Third Pseudo-Boolean Evaluation PB07

Vasco MANQUINHO and Olivier ROUSSEL

Tenth International Conference on
Theory and Applications of Satisfiability Testing,
SAT'07

May 31, 2007

Outline

- Linear and non linear Pseudo-Boolean constraints
- Optimization problem
- Benchmarks and Solvers
- Evaluation Environment
- Results
- Next PB event

Linear Pseudo-Boolean Constraints

- A **linear** pseudo-Boolean constraint may be defined over boolean variables by

$$\sum_i a_i.l_i \geq d \text{ with } a_i, d \in \mathbb{Z}, l_i \in \{x_i, \bar{x}_i\}, x_i \in \mathbb{B}$$

  Example: $3x_1 - 3x_2 + 2\bar{x}_3 + \bar{x}_4 + x_5 \geq 5$
- Extends both clauses and cardinality constraints
  - cardinalities: all $a_i = 1$ and $d > 1$
  - clauses: all $a_i = 1$ and $d = 1$
- PB constraints are more expressive than clauses (one PB constraint may replace an exponential number of clauses)
- A pseudo-Boolean instance is a conjunction of PB constraints

Non-Linear Pseudo-Boolean Constraints

- A **non-linear** pseudo-Boolean constraint may be defined over boolean variables by

$$\sum_i a_i(\prod_j l_{i,j}) \geq d \text{ with } a_i, d \in \mathbb{Z}, l_{i,j} \in \{x_{i,j}, \bar{x}_{i,j}\}, x_{i,j} \in \mathbb{B}$$

  Example: $3x_1\bar{x}_2 - 3x_2x_4 + 2\bar{x}_3 + \bar{x}_4 + x_5x_6x_7 \geq 5$
- A product is a AND
- Compact encoding for several problems (e.g. factoring problem encoded by one constraint)
- Can be easily translated into linear pseudo-Boolean by introducing new variables and constraints s.t.

$$p \leftrightarrow x_0 \wedge x_1 \wedge \ldots \wedge x_n$$

  (requires 2 PB constraints or n+1 clauses)

Optimization

- Another difference with SAT is that most PB problems contain a linear cost function to optimize. For example,

$$\text{minimize } f = \sum_i c_i.x_i \text{ with } c_i \in \mathbb{Z}, x_i \in \mathbb{B}$$

- Example of an optimization instance

$$\begin{cases} \text{minimize} & 5x_1 + x_2 + 8x_3 + 2x_4 + 3x_5 \\ \text{subject to} & x_1 + \bar{x}_2 + x_3 \geq 1 \\ & \bar{x}_1 + x_2 + \bar{x}_3 + x_4 \geq 3 \\ & 2\bar{x}_1 + 4x_2 + 2x_3 + x_4 + 5x_5 \geq 5 \\ & 5x_1 + 4x_2 + 6x_3 + x_4 + 3x_5 \geq 10 \end{cases}$$

  Optimum: 8

  $x_1 = x_2 = x_4 = 1$

  $x_3 = x_5 = 0$

- The cost function may contain products (no such instance in the PB07 evaluation)

Benchmark categories (1)

Based on the objective function

SATUNSAT
: No objective function to optimize (decision problem). The solver must simply find a solution.

OPT
: An objective function is present. The solver must find a solution with the best possible value of the objective function.

Benchmark categories (2)

Based on the size of coefficients

SMALLINT small integers: no constraint with a sum of coefficients greater than 2^{20} (20 bits)

- Expected to be safe for solvers using 32 bits integers and simple techniques
- Strong limit to the encoding of concrete problems.

BIGINT big integers: at least one constraint with a sum of coefficients greater than 2^{20} (20 bits) requires handling of big integers

MEDINT categories from PB'05 and PB'06 merged with BIGINT categories

Benchmark categories (3)

Based on the linearity of constraints

- **LIN** All constraints are linear
- **NLC** At least one constraint is non linear (contains products of literals)

Additional special category

PURE-SAT All constraints are clauses.

This is a proper subset of SATUNSAT-SMALLINT-LIN

Contains pigeon-hole instances and some easy instances from the SAT07 competition (which were solved by at least 10 solvers in the first phase).

From one category to another...

Unfortunately, few solvers have support for all categories

NLC to LIN
: Non linear instances were translated into equivalent linear instances for solvers which did not have native support for non linear constraints.

