Appears in A. Balint, A. Belov, M.J.H. Heule, and M. Järvisalo (eds.), *Proceedings of SAT Competition 2013: Solver and Benchmark Descriptions*, volume B-2013-1 of Department of Computer Science Series of Publications B, University of Helsinki 2013. ISBN 978-952-10-8991-6.

Two Pigeons per Hole Problem

Armin Biere
Institute for Formal Models and Verification
Johannes Kepler University Linz

In the newest version of our SAT solver Lingeling we included a simple algorithm for solving large trivially encoded pigeon hole problems. The algorithm is based on cardinality reasoning. More information about the algorithm can be found in our solver description [1].

One phase of the algorithm consists of extracting *at-most-one* constraints, which we extended to extract *at-most-two* constraints too. This extension allowed us to solve the following simple extension of the pigeon hole problem.

> Given h holes, we ask whether it possible to fit $n = 2 \cdot h + 1$ pigeons into these holes, where each hole can fit at most two pigeons.

We submitted a C program gentph.c as benchmark generator, which takes the number of holes as one argument. For each hole there is an *at-most-two* constraint over n pigeons, which is encoded with $(\text{n3}) = n \cdot (n-1) \cdot (n-2)/6$ clauses of length 3. In addition, for each pigeon there is a clause of length n requiring that the pigeon is at least in one hole.

For $h = 6$ holes the problem becomes difficult for standard CDCL solvers. Glucose 2.1 needs 420 seconds, while Lingeling 587f needs 970 seconds, both on an Intel i7-3930K CPU running at 3.20GHz. Lingeling as submitted to this year's competition, but without cardinality reasoning needs 291 seconds. More holes seem to be out of reach. With cardinality constraint reasoning this problem is trivial and can be solved for up to 20 holes instantly.

We list the sizes of these new benchmarks in Table I. Compared to the well-known original pigeon hole benchmarks, with sizes listed in Table II, we observed that the benchmarks become more difficult for a smaller number of variables.

holes h	pigeons n	variables	clauses
1	3	3	4
2	5	10	25
3	7	21	112
4	9	36	345
5	11	55	836
6	13	78	1729
7	15	105	3200
8	17	136	5457
9	19	171	8740
10	21	210	13321
11	23	253	19504
12	25	300	27625
13	27	351	38052
14	29	406	51185
15	31	465	67456
16	33	528	87329
17	35	595	111300
18	37	666	139897
19	39	741	173680
20	41	820	213241

TABLE I
SUBMITTED "TWO PIGEON PER HOLES" BENCHMARKS TPH_h.

holes h	pigeons n	variables	clauses
1	2	2	3
2	3	6	9
3	4	12	22
4	5	20	45
5	6	30	81
6	7	42	133
7	8	56	204
8	9	72	297
9	10	90	415
10	11	110	561
11	12	132	738
12	13	156	949
13	14	182	1197
14	15	210	1485
15	16	240	1816
16	17	272	2193
17	18	306	2619
18	19	342	3097
19	20	380	3630
20	21	420	4221

TABLE II
WELL-KNOWN PIGEON HOLE BENCHMARKS PH_n.

REFERENCES

[1] A. Biere, "Lingeling, plingeling and treengeling entering the SAT Competition 2013," in *Proc. of SAT Competition 2013*, 2013.

Supported by FWF, NFN Grant S11408-N23 (RiSE).