploring the available literature for examples of how the various existing SLS methods can be adapted to the LOP. We suggest to continue from this article the search for specific examples of SLS algorithms that have been applied to the LOP. Note that the article below and most other journal articles on the LOP can be downloaded from the respective journals at ULB or through the EZ-proxy.

- Rafael Martí, Gerhard Reinelt, and Abraham Duarte. *A benchmark library and a comparison of heuristic methods for the linear ordering problem*. Computational Optimization and Applications. 51(3):1297-1317 (2012).

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 `jeremie.dubois-lacoste@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 22, 2015**