MaxSAT-Based Bi-Objective Boolean Optimization

Christoph Jabs Jeremias Berg Andreas Niskanen Matti Järvisalo

HIIT, Department of Computer Science, University of Helsinki, Finland

August 2nd @ SAT'22 @ FLoC'22

What?

BiOptSat Algorithm

- NP-hard bi-objective optimization
- Enumeration of Pareto-optimal solutions
- Using ideas from MaxSAT solving
- 5 variants

Links to all materials:

- `christophjabs.info/sat22`

Implementation

Open source

Empirical evaluation

- BiOptSat outperforms 3 key competitors
- Hybrid of MaxSAT algorithms performs best in all cases

So What?

Applications for Bi-Optimization

- Interpretable classifiers [Malioutov and Meel 2018]
- Solver portfolios [Janota et al. 2021]
- Network routing [Silvério et al. 2022]
- Supply chain optimization [Pinto-Varela et al. 2011]

Scalarizing is not enough

$$O = O_1 + \lambda O_2$$

Conclusion

Need for SAT-based algorithms for bi-objective optimization

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

$$r_1 = x_1 \wedge x_2$$

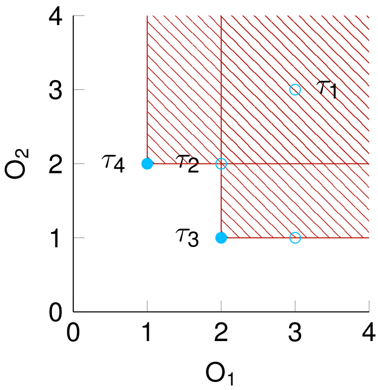
$$r_2 = x_1$$

Pareto Optimality

Two objectives require specific definition of optimality
Commonly used: Pareto optimality

Definition (Pareto Optimality)

Every solution where no objective can be improved without making the other one worse is Pareto-optimal.



The Problem

Similar to MaxSAT

- Hard clauses F
  (SAT encoding of the constraints)
- 2 (multi) sets of soft literals O_I, O_D
  (setting a soft literal incurs unit cost)

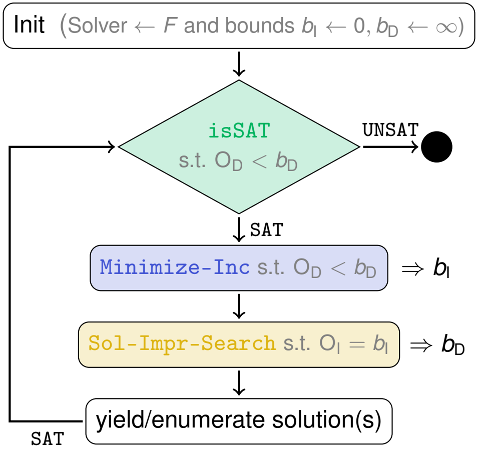
Toy Instance

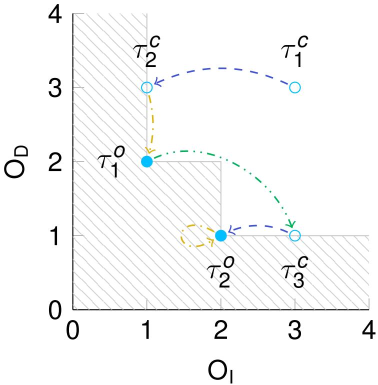
$$F = \texttt{As-CNF}\left(\sum_{\tau(l) \in \mathrm{O}_\mathrm{I} \cup \mathrm{O}_\mathrm{D}} l \geq 3 \right),$$

$$\mathrm{O}_\mathrm{I} = \{i_1, i_2, i_3\},$$

$$\mathrm{O}_\mathrm{D} = \{d_1, d_2, d_3\}$$

BiOptSat: The Lexicographic Method





BiOptSat: Implementation Details

- Single SAT solver instance (CaDiCaL [Biere et al. 2020])
- Heavy use of incremental totalizers [Martins et al. 2014]
- C++ implementation

