Translating Pseudo-Boolean Constraints into CNF

Amir Aavani
aaa78@sfu.ca

Simon Fraser University

1 Introduction

A Pseudo-Boolean constraint (PB-constraint) is a generalization of a clause. A PB-constraint is an inequality (equality) on a linear combination of Boolean literals ($\sum_{i=1}^{n} a_i l_i\ OP\ b$) where $a_1, \cdots a_n$ and b are constant integers, $l_1, \cdots, l_n$ are literals and OP is a comparison operator. The left-hand side of a PB-constraint under assignment $\mathcal{A}$ is equal to the sum of the coefficients whose corresponding literals are mapped to true by $\mathcal{A}$. This kind of constraints has been widely used in expressing NP-complete problems. Several approaches have been proposed to translate a PB-constraint to CNF, [3], [2].

In this paper, we propose a new encoding for translating PB-constraints whose comparison operator is "=" to CNF. The CNF produced by the proposed encoding has small size, and also the constraints for which one can expect the SAT solvers to perform well on the produced CNF can be characterized. We show that there are many constraints for which the proposed encoding has a good performance. It worths mentioning that an arbitrary PB-constraint can be rewritten as a single equivalent PB-constraint whose comparison operator is "=" and all its constant integers are positive.

Definition 1. Given constraint Q on set of variables X, we call the pair $\langle v, C\rangle$, where v is a Boolean variable, C is a set of clauses on $X \cup Y \cup \{v\}$ and Y is a set of propositional variables, a *valid translation* if for every satisfying total assignment $\mathcal{A}$ to $X \cup Y \cup \{v\}$ for C, $\mathcal{A}$ satisfies Q iff it maps v to *true*, i.e., $C \models v \Leftrightarrow Q$.

2 Proposed Method

Let a PBMod-constraint be an equation in the following form:

$$\sum_{i=1}^{n} a'_i l_i = b' \text{ (mod M)}. \tag{1}$$

where $0 \le a'_i < M$ for all $1 \le i \le n$ and $0 \le b' < M$. Total Assignment $\mathcal{A}$ is a solution to (1) iff the value of left-hand side summation under $\mathcal{A}$ minus the value of right-hand side of the equation, b', is a multiple of M.

Definition 2. The PBMod-constraint $Q(M) : \sum a'_i l_i = b'(\text{mod M})$ is called to be the *conversion* of the PB-constraint $Q : \sum a_i l_i = b$, modulo M iff

1. $a'_i = a_i$ mod M
2. $b' = b$ mod M

Proposition 1. Let $\mathtt{M} = \{M_1, \cdots, M_m\}$ be a set of m relatively prime integers. The set of assignments satisfying $Q : \sum a_i l_i = b$ is exactly the same as the set of assignments satisfying all the m PBMod-constraints $Q(M_k)$ if $\prod_{k=1}^{m} M_k > S = \sum a_i$.

One candidate for the set $\mathtt{M}$ is a subset of prime numbers. One can enumerate the prime numbers and add them to the set of modulos, $\mathtt{M}^{\mathtt{P}} = \{2, 3, ..., P_m\}$, until their multiplication exceeds S. The next proposition gives us an estimation for the size of set $\mathtt{M}^{\mathtt{P}}$ as well as the maximum value in $\mathtt{M}^{\mathtt{P}}$.

Proposition 2. Let $\mathtt{M}^{\mathtt{P}} = \{2, \cdots, P_m\}$ be the set of primes s.t. $\prod_{p \in \mathtt{M}^{\mathtt{P}}} p \geq S$. Then:

1. $m = |\mathtt{M}^{\mathtt{P}}| \leq \log S$.
2. $P_m < (\log S)^2$.

Theorem 1. Let $Q : \sum a_i l_i = b$ be a PB-constraint. Also let $\mathtt{M}^{\mathtt{P}} = \{P_1, \cdots, P_m\}$ be as above, and the pair $\langle v_k, C_k \rangle$ be a *valid translation* for PBMod-constraint $Q(P_k)$. Then, the pair $\langle v, C \rangle$ is a valid translation for PB-constraint Q where $C = \cup_k C_k \cup C'$ and C' is the set of clauses describing $v \Leftrightarrow (v_1 \wedge v_2 \cdots \wedge v_m)$.

Translation of PBMod-constraint Through DP The translation presented here is similar to translation through BDD, described in [3]. Tseitin variable, D^l_m, is defined inductively as follows:

$$D^l_m = \begin{cases} \top & \text{if } l \text{ and } m \text{ are both zero;} \\ \bot & l = 0 \text{ and } m > 0; \\ (D^{l-1}_{(m-a_l) \bmod \mathrm{M}} \wedge x_l) \vee (D^{l-1}_m \wedge \neg x_l) & \textit{Otherwise} \end{cases}$$

3 Performance of Unit Propagation

There are three situations in which UP is able to infer the input variables values of a PB-constraint Q:

1. Unit Propagation Detects Inconsistency: If Q is unsatisfiable, UP may be able to infer that there is no assignment satisfying Q.
2. Unit Propagation Solves Constraint: UP may be able to infer the whole solution for Q if there is just a single satisfying solution to Q.
3. Unit Propagation Infers the Value for an Input Variable: UP may be able to infer that the value of input variable x_k is *true*/*false* if x_k takes the same value in all the solutions to Q. This is a generalization of previous case.

It can be shown that for each of the above cases, there are at least $(\frac{\sum a_i}{\log \sum a_i})^{n+1} = \frac{2^{nPoly(n)}}{Poly(n)^{n+1}}$ different PB-constraints in the form $\sum a_1 l_i = b$ such that CNFs, produced using the proposed approach, allow UP to infer input variables.

4 Conclusion

Our translation produces a polynomial size CNF w.r.t. the input size. We also argued that for exponentially many instances, produced CNFs are arc-consistent. This number is much bigger for our encoding comparing to the existing encodings. Interested readers are invited to read the complete version of this paper [1].

References

1. A. Aavani. Translating Pseudo-Boolean Constraints into CNF. *http://arxiv.org/abs/1104.1479.*
2. O. Bailleux, Y. Boufkhad, and O. Roussel. New Encodings of Pseudo-Boolean Constraints into CNF. *Theory and Applications of Satisfiability Testing-SAT 2009*, pages 181–194, 2009.
3. N. Eén and N. Sörensson. Translating pseudo-boolean constraints into SAT. *Journal on Satisfiability, Boolean Modeling and Computation*, 2(3-4):1–25, 2006.