WITH THE POWER OF KNOWLEDGE – FOR THE WORLD

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Multi-Objective MaxSAT Solving

Christoph Jabs Jeremias Berg Matti Järvisalo

Department of Computer Science, University of Helsinki, Finland

August 12th 2024

Overview

What will I be talking about?

An overview of our recent work on multi-objective combinatorial optimization

- Algorithms for multi-objective optimization
  - *P*-minimal: multi-objective solution-improving search
  - BiOptSat: specifically for two objectives
- Preprocessing
  - Techniques from (Max)SAT
  - Core boosting

`christophjabs.info`

Multi-Objective Optimization

Conflicting objectives are important, but hard

- Optimization is commonplace
- Conflicting objectives are common
- Optimality matters

Problem setting

Finding optimal solutions to NP-hard multi-objective optimization problems

Real-world multi-objective problems

- Supply chain optimization [Pinto-Varela et al. 2011]
- Interpretable classifiers [Malioutov and Meel 2018]
- Network routing [Silvério et al. 2022]
- ...

Example — Interpretable Decision Rules

Task: Find decision rules that minimize size and classification error

Definition (Decision Rule)

Propositional formula over binary features. Result of formula represents binary classification.

Pareto-optimal decision rules

Dataset

x_1	x_2	y
1	1	1
0	1	0
1	0	0

Decision rules

$$r_1 = x_1 \land x_2$$

$$r_2 = x_1$$

Single- vs. Multi-Objective Optimization

[Jabs et al. 2024b]

Single-Objective Optimization

- Unique optimal cost value

Multi-Objective Optimization

- What even is optimality?

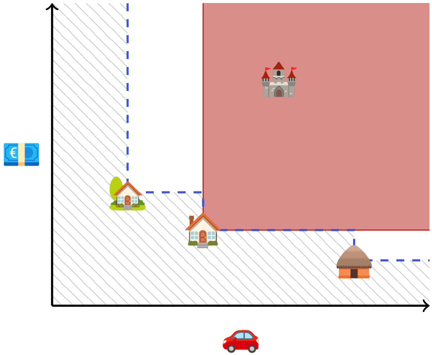
MO Notions of Optimality

- Pareto optimality
  (Every solution without an *obviously* better one is optimal)
- Lexicographic optimality
  (Optimize in order of importance / Lexicographically compare solutions)
- Leximax optimality
  (Optimize worst objective first / *Sort* and lexicographically compare)

What Even Is "Optimal"?

Pareto-optimal solutions

	💶	🚗
🏠	1000 €	10 km
🏰	3000 €	15 km
🛖	600 €	20 km
🏡	1500 €	5 km



Bi/Multi-Objective MaxSAT

Definition (Bi-Objective MaxSAT)

- (Hard) clauses F
  (SAT encoding of the constraints)
- 2 linear PBO objectives O_I, O_D
  (Weighted relaxation literals)

Toy Instance

$$
\begin{aligned}
F &= \texttt{As-CNF}\left(\sum_{l \in O_I \cup O_D} \tau(l) \geq 3\right), \\
O_I &= i_1 + i_2 + i_3, \\
O_D &= d_1 + d_2 + d_3
\end{aligned}
$$

Our Solvers

What you can use to solve a multi-objective MaxSAT instance

BiOptSat 🛠️

- *Bi*-objective solver
- C++ implementation
- Implements BiOptSat and *P*-minimal
- `bitbucket.org/coreo-group/bioptsat`

Scuttle 🛠️

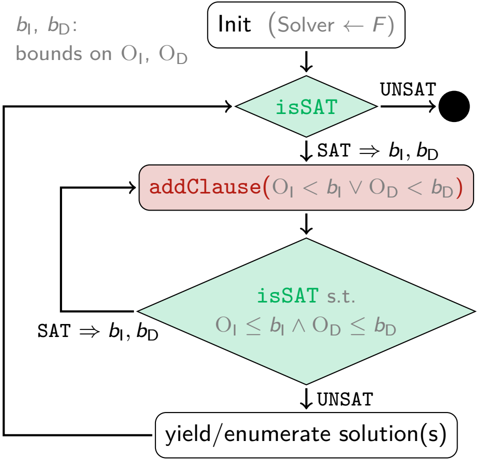
- *Multi*-objective solver
- Rust implementation
- Implements various algorithms, including BiOptSat and *P*-minimal
- `bitbucket.org/coreo-group/scuttle`