BIGINT to SMALLINT
: from a instance with big integers, we generated a **non equivalent** intance with reduced coefficients

Categories

- **OPT-SMALLINT-LIN (807 instances)**
- **OPT-SMALLINT-NLC (405 instances)**
- **OPT-BIGINT-LIN (388 instances)**
- OPT-BIGINT-NLC
- **SATUNSAT-SMALLINT-LIN (371 instances)**
- **SATUNSAT-SMALLINT-NLC (100 instances)**
- SATUNSAT-BIGINT-LIN (14 instances)
- SATUNSAT-BIGINT-NLC
- **PURE-SAT (166 instances)**

2251 benchmarks (almost 1.5 GB).

Submitted solvers (1/2)

10 submitted solvers (and a few more versions)

absconPseudo	Fred Hemery & Christophe Lecoutre a CSP based solver
bsolo	J. Marques-Silva & V. Manquinho integrates SAT-based techniques with estimation procedures on the value of the cost function
glpPB	Hossein Sheini & K. Sakallah simple use of an integer linear programming toolkit (2006)
minisat+	Niklas Een & Niklas Sörensson translates PB constraints to SAT (2006)
PBS	Bashar AlRawi & Fadi Aloul an extension of the *zchaff 2004* SAT solver to handle pseudo-Boolean constraints
PB-clasp	Gayathri Namasivayam

Submitted solvers (2/2)

Pueblo Hossein Sheini & K. Sakallah
an extension of the *minisat* SAT solver to handle pseudo-Boolean constraints; uses a general pseudo-Boolean learning mechanism (2006)

oree Olivier Roussel
an experimental CDCL solver which uses CP and a specific simplification of reasons to learn PB constraints

SAT4JPseudo Daniel Le Berre & Anne Parrain
a *Galena* like CDCL (Constraint Driven Constraint Learning) solver written in Java (3 versions)

wildcat Lengning Liu & Miroslaw Truszczynski
local search solver based on *wsat* generalized for pseudo-Boolean constraints (2 versions)

Evaluation environment

- Cluster of bi-Xeon 3 GHz, 2MB cache, 2GB RAM (but all solvers were run in 32 bits mode)
  kindly provided by the CRIL, University of Artois, France
  The same environment was used for the SAT competition
- Each solver was given a time limit of 30 minutes (1800s) and a memory limit of 1800 MB (to avoid swapping).
- 414 days of CPU time used in the final phase

Comparing the solvers

Several ways with different point of views

- number of instances they solve completely (UNSAT answer and OPT answers (or SAT answers for decision problems))
- number of instances they solve partially (timeout, but a solution found)
- number of best solutions found
- number of times they are the fastest to give the best solution
- comparison of execution time
- ...

In this presentation, we focus on the number of instances completely solved.

Top solvers: linear categories

A first approach on the number of solved instances in Linear Pseudo-Boolean Categories

Category	UNSAT answers	SAT/OPT answers	Both answers
SATUNSAT-SMALLINT	Pueblo 1.4 PBS4	wildcat-skc wildcat-rnp	Pueblo 1.4 PBS4
OPT-SMALLINT	SAT4jCP SAT4jCPClause	bsolo glpPB	bsolo glpPB
OPT-BIGINT	SAT4jResolution minisat+, oree, SAT4jCP	SAT4jResolution SAT4jCP	SAT4jResolution SAT4jCP

Top solvers: non linear categories

A first approach on the number of solved instances in Non-Linear Pseudo-Boolean Categories

Category	UNSAT answers	SAT/OPT answers	Both answers
SATUNSAT-SMALLINT	glpPB PBS4, minisat+, PB-clasp	wildcat-skc wildcat-rnp	glpPB, minisat+, PBS4 PB-clasp, Pueblo 1.4
OPT-SMALLINT	– –	minisat+ Pueblo 1.4	minisat+ Pueblo 1.4

Comparing SAT and PB solvers

Since

- PB is an extension of SAT
- the SAT competition and the PB evaluation were run in the same environment
- it's easy to translate a SAT instance in the PB syntax

we may try to compare SAT and PB solvers.

