# ParLS-PBO: A Parallel Local Search Solver for Pseudo Boolean Optimization

Zhihan Chen, Peng Lin[presenter], Hao Hu, and Shaowei Cai*

Institute of Software, Chinese Academy of Sciences, Beijing, China

School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China

{chenzh, linpeng, huhao, caisw}@ios.ac.cn

2024.09.03

Presenter

* Corresponding author

中国科学院软件研究所 Institute of Software Chinese Academy of Sciences

中国科学院大学 University of Chinese Academy of Sciences

# Outline

## Background

- Pseudo Boolean Optimization
- Local Search
- LS-PBO

## DLS-PBO

- Dynamic Scoring Mechanism
- Comparison Results

## ParLS-PBO

- Framework
- Solution Pool
- Polarity Density Weight
- Comparison Results

# Pseudo Boolean Optimization

**Definition 1** *Linear pseudo-Boolean constraint (LPB constraint): Let* $x = \{x_1, x_2, \ldots, x_n\}$ *be a set of boolean variables, a literal* $l_i$ *is either a variable* $x_i$ *or its negation* $\neg x_i$*, a LBP constraint is formed as*

$$\sum_{i=1}^{n} a_i \cdot l_i \triangleright b, \quad a_i, b \in \mathbb{Z}, \quad \triangleright \in \{=, \leq, <, \geq, >\}$$

Normalized

$$\sum_{i=1}^{n} a_i \cdot l_i \geq b, \quad a_i, b \in \mathbb{N}_0^+$$

**Definition 2** *Pseudo-Boolean Optimization (PBO) problem: Find an assignment satisfying all constraints with the minimal value of a given objective function. A PBO instance subjecting to m LPB constraints is formed as*

$$\min_{\{x_1,\ldots,x_n\}} \sum_{i=1}^{n} c_i \cdot l_i, \quad c_i \in \mathbb{Z}$$

$$\text{subject to:} \bigwedge_{j=1}^{m} \sum_{i=1}^{n} a_{ji} \cdot l_i \geq b_j, \quad a_{ji}, b_j \in \mathbb{N}_0^+$$

- Solution: values assigned for each variable
- Feasible solution ⇔ satisfies all constraints
- Lower objective value ⇔ higher quality

# PBO solvers

**Complete Algorithms**

- **Linear Search** — *Sat4j, RoundingSAT, Hybrid*
- **Branch-and-Bound**
- **Translated into SAT** — *MINISAT+, OpenWBO*
- **MIP solvers** — *SCIP, Gurobi(complete version)*

**Require exponential time in the worst case**

**Pool performance on Real World problems**

**Imcomplete Algorithms (Local Search)** — *LS-PBO, DeciLS-PBO, NuPBO, OraSLS*

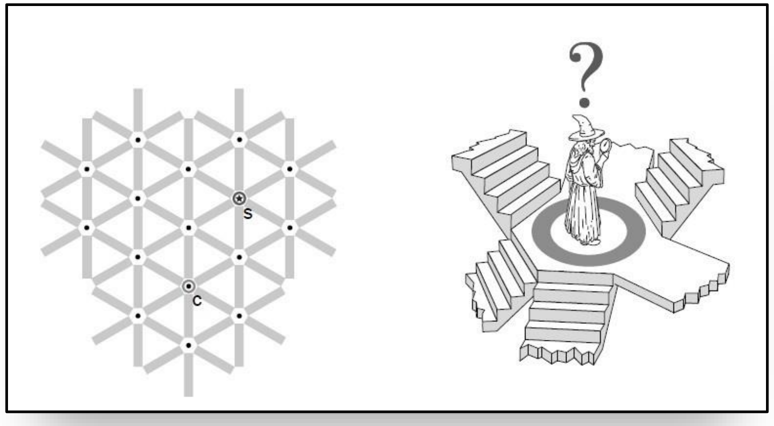
**Parallel Algorithms**

- **Divide-and-conquer**
- **Portfolio**

*FiberSCIP, Gurobi*

**Local Search Process**

- Starting from an initial solution
- Iteratively performing neighbor operation until time limit
- Using the scoring function to guild neighbor operation.




**Local Search for PBO**

- **Solution**: values assigned for each variable.
- **Neighbors** of the current solution $\boldsymbol{S}$: solutions that differ from $\boldsymbol{S}$ in exactly one variable.

**Find high-quality solution quickly!**

- **Constraint Weighting Scheme: help the search avoid stuck in the local optimum**
  - $w(c) = w(c) + 1$ whenever the local search progress is stuck in a local optimal.