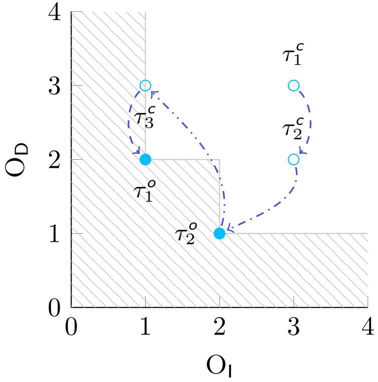
- SAT solver: CaDiCaL [Biere et al. 2020]
- Card/PB encodings: (Generalized) totalizer [Bailleux and Boufkhad 2003; Joshi et al. 2015]

P-Minimal: "Multi-Objective Solution-Improving Search"

[Soh et al. 2017]

b_I, b_D: bounds on O_I, O_D

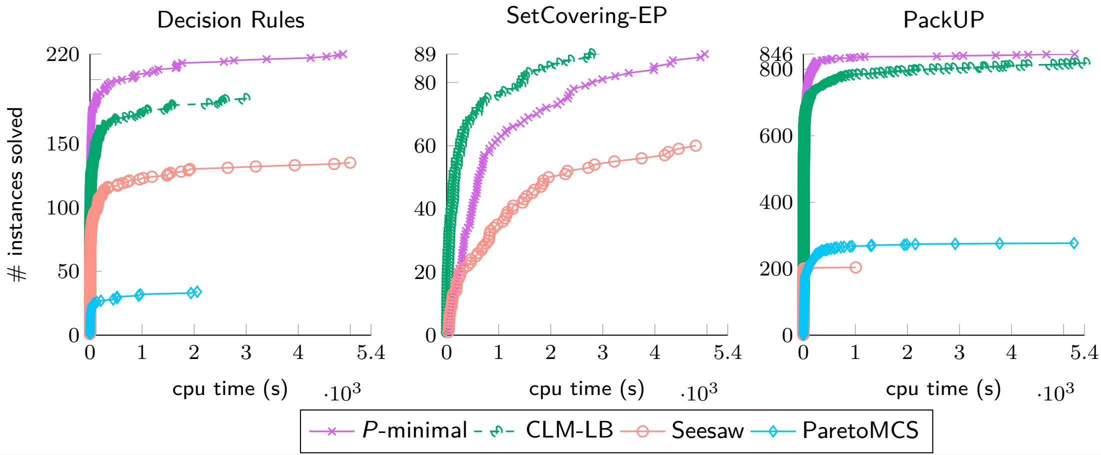




P-Minimal: Results

(BiOptSat 🛠️)

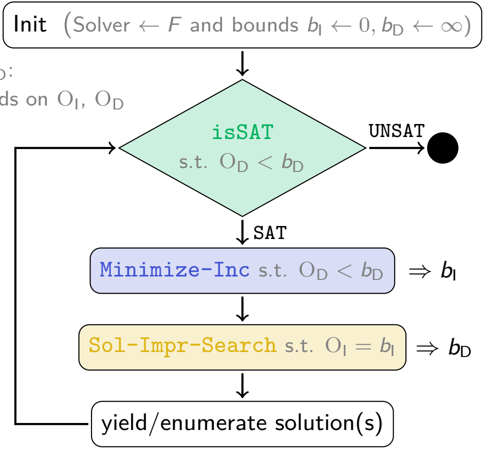
[Jabs et al. 2024b]

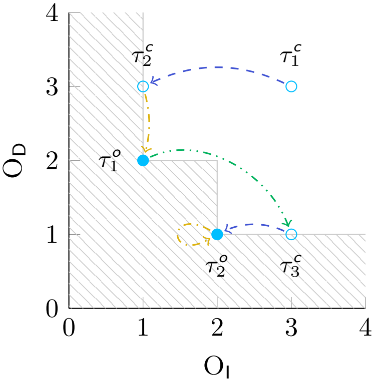


BiOptSat: The Lexicographic Method

[Jabs et al. 2024b]

b_I, b_D:
bounds on O_I, O_D





BiOptSat SAT-UNSAT

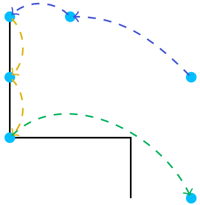
Sol-Impr-Search always solution improving search

