# TT-Open-WBO-Inc: Tuning Polarity and Variable Selection for Anytime SAT-based Optimization

Alexander Nadel
Intel Corporation, P.O. Box 1659, Haifa 31015 Israel
Email: alexander.nadel@intel.com

***Abstract*—This document is a description of the solver TT-Open-WBO-Inc, submitted to the weighted incomplete tracks of MaxSAT Evaluation 2019. We tuned the polarity and variable selection strategies of the underlying SAT solver in the best-performing MaxSAT solver in the Weighted-Incomplete-60-Second track of MaxSAT Evaluation 2018 – Open-WBO-Inc-BMO [6], [7].**

## I. Introduction

The main goal of this submission is to experiment with a new polarity selection heuristic and an enhancement to the variable decision strategy for SAT-based anytime Weighted MaxSAT solving in the state-of-the-art algorithm Open-WBO-Inc-BMO [6], [7]. In principle, our heuristics can be applied to solving any optimization problem with a SAT-based anytime algorithm.

We call our polarity selection heuristic Target-Optimum-Rest-Conservative (TORC) and the enhancement to the variable selection strategy Target-Score-Bump (TSB). Our heuristics are detailed in a paper under submission. We provide a brief (yet precise) description in this document.

## II. Preliminaries

A Weighted MaxSAT instance comprises a set of *hard* satisfiable clauses $H$ and a set of weighted *soft* constraints $T = \{t_{n-1}, t_{n-2}, \dots, t_0\}$, where each constraint $t_i$ is associated with a strictly positive integer weight $w_i$. The *weight of a variable assignment* $\mu$ is $unsWt(T, \mu) = \sum_{i=0}^{n-1} \neg\mu(t_i) * w_i$, that is, the overall weight of $T$'s bits, falsified by $\mu$. Given a Weighted MaxSAT instance, a Weighted MaxSAT solver is expected to return a model having the minimum possible weight. For the rest of this document, for convenience and without restricting generality, it is assumed that every soft constraint is a unit clause (that is, a clause containing one literal). An arbitrary soft constraint $t_i$, reducible to a set of clauses $F$, can be transformed to a unit clause $s'$, where $s'$ is a fresh variable, by adding the clause $\neg s' \vee c$ to $H$ for each clause $c \in F$. Thus, $T$ can be thought of as a bit-vector, where $t_0$ is its Least Significant Bit (LSB) and $t_{n-1}$ is its Most Significant Bit (MSB). $T$ is called the *target bit-vector*, or, simply, the *target* and every $t_i \in T$ is called a *target bit*.

Recall that modern SAT solvers apply phase saving [10] as their polarity selection heuristic. In phase saving, once a variable is picked by the variable decision heuristic, the literal is chosen according to its latest value, where the values are normally initialized with 0.

It turned out that overriding phase saving in the context of anytime SAT-based optimization algorithms, which generate an improving set of models $\{\mu_1, \mu_2, \dots, \mu_n\}$ over time, is advantageous. In this context, one can distinguish between the optimistic and the conservative approaches to polarity selection. The optimistic approach [3], [5], [9] sets the polarity of the target bits to 1; it works well when the actual solution is close to the optimum. The conservative approach [1], [4], [11] sets the polarity of all the variables (or all the original variables) to the previous best solution.

## III. Target-Optimum-Rest-Conservative (TORC) Polarity Selection

We propose a new polarity selection heuristic, which we call *Target-Optimum-Rest-Conservative (*TORC*)*.

Before the initial SAT invocation, TORC *fixes* the polarity of all the *target* variables to the optimal value. Then, after each new improving model $\mu_i$ is encountered, the polarity of *all* the *non-target* variables are fixed to their values in $\mu_i$.

