

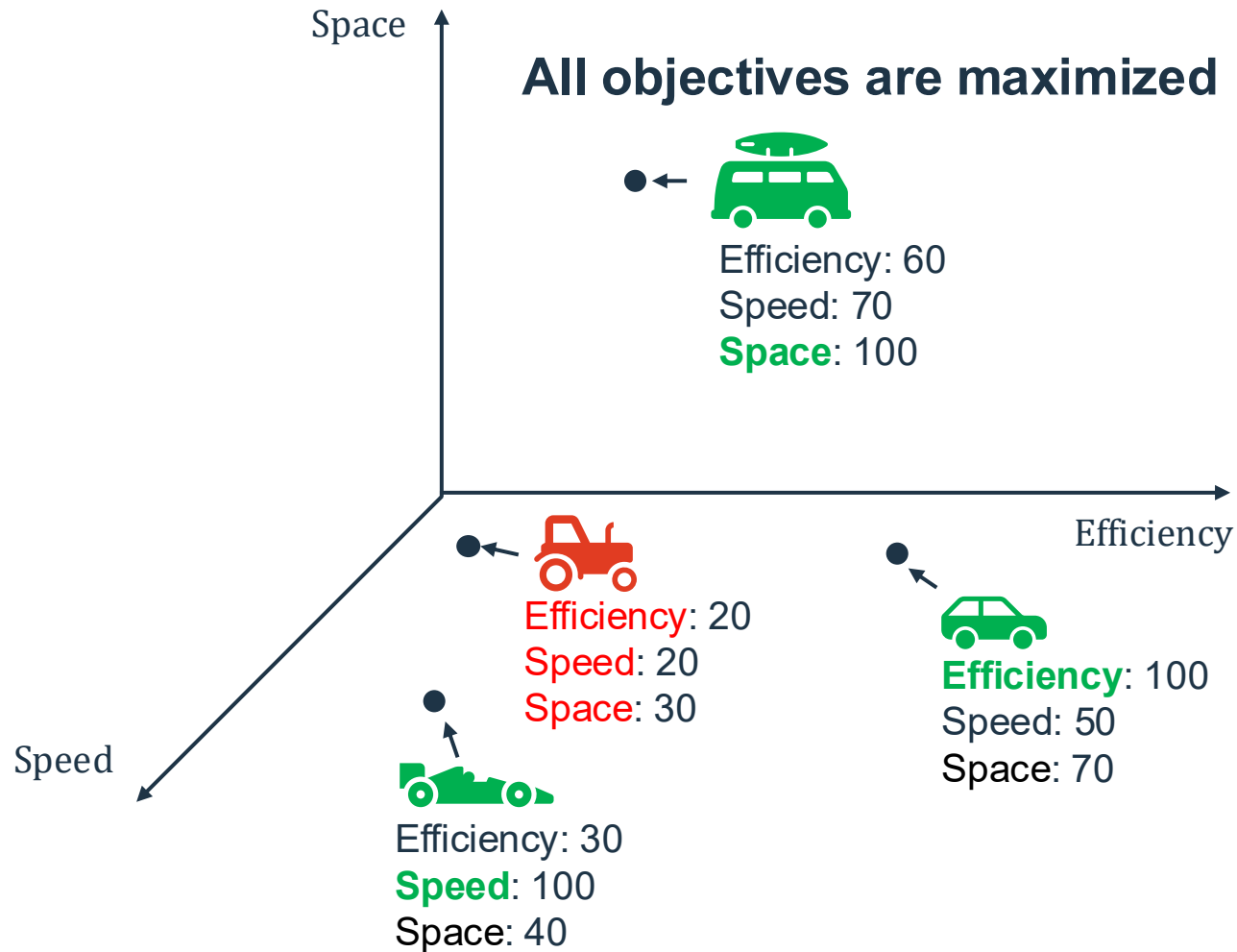
Combining an ϵ -Constraint Method with the Pareto Global Constraint

CP 07/2026, Lisbon

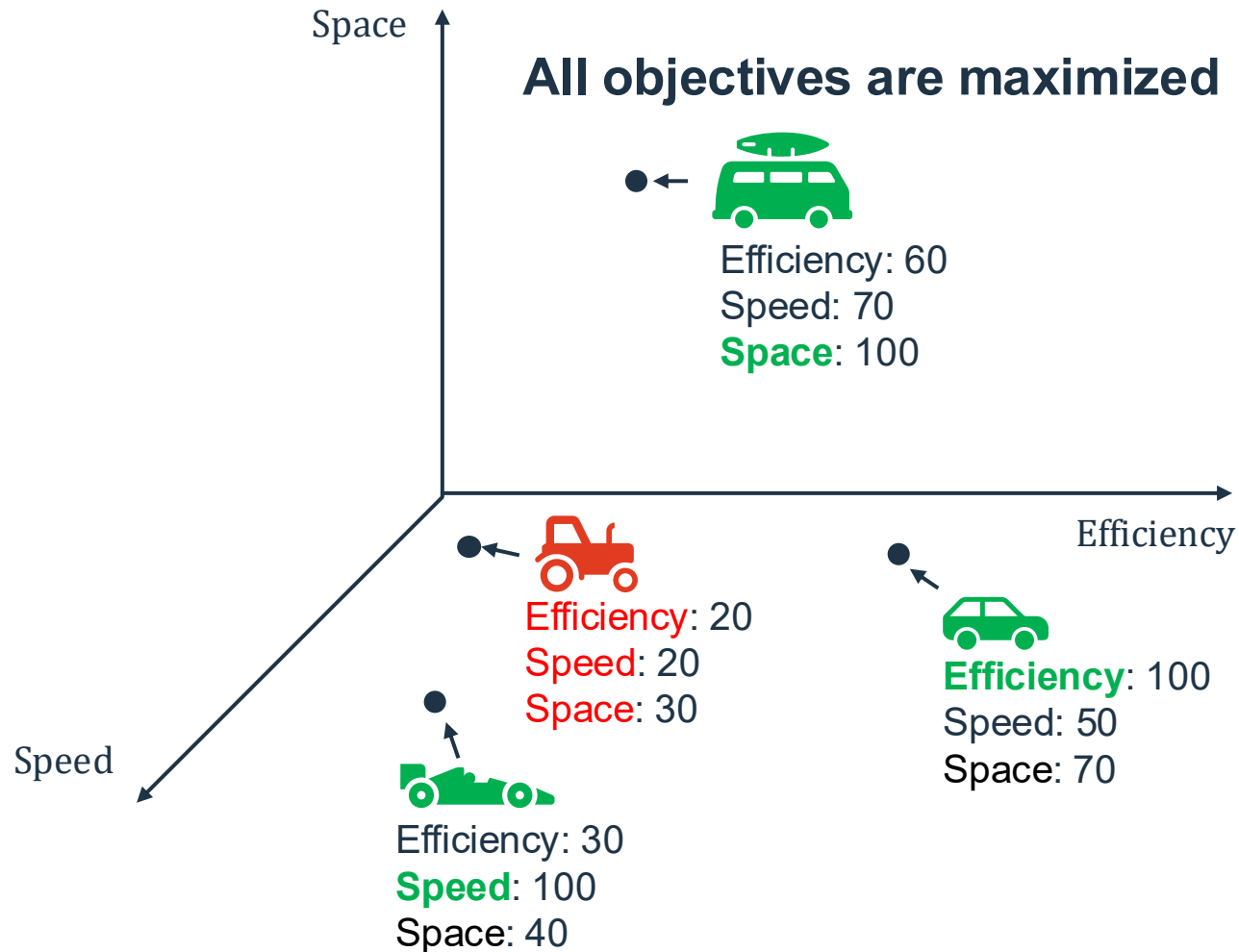
Manuel Combarro Simón, Pierre Talbot, Pascal Bouvry

Interdisciplinary Centre for Security, Reliability and Trust (SnT), Luxembourg
University of Luxembourg

