CPSC 532d: Assignment 2 — Mar 7, 2003

This assignment covers Modules 5–7. It is due on Tue, Mar 18, 2003.

Problem 1 (Problem solving; 5 marks) Model the GRASP algorithm for SAT described in Example 2.7, p. 82 of the book *adequately* as a GLSM. Specify the machine graph and give pseudo-code for each state type used in your model. Briefly discuss any modelling choices you've encountered and justify your solution.

Problem 2 (Hands-on algorithm evaluation; 10 marks) In this exercise, you will perform a limited empirical evaluation of two high-performance SLS algorithms for SAT, *walksat/tabu* and *novelty+*. Some of the tools you need in this context are provided on the BETA file system in */cs/beta/People/Hoos/Resources/CPSC532D-03/Assg-2/*. You are expected to perform all algorithm runs on the BETA compute clusters, preferably on *sifnos, patmos* or *rhodes*.

These are the tools you will use:

- *walksat -tabu 2 -cutoff 10000 -tries 100* $<$ *test.cnf*: Run *walksat/tabu* with tabu list length $tl = 2$ on SAT instance *test.cnt*, performing 100 independent runs with a cutoff of 10000 variable flips each.
- *walksat -novelty+ -noise 30 100 -wp 1 100 -cutoff 10000 -tries 100* $<$ *test.cnf*: Run *novelty+* with noise $p = 30/100$ and walk probability $wp = 1/100$ on SAT instance *test.cnt*, performing 100 independent runs with a cutoff of 10000 variable flips each.
- *ws2rtd.pl test.out*: Extract RTD data from walksat ouput *test.out*
- *getstats.pl rtd.dat*: Extract basic descriptive statistics from RTD data in *rtd.dat*

Furthermore, you will need a programme for plotting RTDs from RTD data obtained from *ws2rtd.pl*; you can use *gnuplot* on the BETA machines or Excel on other departmental machines.

Finally, you will need to obtain the following SAT benchmark instances from the SATLIB webpage (www.satlib.org):

- *uf250-099.cnf* from the Uniform Random-3-SAT set uf250-1065;
- *logistics.b* from the planning series;
- *g250.15.cnf* from the DIMACS set GCP.

(a) Measure an RTD for *walksat/tabu* with $tl = 1$ on instance *uf250-099.cnf* based on 100 runs and graphically show the correlation between run-length (number of flips, second column of RTD file) and run-time (in CPU seconds, third column of RTD file). Based on this result, is it reasonable to use the number of flips to measure run-length?

(b) Measure RLDs for *walksat/tabu* with $tl = 1$ on instance *uf250-099.cnf* based on 10, 100, and 1000 runs. Show all three RTDs in a semilog plot and compare the three respective values for the mean and median number of flips.

(c) Manually fit an exponential function to the highest resolution RLD from part (b) and estimate the optimal cutoff for static restart. (Explain the method you use and illustrate your results graphically.)

(d) Empirically study the impact of tabu list length on the performance of *walksat/tabu* on *uf250-099.cnf*. Explain and justify your experimental protocol and show and interpret the results. (You do not have to determine an optimal tabu-list length with accuracy.)

(e) Compare the performance of *walksat/tabu* and *novelty+* on *uf250-099.cnf*, *logistics.b*, and *g250.15.cnf* in a fair way. Describe and justify your experimental protocol, show and interpret the results, and discuss potential weaknesses of your experimental analysis.

(f) Explain how the comparative analysis you performed in part (e) can be extended to deal with the entire test-set *uf250-1065*. Devise and describe an accurate experimental protocol for this purpose. (You are not expected to do the actual experiment.)

Problem 3 (Knowledge test; 6 marks)

(a) Are there non-neutral search landscapes in which a gradient walk from a given point is not uniquely defined? Give a proof or (very simple) counter-example.

(b) Give an example for a landscape that has no local minimum other than the global optimum and is yet very hard to search for all of the SLS algorithms covered in the course.

(c) Explain how you would obtain a fitness distance plot for the landscape searched by GSAT on a given SAT instance.

(d) Show a (fictitious) fitness distance plot that indicates an FDC close to zero and explain why in this situation random restarts can still be detrimental to the performance of a given SLS algorithm.