In other words, whenever the variable decision heuristic chooses:

1) A target variable: TORC sets its polarity to 1 (to be optimistic).
2) A non-target variable: TORC sets its polarity to its value in the best model so far (to be conservative; only after the first SAT invocation is completed)

Note that, after the initial SAT call, TORC sets the polarity every single time a new decision variables is picked.

TORC has been designed to leverage the best of both the conservative and the optimistic worlds. On one hand, we are interested in taking advantage of the conservative heuristic, which is known to find the next improved model more quickly than the default heuristic by looking near the previous model. At the same time, however, we would like to encourage the values of the target variables to be as close to the optimum as possible in order to move more quickly towards the optimum.

## IV. Target Score Bump (TSB)

We would like to experiment with tuning the SAT solver's variable selection heuristic for anytime SAT-based optimization.

Modern SAT solvers mostly use variants of the VSIDS variable decision heuristic [8]. VSIDS associates a score with every variable and picks as the next decision the variable with the greatest score. `Open-WBO-Inc-BMO` uses Glucose 4.1 SAT solver [2]. Glucose 4.1 has a function *varBumpActivity(v,b)*, which bumps up the score of variable $v$ by $b$.

Our proposed heuristic–`Target-Score-Bump` (`TSB`)– bumps up the variable scores of the *target bit* variables, so as to improve their chances of being picked early. We would also like to give some preference to target bits having greater weight.

We apply `TSB` prior to the initial SAT invocation as follows.

Let the minimal target-bit weight be $\min = \min(w_0, \ldots, w_{n-1})$ and the maximal target-bit weight be $\max = \max(w_0, \ldots, w_{n-1})$. For every variable $t$ of target bit literal $t_i$ of weight $w_i$, we apply the function *varBumpActivity(t,b)*, where $b$ is

$(w_i - \min)/(\max - \min) * weightBump + varBump$.

Both weightBump and varBump are user-given parameters. They regulate the relative importance of the weight in the scores. The default version of `TT-Open-WBO-Inc` uses weightBump = 113 and varBump = 552.

## REFERENCES


[1] C. Ansótegui and J. Gabàs. WPM3: an (in)complete algorithm for weighted partial maxsat. *Artif. Intell.*, 250:37–57, 2017.

[2] G. Audemard and L. Simon. On the glucose SAT solver. *International Journal on Artificial Intelligence Tools*, 27(1):1–25, 2018.

[3] D. L. Berre and A. Parrain. The sat4j library, release 2.2. *JSAT*, 7(2-3):59–6, 2010.

[4] E. Demirovíc, G. Chu, and P. J. Stuckey. Solution-based phase saving for CP: A value-selection heuristic to simulate local search behavior in complete solvers. In *CP 2018*, pages 99–108, 2018.

[5] E. Di Rosa and E. Giunchiglia. Combining approaches for solving satisfiability problems with qualitative preferences. *AI Commun.*, 26(4):395–408, 2013.

[6] S. Joshi, P. Kumar, V. Manquinho, R. Martins, A. Nadel, and S. Rao. Open-WBO-Inc in MaxSAT evaluation 2018. volume B-2018-2 of *Department of Computer Science Series of Publications B*, page 16. University of Helsinki, 2018.

[7] S. Joshi, P. Kumar, R. Martins, and S. Rao. Approximation strategies for incomplete maxsat. In *CP 2018*, pages 219–228, 2018.

[8] M. W. Moskewicz, C. F. Madigan, Y. Zhao, L. Zhang, and S. Malik. Chaff: Engineering an efficient SAT solver. In *DAC 2001*, pages 530–535, 2001.

[9] A. Nadel and V. Ryvchin. Bit-vector optimization. In *TACAS 2016*, pages 851–867, 2016.

[10] K. Pipatsrisawat and A. Darwiche. A lightweight component caching scheme for satisfiability solvers. In *SAT*, pages 294–299, 2007.

[11] I. A. Roig. *Solving hard industrial combinatorial problems with SAT*. Dissertation, Polytechnic University of Catalonia, 2013.