Multi-objective optimization



Multi-objective optimization

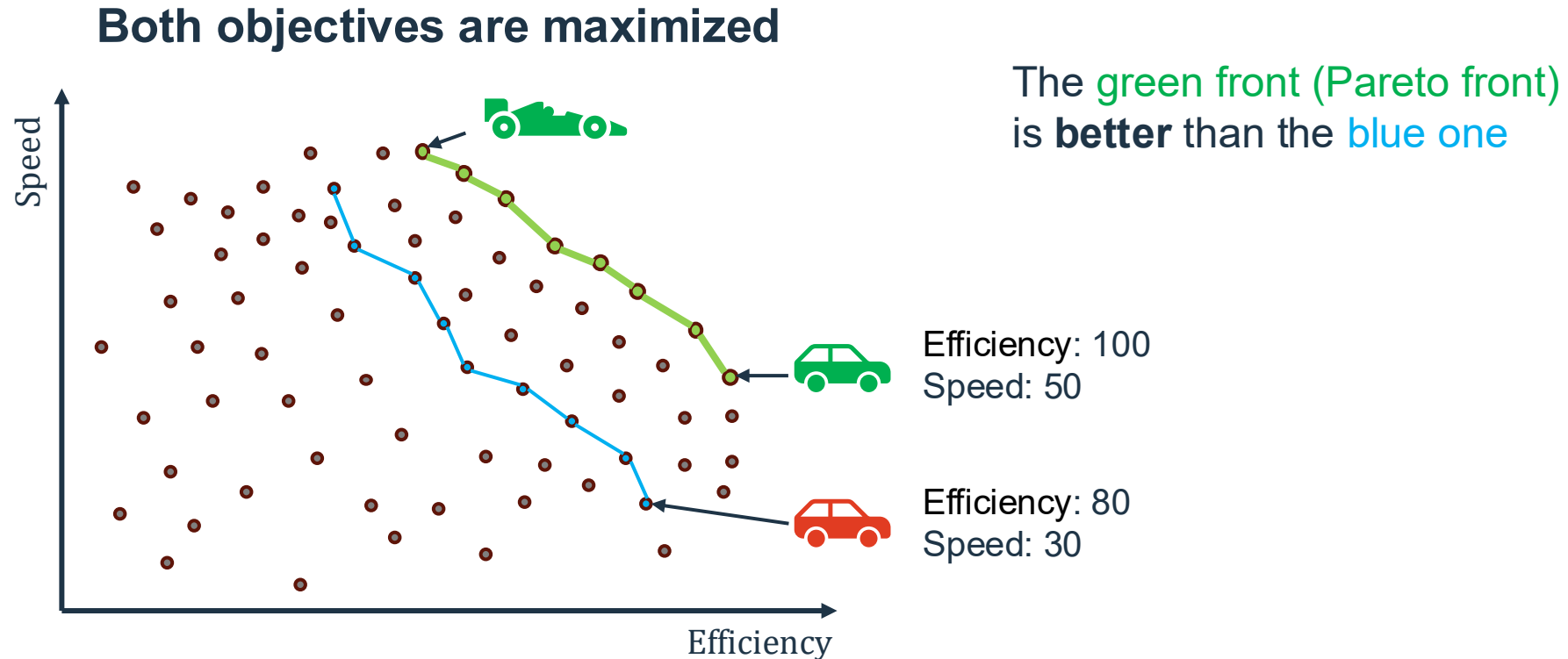


Pareto dominance

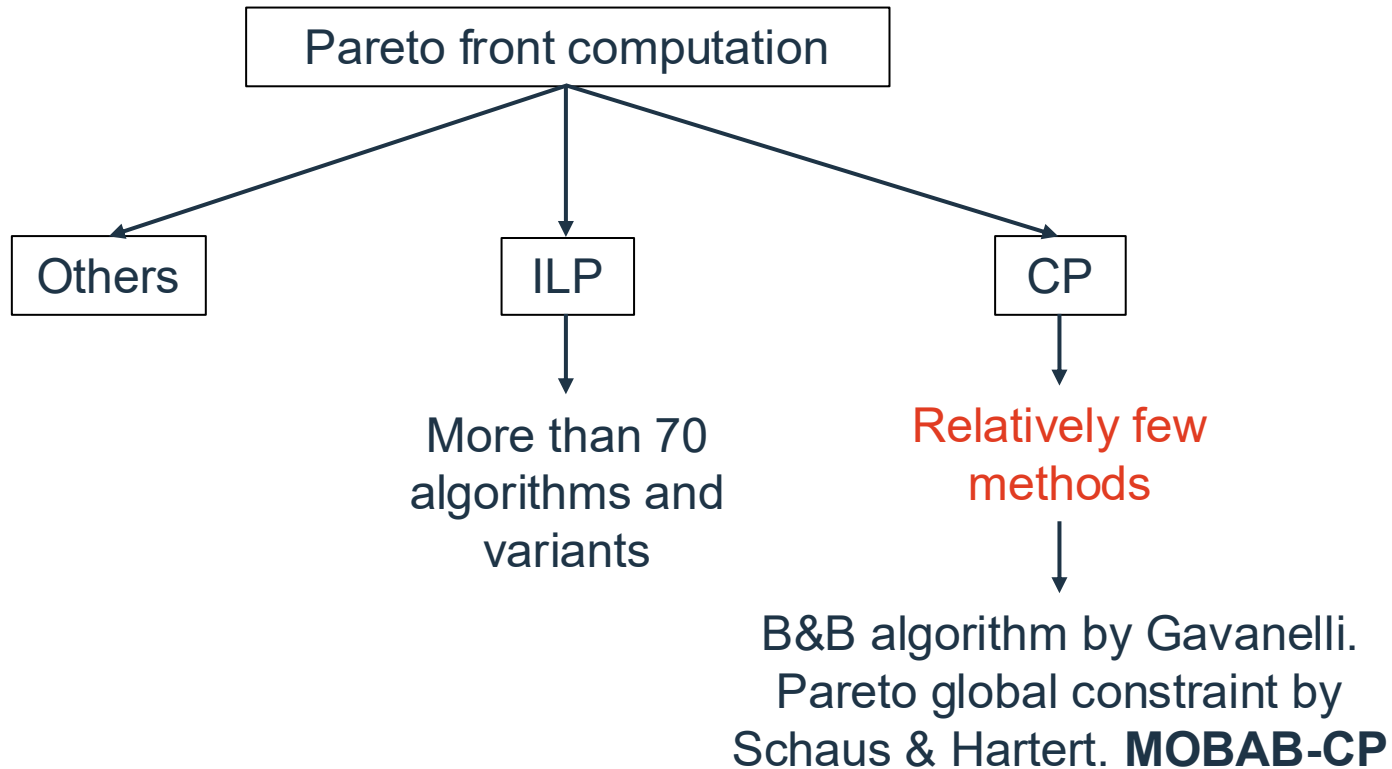
		
Efficiency: 100	$\geq$	Efficiency: 20
Speed: 50		Speed: 20
Space: 70		Space: 30

Pareto front

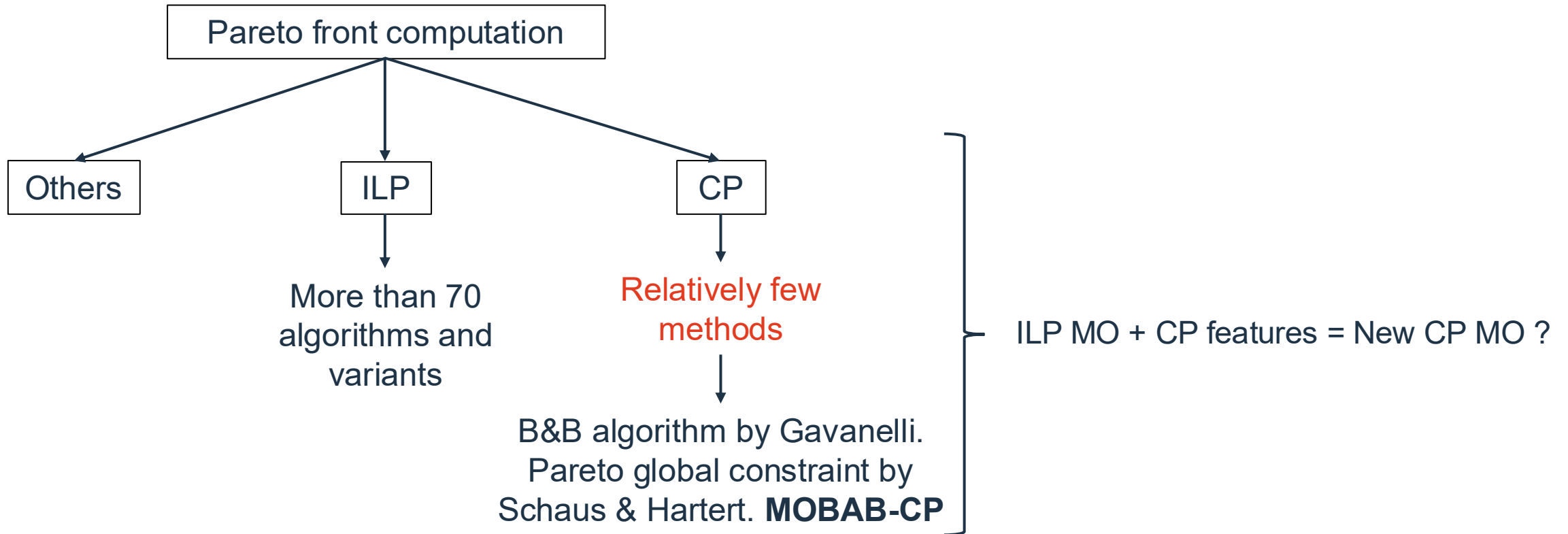
- In multi-objective optimization, the goal is to find **all Pareto-optimal solutions**: the **Pareto front**.



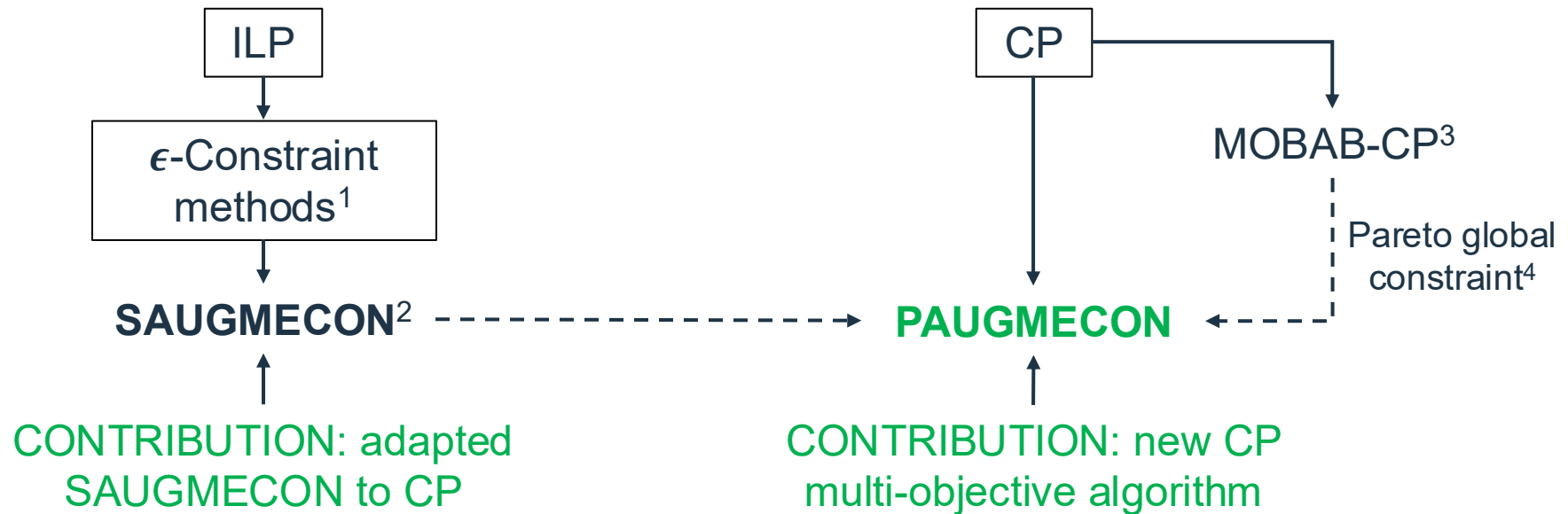
Pareto front computation



Pareto front computation



Contributions



¹Y. Y. Haimes, W. A. Hall, and H. T. Freedman: Developments in Water Science: The Surrogate Worth Trade-off Method. Multi-objective Optimization in Water Resources Systems (1975).

