Different Solving Strategies on PBO Problems from Automotive Industry

Thore Kübart[1] Rouven Walter[1,2] Wolfgang Küchlin[1,2]

[1]Steinbeis Technology Transfer Centre, STZ OIT, Tübingen, Germany

[2]Symbolic Computation Group, WSI Informatics University of Tübingen, Germany

17th International Configuration Workshop
10th – 11th September 2015, Vienna, Austria

Overview

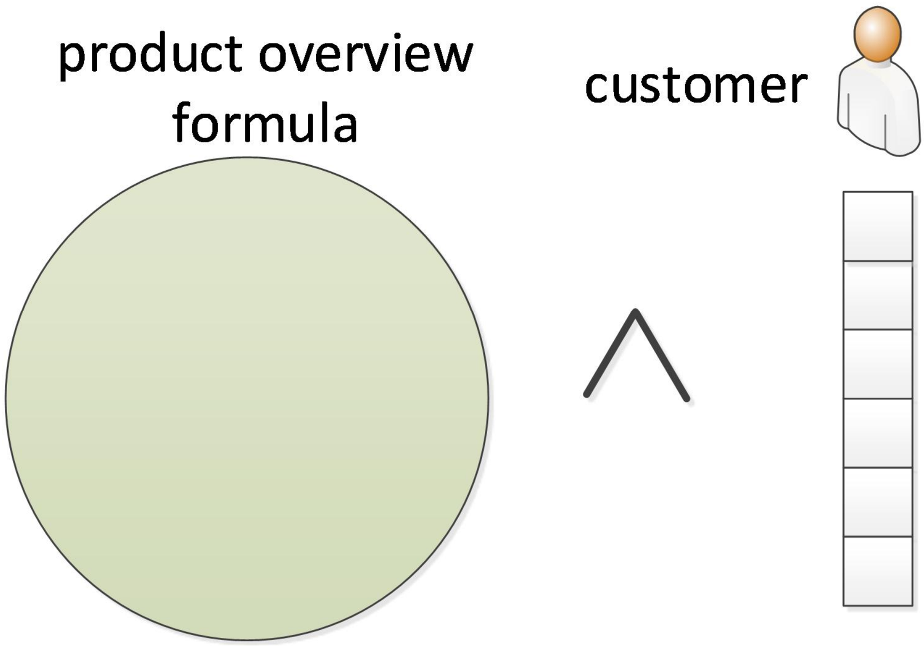
- **PBO Problems from Automotive Industries**
- Different Solving Approaches for PBO
- Experimental Results

Product Overview Formula

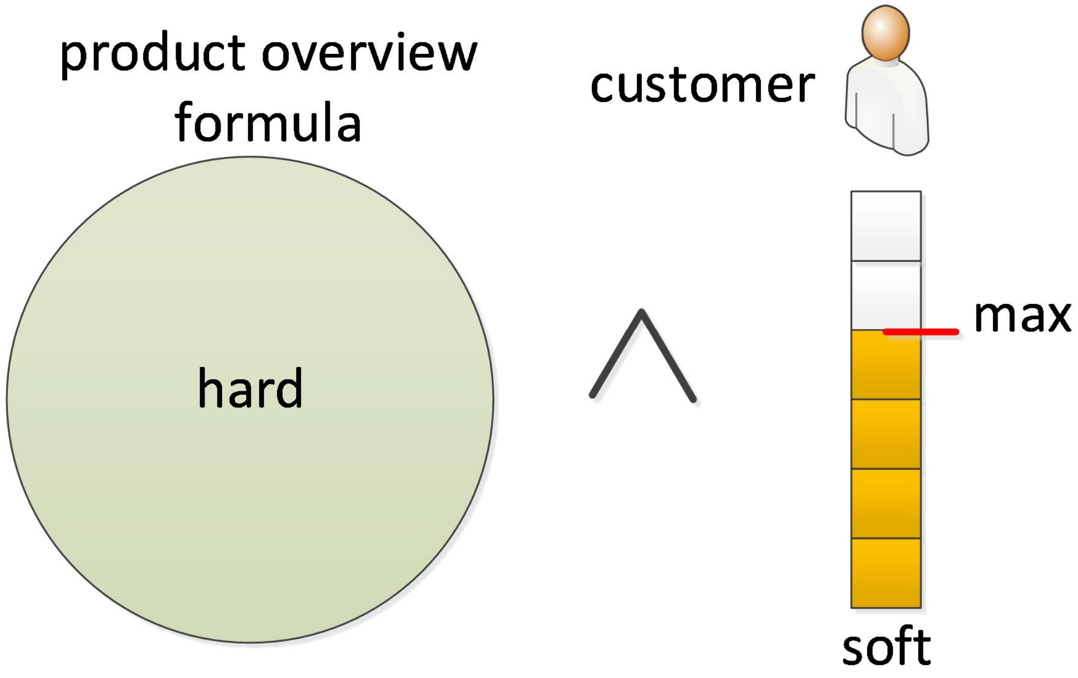
$$M = \{\beta \mid \beta(\phi_{POF}) = 1\}$$

