Evolutionary Computation
Practical Assignment 2

1 MLS, ILS, and GLS for Graph Bipartitioning

The task of this practical assignment is to implement and experimentally compare multi-start local search (MLS), iterated local search (ILS), and genetic local search (GLS) for a graph-bipartitioning (GP) problem. The goal is to divide the set of vertices of a graph in two equally sized subsets such that the number of edges that connect two vertices belonging to two different subsets is minimized.

Local search algorithms iteratively change a solution until no better solution is found in the neighborhood of the current solution. The local search algorithm used is the Fiduccia-Mattheyses (FM) heuristic (see lectures slides and FM paper). MLS, ILS, and GLS are metaheuristic algorithms that improve the performance of the local search algorithm.

1. *Multi-start local search* simply restarts local search from a set of randomly generated initial solutions. The best local optimum found is returned as final solution.
2. *Iterated local search* mutates the current solution found by local search and applies local search from the mutated solution. When the new local optimum is better than the current one, ILS continues its search from this new local optimum, else it simply returns to the previous local optimum. The size of the mutation is a crucial factor in ILS. If the mutation is too small, the local search will return to the local optimum it has just mutated. If the mutation is too large there will be no correlation between the new local optimum and the one that was mutated. ILS is then reduced to a MLS algorithm.
3. *Genetic local search* keeps a population of the best solutions found and generates new starting solutions for local search by recombining solutions from the population.

2 Experimental study

Set up an experimental study that investigates the following algorithms. Note that to experimentally compare two algorithms you need to run the algorithms for at least 25 times each and compare the mean or median results of these 25 runs with each other. Use a statistical significance test (t-test or Wilcoxon-Mann-Whitney test) to check whether the observed differences are indeed statistically significant. Box plots are a useful way to show results (see lecture slides).

1. Implement MLS with the FM local search. In one pass of the FM heuristic all nodes are swapped to the other partitioning in a greedy way. We compare all algorithms here based on the same number of FM passes, specifically 10000 passes. Local search searches for the nearest local optimum so you need to apply multiple FM passes in succession to reach a local optimum (typically 5 to 10 FM passes), so MLS will visit approximately 1000 to 2000 local optima (this is a rough guess !). Measure and report also the computational time required.

2. Investigate the impact of different mutation/perturbation sizes for ILS with the FM local search. Start with the smallest possible. Keep investigating larger perturbation sizes until the performance of ILS drops. Measure the average amount of times that the newly obtained local optimum is equal to the local optimum before perturbation (meaning that the search has not left the region of attraction of the perturbed local optimum). Are the ILS versions statistically better/worse than MLS ? Are the results obtained with the different mutation/perturbation sizes statistically different from each other ?

3. Implement and test GLS (using the FM local search) with population size 50. The specific genetic algorithm is an incremental (or steady state) GA where there is no explicit notion of generations: each iteration two parents are randomly selected, use uniform crossover to generate one child, do FM local search on the child, let this optimized child compete with the worst solution in the population, if it is better or equal than it replaces the worst solution. Are the GLS results statistically better/worse than MLS or ILS ?

4. Select your own topic to investigate. Clearly specify the research goal and discuss the results. Possible topics might be different selection mechanisms, different population sizes, not allowing duplicate solutions in the population with GLS, combining ILS with GLS by adding mutation/perturbation (how to combine them ?), not performing all passes of FM to reach a local optimum, use more FM passes than 10000, apply an adaptive mutation/perturbation step size in ILS, comparison with other metaheuristics like simulated annealing, ...

   Compare the above algorithms in two ways:

   (a) Compare them on the basis of generating the same number of FM passes (= 10000),

   (b) Compare them on the basis of the same run time.

3 Graph Bipartitioning

- The experiments are tested on a single planar graph of 500 vertices. The graph is specified in a text file with each line giving the id number of the vertex, its coordinates

in the plane, the number of connected vertices, the id numbers of the connected vertices. For instance the line:
1 (0.502987,0.528829) 8 28 102 162 233 360 393 460 500
states that vertex 1 is connected to 8 other vertices with indices 28 102 162 233 360 393 460 500. (note: you do not need the coordinates for this assignment).

- Solutions to the GP problem are represented by binary strings $x_1 \ldots x_\ell$ with ℓ the number of nodes in the graph and $x_i \in \{0,1\}$ specifying to which of the two partitions the node i belongs. Note that the number of ones and zeroes in each solution string needs to be equal to $\ell/2$.
- A vertex swap is simply replacing a 1 by 0, and a 0 by 1 in the binary string.
- Recombination is done by uniform crossover that respects the equality constraint between the number of ones and zeroes:
  1. Compute the hamming distance between the two parents. If the distance is larger than $\ell/2$ invert all bit values of one parent.
  2. Generate a single child that has the same bit value as the parents whenever the two parents have the same value for that vertex.
  3. The positions in the child string where the two parents disagree are randomly filled in under the constraint that the total number of ones and zeroes in each solution needs to be equal to $\ell/2$.

4 Report

Write a report discussing your results and send it to d.thierens@uu.nl. The report should be in **PDF format !!!** The source code should be in SEPARATE, compressed archive file (program.tar.gz). **Do NOT include** your report.pdf file in this archive !

The report should at least contain the following topics:

1. Give a short description of the implemented program.
2. Tabulate and/or plot the experimental results.
3. Discuss your findings.

5 Deadline

Tuesday, March 31, 2020, 9:00 hrs.

6 Questions

Any remaining questions about the assignment can be asked during the break or after each lecture.