

Bee-colony optimization for the satisfiability problem in probabilistic logic

All our implementations are running on a cluster consisted of nodes with 2 AMD Opteron 2.1GHz 6272 processors with 48 GB RAM under kernel 2.6.32 x64, gcc version 4.4.3.

For a formula $CP_{\approx 1}(p_2, p_1 \vee p_2 \vee \neg p_3) \wedge CP_{< 0.3}(\neg p_3, \neg p_2 \vee p_3) \wedge CP_{]0}(p_1 \vee (p_1 \wedge p_3), (\neg p_1 \wedge p_2 \wedge \neg p_3) \vee (\neg p_1 \wedge \neg p_2 \wedge p_3))$ input file is:

```
3 3 // Number of letters and number of CP formulas

// Formula A
1 // Number of conjunctions
1 //Number of literals in conjunction
2

// Formula B
3 // Number of conjunctions
1 //Number of literals in conjunction
1
1 //Number of literals in conjunction
2
1 //Number of literals in conjunction
-3

~ // Symbol
1.0000000000 // Value

// Formula A
1 // Number of conjunctions
1 //Number of literals in conjunction
```

```

-3

// Formula B
2 // Number of conjunctions
1 //Number of literals in conjunction
-2
1 //Number of literals in conjunction
3

< // Symbol
0.3000000000 // Value

// Formula A
2 // Number of conjunctions
1 //Number of literals in conjunction
1
2 //Number of literals in conjunction
1 3

// Formula B
2 // Number of conjunctions
3 //Number of literals in conjunction
-1 2 -3
3 //Number of literals in conjunction
-1 -2 3

] // Symbol
0.0000000000 // Value

```

Table 1: The test results for small, randomly generated, examples using Fourier-Motzkin and BCOi methods

N, M, Inst	Elapsed time(s)		
	Polynom	Rational	BCOi
3, 3, 0	0.0000	0.0000	0.0010
3, 3, 1	DIV 0	1.7600	–
3, 3, 2	0.0000	0.0000	0.0010
3, 4, 0	0.0600	0.5600	0.0000
3, 4, 1	0.0000	0.0300	0.0010
3, 4, 2	DIV 0	683.580	–
4, 3, 0	0.2800	1.1100	0.0020
4, 3, 1	26.1900	88.7700	0.0010
4, 3, 2	166.5300	2116.7100	0.0020
4, 4, 0	0.0000	0.0000	0.0020
4, 4, 1	DIV 0	0.3400	–
4, 4, 2	0.1700	0.6800	0.0020
4, 5, 0	MEM	MEM	0.0020
4, 5, 1	MEM	MEM	0.0020
4, 5, 2	MEM	MEM	0.0040
5, 5, 0	MEM	MEM	0.0280
5, 5, 1	0.0200	0.0000	0.0030
5, 5, 2	MEM	MEM	0.0050

Table 2: BCOi results for all test examples

N, L, Inst	Solved	Num. Fun. Eval.	CPU time
10, 30, 0	10/10	10	7.768
10, 30, 1	10/10	10	8.000
10, 30, 2	4/10	11946	1819.987
10, 50, 0	10/10	34	82.972
10, 50, 1	10/10	178	378.609
10, 50, 2	10/10	19	41.336
10, 100, 0	0/10	15000	–
10, 100, 1	10/10	20	208.448
10, 100, 2	10/10	10	14.491
20, 20, 0	10/10	17	3.856
20, 20, 1	10/10	51	16.585
20, 20, 2	10/10	138	59.054
20, 50, 0	10/10	32	44.816
20, 50, 1	10/10	10	19.229
20, 50, 2	9/10	6651	10771.538
20, 100, 0	10/10	120	128.694
20, 100, 1	10/10	56	541.613
20, 100, 2	10/10	10	75.885
20, 250, 0	10/10	360	7571.212
20, 250, 1	0/10	15000	–
20, 250, 2	0/10	15000	–
30, 30, 0	0/10	15000	–
30, 30, 1	10/10	40	18.167
30, 30, 2	10/10	83	53.004
30, 60, 0	10/10	59	153.523
30, 60, 1	10/10	150	408.450
30, 60, 2	10/10	24	74.316
30, 100, 0	0/10	15000	–
30, 100, 1	10/10	1260	4497.525
30, 100, 2	10/10	480	3793.997
30, 250, 0	10/10	10	442.865
30, 250, 1	10/10	62	3369.639
30, 250, 2	0/10	15000	–
40, 40, 0	0/10	15000	–
40, 40, 1	10/10	14	8.922
40, 40, 2	10/10	360	412.283
50, 50, 0	0/10	15000	–
50, 50, 1	10/10	3420	4118.458
50, 50, 2	3/10	11517	23622.824