Programming assignment 1

Heuristic Optimization Techniques, 2017WS

October 9, 2017

Single-solution-based Metaheuristics [30 points]

The first programming assignment is to develop your *own* single-solution-based metaheuristics for the K-page crossing number minimization problem. The subtasks for this exercise are:

1. Develop a deterministic construction heuristic.
2. Develop a randomized construction heuristic.
3. Develop a framework for simple local search which is able to deal with
   - different neighborhood structures
   - different step functions (first-improvement, best-improvement, random)
4. Develop at least two different neighborhood structures.
5. Develop a VND framework which uses the above neighborhood structures.
6. Implement one of the following metaheuristics:
   - General Variable Neighbourhood Search (GVNS) which uses your VND
   - Simulated Annealing (SA)
   - Tabu Search (TS)
7. Run experiments and compare all your algorithms on the given instances:
   (a) determinstic and randomized construction heuristic
   (b) Use the solution of the deterministic construction heuristic to test the other implementations:
      i. Local search for each of your neighborhood structures using each of the three step functions (at least 6 different algorithms)
      ii. VND
      iii. GVNS or SA or TS
8. **Use incremental evaluation whenever possible**. Test your approaches once with incremental evaluation and once without incremental evaluation and compare the performance and running times.

9. Write a report containing the description of your algorithms and the experiment results (see the general problem description)

For the development and the report consider the following points:

- How is your solution represented?
- 1st task: Do you construct the spine order and the edge partitioning separately or in one step? Argue on the expected benefits of your approach.
- 2nd task: How do you generate different solutions? Which parts of your algorithm can be reasonably randomized and how can you control the degree of randomization?
- 1st + 2nd task: Does randomization improve the generated solutions?
- What are the building blocks of the metaheuristics you implemented?
- Can subsequent (possibly random) moves in your neighbourhood structures reach any solution in the search space?
- local search: How many iterations does it take to reach local optima? What does this say about your neighbourhood structures?
- How does incremental evaluation work for your neighbourhood structures
- VND: Does the order of your neighbourhoods affect the solution quality?

Hand in your report via TUWEL until *2017-11-20, 23:55.* For further questions send an e-mail to: `heuopt@ac.tuwien.ac.at`