- M = Set of technically feasible vehicle configurations
- ϕ_{POF} = Boolean Product Overview Formula [KS2000]

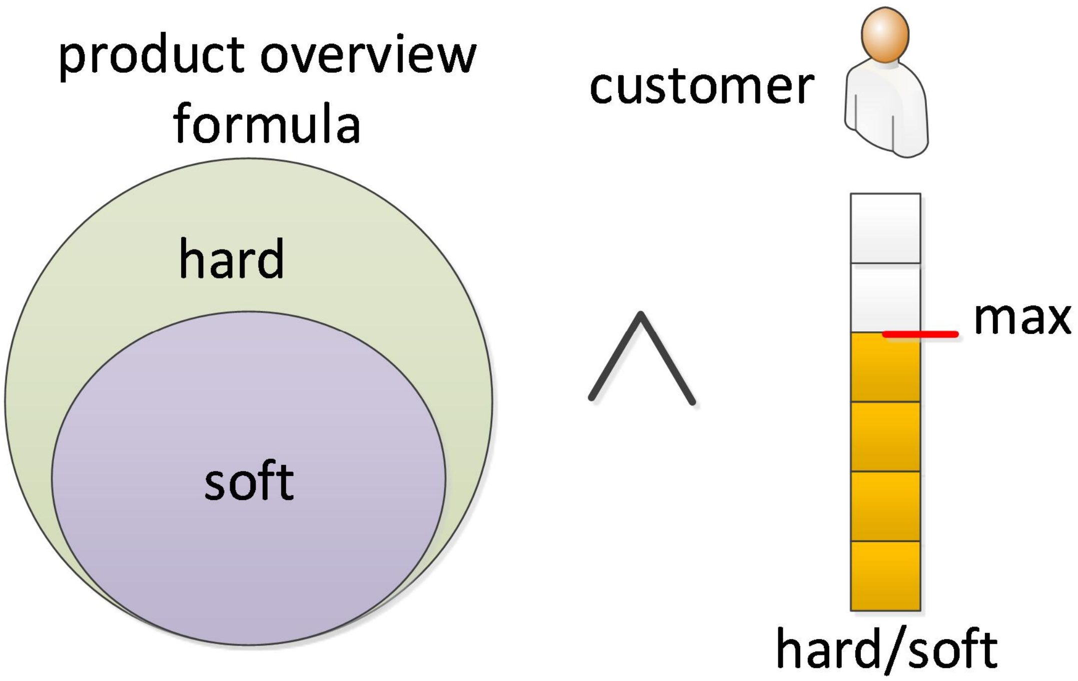
Re-Configuration



Re-Configuration (cont'd)