- **Scoring Function: measures the benefits of flipping a Boolean variable**
  - $score(x) = hscore(x) + oscore(x)$.
  - $hscore(x)$ ($oscore(x)$) indicates the decrease of the total penalty of falsified hard constraints (objective function) caused by flipping $x$.

- **Example**

$$min: 10 \cdot x_1 + 20 \cdot x_2 + 30 \cdot x_3$$
$$\text{st. } c: 2 \cdot x_1 + 3 \cdot x_2 + 4 \cdot x_3 \geq 5$$

Suppose $w(c) = 2, w(oc) = 1$, current assignment $(x_1, x_2, x_3) = (1,0,0)$

| | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| $hscore$ | $2 \times (-2)$ | $2 \times 3$ | $2 \times 3$ |
| $oscore$ | $1 \times 10$ | $-1 \times 20$ | $-1 \times 30$ |

# LS-PBO

**Algorithm 1:** LS-PBO

**Input**: PBO instance $F$, cutoff time *cutoff*

**Output**: A solution $\alpha$ of $F$ and its objective value

**begin**
- $\alpha^* := \emptyset, \quad obj^* := +\infty;$
- $\alpha :=$ all variables are set to 0;
- **while** *elapsed time* $<$ *cutoff* **do**
  - **if** $\alpha$ *is feasible* **and** $obj(\alpha) < obj^*$ **then** $\alpha^* := \alpha;\ obj^* := obj(\alpha)$ ;
  - **if** $D := \{x | score(x) > 0\} \neq \emptyset$ **then**
    - $x :=$ a variable in $D$ with the highest score;
  - **else**
    - update constraint weights using Weighting-PBO;
    - **if** $\exists$ *unsatisfied hard constraints* **then**
      - $c :=$ a randomly chosen unsatisfied hard constraint;
      - $x :=$the variable with highest score in $c$;
    - **else**
      - $x :=$a randomly chosen variable with $oscore(x) > 0$;
  - $\alpha := \alpha$ with $x$ flipped;
- **return** $(\alpha^*,\ obj^*)$

# Dynamic Scoring Mechanism

$$score(x) = hscore(x) + oscore(x)$$

↓

$$score^*(x) = hscore(x) + p \cdot oscore(x)$$

Lack of dynamic adjustments to the ratio of the soft and hard constraints

- $p$ is initially set to 1 and updated every $K$ steps.
- If no feasible solution is found during the recent $K$ steps, $p = p \div inc$ $(inc > 1)$.
- Otherwise $p = p \times inc$.

# Dynamic Scoring Mechanism

- **Example**

$$min: \ 10 \cdot x_1 + 20 \cdot x_2 + 30 \cdot x_3$$
$$\text{st.} \ \ c: 2 \cdot x_1 + 3 \cdot x_2 + 4 \cdot x_3 \geq 5$$

Suppose $w(c) = 2, w(oc) = 1$, current assignment $(x_1, x_2, x_3) = (1,0,0)$

| | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| $hscore$ | $-4$ | $6$ | $6$ |
| $oscore$ | $10$ | $-20$ | $-30$ |

- Situation 1: Feasible solutions are found frequently in recent period. (Suppose $p = 2$)

| | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| $score^*$ | $16$ | $-34$ | $-54$ |

- Situation 2: No feasible solution can be found in recent period. (Suppose $p = 0.01$ )

| | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| $score^*$ | $-3$ | $4$ | $3$ |