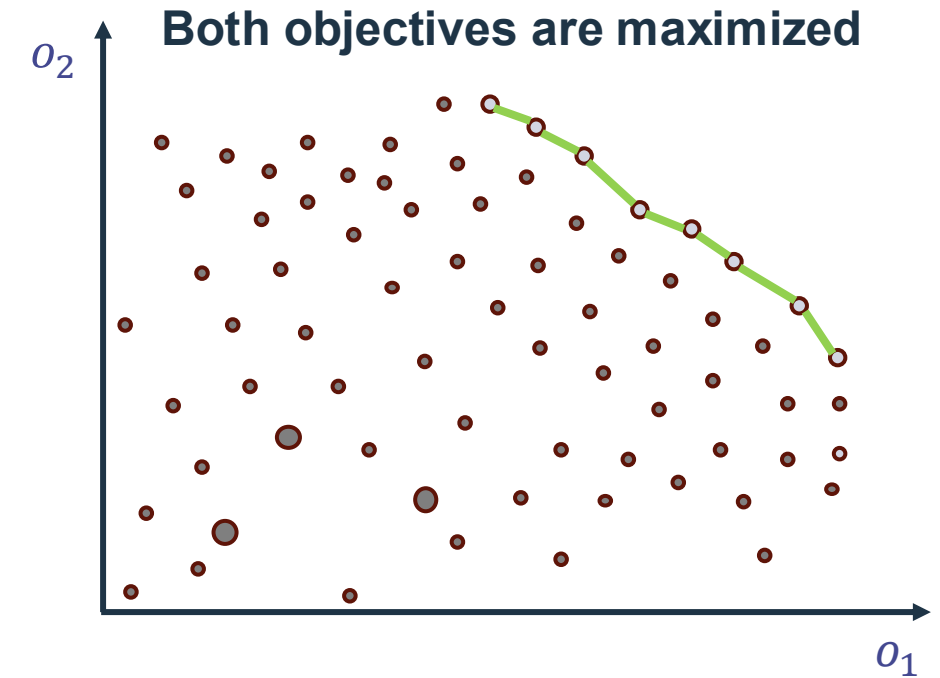
²W. Zhang and M. Reimann: 'A simple augmented ϵ -constraint method for multi-objective mathematical integer programming problems'. European Journal of Operational Research (2014).

³M. Gavanelli: 'An Algorithm for Multi-Criteria Optimization in CSPs'. ECAI (2002).

⁴Pierre Schaus and Renaud Hartert: 'Multi-Objective Large Neighborhood. Search'. CP (2013)

SoTA: MOBAB-CP – CP baseline¹

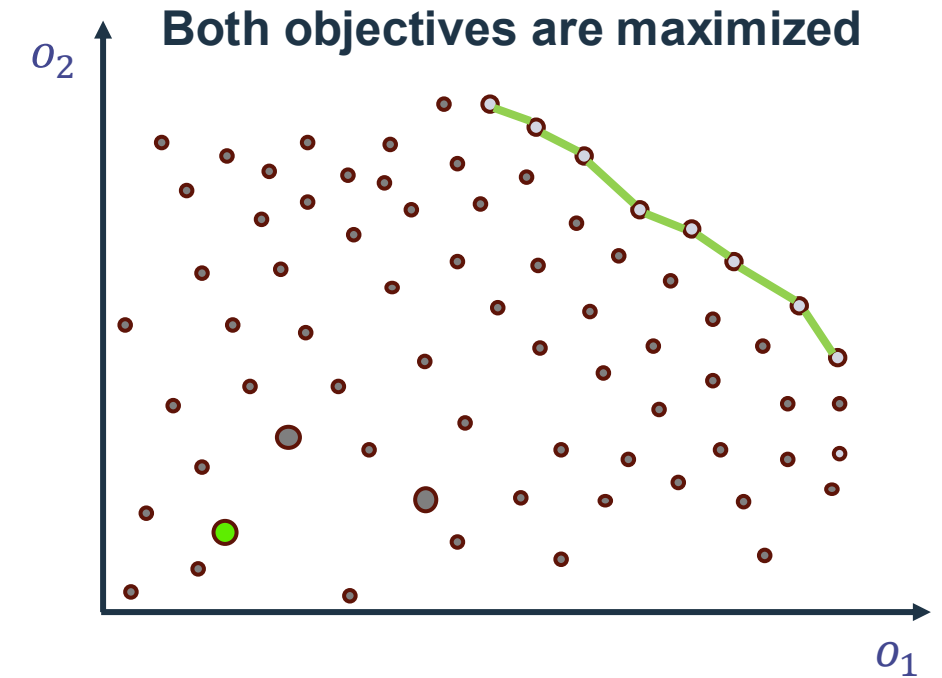
Intuition:



SoTA: MOBAB-CP – CP baseline¹

Intuition:

- 1. Find the first solution that satisfies the problem, *sol*.

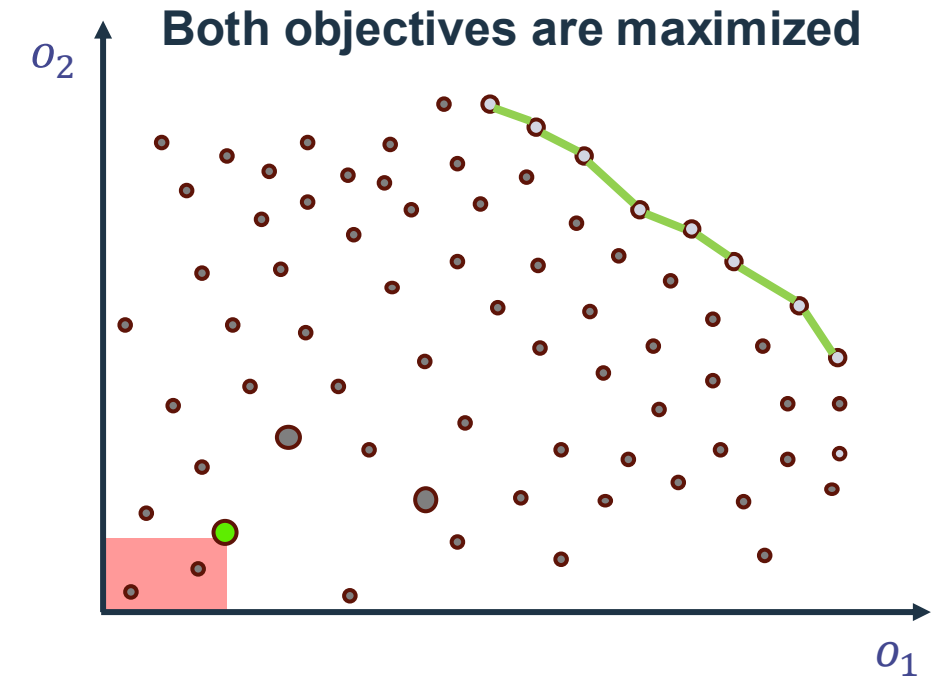


SoTA: MOBAB-CP – CP baseline¹

Intuition:

1. Find the first solution that satisfies the problem, sol .
- 2. Constrain the problem so the next solution is not dominated by sol .

Ex: Add $(o_1 > sol_1 \vee \dots \vee o_p > sol_p)$

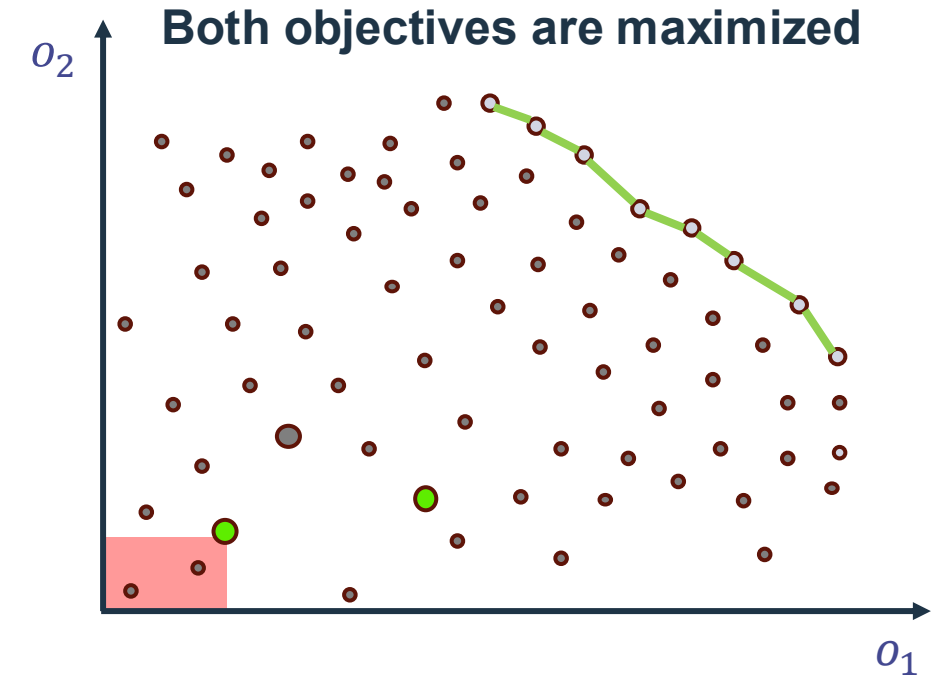


SoTA: MOBAB-CP – CP baseline¹

Intuition:

- 1. Find the first solution that satisfies the problem, sol .
- 2. Constrain the problem so the next solution is not dominated by sol .

Ex: Add $(o_1 > sol_1 \vee \dots \vee o_p > sol_p)$

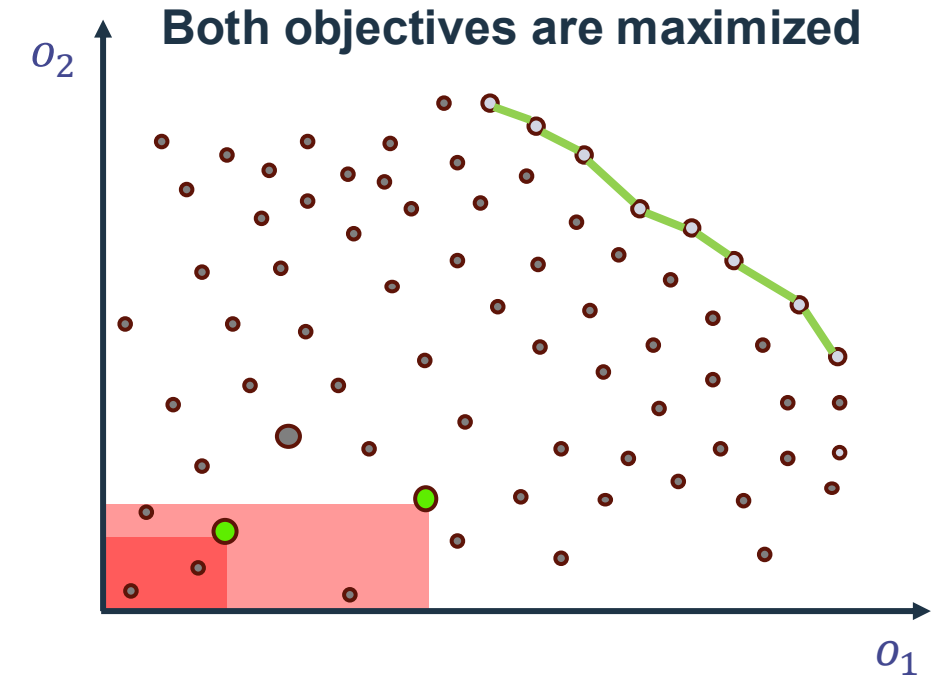


SoTA: MOBAB-CP – CP baseline¹

Intuition:

1. Find the first solution that satisfies the problem, sol .
- 2. Constrain the problem so the next solution is not dominated by sol .