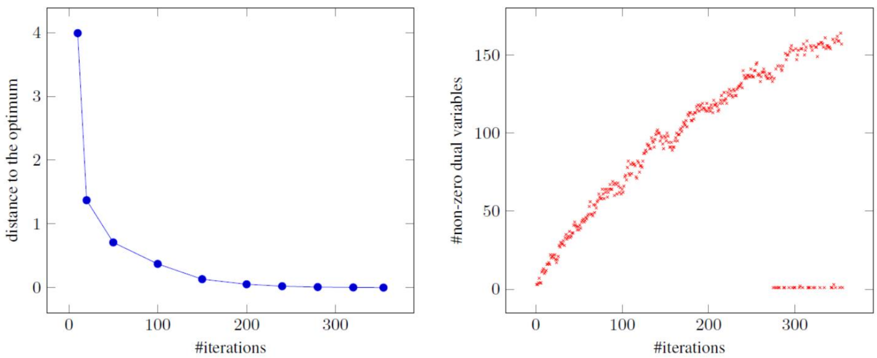


Re-Configuration (cont'd)



Production Planning

- Calculation of configurations before sales orders are received [SR2013]
- Applying Branch & Price to an ILP:
  - $A \in \{0,1\}^{m \times n}$ contains all models of ϕ_{POF} as 0,1-columns
  - min $\sum_i c_i \cdot x_i$, s.t. $A \cdot x = b,\ x \in \{0,1\}^n$



Column Generation Call

Overview

- PBO Problems from Automotive Industries
- **Different Solving Approaches for PBO**
- Experimental Results

Partial Weighted MaxSAT

- Product overview formula: ϕ_{POF}
- Objective Function: $\sum_{i=1}^{m} w_i \cdot o_i$

- Hard clauses: $\text{Tseitin}(\phi_{POF}) = \{c_1, \dots, c_n\}$
- Weighted Soft Clauses: $\{(w_1, o_1), \dots, (w_m, o_m)\}$

- Unsatisfiable Core Guided approach: Open-WBO

Pseudo-Boolean Optimization (PBO)

- Product overview formula: ϕ_{POF}
- Objective Function: $\sum_{i=1}^{m} w_i \cdot o_i$

- Linear Pseudo-Boolean Constraint:

$$\sum_{i} d_i l_i \geq b\,, \qquad d_i \in \mathbb{Z}, l_i\ literal$$

- PBO: Optimize objective function $\sum_i a_i l_i$

Algorithm 1: PBO: PBS-Based Linear Search

Input: PBS formula φ, target function $z = \sum_i a_i l_i$
Output: Model β of φ which maximizes z

while $SAT(\varphi)$ **do**
  $\beta \leftarrow \text{getModel}(\varphi)$
  $\varphi \leftarrow \varphi \wedge (\sum_i a_i l_i \geq z(\beta) + 1)$
return β

Integer Linear Programming

- Product overview formula: ϕ_{POF}
- Objective Function: $\sum_{i=1}^{m} w_i \cdot o_i$

- Convert ϕ_{POF} to ILP inequalities
- Example: Conversion of a clause:
  - $(l_1 \vee \neg l_2 \vee l_3)$
  - $l_1 + \neg l_2 + l_3 \geq 1$
  - $l_1 + (1 - l_2) + l_3 \geq 1$
  - $l_1 - l_2 + l_3 \geq 0$

- State-of-the-art ILP solver: CPLEX / Gurobi (Branch and Cut)

Overview

- PBO Problems from Automotive Industries
- Different Solving Approaches for PBO
- **Experimental Results**