BiOptSat: Variants

- 5 variants of `Minimize-Inc`
- No other variants for `Sol-Impr-Search` (loosening of constraints on O_I can invalidate cores)

MaxSAT algorithm-based

- `SAT-UNSAT`
- `UNSAT-SAT`
- `MSU3`
- `OLL`

Novel hybrid

- `MSHybrid`

MSHybrid: Avoiding UNSAT-SAT in MSU3

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

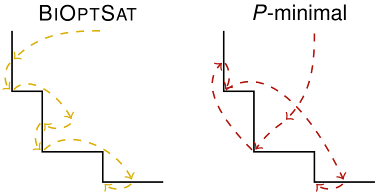
MSHybrid

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

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]

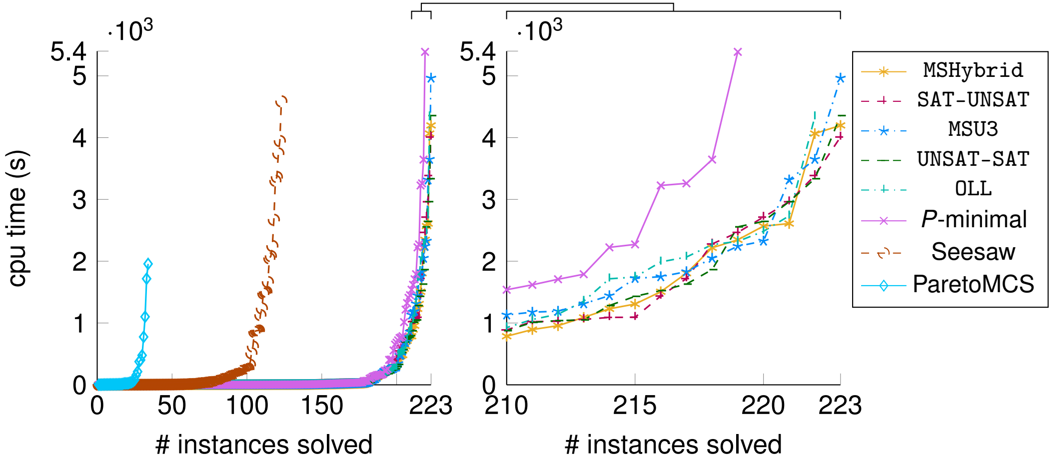
- Enumerate *all* MCSes and filter non-optimal solutions
- Available Sat4j implementation

Benchmark I: Interpretable Decision Rules — Description

Learning interpretable decision rules from binary data

- MLIC encoding [Malioutov and Meel 2018]
- Datasets from UCI MLR and Kaggle (and random subsets)
- Objectives
  - Rule size (number of literals)
  - Classification error