- Lower-bounding impractical because of decreasing objective values
- Loosening of constraints on O_l can invalidate cores
- Totalizer over all objective literals

Simplest variant of Minimize-Inc: SAT-UNSAT

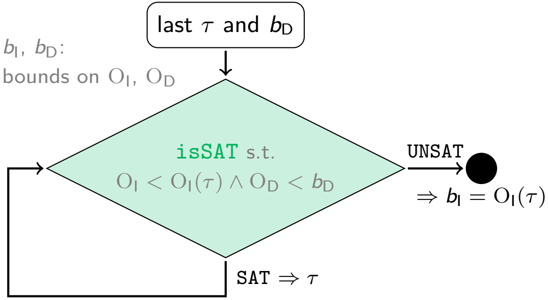
- Totalizer over objective literals
- Enforce $O_l < b_l$ with assumptions until UNSAT

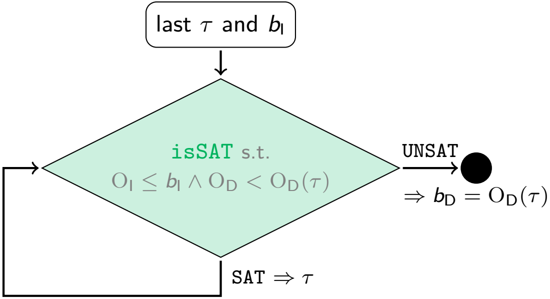
BiOptSat SAT-UNSAT



BiOptSat SAT-UNSAT: Details

Minimize-Inc (SAT-UNSAT) **Sol-Impr-Search**

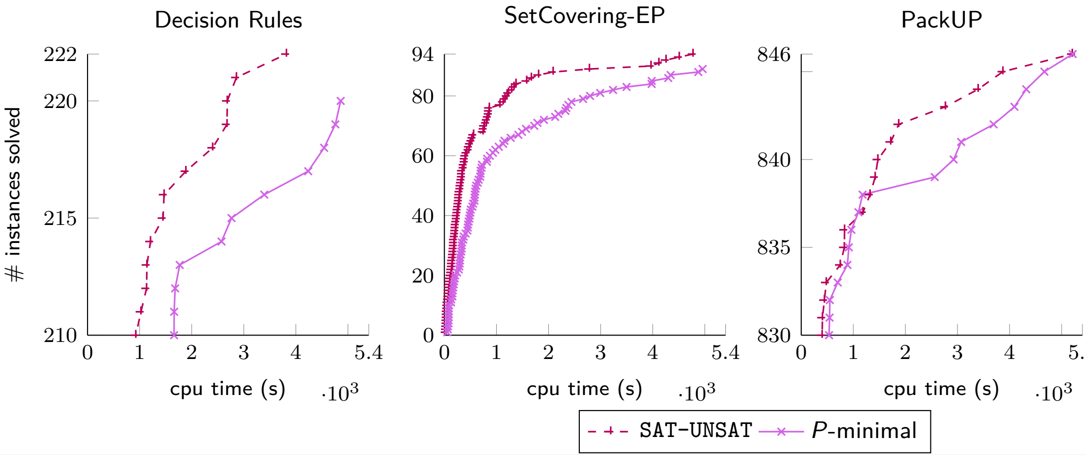




BiOptSat SAT-UNSAT: Results

BiOptSat 🛠️

[Jabs et al. 2024b]



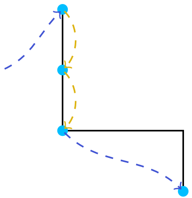
BiOptSat Lower-Bounding

Lower-bounding `Minimize-Inc`

- Incrementally built totalizer over objective literals
- `UNSAT-SAT`: increase bound until `SAT`
- `MSU3` & `OLL`: core guided search
- Next iteration, continue from last found optimum