# Comparison Results

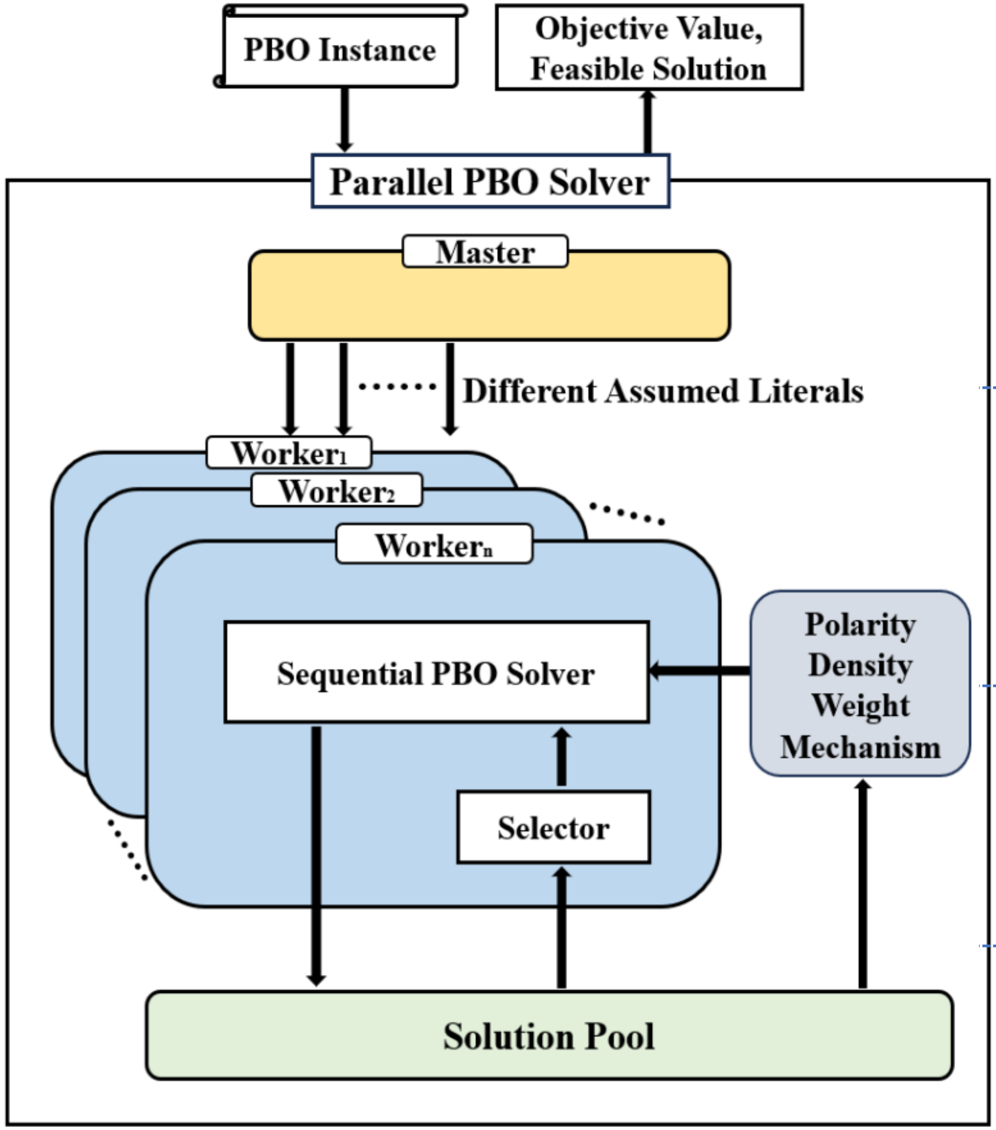
- **Compare with LS-PBO**

| Benchmark | #Ins | *LS-PBO* | | *DLS-PBO* | |
|---|---|---|---|---|---|
| | | $\#win$ | $avg_{sc^*}$ | $\#win$ | $avg_{sc^*}$ |
| cutoff=300s | | | | | |
| Real-World | 63 | 25 | 0.976 | **48** | **0.996** |
| miplib | 252 | 118 | 0.777 | **182** | **0.836** |
| PB16 | 1524 | 711 | 0.692 | **1124** | **0.776** |
| Total | 1839 | 854 | 0.713 | **1354** | **0.792** |
| cutoff=3600s | | | | | |
| Real-World | 63 | **38** | **0.991** | 35 | 0.988 |
| miplib | 252 | 121 | 0.81 | **186** | **0.863** |
| PB16 | 1524 | 829 | 0.753 | **1189** | **0.825** |
| Total | 1839 | 988 | 0.769 | **1410** | **0.836** |