Ex: Add $(o_1 > sol_1 \vee \dots \vee o_p > sol_p)$

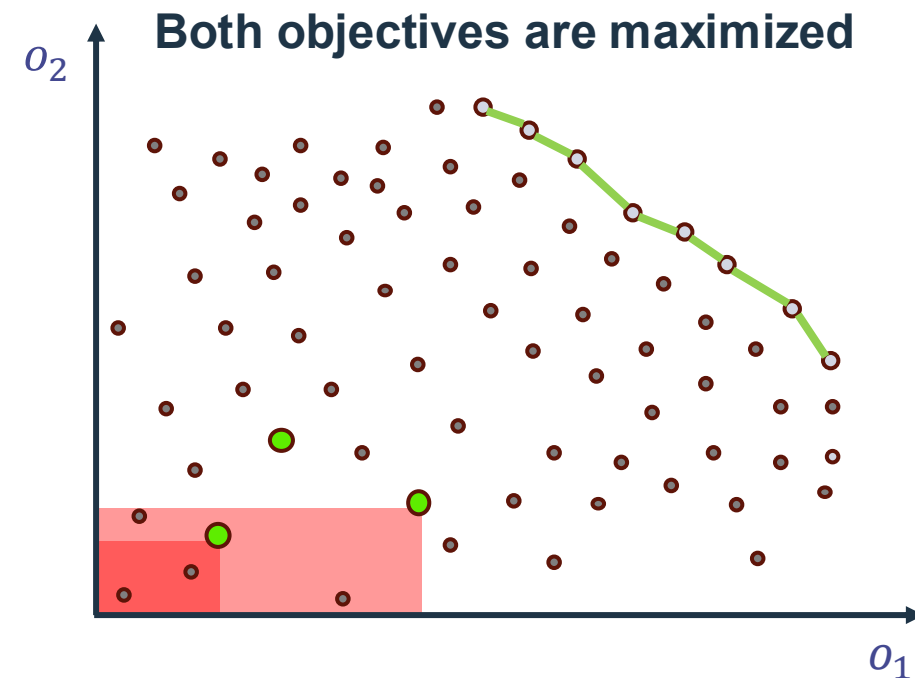


SoTA: MOBAB-CP – CP baseline¹

Intuition:

- 1. Find the first solution that satisfies the problem, sol .
- 2. Constrain the problem so the next solution is not dominated by sol .

Ex: Add $(o_1 > sol_1 \vee \dots \vee o_p > sol_p)$

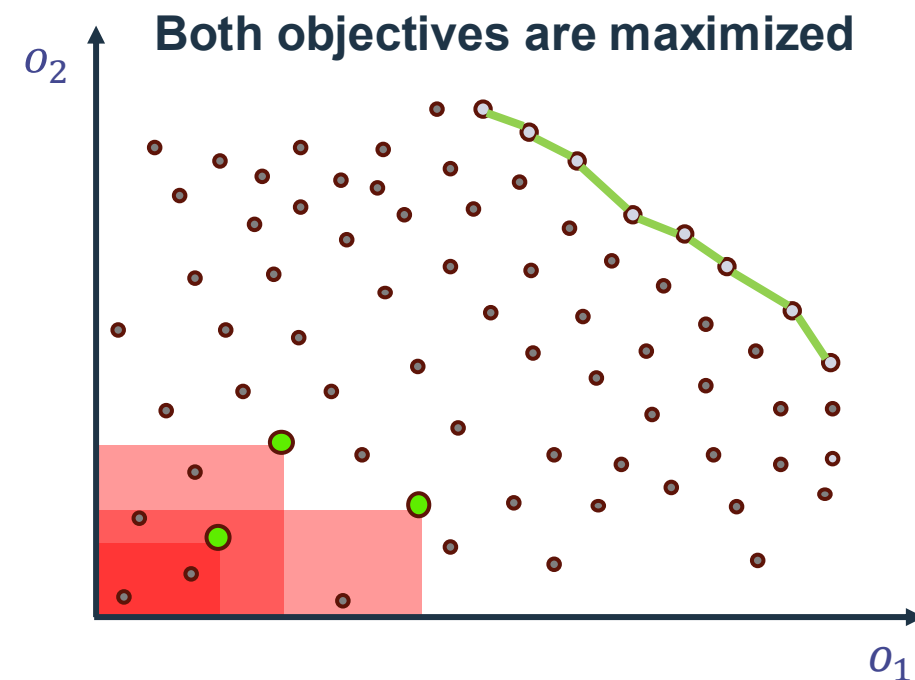


SoTA: MOBAB-CP – CP baseline¹

Intuition:

1. Find the first solution that satisfies the problem, sol .
- 2. Constrain the problem so the next solution is not dominated by sol .

Ex: Add $(o_1 > sol_1 \vee \dots \vee o_p > sol_p)$



SoTA: MOBAB-CP – CP baseline¹

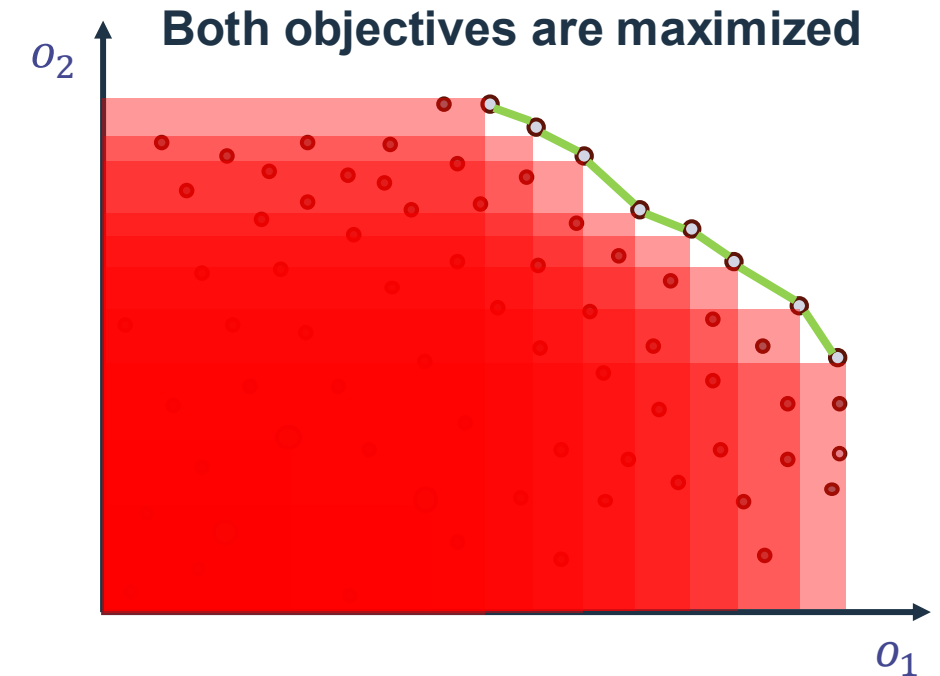
Intuition:

1. Find the first solution that satisfies the problem, sol .
2. Constrain the problem so the next solution is not dominated by sol .

Ex: Add $(o_1 > sol_1 \vee \dots \vee o_p > sol_p)$

3. Continue until cannot find another solution (search tree has been completely explored) -> Pareto front.

Ex: $(o_1 > sol_1^1 \vee \dots \vee o_p > sol_p^1) \wedge \dots \wedge (o_1 > sol_1^M \vee \dots \vee o_p > sol_p^M)$ for M solutions found



SoTA: MOBAB-CP – CP baseline¹

Intuition:

1. Find the first solution that satisfies the problem, sol .
2. Constrain the problem so the next solution is not dominated by sol .

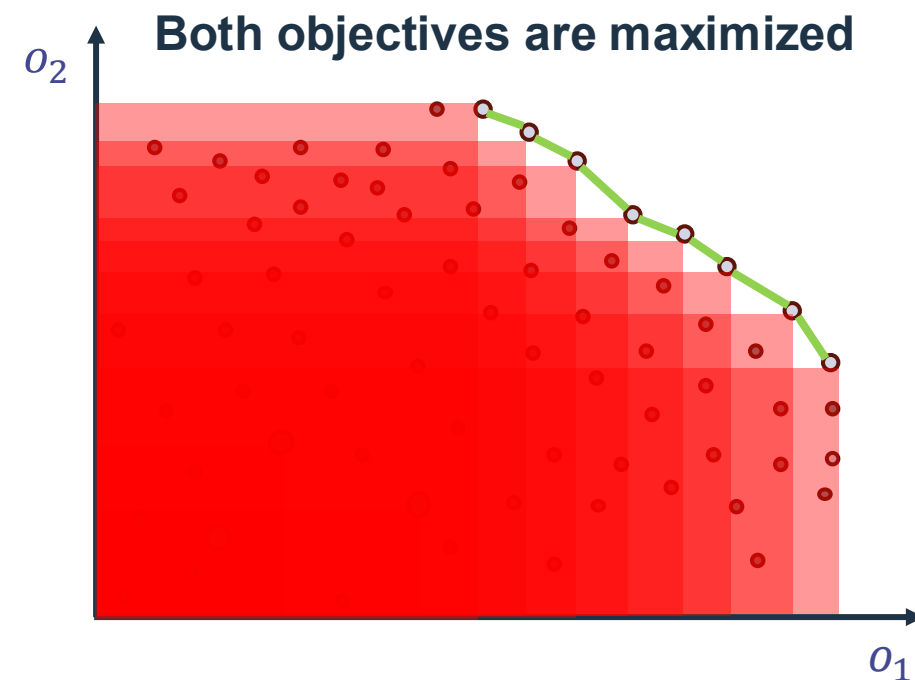
Ex: Add $(o_1 > sol_1 \vee \dots \vee o_p > sol_p)$

3. Continue until cannot find another solution (search tree has been completely explored) -> Pareto front.

Ex: $(o_1 > sol_1^1 \vee \dots \vee o_p > sol_p^1) \wedge \dots \wedge (o_1 > sol_1^M \vee \dots \vee o_p > sol_p^M)$ for M solutions found

Pareto global constraint²

(use only the non-dominated solutions)



SoTA: MOBAB-CP – CP baseline¹

Intuition:

1. Find the first solution that satisfies the problem, sol .
2. Constrain the problem so the next solution is not dominated by sol .

