Bi-Objective MaxSAT and the BIOPTSAT Algorithm

Christoph Jabs Jeremias Berg Andreas Niskanen Matti Järvisalo

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

December '22 @ DIKU / LTH

What?

Bi-Objective MaxSAT

- NP-hard bi-objective optimization
- Notions of optimality

P-Minimal

For comparison [Soh et al. 2017]

BiOptSat Algorithm

- Enumeration of Pareto-optimal solutions
- Using ideas from MaxSAT solving
- 5 variants

Links to all materials:

- `christophjabs.info/cph22`

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

$$r_1 = x_1 \land x_2$$

$$r_2 = x_1$$

Single- vs. Multi-Objective Optimization

Single-Objective Optimization

- Unique optimal cost value
- Possibly multiple optimal solutions

Multi-Objective Optimization

- What even is optimality?

MO Notions of Optimality

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

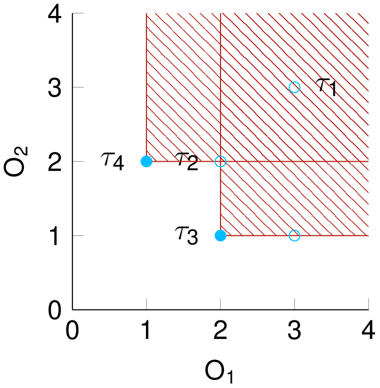
Pareto Optimality

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.



The Problem

Definition (Bi-Objective MaxSAT)

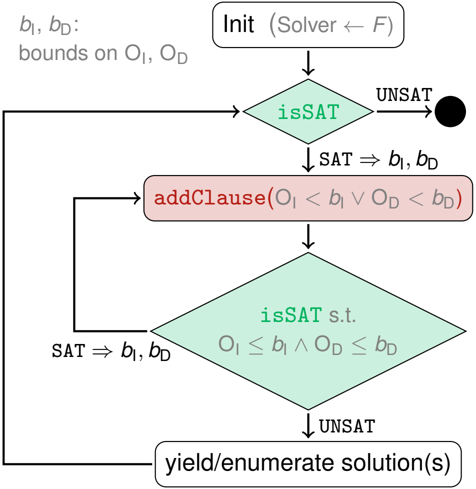
- Hard clauses F
  (SAT encoding of the constraints)
- 2 linear PBO objectives O_I, O_D
  (Weighted relaxation literals)

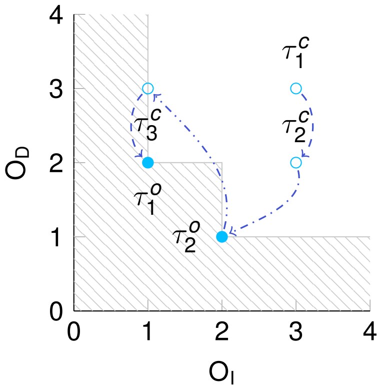
Toy Instance

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

P-Minimal: "Multi-Objective Solution-Improving Search" [Soh et al. 2017]

