Second DIMACS Challenge

Satisfiability Benchmark Results

GENERAL INFORMATION

Authors: Bart Selman, Henry Kautz, and Bram Cohen

Title: Local Search Strategies for Satisfiability Testing

Name of Algorithm: GSAT

Brief Description of Algorithm:

Incomplete: randomized greedy local search method with restarts and a mixed random walk strategy.

Type of Machine: SGI Challenge (Irix 5.1.1.3)

Compiler and flags used: /usr/bin/cc -O

MACHINE BENCHMARKS

User time for instances:

r100.5	r200.5	r300.5	r400.5	r500.5
0.03	0.58	4.88	30.24	116.98

ALGORITHM BENCHMARKS

Authors' Comments:

GSAT's parameters were tuned by trying various settings for a few tries. The following settings were used. (N is the number of variables in the problem instance.)

- The aim-* instances: weights, averaging, and downwards moves only (Selman and Kautz 1993; "break-out" Morris 1993). Max-flips $5N$, max-tries 20,000.
- The f* instances: Max-flips $10N^2$, max-tries 2, walk ($p = 0.5$).
- The g125* instances: Max-flips 200000, max-tries 10, tabu list (length 20).
- The g250* instances: Max-flips 700000, max-tries 10, tabu list (length 20).
- The ii* instances: Max-flips $10N$, max-tries 10, walk ($p = 0.3$).
- The par* instances Max-flips $100N$, max-tries 20, walk ($p = 0.5$).
- The ssa* instances: Max-flips $100N$, max-tries 50, walk ($p = 0.4$).

We give only results for runs on the satisfiable instances. All timings are in seconds. We also ran experiments with WSAT (walksat) on some of the hardest instances. WSAT implements GSAT's random walk with subtle but significant modifications (see main text). Our WSAT implementation is also more efficient than our GSAT implementation (more "flips" per second). We used a "*" to indicate the WSAT timings.

Results on Benchmark Instances:

Name	Runs (Fail)	Time Min	Time Avg (Std. Dev.)	Time Max	Result
aim-100-2_0-no-1					
aim-100-2_0-no-2					
aim-100-2_0-no-3					
aim-100-2_0-no-4					
aim-100-2_0-yes1-1	10 (1)	0.27	1.96 (2.14)	6.66	yes
aim-100-2_0-yes1-2	10 (0)	0.31	1.6 (1.53)	4.58	yes
aim-100-2_0-yes1-3	10 (0)	0.38	1.09 (0.74)	2.8	yes
aim-100-2_0-yes1-4	10 (0)	0.07	1.54 (1.5)	5	yes
bf0432-007.cnf					
bf2670-001.cnf					
dubois20.cnf					
dubois21.cnf					
f400.cnf	10 (0)	1.25	9.56 (8.9)	26.26	yes
f800.cnf	10 (1)	39.55	699.81 (628.7)	1870.47	yes
f1600.cnf	10 (0)	107.73	1172.93 (947.14)	2632.06	yes
f3200.cnf	10 (0)*	122.29	600.95 (437.29)	1152.81	yes
f6400.cnf	1 (0)*		6268		yes
g125.17.cnf	10 (3)	152.0	264.07 (89.61)	384.12	yes
g125.18.cnf	10 (0)	0.94	1.91 (0.84)	3.66	yes
g250.15.cnf	10 (0)	2.53	4.41 (2.18)	8.99	yes
g250.29.cnf	10 (1)	124.7	1219.88 (841.51)	2396.59	yes
ii32b3.cnf	10 (0)	0.29	0.61 (0.44)	1.78	yes
ii32c3.cnf	10 (0)	0.11	0.27 (0.09)	0.43	yes
ii32d3.cnf	10 (0)	0.84	2.24 (1.47)	5	yes
ii32e3.cnf	10 (0)	0.29	0.49 (0.16)	0.87	yes
par16-2-c.cnf	1(0)*		49920		yes
par16-4-c.cnf					
par32-2-c.cnf					
par32-4-c.cnf					
par8-2-c.cnf	10 (0)	0.04	1.33 (1.31)	3.7	yes
par8-4-c.cnf	10 (0)	0.01	0.2 (0.24)	0.83	yes
pret150_25.cnf					
pret150_75.cnf					
pret60_25.cnf					
pret60_75.cnf					
ssa0432-003.cnf					
ssa2670-141.cnf					
ssa7552-038.cnf	10 (0)	0.92	10.14 (7.58)	23.98	yes
ssa7552-158.cnf	10 (2)	2.5	34.92 (36.64)	117	yes
ssa7552-159.cnf	10 (0)	0.49	3.57 (4.29)	12.51	yes
ssa7552-160.cnf	10 (0)	0.78	3.07 (1.55)	5.46	yes