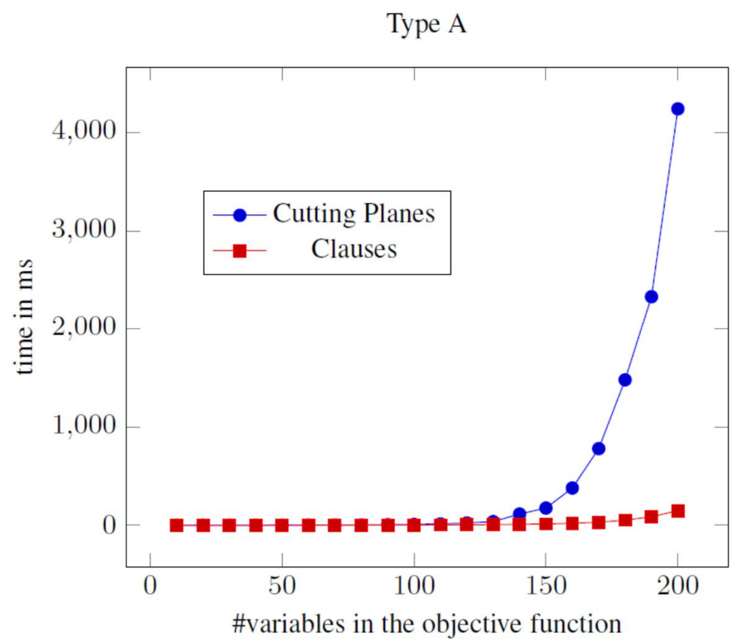
Test Environment

- Windows 7 Professional 64 Bit on an Intel(R) Core(TM) i7-4800MQ, CPU with 2.70 GHz and 2 GB main memory

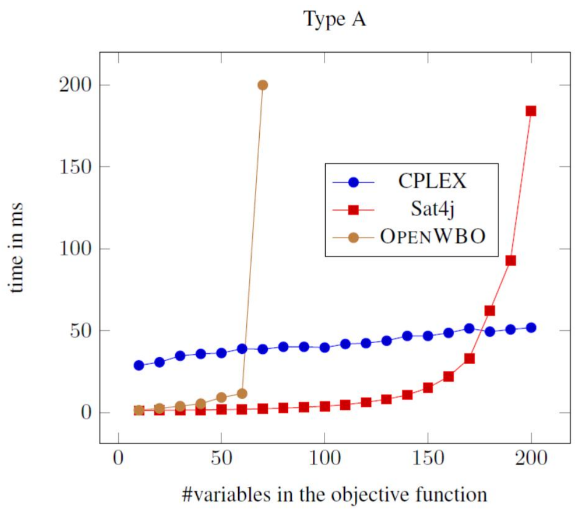
Type	Fixed Attributes	#Clauses
A	market, model, body type, engine, steering type	1200-5900
B	market, model, body type	19100-137700

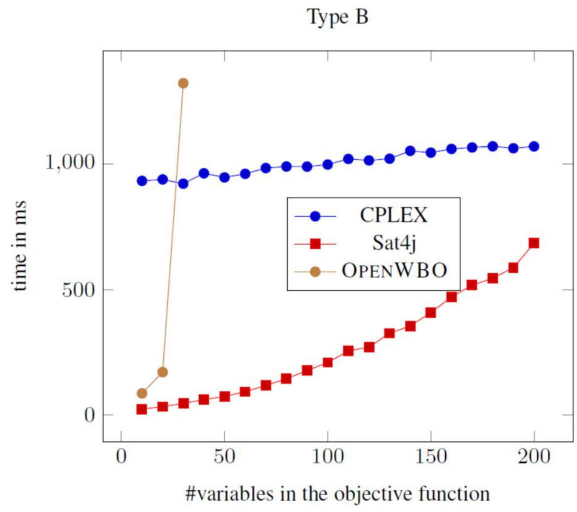
- Test Instances: 13 POFs of type A, 6 POFs of type B
- Random objective functions:
  - Objective function size $n = 10, 20, \ldots, 200$
  - Random integer coefficients from $[-10^7, 10^7]$
  - 20 objective functions for each n