- **Compare with Other Sequential Sota Solvers**

| Benchmark | #Ins | *SCIP* | | *HYBRID* | | *PBO-IHS* | | *NuPBO* | | *Gurobi*(Comp.) | | *Gurobi*(Heur.) | | *DLS-PBO* | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\#win$ | $avg_{sc^*}$ | $\#win$ | $avg_{sc^*}$ | $\#win$ | $avg_{sc^*}$ | $\#win$ | $avg_{sc^*}$ | $\#win$ | $avg_{sc^*}$ | $\#win$ | $avg_{sc^*}$ | $\#win$ | $avg_{sc^*}$ |
| cutoff=300s | | | | | | | | | | | | | | | |
| Real-World | 63 | 0 | 0.126 | 3 | 0.109 | 2 | 0.266 | **46** | **0.972** | 4 | 0.289 | 4 | 0.292 | 29 | 0.977 |
| MIPLIB | 252 | 88 | 0.614 | 53 | 0.572 | 81 | 0.741 | 116 | 0.854 | 152 | 0.838 | **165** | **0.849** | 101 | 0.803 |
| PB16 | 1524 | 810 | 0.687 | 663 | 0.624 | 882 | 0.804 | 980 | 0.813 | 1071 | 0.84 | **1072** | **0.84** | 842 | 0.741 |
| cutoff=3600s | | | | | | | | | | | | | | | |
| Real-World | 63 | 0 | 0.171 | 11 | 0.494 | 5 | 0.401 | **43** | **0.997** | 11 | 0.34 | 9 | 0.38 | 27 | 0.974 |
| MIPLIB | 252 | 113 | 0.675 | 65 | 0.697 | 96 | 0.789 | 113 | 0.859 | 171 | 0.893 | **181** | **0.895** | 106 | 0.831 |
| PB16 | 1524 | 906 | 0.734 | 729 | 0.715 | 939 | 0.814 | 1012 | 0.822 | 1138 | 0.86 | **1144** | **0.862** | 940 | 0.797 |

# Framework



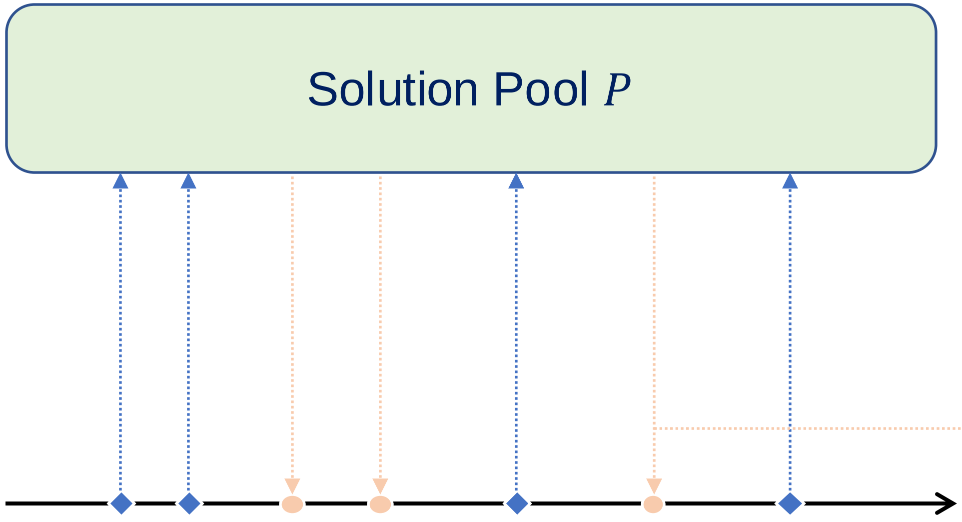

**Literal Assume**

the occurrence of polarities of variables in high-quality solution

**Information to guild the search process**

high-quality solution

# Solution Pool




Search progress of a single thread

- Find a feasible solution → Update $P$
- Get stuck in a local optimal → Replacing by a better solution from $P$

$$div(S) = \sum_{\{S' \in P\}} Hamming(S, S')$$

$$r_{mix}(S) = rank_{obj}(S) \cdot p^* + rank_{div}(S) \cdot (1 - p^*)$$

Let $\{S_1, \ldots, S_k\}$ denotes the better solution from $P$, the probability of selecting $S_i$ is

$$\frac{Obj(S_{cur}) - Obj(S_i)}{\sum_{j=1}^{k} Obj(S_{cur}) - Obj(S_j)}$$

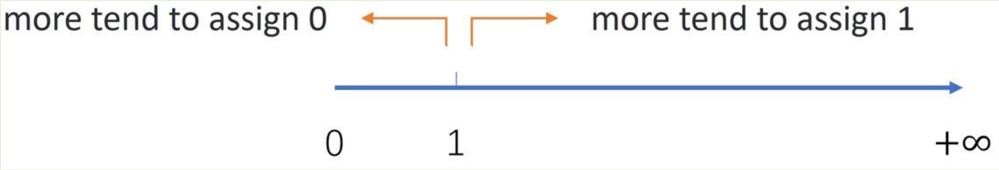
# Polarity Density Weight

$w_{pd}(x)$ : polarity density weight for a variable $x$ ⇒ Reflects the preference of certain polarity of $x$ appearing in high-quality solutions.