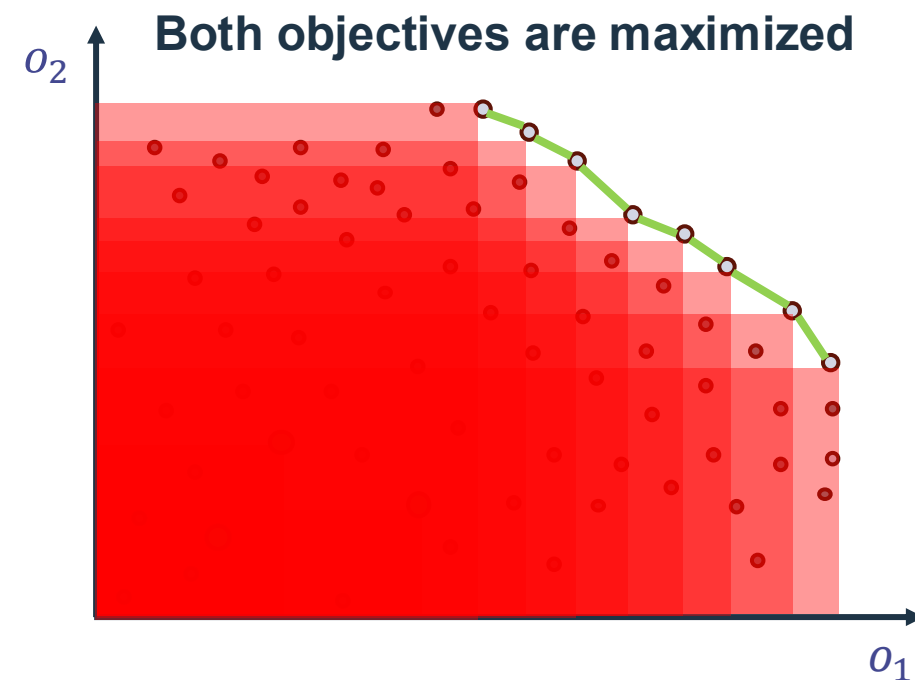
Ex: Add $(o_1 > sol_1 \vee \dots \vee o_p > sol_p)$

3. Continue until cannot find another solution (search tree has been completely explored) -> Pareto front.

Ex: $(o_1 > sol_1^1 \vee \dots \vee o_p > sol_p^1) \wedge \dots \wedge (o_1 > sol_1^M \vee \dots \vee o_p > sol_p^M)$ for M solutions found

Pareto global constraint²

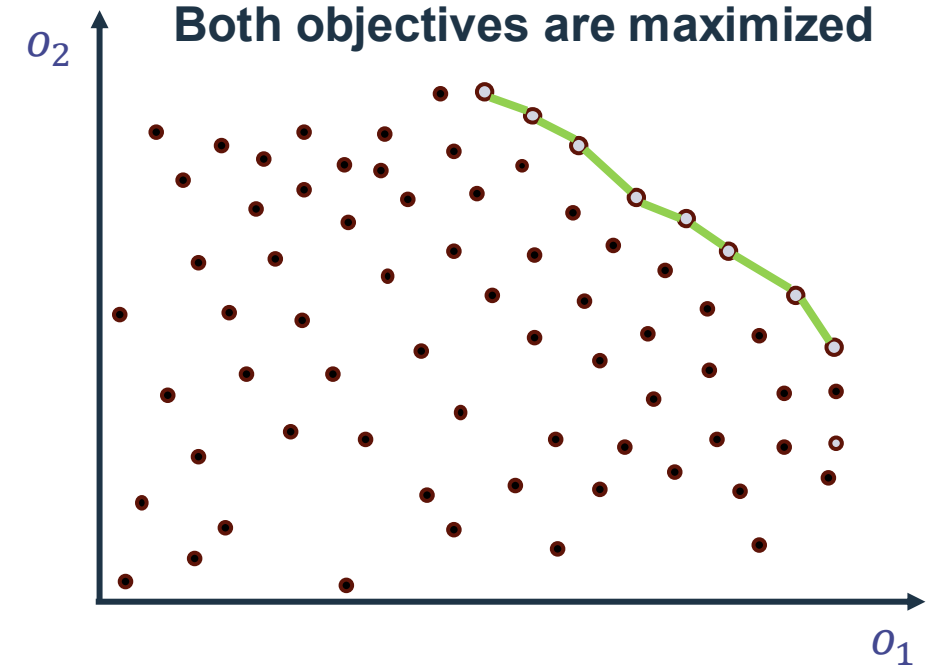
(use only the non-dominated solutions)



Can find all the Pareto-optimal solutions without restarting the search.

SoTA: SAUGMECON¹

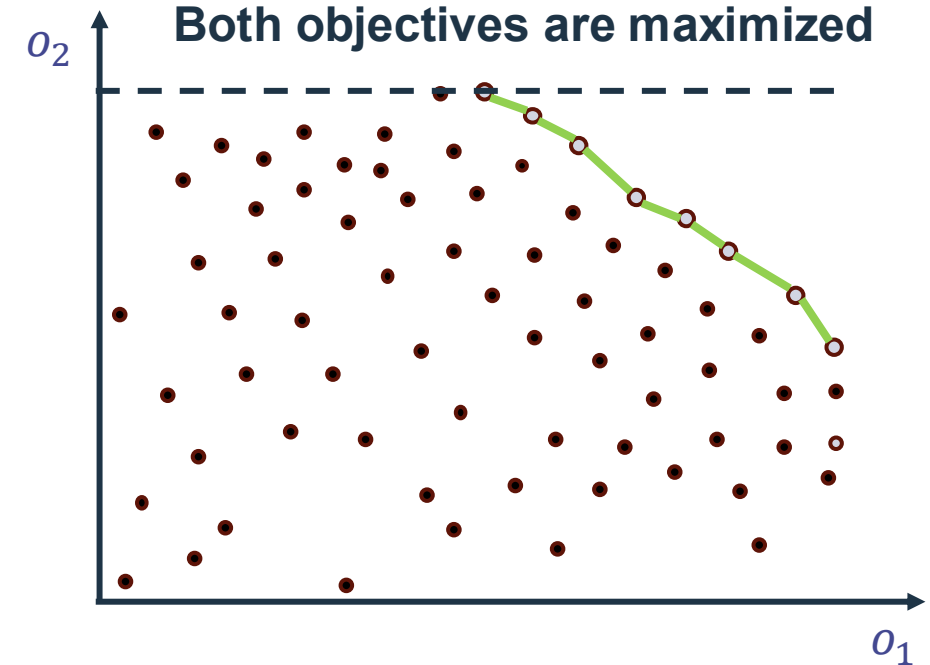
1. **Aggregate objective function.** One objective more important than the rest. $o_1 + \delta(w_2 o_2 + \dots + w_p o_p)$.
2. **Information from previous solutions.** Reduce search space and avoid redundancy. $o_2 > \text{solution}_{o_2}, \dots, o_p > \text{solution}_{o_p}$



SoTA: SAUGMECON¹

1. **Aggregate objective function.** One objective more important than the rest. $o_1 + \delta(w_2 o_2 + \dots + w_p o_p)$.
2. **Information from previous solutions.** Reduce search space and avoid redundancy. $o_2 > \text{solution}_{o_2}, \dots, o_p > \text{solution}_{o_p}$

Preprocessing: get the best value for $p - 1$ objectives.



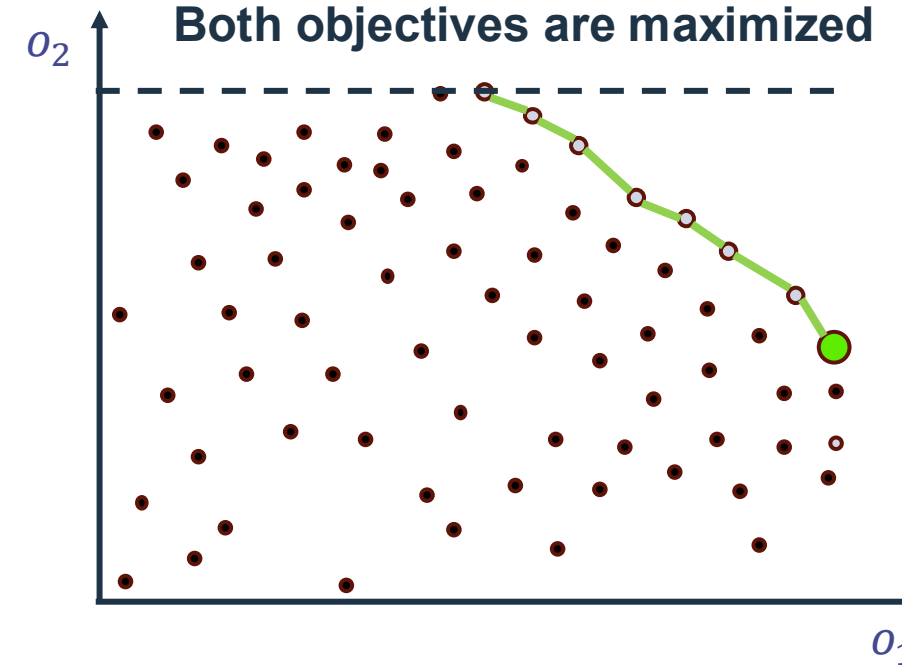
SoTA: SAUGMECON¹

1. **Aggregate objective function.** One objective more important than the rest. $o_1 + \delta(w_2 o_2 + \dots + w_p o_p)$.
2. **Information from previous solutions.** Reduce search space and avoid redundancy. $o_2 > \text{solution}_{o_2}, \dots, o_p > \text{solution}_{o_p}$

Preprocessing: get the best value for $p - 1$ objectives.

Solving phase:

- a. Optimize objective function. Solution is Pareto-optimal.