SAT4J: Learning Cutting Planes vs. Clauses

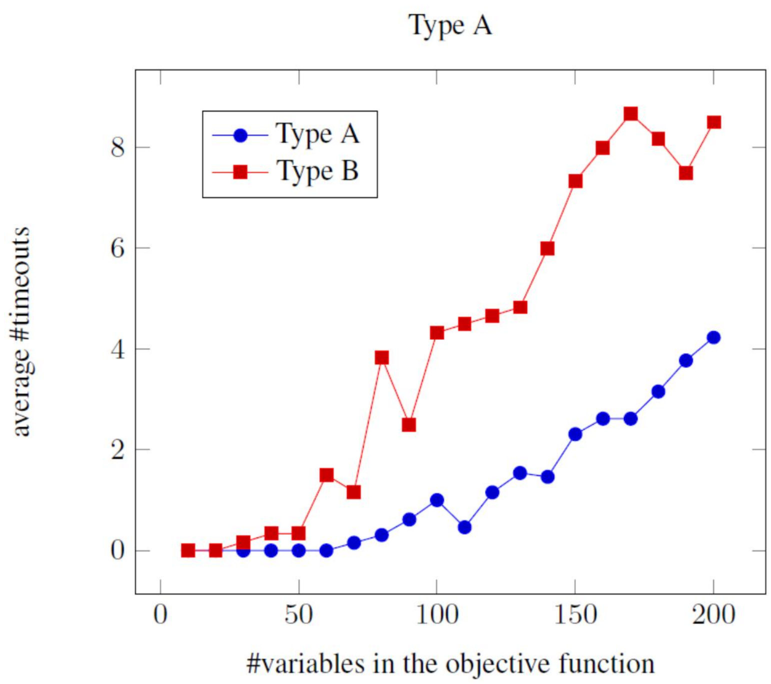


P.W. MaxSAT vs. PBO (Linear Search) vs. ILP

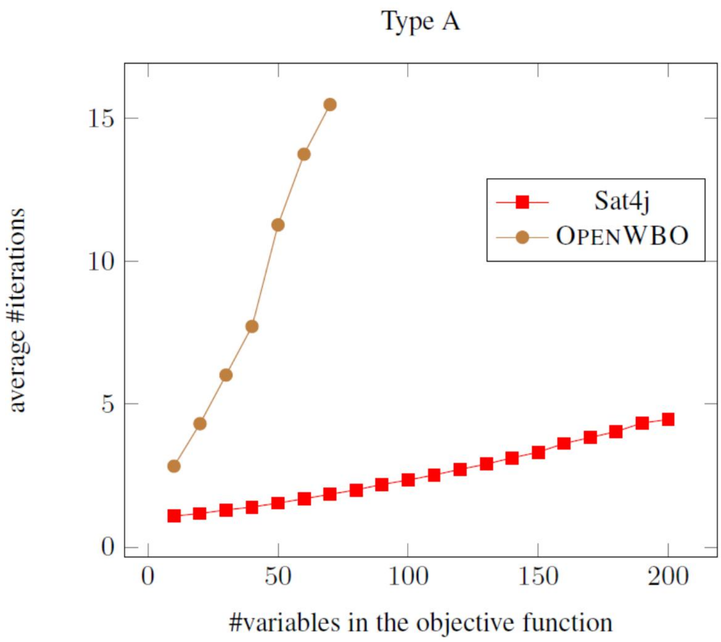




Open-WBO: Timeouts (60s)



Consistency Calls for MaxSAT and PBO (Linear Search)



Summary

- DPLL-based methods:
  - Quick calculation of a model of ϕ_{POF}
  - SAT4J: Learning clauses is more effective than learning Cutting Planes
  - Strong increase of running times, once a certain length of objective functions has been reached
- CPLEX can handle more complex objective functions (more robust)
- Number of iterations for a linear search increases only linearly with the numbers of variables in the objective function
- Wanted: More efficient PBS-Solver for a set of product overview clauses and one extensive Linear PBC.

Summary

- DPLL-based methods:
  - Quick calculation of a model of ϕ_{POF}
  - SAT4J: Learning clauses is more effective than learning Cutting Planes
  - Strong increase of running times, once a certain length of objective functions has been reached
- CPLEX can handle more complex objective functions (more robust)
- Number of iterations for a linear search increases only linearly with the numbers of variables in the objective function
- Wanted: More efficient PBS-Solver for a set of product overview clauses and one extensive Linear PBC.

Thank you for your attention

Bibliography

- [SR2013] Singh, Rangaraj. **Generation of predictive configurations for production planning.** In *Proc. of the 15th Intern. Configuration Workshop*, 2013
- [KS2000] Küchlin, Sinz. **Proving consistency assertions for automotive product data management**. *Journal of Automated Reasoning*, 2000