Benchmark I: Interpretable Decision Rules — Results

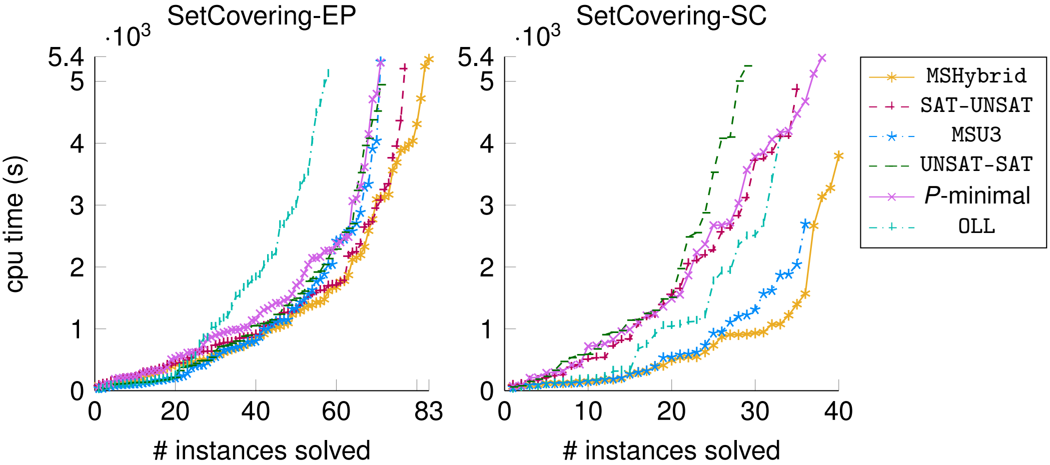


Benchmark II: Bi-Objective Set Covering — Description

Set covering with two costs per element

- Find lowest cost cover that intersects with all sets
- Randomly generated instances
  - SetCovering-EP: fixed probability for an element to be in a set
  - SetCovering-SC: fixed cardinality of sets
- Competitors
  - Seesaw cannot be reasonably instantiated (no anti-monotone objective)
  - ParetoMCS did not solve any instance

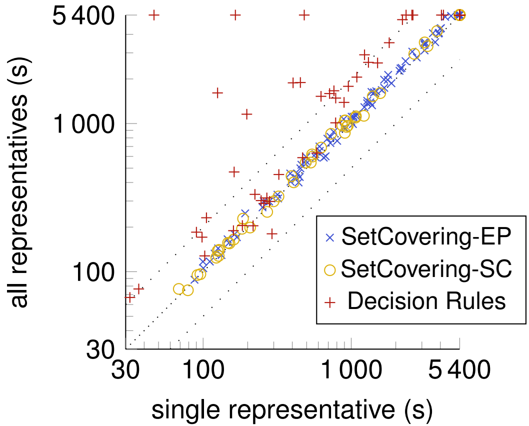
Benchmark II: Bi-Objective Set Covering — Results



Enumerating Representatives vs. All

Two enumeration tasks

- one representative solution per Pareto-optimal objective value pair
- all Pareto-optimal solutions



Refinement: Domain Specific Blocking

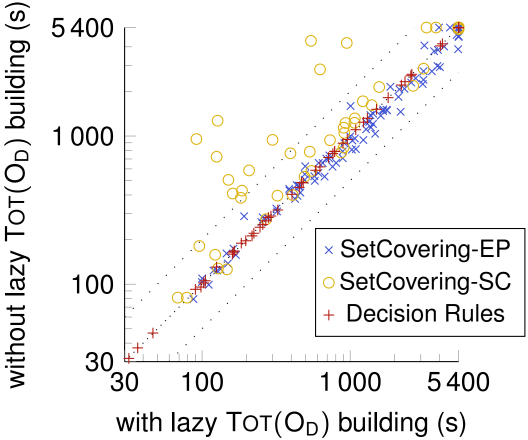
For enumeration of all Pareto-optimal solutions, blocking clauses for solutions are needed

- 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)

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



Summary

- BiOptSat algorithm
  - Enumerate Pareto-optimal solutions of CNFs
- Open-source implementation
- Outperforms 3 key competitors
- Best-performing variant: `MSHybrid`

Paper, slides, code, and contact:

`christophjabs.info/sat22`

Bibliography I

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.

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.

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.

Bibliography II

- 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].
- 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.

Bibliography III

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.

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

Instance Type	SAT-UNSAT		UNSAT-SAT		MSU3		OLL		MSHybrid		*P*-minimal	
	single	all	single	all	single	all	single	all	single	all	single	all
Decision Rules	**223**	**215**	**223**	**215**	**223**	**215**	222	213	**223**	**215**	219	213
SetCovering-EP	77	75	71	71	71	70	58	58	**83**	**81**	71	68
SetCovering-SC	35	35	29	29	36	36	34	34	**40**	**40**	38	26