Deep Cooperation of Local Search and Unit Propagation Techniques

Xiamin Chen, Zhendong Lei, and Pinyan Lu

Shanghai University of Finance and Economics
Institution of Theoretical Computer Science
Huawei Taylor Lab

September 6, 2024

Overview

1. Introduction
2. Main Idea
3. Experimental Results

Notations

- **Variables:** $x_1, x_2, \ldots, x_n \in \{0, 1\}$
- **Literals:** $l_i \equiv x_*$ or $l_i \equiv \neg x_*$ for some variable x_*
- **Clauses:** $C_j = l_1 \vee l_2 \vee \cdots \vee l_k$ with a weight w_j.
  - is **hard** if $w = \infty$
  - is **soft** if $w < \infty$
- **Assignment** α: maps variables x_* to 0 or 1.
  - is **complete** if all variables are mapped.
  - is **partial** if some variables are undecided.

Partial MaxSAT and Pseudo-Boolean Optimization

Partial MaxSAT

- hard clauses $H = (C_1, C_2, \ldots, C_n)$.
- Soft clauses $S = (S_1, S_2, \ldots, S_m)$ with weight $w_1, w_2, \ldots, w_m$.
- Objective: Find assignment α^* satisfying H, minimizing unsatisfied weight sum of S.

Pseudo-Boolean Optimization (PBO)

minimize:

$$\sum_j c_{oj} l_{oj}$$

subject to

$$\sum_j c_{ij} l_{ij} \geq d \text{ ,for } i = 1, 2, \ldots, m$$

where l_{ij} are literals based on variables $x_1, \ldots, x_n \in \{0, 1\}$.

Transformations between PMS and PBO

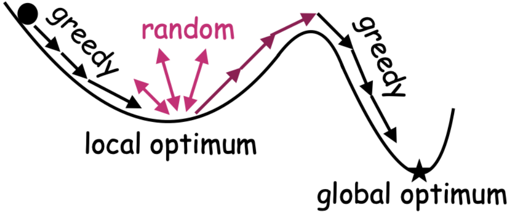
Clauses can be written in PB constraints easily because the latter is more expressive.

hard clause		constraint
$x_1 \vee x_2 \vee x_3$	$\longrightarrow$	$x_1 + x_2 + x_3 \geq 1$

soft clause		objective + constraint	
$x_1 \vee x_2 \vee x_3 \; (w_i)$	$\longrightarrow$	$+ \, w_i \, x_4$	$x_1 + x_2 + x_3 + x_4 \geq 1$

For clarification, we use the term constraints and objective in the rest of this presentation.

The Local Search Algorithm



- Begins with a complete assignment.
- Improves by flipping (local view).
- Quick convergence to good solutions.
- Difficult to leave local optimum and reach global optimum.

Scoring System of Local Search in PBO

The score of flipping a variable is the weighted sum of contributes in each constraints. Here is an example:

	before	after	bound	contribute	weight
cons1	4	6	5	1 (make)	x2
cons2	1	3	10	2 (improve)	x1
cons3	3	1	3	-2 (break)	x1
cons4	7	5	5	0 (keep)	x1

flip score = 2

Trouble: Local search is trapped in local optima if no variable has positive score.

Existing Techniques

- Tabu-Search-related candidate filtering. CCAnr [Cai et al. 2015].

  To avoid looping, any variable should not be flipped again, until its neighborhood has changed.

- Dynamic clause weighting. SATLike [Cai & Lei 2020]

  Add weights to unsatisfied constraints, to emphasize important constraints, as well as to find new greedy flips.

- Reinforcement-Learning-based variable selection. BandMaxSAT [Zheng et al. 2022]

  Instead of referring to the scores, choose the variable with highest UCB to flip at local optima.

These methods are highly effective, but the challenge persists ...

Decreasing Effectiveness of Local Search Solvers

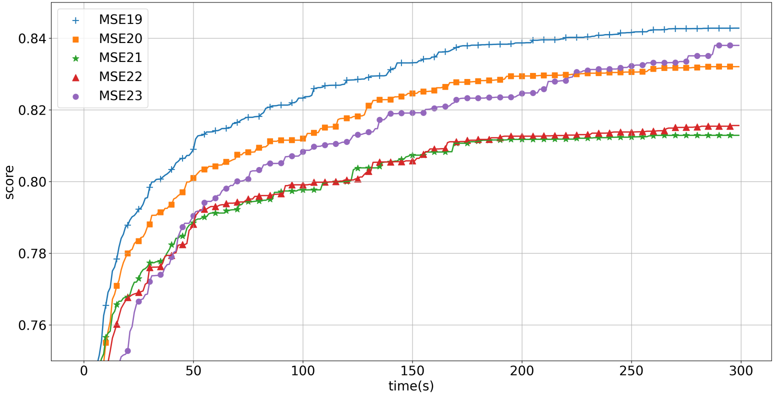


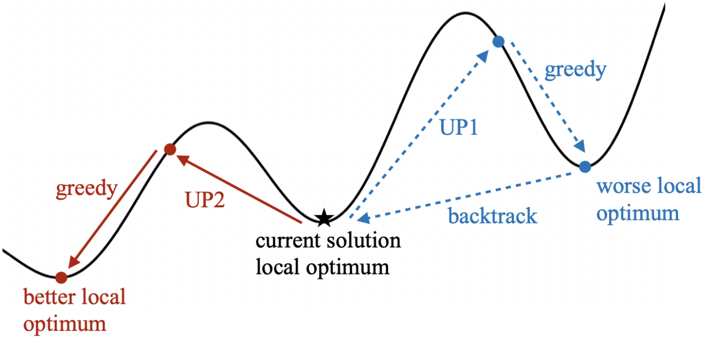
Figure: Tracing the anytime-results of NuWLS in different cutoff times