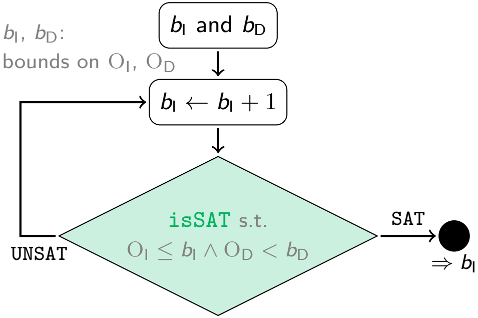
`Sol-Impr-Search` unchanged

BiOptSat `UNSAT-SAT` / `MSU3` / `OLL`



BiOptSat Lower-Bouning: Details

`Minimize-Inc` (UNSAT-SAT)



`Minimize-Inc` (MSU3)

$(\text{res}, \tau, \kappa) \leftarrow \texttt{isSAT}(\{\langle \mathrm{O_D} < b_\mathrm{D} \rangle, \langle \mathrm{O_I} \leq b_\mathrm{I} \rangle\} \cup \{\neg l \mid l \in \mathrm{O_I} \setminus \texttt{Act}\})$

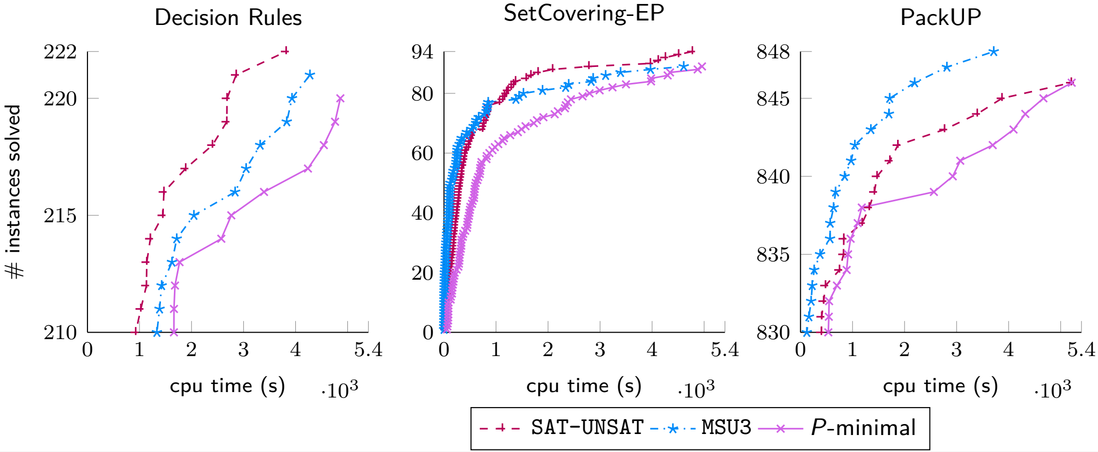
while res $=$ UNSAT **do**

- $b_\mathrm{I} \leftarrow b_\mathrm{I} + 1$
- $\kappa \leftarrow \kappa \setminus \{\neg\langle \mathrm{O_D} < b_\mathrm{D} \rangle, \neg\langle \mathrm{O_I} \leq b_\mathrm{I} \rangle\}$
- $\texttt{Act} \leftarrow \texttt{Act} \cup \kappa$
- build or extend $\textsc{Tot}(\texttt{Act}, b_\mathrm{I})$
- $(\text{res}, \tau, \kappa) \leftarrow \texttt{isSAT}(\{\langle \mathrm{O_D} < b_\mathrm{D} \rangle, \langle \mathrm{O_I} \leq b_\mathrm{I} \rangle\} \cup \{\neg l \mid l \in \mathrm{O_I} \setminus \texttt{Act}\})$

return b_I

BiOptSat Lower-Bounding: Results

(BiOptSat 🛠️) [Jabs et al. 2024b]



