PB-smodels a Pseudo-Boolean Solver

(original, unpublished research)

Gayathri Namasivayam
University of Kentucky
Computer Science, Lexington, Ky. 40506-0046
(859) 257-3961, gayathri@cs.uky.edu

Introduction

A *pseudo-boolean constraint* (PB-constraint) is a linear inequality with 0-1 variables and integer coefficients. Search problems can be modeled as sets of PB-constraints, so that, solutions to a set of PB-constraints map to solutions of the search problem. An optimization problem can be modeled using a set of PB-constraints along with an optimization statement over a set of 0-1 variables.

Pseudo-boolean solvers (PB-solvers) are programs designed to find solutions to a set of PB-constraints, and hence can be used to solve search problems. PBS4 (Aloul 2005), MiniSat+ (Eén & Sörensson 2005), BSOLO (Manquinho & Marques-Silva 2005) and Pueblo (Sheini & Sakallah 2005) are recent PB-solvers.

Our contribution is to show that *smodels*, a solver originally designed to solve search problems encoded as logic programs with weight constraints (Niemelä & Simons 1997), can be used as a competitive PB-solver. We call this PB-solver *PB-smodels*.

PB-smodels uses a procedure we designed, *PB2smodels*, to translate a set of PB-constraints into a logic program that is accepted by *smodels*. We show that solutions of a search problem modeled as a set of PB-constraints T correspond to stable models of the logic program $P(T)$ produced from T by PB2smodels.

We compare the performance of PB-smodels with two PB-solvers MiniSat+ and Pueblo on search problems encoded as sets of PB-constraints. In particular, we compare PB-smodels to MiniSat+ and Pueblo on benchmarks obtained from PB 2005 competition (Manquinho & Roussel 2006). We also compare the PB-solvers on benchmarks derived from randomly generated instances of problems such as weighted N-queens, traveling salesperson problem (TSP), blocked N-queens and weighted dominating set problem.

Pseudo-boolean constraints

A PB-constraint is an expression of the form

$$w_1 * x_1 + \ldots + w_k * x_k R c,$$

where $w_1, \ldots, w_k, c$ are integers, $R \in \{\leq, \geq\}$ and $x_1, \ldots, x_k$ are 0-1 variables. Let T be a set of PB-constraints. An assignment of 0s and 1s to variables occurring in constraints in T is a *solution* to T if each PB-constraint in T holds (as a linear inequality).

A set of PB-constraints T may also contain an optimization statement

$$ST : v_1 * y_1 + \ldots + v_l * y_l,$$

where $ST \in \{minimize, maximize\}$, $v_1, \ldots, v_l$ are 0-1 variables and $y_1, \ldots, y_l$ are integers. An optimal solution is one that minimizes (maximizes) $v_1 * y_1 + \ldots + v_l * y_l$.

Logic Programming

A weight constraint W is an expression of the form

$$W = l[x_1 = w_1, \ldots, x_k = w_k]u,$$

where $l, u, w_1, \ldots, w_k$ are integers and $x_1, \ldots, x_k$ are propositional atoms. We use l to denote the lower bound, u to denote the upper bound, and w_i to denote the weight assigned to atom x_i.

A logic program accepted by *smodels* (*smodels program* for short) (Niemelä & Simons 1997), is a set of rules, of the form

$$r = A_0 \leftarrow A_1, \ldots, A_n,$$

where $n \geq 0$, A_0 is either a weight constraint or an atom, and each A_i, $1 \leq i \leq n$, is a *literal* (an expression a or $not(a)$, where a is an atom) or a weight constraint. An smodels program may also contain optimization rules of the form

$$ST[v_1 = y_1, \ldots, v_l = y_l],$$

where $ST \in \{minimize, maximize\}$, $v_1, \ldots, v_l$ are atoms and $y_1, \ldots, y_l$, are integers. We refer the reader to (Niemelä & Simons 1997) for the definition of the semantics of smodels programs.

Translation procedure from PB-constraints to an smodels program

We describe now a procedure PB2smodels that takes as input a set of PB-constraints T and translates it to an smodels program $P(T)$. In the translation, we identify 0-1 integer variables with boolean propositional variables. That is, we view x appearing in T as a 0-1 variable and the same x appearing in $P(T)$ as a propositional variable.

Solvers	PB'05	Random instances
PB-smodels	144	400
MiniSat+	184	298
Pueblo	172	380

Figure 1: Total number of instances solved by the solvers

First, for each 0-1 variable x in T, the procedure includes in $P(T)$ a *choice* atom $\{x\}$ in $P(T)$ (Niemelä & Simons 1997). Next, for each PB-constraint $r \in T$ we do the following.