b_I, b_D: bounds on O_I, O_D

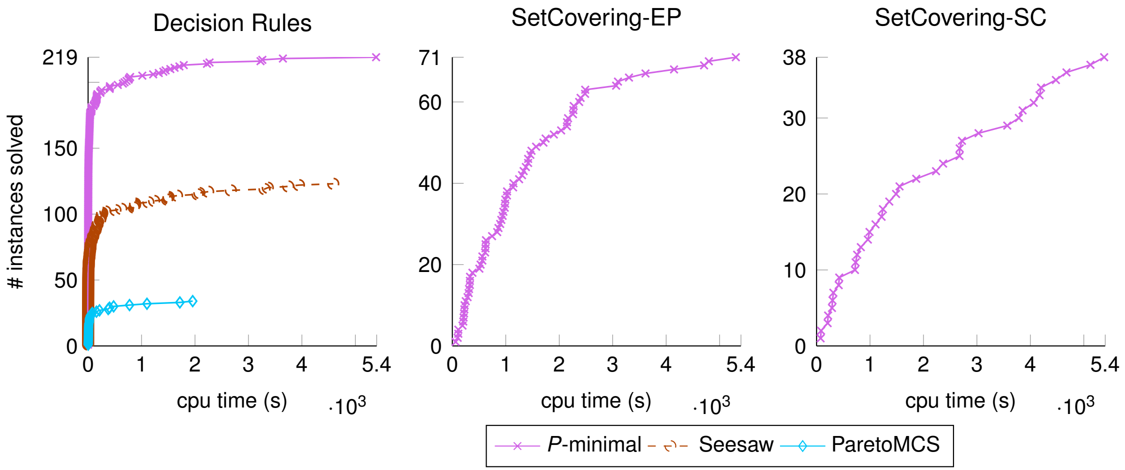




P-Minimal: Implementation Details

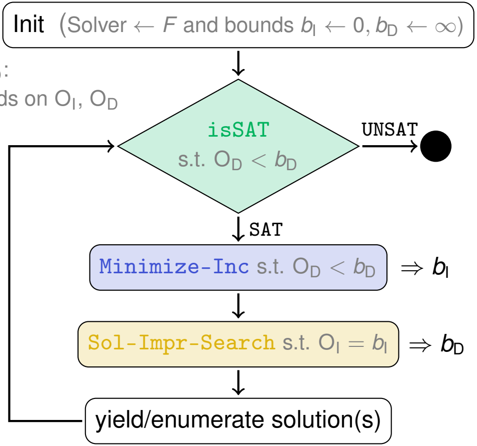
- Single SAT solver instance (CaDiCaL [Biere et al. 2020])
- C++ implementation
- Instead of order encoding as in [Soh et al. 2017], totalizers [Bailleux and Boufkhad 2003]

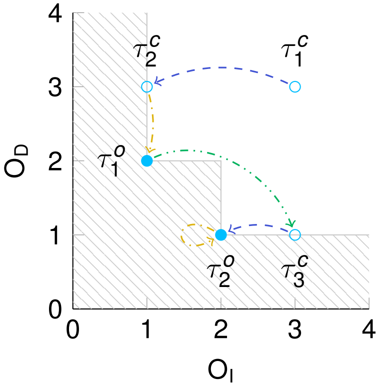
P-Minimal: Results



BiOptSat: The Lexicographic Method

b_I, b_D:
bounds on O_I, O_D





BIOPTSAT: Implementation Details

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

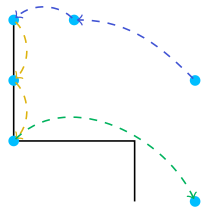
BiOptSat SAT-UNSAT

Simplest variant of Minimize-Inc:

SAT-UNSAT

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

BiOptSat SAT-UNSAT



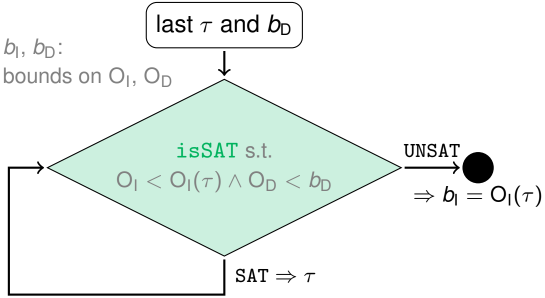
Sol-Impr-Search always solution improving search

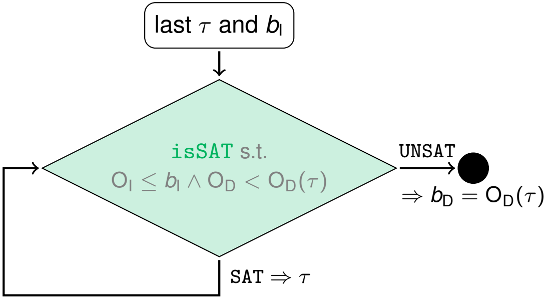
- Lower-bounding impractical because of decreasing objective values
- Loosening of constraints on O_l can invalidate cores
- Totalizer over all objective literals

BiOptSat SAT-UNSAT: Details

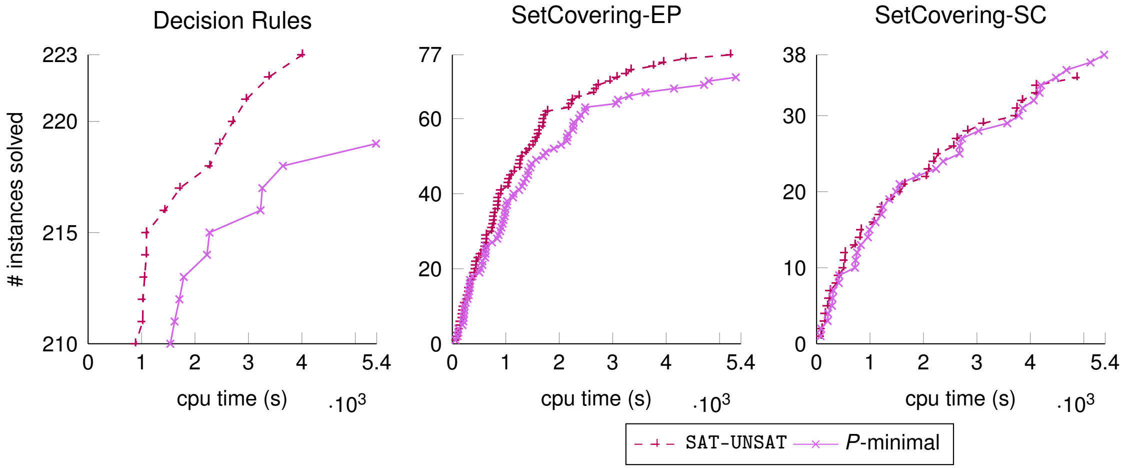
Minimize-Inc (SAT-UNSAT)