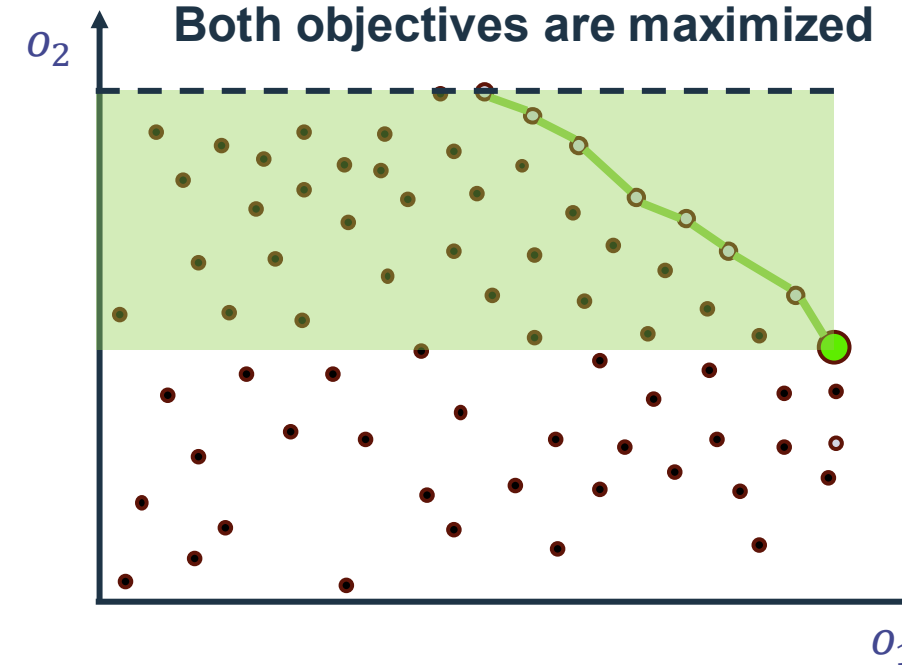
SoTA: SAUGMECON¹

1. **Aggregate objective function.** One objective more important than the rest. $o_1 + \delta(w_2 o_2 + \dots + w_p o_p)$.
2. **Information from previous solutions.** Reduce search space and avoid redundancy. $o_2 > \text{solution}_{o_2}, \dots, o_p > \text{solution}_{o_p}$

Preprocessing: get the best value for $p - 1$ objectives.

Solving phase:

- a. Optimize objective function. Solution is Pareto-optimal.
- b. Use the solution to constrain the $p - 1$ objectives.



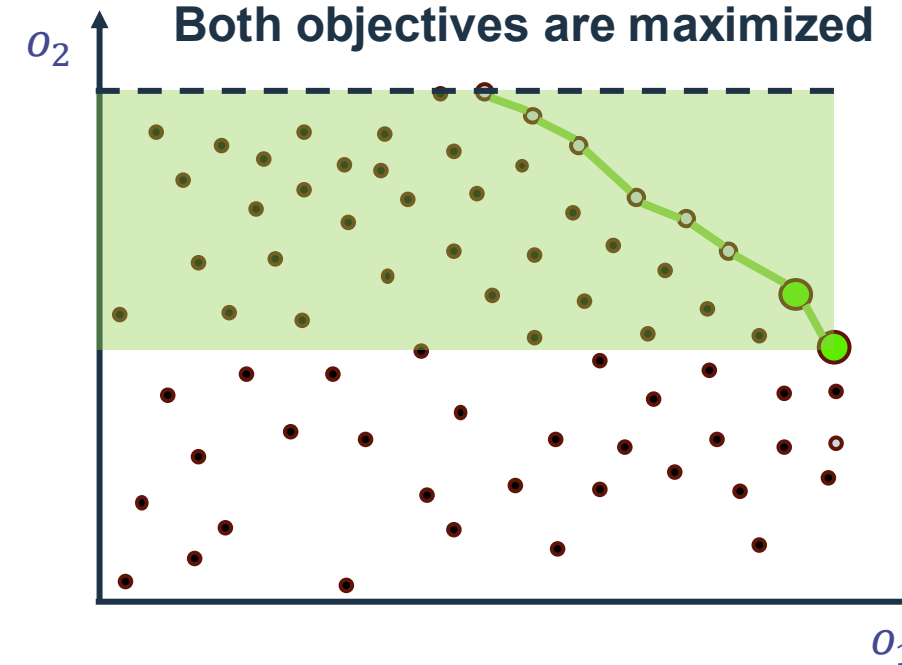
SoTA: SAUGMECON¹

1. **Aggregate objective function.** One objective more important than the rest. $o_1 + \delta(w_2 o_2 + \dots + w_p o_p)$.
2. **Information from previous solutions.** Reduce search space and avoid redundancy. $o_2 > \text{solution}_{o_2}, \dots, o_p > \text{solution}_{o_p}$

Preprocessing: get the best value for $p - 1$ objectives.

Solving phase:

- a. Optimize objective function. Solution is Pareto-optimal.
- b. Use the solution to constrain the $p - 1$ objectives.



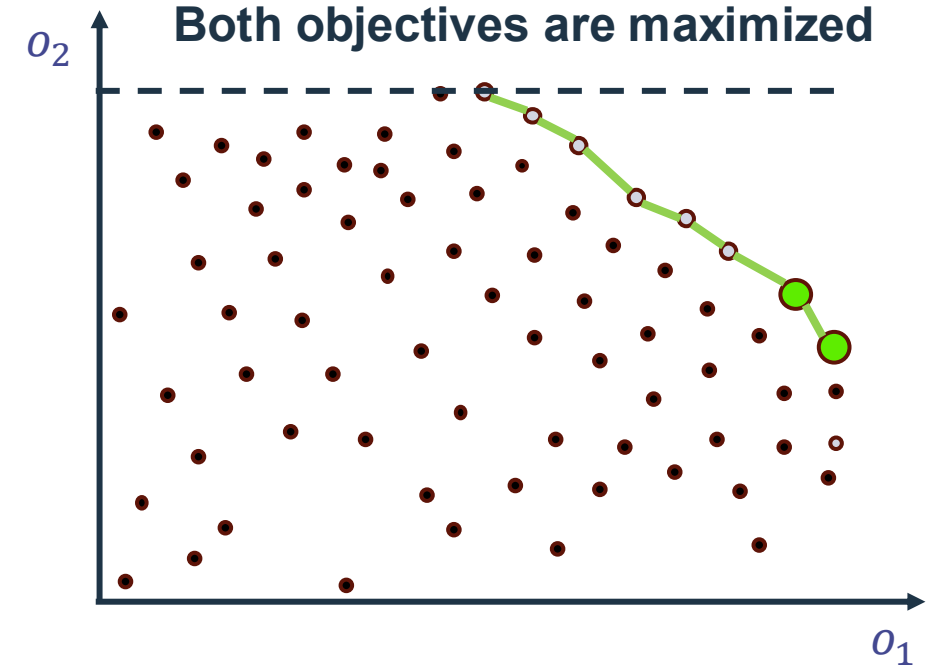
SoTA: SAUGMECON¹

1. **Aggregate objective function.** One objective more important than the rest. $o_1 + \delta(w_2 o_2 + \dots + w_p o_p)$.
2. **Information from previous solutions.** Reduce search space and avoid redundancy. $o_2 > \text{solution}_{o_2}, \dots, o_p > \text{solution}_{o_p}$

Preprocessing: get the best value for $p - 1$ objectives.

Solving phase:

- a. Optimize objective function. Solution is Pareto-optimal.
- b. Use the solution to constrain the $p - 1$ objectives.



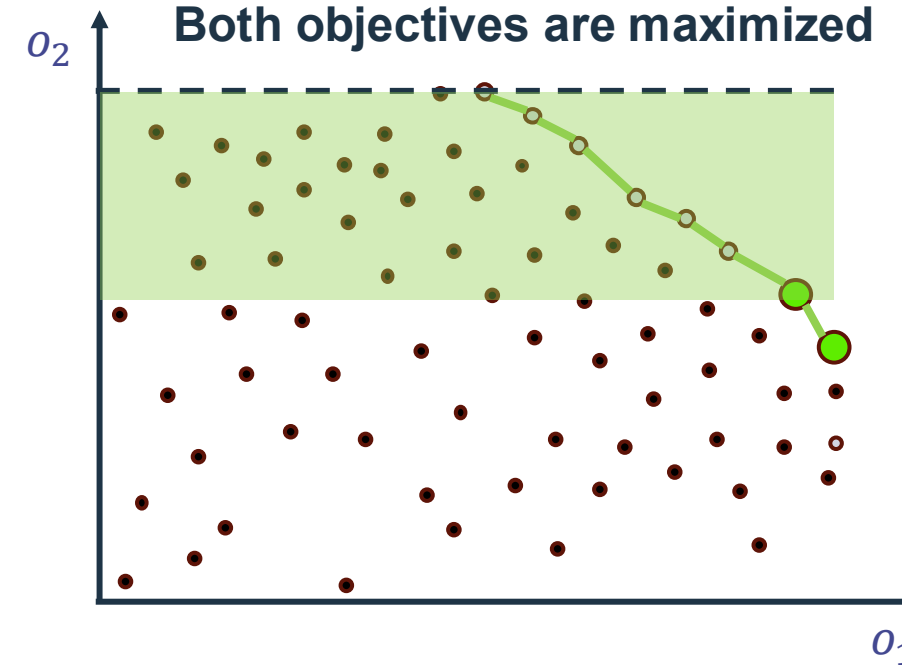
SoTA: SAUGMECON¹

1. **Aggregate objective function.** One objective more important than the rest. $o_1 + \delta(w_2 o_2 + \dots + w_p o_p)$.
2. **Information from previous solutions.** Reduce search space and avoid redundancy. $o_2 > \text{solution}_{o_2}, \dots, o_p > \text{solution}_{o_p}$

Preprocessing: get the best value for $p - 1$ objectives.

Solving phase:

- a. Optimize objective function. Solution is Pareto-optimal.
- b. Use the solution to constrain the $p - 1$ objectives.



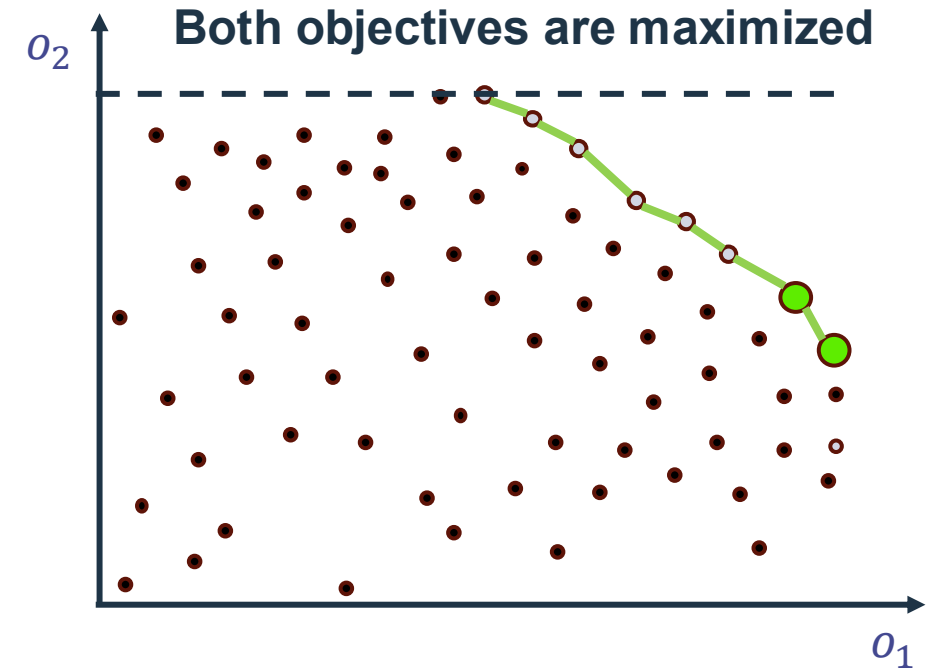
SoTA: SAUGMECON¹

1. **Aggregate objective function.** One objective more important than the rest. $o_1 + \delta(w_2 o_2 + \dots + w_p o_p)$.
2. **Information from previous solutions.** Reduce search space and avoid redundancy. $o_2 > \text{solution}_{o_2}, \dots, o_p > \text{solution}_{o_p}$

Preprocessing: get the best value for $p - 1$ objectives.