Hybrids: Avoiding UNSAT-SAT in Core-Guided Search

MSHybrid and OSHybrid

Search in Minimize-Inc continues after first optimum

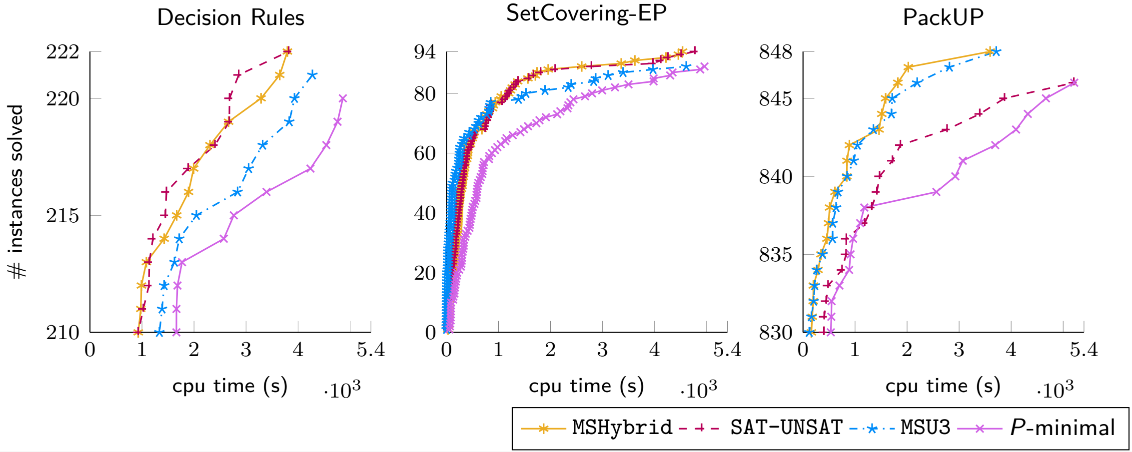
- Often all literals active (core-guided variants)
- MSU3 with all literals active = UNSAT-SAT
- Performance suffers
- $\Rightarrow$ Switch to SAT-UNSAT

Hybrid

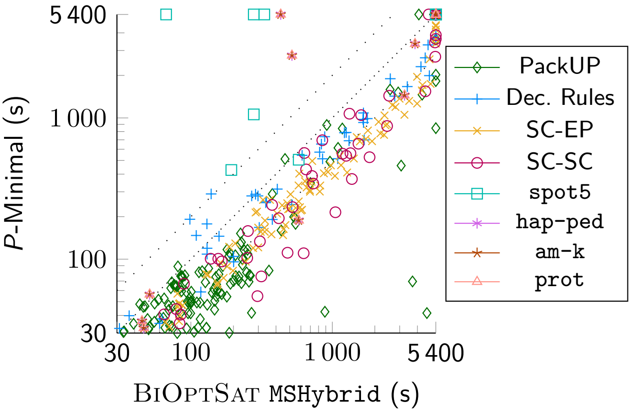
1: **if** $|\mathtt{Act}| < \mathtt{thr} \cdot |O_I|$ **then**
2: MSU3 / OLL
3: **else**
4: SAT-UNSAT

BiOptSat Hybrid: Results

(BiOptSat 🛠️) [Jabs et al. 2024b]



BiOptSat vs. *P*-Minimal

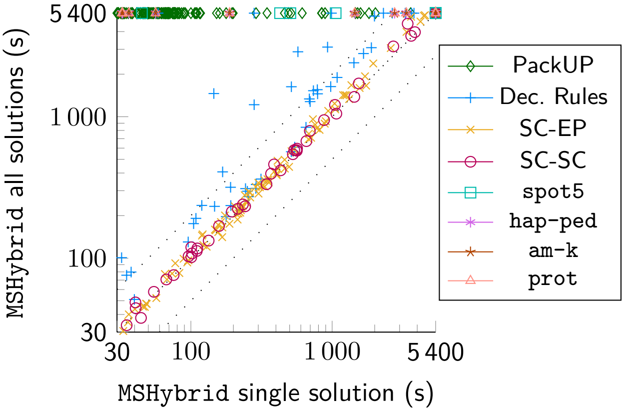


- BiOptSat restricts the SAT solver more
- This restriction allows for improving on the used algorithms
  - Allows lower-bounding (including core-guided) search
  - Hardening of the decreasing objective
- Most of the advantages of BiOptSat limited to two objectives

Enumeration of *All* Pareto-Optimal Solutions

[Jabs et al. 2024b]

- Every cost point can have multiple solutions associated with it
- Both BiOptSat and *P*-Minimal can enumerate all solutions
- Feasibility depends on the instance



Refinement: Domain Specific Blocking

For enumeration of all Pareto-optimal solutions, blocking clauses for solutions are needed

- Generally: block over all variables in F (or all decisions of the SAT solver)
- Improved blocking clauses using domain knowledge
  - Shorter clauses (omitting functionally defined variables)
  - Block multiple symmetric solutions at once (omitting auxiliary variables)