Motivation

1. We want to introduce some systematic methods to local search to boost its exploration.
2. The power of local search decreases after 60 seconds, we need to do something to make good use of the rest of time for new breakthroughs.
3. We will focus on multi-step flips guided by unit propagation to escape from local optima.
4. The implementation of unit propagation must be light-weighted and time-efficient, so it can be planted into many local search solvers.

Main Contribution

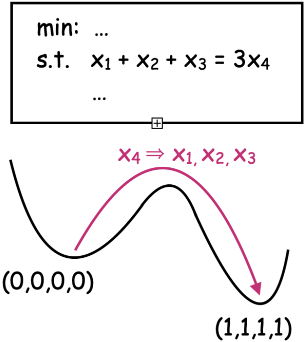
1. Use unit propagation to guide a multi-step flip to leave the local optimum.
2. Following the greedy steps to a new local optimum.
3. Reject worse new local optima (waste of time) or accept better ones (success).



Invocation Circumstances

Meeting local optima.

- Calls UP every k times of local optima. (k ranges dynamically in $[10, 2560]$.)
- Balancing gains from UP and time cost.
- UP is just one option. We need to allow original heuristics to take effect, such as clause-weighting schemes.



When restarts: revisit the best-found solution.

- Solver needs to restart because it is drowned in hopeless neighborhoods.
- Good solutions cluster together, so better ones can be found nearby.
- Stress more on intensification.

The Implementation of Unit Propagation

The relationships of unit propagation are derived from binary constraints.

in PMS:	in PBO:
$x_1 \vee x_2$	$3x_1 + x_2 + x_3 + x_4 \geq 3$

Once the propagation of some variable x_i is needed, check all binary constraints containing x_i, then restore propagated literals in cache.

Pros:

- Light-weighted. No calculation until being needed. For the first time, the complexity is O(n). For the next time, call it from cache.
- Independent. Only related to the model itself, no further CDCL structure is needed.

Cons:

- Insufficient. Some secondary relationships may be omitted. But the subsequent greedy steps will compensate for this.

Planting Our Framework in Different Algorithms

We have implemented our approach in LS-PBO, NuPBO, Satlike3.0 and NuWLS. Actually, it can be used in many other local search solvers because

- The strategy is basically doing multi-step flips, can be regarded as an alternative of a common flip. Call it as long as you want.
- It is easy and quick to derive unit propagation relationships through searching its existing constraints.
- (Future work) For other problems without linear constraints as in PMS and PBO, but with other means of constraints, as long as propagation exists, one can always keep this strategy as a weapon.

Experimental Results

benchmark	#inst.	#win.	#lose.	*nuwls*$_{ss}$	*nuwls*	*accept*(%)	*dist*	*step*(%)
MSE19	299	**74**	38	**0.8322**	0.7963	28	4.5	85
MSE20	262	**75**	30	**0.8219**	0.7882	30	5.0	86
MSE21	155	**34**	19	**0.7991**	0.7723	27	4.7	84
MSE22	179	**46**	25	**0.8044**	0.7856	28	4.0	86
MSE23	179	**61**	30	**0.8158**	0.8032	25	5.3	86

Table: Experiment results of NuWLS on MSE benchmarks.

- *accept(%)* is the rate of accepted local optima over the total number UP-flip called.
- *dist* denotes the average Hamming distance between two local optima before and after an accepted UP-flip.
- *step(%)* is the rate of valid flips compared with the original NuWLS solver.

Experimental Results

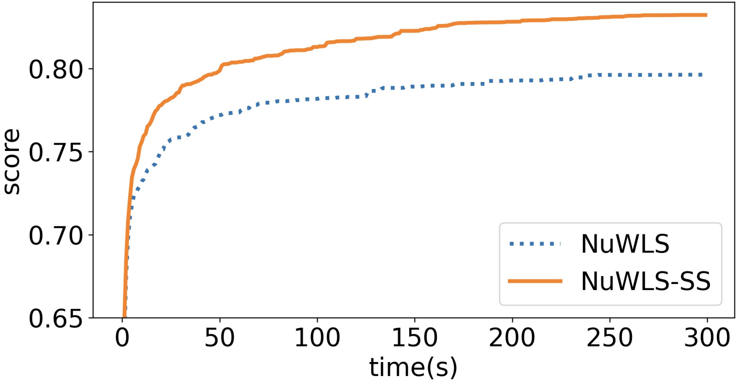


Figure: The average score of testing on benchmark MSE19, tracing the anytime-results of modified and original NuWLS in different cutoff times.

Thank you.