But of course

- PB solvers generally are disadvantaged (must handle coefficients)
- PB solvers may not be optimized for clauses

Context of the comparison

- subset of 146 instances from the CRAFTED and INDUSTRIAL categories which were solved by at least 10 solvers in the first phase of the SAT competition
- 20 minutes timeout for all solvers
- all SAT solvers from the Competition or Demonstration division

Pure-SAT Category (1/4)

Rank	Solver	Version	Solved	Time
1	minisat	SAT 07	137	19211.04
2	minisat	SAT 07 (assertions)	137	19995.93
3	minimarch	2007-04-26 (fixed)	128	18492.29
4	picosat	535	125	30759.73
5	MiraXT	v3	124	22328.80
6	SATzilla	FULL	124	27070.53
7	Barcelogic Fixed	2007-04-13	123	24131.13
8	SATzilla	CRAFTED	123	28322.52
9	MiraXT	v1	122	18922.25
10	Rsat	2007-02-08	120	19330.37
11	CMUSAT	2007-02-08	119	16710.13
12	MiraXT	v2	117	20028.29
13	MXC	2007-02-08	117	21222.30
14	Spear FHS	1.0	115	25572.86
15	Spear FH	1.0	115	28259.06
16	PB-clasp	2007-04-10	114	27573.84
17	TiniSatELite	2007-02-08	113	21261.11

Pure-SAT Category (2/4)

Rank	Solver	Version	Solved	Time
18	tinisat	2007-02-08	108	22752.39
19	SAT7	2007-02-08	108	29724.96
20	minisat+	1.14	108	31334.65
21	CMUSAT BASE	2007-02-08	107	25669.27
22	Spear	2007-02-12	104	22716.79
23	SATzilla	RANDOM	92	21035.51
24	PB-clasp	2007-03-23	86	25391.63
25	SAT4JPseudoResol.	2007-03-23	86	26951.92
26	SAT4J	SAT 07	85	29785.08
27	Pueblo	1.4	70	29118.02
28	SAT4J JVM changed	SAT 07	62	21893.82
29	bsolo	3.0.17	55	11759.12
30	bsolo	3.0.16	51	9870.80
31	ornithorynque	0.1 alpha	41	8711.92
32	DEWSATZ 1A	2007-02-08	40	9549.07
33	sat4jPseudoCP	2007-03-23	38	11466.30
34	sat4jPseudoCPClause	2007-03-23	38	11541.60

Pure-SAT Category (3/4)

Rank	Solver	Version	Solved	Time
35	March KS	2007-02-08	34	6491.74
36	DEWSATZ	2007-04-26 (fixed)	32	7690.29
37	KCNFS	2006	29	5307.28
38	TTS	4.0	27	1316.12
39	PBS4_v2	2007-03-23	25	4811.30
40	PBS4	2007-03-23	25	4849.70
41	KCNFS	SMP	24	6452.26
42	KCNFS	2004	16	3281.07
43	Hybrid1	2007-02-08	14	1593.20
44	adaptg2wsat+	2007-02-08	14	1604.71
45	FH	2007-02-08	13	1828.24
46	oree	0.1.2 alpha	13	3367.72
47	adaptg2wsat	2007-02-08	12	675.70
48	ranov	2007-02-08	12	4799.76
49	adaptg2wsat0	2007-02-08	11	940.70
50	adaptg2wsatp	2007-02-08	11	1468.26
51	absconPseudo	102	9	1543.97

Pure-SAT Category (4/4)

Rank	Solver	Version	Solved	Time
52	UnitMarch	2007-02-08	7	2156.12
53	adaptnovelty	2007-02-08	6	2911.42
54	wildcat-skc	2007-03-21	4	1634.57
55	saps	2007-02-08	4	2319.09
56	gnovelty+	2007-02-08	3	637.12
57	sapsrt	2007-02-08	3	1439.19
58	Mmisat	2007-02-08	2	757.87
59	wildcat-rnp	2007-03-21	0	-
60	glpPB	0.2	0	-

Next PB event

- No evaluation in 2008 to give time to
  - write new solvers (complete and **incomplete**)
  - improve current solvers
    - support for big integers
    - non linear constraints
    - ....
  - move to 64 bits binaries
- A competition in 2009
  - will introduce multi-objective pseudo-Boolean categories **iff** enough solvers and benchmarks are provided
- **You have approximately 18 months to write your solver and submit benchmarks!**

More information

- See our nice posters!
- Or see the web site http://www.cril.univ-artois.fr/PB07