What is Preprocessing

- So far: solving stage
- Now: simplifying the instance before solving

Problem ❓

↓

Encoding

$(x_1 \lor x_2) \land (x_3 \lor \neg x_4) \land \ldots$
$\min x_1 + 3x_2 + x_3 + 2x_4$

↓

Preprocessing

↓

Solving

Solution ❗

↓

"Traditional" Preprocessing

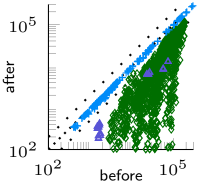
Extending (Max)SAT preprocessing to multiple objectives [Jabs et al. 2023]

- Extend SAT and MaxSAT preprocessing to multi-objective (MO-)MaxSAT
- Open-source preprocessor

Redundancy Notions

- Basis for formal correctness of techniques
- Analysis of expressiveness

Clauses

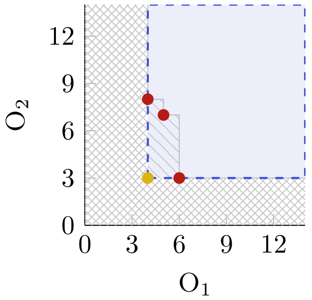


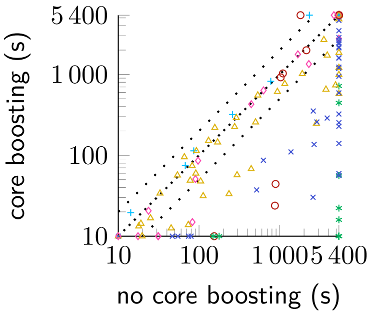
Solver	PackUP	LIDR	DAL
BiOptSat*	$+2.4\%$	$+1.3\%$	–
Scuttle$^+$	$+0.5\%$	-0.4%	$+1.5\%$
CLM-LB$^+$	-0.3%	$\pm 0\%$	-11.7%
leximaxIST†	$+3.1\%$	$\pm 0\%$	$\pm 0\%$

Core Boosting

Using single-objective optimization as reformulation [Jabs et al. 2024a]

- Reformulate via single-objective core-guided search





Summary

- MO-MaxSAT: enumerate Pareto-optimal solutions
- Algorithms
  - BiOptSat
  - *P*-minimal
- Open-source solvers
  - BiOptSat 🛠️
  - Scuttle 🛠️
- Preprocessing
  - Techniques from (Max)SAT
  - Core boostin

Paper, slides, code, and contact:

`christophjabs.info`

Bibliography I

Bailleux, Olivier and Yacine Boufkhad (2003). 'Efficient CNF Encoding of Boolean Cardinality Constraints'. In: *Principles and Practice of Constraint Programming – CP 2003*. Springer Berlin Heidelberg, pp. 108–122. DOI: 10.1007/978-3-540-45193-8_8.

Biere, Armin et al. (2020). 'CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling Entering the SAT Competition 2020'. In: *Proc. of SAT Competition 2020 – Solver and Benchmark Descriptions*. Ed. by Tomas Balyo et al. Vol. B-2020-1. Department of Computer Science Report Series B. University of Helsinki, pp. 51–53.

Jabs, Christoph et al. (2023). 'Preprocessing in SAT-Based Multi-Objective Combinatorial Optimization'. In: *29th International Conference on Principles and Practices of Constraint Programming (CP 2023)*. Ed. by Roland H. C. Yap. Vol. 280. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl—Leibniz-Zentrum für Informatik, 18:1–18:19. DOI: 10.4230/LIPIcs.CP.2023.18.

Bibliography II

Jabs, Christoph et al. (2024a). 'Core Boosting in SAT-Based Multi-objective Optimization'. In: *Integration of Constraint Programming, Artificial Intelligence, and Operations Research—21st International Conference, CPAIOR 2024, Uppsala, Sweden, May 28–31, 2024, Proceedings, Part II*. Ed. by Bistra Dilkina. Vol. 14743. Lecture Notes in Computer Science. Springer, pp. 1–19. ISBN: 978-3-031-60599-4. DOI: 10.1007/978-3-031-60599-4_1. URL: https://link.springer.com/chapter/10.1007/978-3-031-60599-4_1.