1. If r is a constraint rule $w_1 * x_1 + \ldots + w_k * x_k \leq c$, we translate r to an smodels rule $\leftarrow (c+1)[x_1 = w_1, \ldots, x_k = w_k]$.
2. If r is a constraint rule $w_1 * x_1 + \ldots + w_k * x_k \geq c$, we translate r to an smodels rule $\leftarrow [x_1 = w_1, \ldots, x_k = w_k](c-1)$. As a special case, we translate a PB-constraint $1 * x_1 + \ldots + 1 * x_k \geq 1$ into an smodels rule $\leftarrow not(x_1), \ldots, not(x_k)$.
3. If r is an optimization statement $ST : w_1 * x_1 + \ldots + w_k * x_k$, we translate r to an smodels optimization rule $ST[x_1 = w_1, \ldots, x_k = w_k]$.

Theorem 1 *Let T be a set of PB-constraints and let $P(T) = PB2smodels(T)$ be a logic program. An assignment I of 0s and 1s to variables in T is a solution to T if and only if the set $M_I = \{x_i : I(x_i) = 1\}$ is a stable model of $P(T)$. Moreover, if T contains an optimization statement, I is an optimal solution to T if and only if M_I is an optimal stable model of $P(T)$.*

Experiment Methodology and Results

We compared the performance of the three pseudo-boolean solvers PB-smodels, MiniSat+ and Pueblo on instances of search problems that are modeled as PB-constraints. We used two sets of benchmarks in our experiments. The first benchmark consists of 386 instances of optimization problems used in PB 2005 competition (Manquinho & Roussel 2006). The second benchmark consists of random, satisfiable and unsatisfiable instances of weighted N-queens, blocked N-queens, traveling salesperson and weighted dominating set problems (Namasivayam 2006).

All three algorithms were executed on the same set of instances, on the same machine and with 500 second time limit. We recorded the time taken by each algorithm to find a solution to decide that there is no solution, or to determine the optimum value for an optimization instance.

Figure 1 summarizes the results. It shows that PB-smodels solved (with the 500 second time limit) more random instances than other solvers. The run-time distribution (RTD) graph for instances of the TSP problem (with 11 cities, and a bound on the total length set at 25) shown in Figure 2, provides an additional evidence for good performance of PB-smodels on random instances. RTDs at the web-site (Namasivayam 2006) show similar good performance also

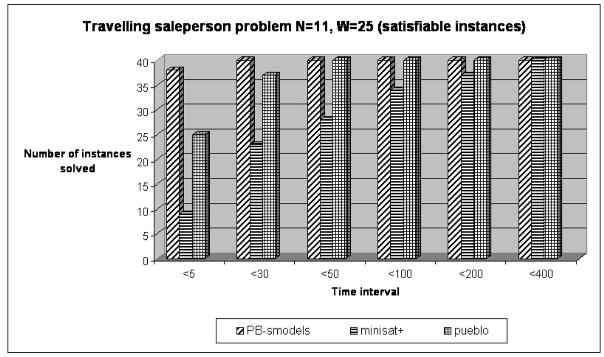

Figure 2: Performance of satisfiable instances of TSP

on the blocked N-queens problem. The other solvers perform better on the remaining two problems.

PB-smodels was also able to solve 144 optimization instances out of the total of 386. The other solvers solved more instances in this category, but neither performed significantly better.

The performance of the algorithms on the benchmarks is available in the web site (Namasivayam 2006).

Conclusion

We developed a PB-solver PB-smodels and demonstrated that in many cases it outperforms existing PB-solvers such as MiniSat+ and Pueblo.

Acknowledgments

I thank Mirosław Truszczyński for his helpful guidance and suggestions during the course of this research. This work was partially supported by the NSF grant IIS-0325063.

References

Aloul, F. 2005. PBS4 solver. `http://www.eecs.umich.edu/~faloul/Tools/pbs4/`.

Eén, N., and Sörensson, N. 2005. Minisat+ solver. `http://www.cs.chalmers.se/Cs/Research/FormalMethods/MiniSat/Papers.html`.

Manquinho, V., and Marques-Silva, J. P. 2005. BSOLO solver. `http://sat.inesc-id.pt/~vmm/`.

Manquinho, V. M., and Roussel, O. 2006. The first evaluation of pseudo-boolean solvers (PB'05). *Journal on Satisfiability, Boolean Modeling and Computation* 2:103–143.

Namasivayam, G. 2006. PB-smodels solver. `http://www.csr.uky.edu/~gayathri/pbsmodels.htm`.

Niemelä, I., and Simons, P. 1997. Smodels – an implementation of the stable model and well-founded semantics for normal logic programs. In *Proceedings of LPNMR*, 420–429. Springer-Verlag.

Sheini, H. M., and Sakallah, K. A. 2005. Pueblo: A modern pseudo-boolean sat solver. In *DATE*, 684–685. IEEE Computer Society.