Solving phase:

- a. Optimize objective function. Solution is Pareto-optimal.
- b. Use the solution to constrain the $p - 1$ objectives.



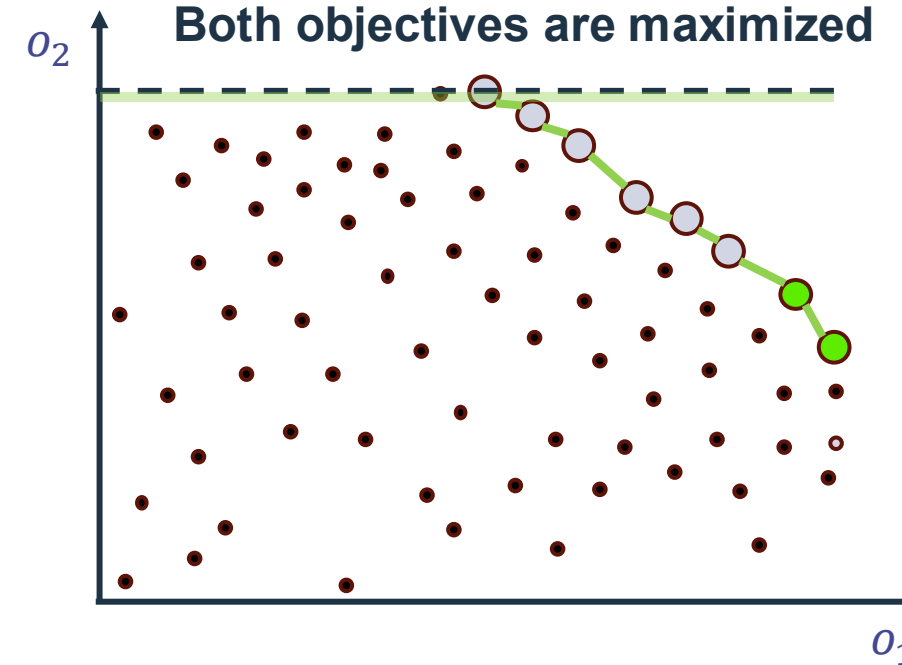
SoTA: SAUGMECON¹

1. **Aggregate objective function.** One objective more important than the rest. $o_1 + \delta(w_2 o_2 + \dots + w_p o_p)$.
2. **Information from previous solutions.** Reduce search space and avoid redundancy. $o_2 > \text{solution}_{o_2}, \dots, o_p > \text{solution}_{o_p}$

Preprocessing: get the best value for $p - 1$ objectives.

Solving phase:

- a. Optimize objective function. Solution is Pareto-optimal.
- b. Use the solution to constrain the $p - 1$ objectives.
- c. Stops when objective p cannot be further improved/constrained.



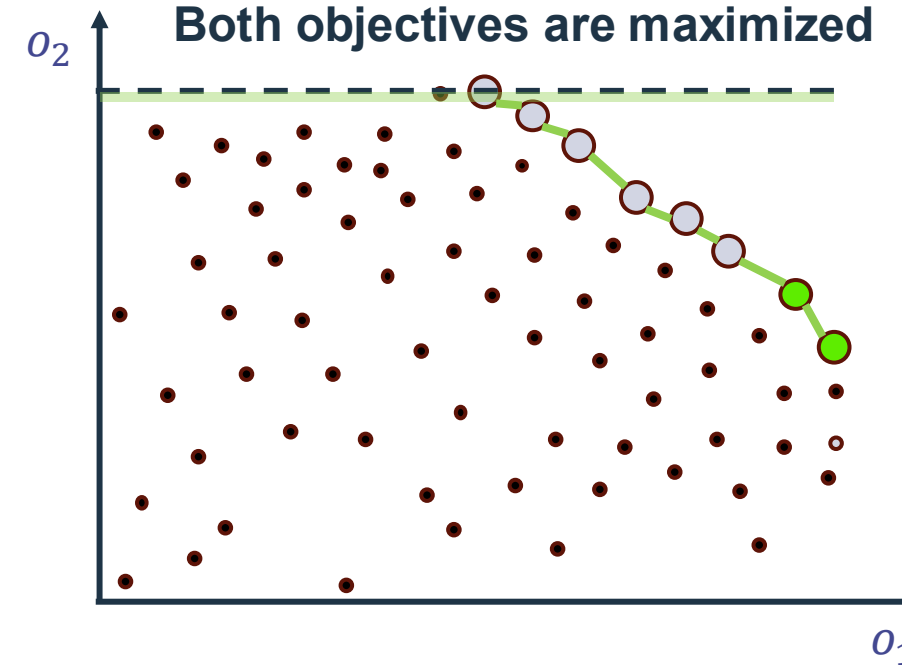
SoTA: SAUGMECON¹

1. **Aggregate objective function.** One objective more important than the rest. $o_1 + \delta(w_2 o_2 + \dots + w_p o_p)$.
2. **Information from previous solutions.** Reduce search space and avoid redundancy. $o_2 > \text{solution}_{o_2}, \dots, o_p > \text{solution}_{o_p}$

Preprocessing: get the best value for $p - 1$ objectives.

Solving phase:

- a. Optimize objective function. Solution is Pareto-optimal.
- b. Use the solution to constrain the $p - 1$ objectives.
- c. Stops when objective p cannot be further improved/constrained.



Cons:

- Need at least one search restart per Pareto-optimal solution.
- Can re-explore parts of the search tree already visited.

Contribution: avoiding redundancy in SAUGMECON

Problem: reduce the number of search restarts needed to find the Pareto front.

- Ideally every restart should return a different Pareto point.
- SAUGMECON can still return a Pareto point already found.

Contribution: avoiding redundancy in SAUGMECON

Problem: reduce the number of search restarts needed to find the Pareto front.

- Ideally every restart should return a different Pareto point.
- SAUGMECON can still return a Pareto point already found.

Solution:

Combine SAUGMECON with the Pareto global constraint.

- a. Add each Pareto-optimal solution found to the Pareto global constraint. This avoids obtaining the same Pareto point again.

Contribution: avoiding redundancy in SAUGMECON

Problem: reduce the number of search restarts needed to find the Pareto front.

- Ideally every restart should return a different Pareto point.
- SAUGMECON can still return a Pareto point already found.

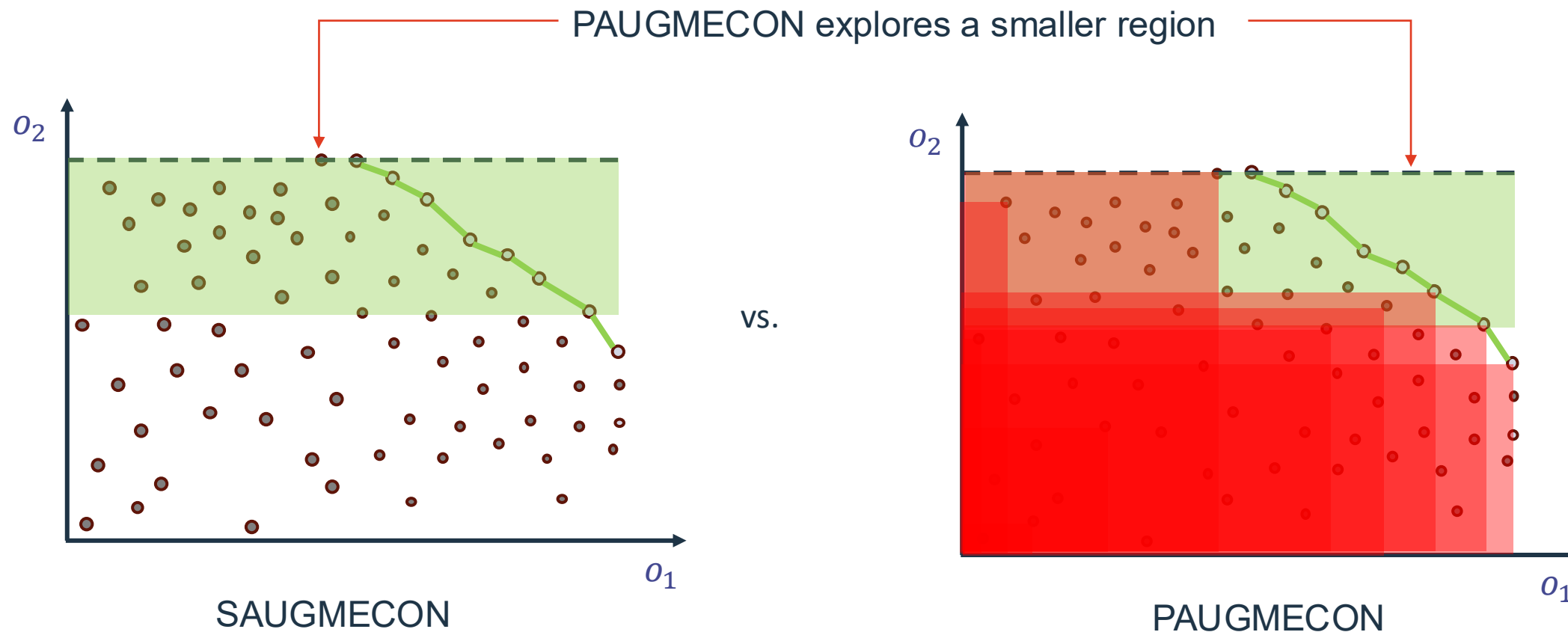
Solution:

Combine SAUGMECON with the Pareto global constraint.