Jabs, Christoph et al. (Aug. 2024b). 'From Single-Objective to Bi-Objective Maximum Satisfiability Solving'. In: *Journal of Artificial Intelligence Research* 80, pp. 1223–1269. DOI: 10.1613/jair.1.15333. URL: http://dx.doi.org/10.1613/jair.1.15333.

Janota, Mikolás et al. (2021). 'The Seesaw Algorithm: Function Optimization Using Implicit Hitting Sets'. In: *27th International Conference on Principles and Practice of Constraint Programming, CP 2021, Montpellier, France (Virtual Conference), October 25-29, 2021*. Ed. by Laurent D. Michel. Vol. 210. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 31:1–31:16. DOI: 10.4230/LIPIcs.CP.2021.31. URL: https://doi.org/10.4230/LIPIcs.CP.2021.31.

Bibliography III

Joshi, Saurabh et al. (2015). 'Generalized Totalizer Encoding for Pseudo-Boolean Constraints'. In: *Principles and Practice of Constraint Programming - 21st International Conference, CP 2015, Cork, Ireland, August 31 - September 4, 2015, Proceedings*. Ed. by Gilles Pesant. Vol. 9255. Lecture Notes in Computer Science. Springer, pp. 200–209. DOI: 10.1007/978-3-319-23219-5_15. URL: https://doi.org/10.1007/978-3-319-23219-5_15.

Malioutov, Dmitry and Kuldeep S. Meel (2018). 'MLIC: A MaxSAT-Based Framework for Learning Interpretable Classification Rules'. In: *Principles and Practice of Constraint Programming - 24th International Conference, CP 2018, Lille, France, August 27–31, 2018, Proceedings*. Ed. by John N. Hooker. Vol. 11008. Lecture Notes in Computer Science. Springer, pp. 312–327. DOI: 10.1007/978-3-319-98334-9_21. URL: https://doi.org/10.1007/978-3-319-98334-9_21.

Martins, Ruben et al. (Aug. 2014). 'Incremental Cardinality Constraints for MaxSAT'. In: *Lecture Notes in Computer Science*. Springer International Publishing, pp. 531–548. DOI: 10.1007/978-3-319-10428-7_39. arXiv: 1408.4628 [cs.LO].

Bibliography IV

- Pinto-Varela, Tânia et al. (2011). 'Bi-objective optimization approach to the design and planning of supply chains: Economic versus environmental performances'. In: *Comput. Chem. Eng.* 35.8, pp. 1454–1468. DOI: 10.1016/j.compchemeng.2011.03.009. URL: https://doi.org/10.1016/j.compchemeng.2011.03.009.
- Silvério, Antonio José et al. (Feb. 2022). 'A bi-objective optimization model for segment routing traffic engineering'. en. In: *Ann. Telecommun.* DOI: 10.1007/s12243-022-00907-w. URL: https://doi.org/10.1007/s12243-022-00907-w.
- Soh, Takehide et al. (2017). 'Solving Multiobjective Discrete Optimization Problems with Propositional Minimal Model Generation'. In: *Principles and Practice of Constraint Programming—23rd International Conference, CP 2017, Melbourne, VIC, Australia, August 28 – September 1, 2017, Proceedings*. Ed. by J. Christopher Beck. Vol. 10416. Lecture Notes in Computer Science. Springer, pp. 596–614. DOI: 10.1007/978-3-319-66158-2_38. URL: https://doi.org/10.1007/978-3-319-66158-2_38.
- Terra-Neves, Miguel et al. (July 2018). 'Multi-Objective Optimization Through Pareto Minimal Correction Subsets'. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization. DOI: 10.24963/ijcai.2018/757.

Pareto Optimality

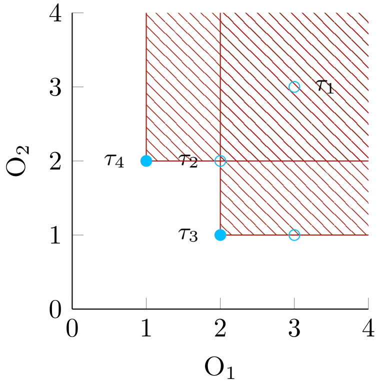
A more formal definition

Definition (Dominance)

A solution dominates another one if it is *not worse* on all objectives and *better* on at least one.