**Global Function !**

- Update: once a solution $S$ is added to the $P$

$$w_{pd}(x) = \begin{cases} \max(w_{pd}(x) - \beta, 1 - \epsilon), & \text{if } x = 0 \text{ in } S \\ \min(w_{pd}(x) + \beta, 1 + \epsilon), & \text{if } x = 1 \text{ in } S \end{cases}$$




$score^{**}(x)$ : an enhanced scoring function

$$score^{**}(x) = \begin{cases} score^{*}(x) \cdot w_{pd}(x), & \text{if } x = 0 \text{ in } S_{cur} \\ score^{*}(x) / w_{pd}(x), & \text{if } x = 1 \text{ in } S_{cur} \end{cases}$$

# Polarity Density Weight

- **Example**

$$min: \ 10 \cdot x_1 + 20 \cdot x_2 + 30 \cdot x_3$$
$$\text{st.} \ \ c: 2 \cdot x_1 + 3 \cdot x_2 + 4 \cdot x_3 \geq 5$$

Suppose $w(c) = 2, w(oc) = 1$, current assignment $(x_1, x_2, x_3) = (1,0,0), \ p = 1$

| | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| $score^*$ | $6$ | $-14$ | $-24$ |

Suppose $w_{pd}(x_1) = 1.1$, $w_{pd}(x_2) = 1.1$, $w_{pd}(x_3) = 0.9$

| | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| $score^{**}$ | $6 \div 1.1$ | $(-14) \times 1.1$ | $(-24) \times 0.9$ |

# Comparison Results

| Benchmark | Category | #ins | *FiberSCIP* | *Gurobi* Comp. | *Gurobi* Heur. | *ParLS-PBO* |
|---|---|---|---|---|---|---|
| cutoff=300s | | | | | | |
| Real-World | MWCB | 24 | 0 | 0 | 0 | **24** |
| | WSNO | 18 | 0 | 4 | 4 | **18** |
| | SAP | 21 | 0 | 0 | 0 | **21** |
| | Total | 63 | 0 | 4 | 4 | **63** |
| MIPLIB | Total | 252 | 113 | **190** | 180 | 129 |
| PB16 | Factor | 192 | **186** | **186** | **186** | 172 |
| | Kexu | 40 | 6 | 10 | 7 | **40** |
| | Logic synthesis | 74 | 71 | **73** | 72 | **73** |
| | Market split | 40 | 12 | **21** | 13 | 5 |
| | Mps | 35 | 30 | 33 | **34** | 23 |
| | Numerical | 34 | 13 | 18 | **21** | 8 |
| | Prime | 156 | 123 | 128 | 129 | **131** |
| | Reduced mps | 273 | 76 | 145 | **150** | 39 |
| | Total | 1524 | 898 | **1147** | 1143 | 1100 |

| Benchmark | Category | #ins | *FiberSCIP* | *Gurobi* Comp. | *Gurobi* Heur. | *ParLS-PBO* |
|---|---|---|---|---|---|---|
| cutoff=3600s | | | | | | |
| Real-World | MWCB | 24 | 0 | 5 | 2 | **20** |
| | WSNO | 18 | 0 | 10 | 10 | **18** |
| | SAP | 21 | 0 | 0 | 0 | **21** |
| | Total | 63 | 0 | 15 | 12 | **59** |
| MIPLIB | Total | 252 | 129 | 184 | **193** | 140 |
| PB16 | Factor | 192 | **186** | **186** | **186** | 182 |
| | Kexu | 40 | 14 | 17 | 14 | **40** |
| | Logic synthesis | 74 | 72 | 72 | 72 | **74** |
| | Market split | 40 | 16 | **22** | 12 | 8 |
| | Mps | 35 | 30 | **33** | **33** | 25 |
| | Numerical | 34 | 13 | 19 | **25** | 8 |
| | Prime | 156 | 127 | 130 | 131 | **132** |
| | Reduced mps | 273 | 100 | 150 | **160** | 43 |
| | Total | 1524 | 995 | 1198 | **1201** | 1107 |