Sol-Impr-Search





BiOptSat SAT-UNSAT: Results



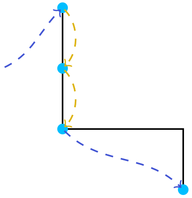
BiOptSat Lower-Bounding

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

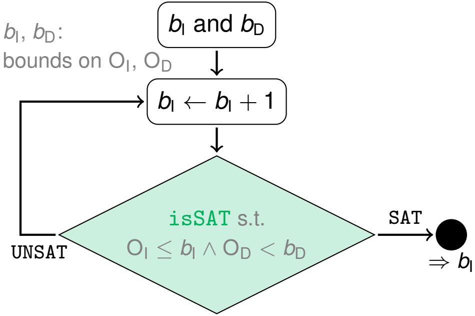
Sol-Impr-Search unchanged

BiOptSat UNSAT-SAT / MSU3 / OLL



BiOptSat Lower-Bouning: Details

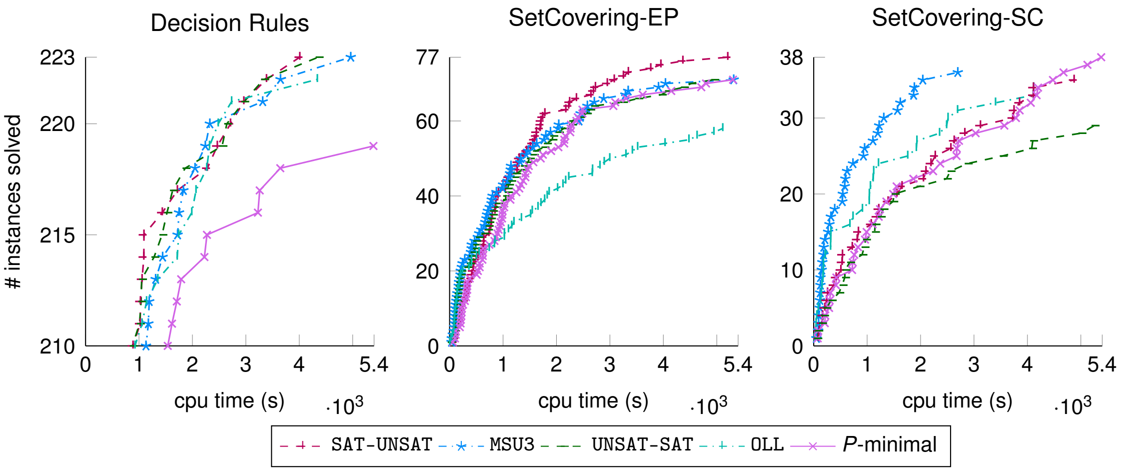
`Minimize-Inc` (UNSAT-SAT)



`Minimize-Inc` (MSU3)

(res, $\tau, \kappa) \leftarrow$ `isSAT`$(\{\langle O_D < b_D \rangle, \langle O_I \leq b_I \rangle\} \cup \{\neg l \mid l \in O_I \setminus \texttt{Act}\})$
while res = UNSAT **do**
  $b_I \leftarrow b_I + 1$
  $\kappa \leftarrow \kappa \setminus \{\neg\langle O_D < b_D \rangle, \neg\langle O_I \leq b_I \rangle\}$
  $\texttt{Act} \leftarrow \texttt{Act} \cup \kappa$
  build or extend Tot$(\texttt{Act}, b_I)$
  (res, $\tau, \kappa) \leftarrow$ `isSAT`$(\{\langle O_D < b_D \rangle, \langle O_I \leq b_I \rangle\} \cup \{\neg l \mid l \in O_I \setminus \texttt{Act}\})$
return b_I