Definition (Pareto Optimality)

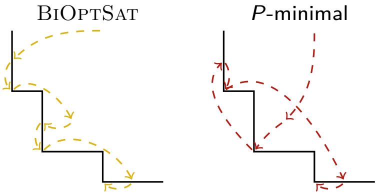
Every solution that is not dominated by any other solution is Pareto-optimal.



Competing Approaches

P-minimal [Soh et al. 2017]

- Closest to BiOptSat
- Unstructured order of enumeration



Seesaw [Janota et al. 2021]

- IHS framework generalized to two objectives
- Instantiating black box with solution improving search

ParetoMCS [Terra-Neves et al. 2018]

- Enumerate *all* MCSes and filter non-optimal solutions
- Available Sat4j implementation

Seesaw Instantiation

[Janota et al. 2021]

- Oracle: solution improving search in SAT solver
  - Minimal decision rule size to misclassify at most samples in hitting set
  - CaDiCaL [Biere et al. 2020]
- Cost function (hitting set extraction): MILP solver (CPLEX)
- (Optimization) core extraction: version for anti-monotone oracles in paper [Janota et al. 2021]

Number of Solved Instances

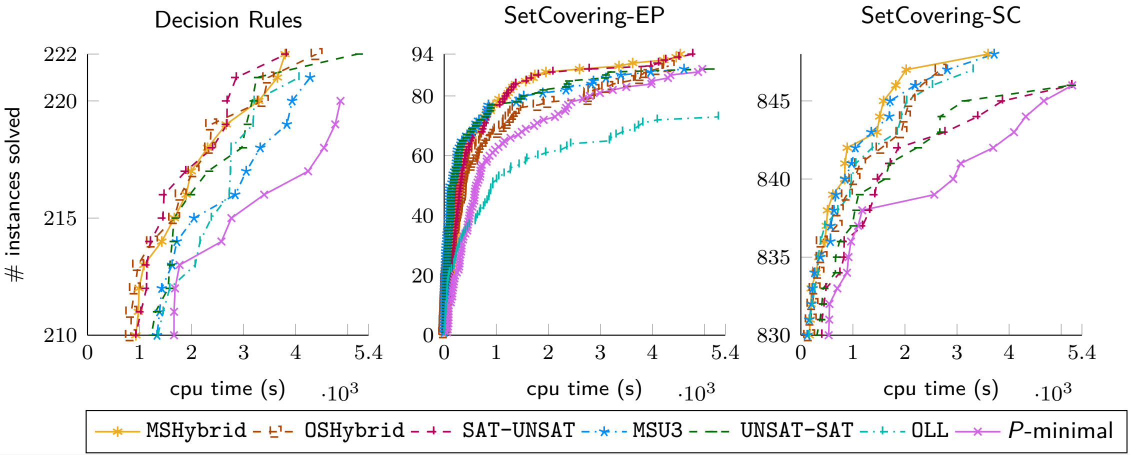
[Jabs et al. 2024b]

	Set Covering			MaxSAT Lib				
Domain	LIDR	SC-EP	SC-SC	PackUP	spot5	hap-ped	am-k	prot
# Inst.	371	120	120	1057	32	100	36	11
SAT-UNSAT	**222**	**94**	42	846	20	21	**24**	**11**
UNSAT-SAT	**222**	89	38	846	20	21	23	**11**
MSU3	221	89	38	**848**	18	22	23	**11**
OLL	221	73	41	847	**24**	22	23	**11**
MSHybrid	**222**	**94**	42	**848**	16	22	**24**	**11**
OSHybrid	**222**	90	**47**	847	10	21	23	**11**
P-minimal	220	89	40	846	20	**23**	23	**11**
CLM-LB	185	93	43	819	5	19	1	0
CLM-IHS	86	86	39	626	4	7	0	0
Seesaw	135	60	39	204	6	18	0	2
Pareto-MCS	34	0	0	277	0	1	0	0
leximaxIST	220	52	42	962	8	23	18	11

BiOptSat All Variants: Results

(BiOptSat 🛠️)

[Jabs et al. 2024b]



Refinement: Lazily Building Both Totalizers

One totalizer over each objective

- Incrementally build "increasing" totalizer
  [Martins et al. 2014]
- Core-guided variant and overlapping objectives
  ⇒ "Decreasing" totalizer can be incrementally built
  - If literal inactive in increasing objective, ignore for "decreasing" totalizer