# Comparison Results

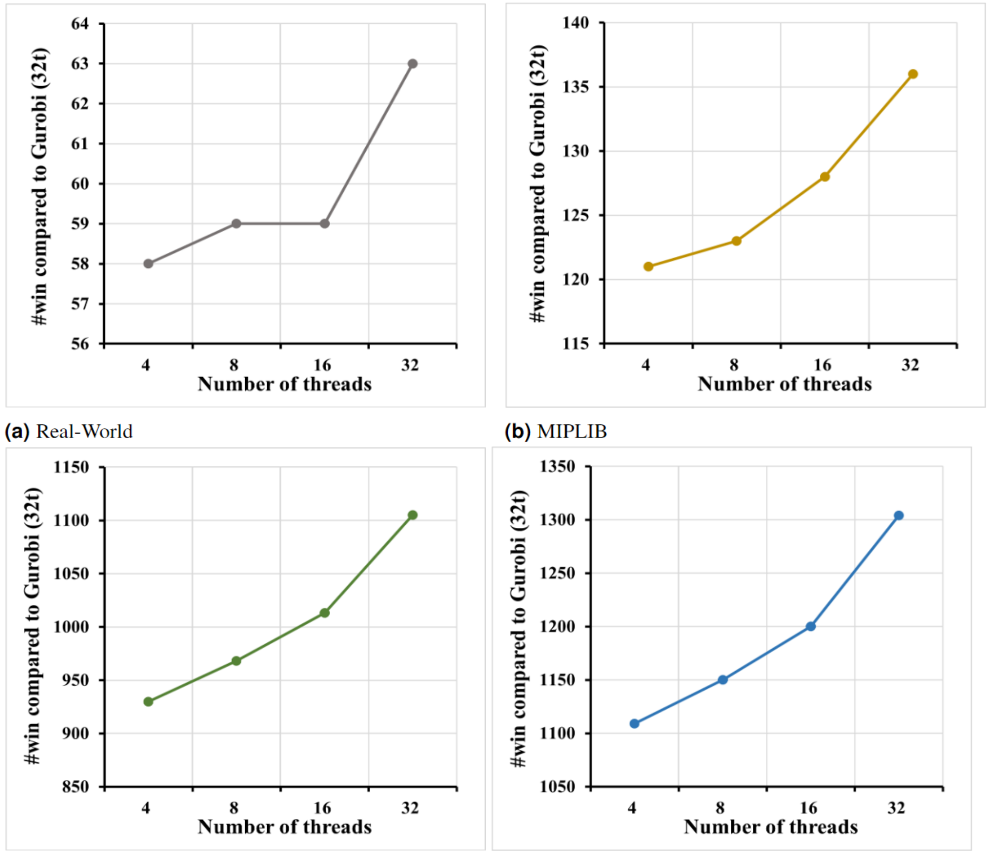
## • Effectiveness Analysis

- $V_1$ : disable the sharing mechanism.
- $V_2$ : disable global score mechanism.

| Benchmark | #Ins | $V_1$ vs. *ParLS-PBO* | | | | $V_2$ vs. *ParLS-PBO* | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $V_1$ | | *ParLS-PBO* | | $V_2$ | | *ParLS-PBO* | |
| | | $\#win$ | $avg_{sc^*}$ | $\#win$ | $avg_{sc^*}$ | $\#win$ | $avg_{sc^*}$ | $\#win$ | $avg_{sc^*}$ |
| cutoff=300s | | | | | | | | | |
| Real-World | 63 | 28 | 0.993 | **52** | **0.998** | 36 | 0.996 | **44** | **0.998** |
| MIPLIB | 252 | 137 | 0.857 | **199** | **0.868** | 166 | **0.871** | **185** | 0.870 |
| PB16 | 1524 | 1095 | 0.835 | **1231** | **0.847** | 1168 | 0.841 | **1182** | **0.844** |
| cutoff=3600s | | | | | | | | | |
| Real-World | 63 | 19 | 0.981 | **62** | **1.0** | 37 | 0.995 | **48** | **0.999** |
| MIPLIB | 252 | 145 | 0.870 | **203** | **0.897** | **181** | **0.897** | 180 | 0.893 |
| PB16 | 1524 | 1111 | 0.840 | **1251** | **0.854** | 1191 | 0.852 | **1214** | **0.852** |

## • Scalability Analysis

- Baseline: *Gurobi* (32 threads).




(a) Real-World
(b) MIPLIB
(c) PB16
(d) Total

# Conclusion and Future Work

Applied the ideas of this work to other problems (SAT, MaxSAT, ……).

Implement a distributed version of ParLS-PBO for cloud computation

ParLS-PBO: A Parallel Local Search Solver for Pseudo Boolean Optimization

Our code: https://github.com/shaowei-cai-group/ParLS-PBO

# Thank You!
# Q&A