BiOptSat Lower-Bounding: Results



MSHybrid: Avoiding UNSAT-SAT in MSU3

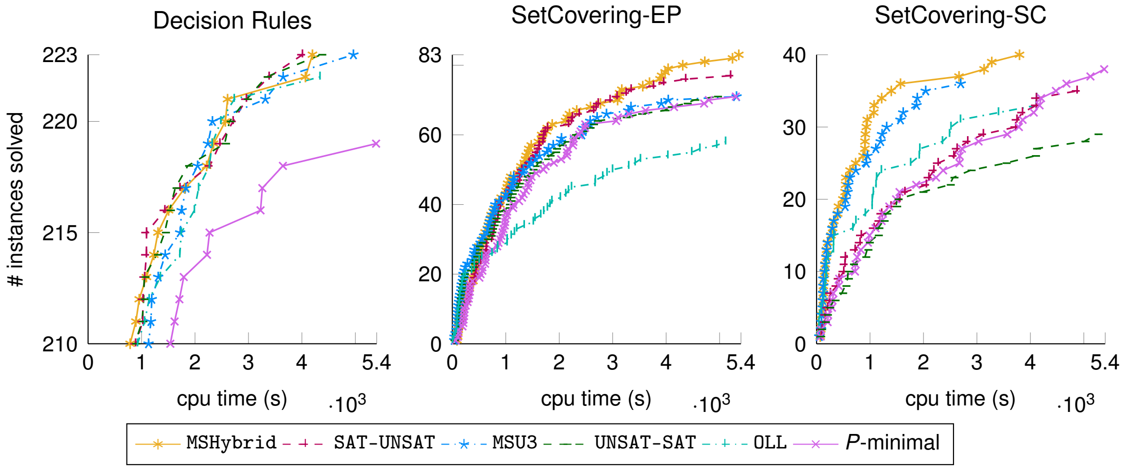
Search in Minimize-Inc continues after first optimum

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

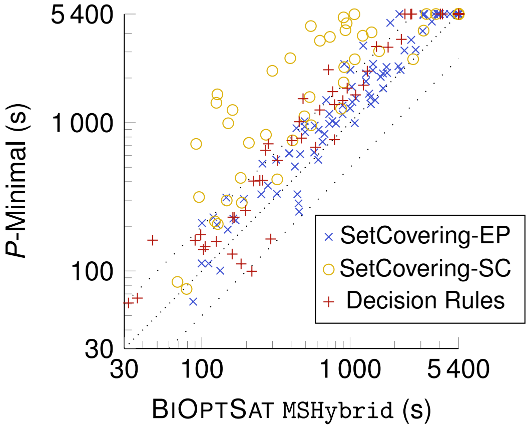
MSHybrid

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

BiOptSat MSHybrid: Results



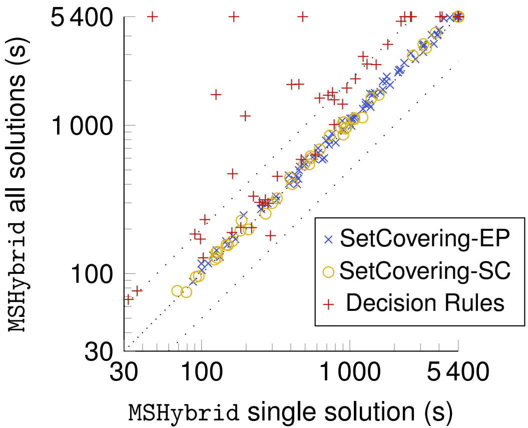
BiOptSat vs. *P*-Minimal



- 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

- 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

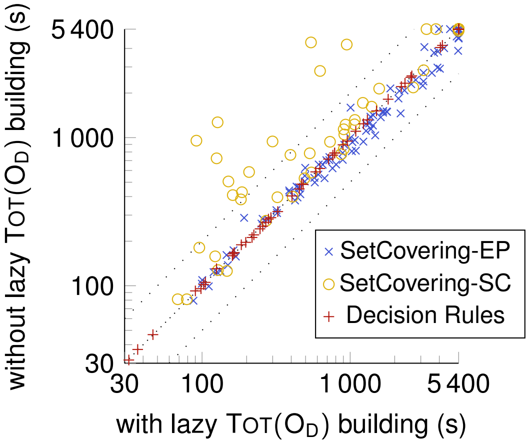
For enumeration of all Pareto-optimal solutions, blocking clauses for solutions are needed

- 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)

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/cph22`

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.

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 II

- 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].
- 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.

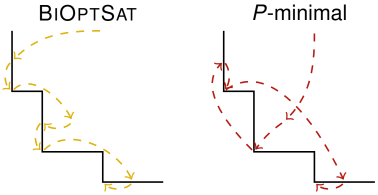
Bibliography III

- 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.

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

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