- a. Add each Pareto-optimal solution found to the Pareto global constraint. This avoids obtaining the same Pareto point again.

Can be further improved?

SAUGMECON vs. PAUGMECON



Contribution: PAUGMECON a new multi-objective CP algorithm

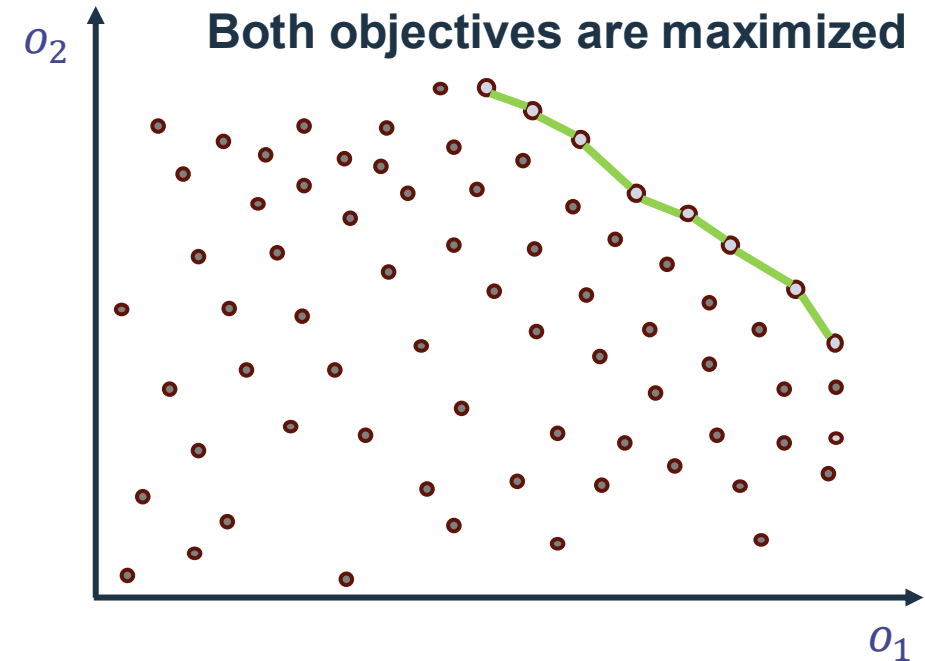
Improvements:

1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.

Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

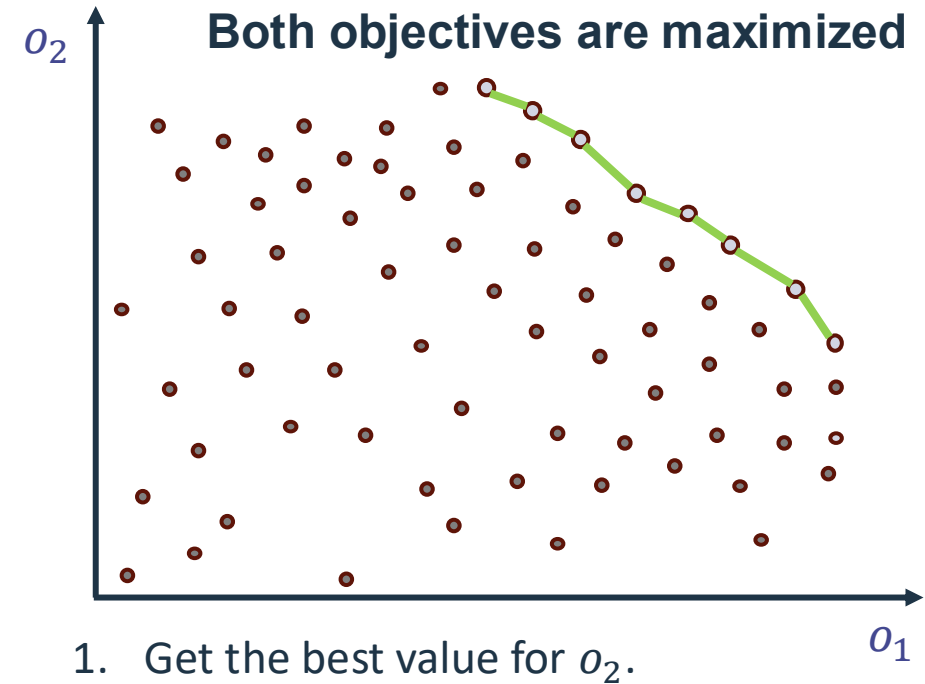
1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.



Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

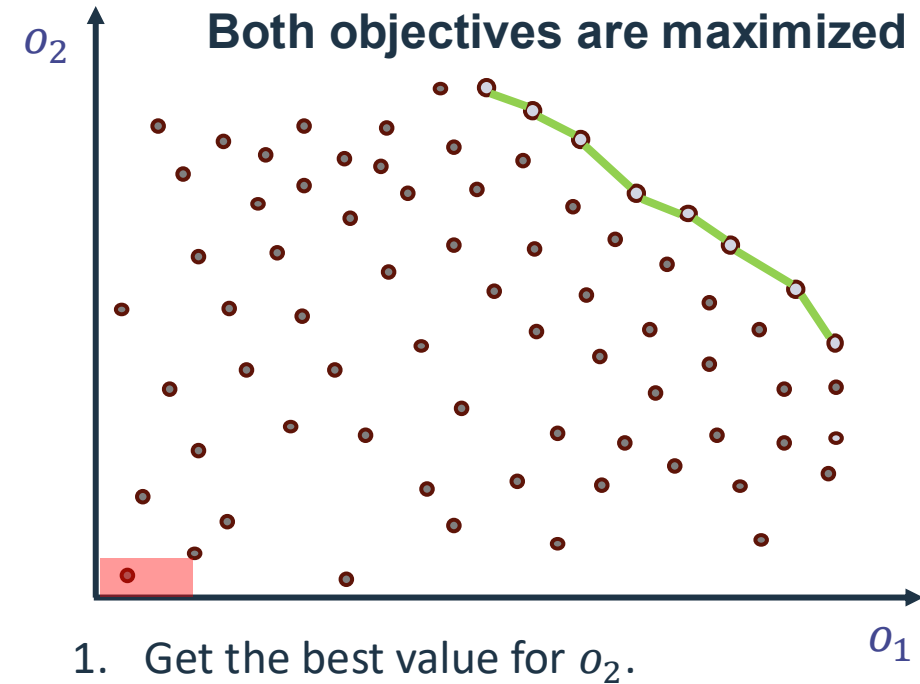
1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.



Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

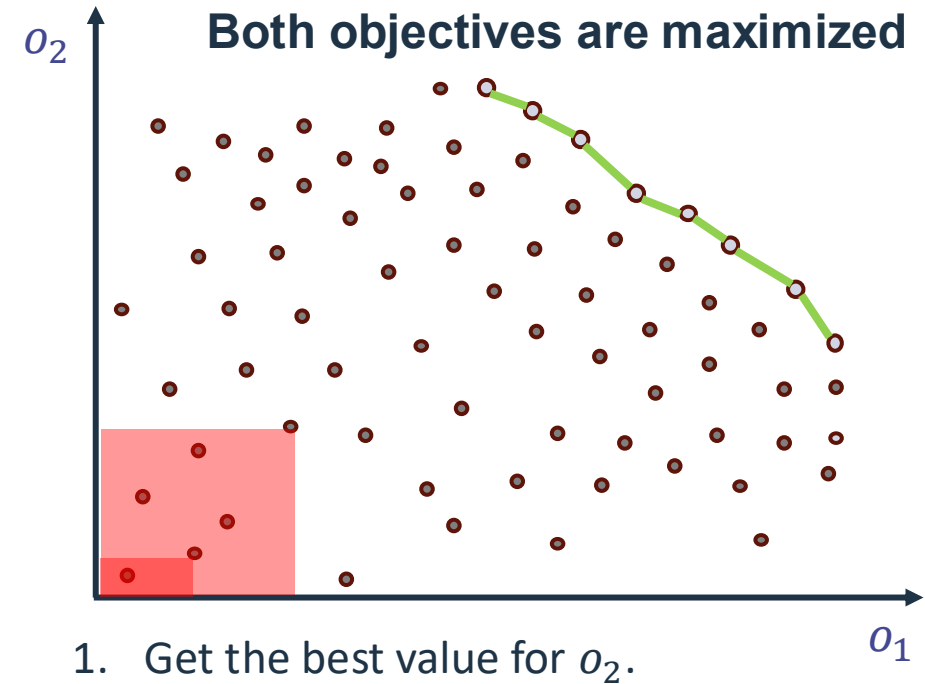
1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.



Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

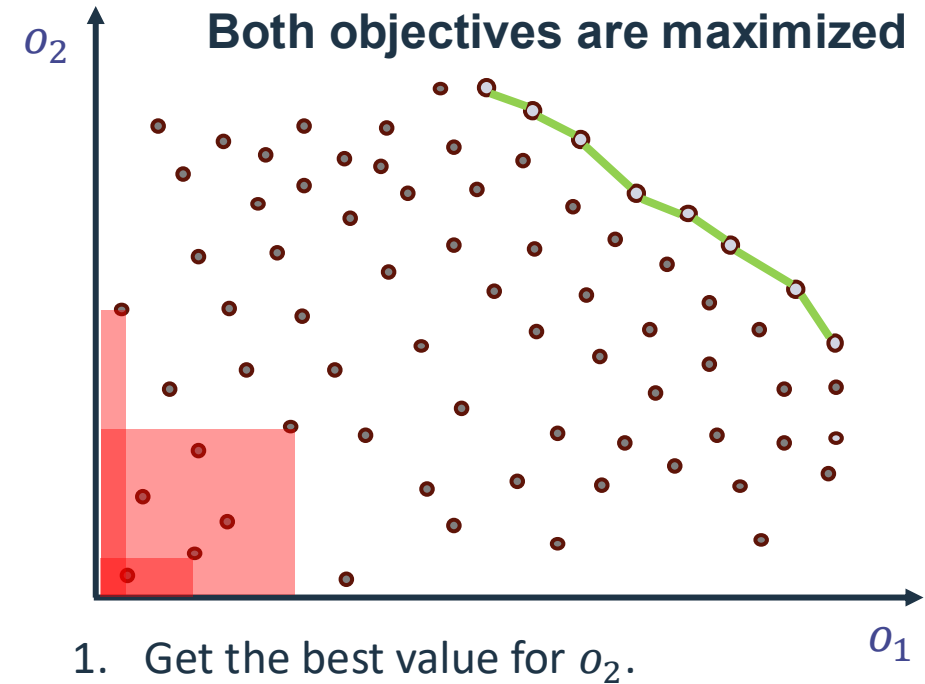
1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.



Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

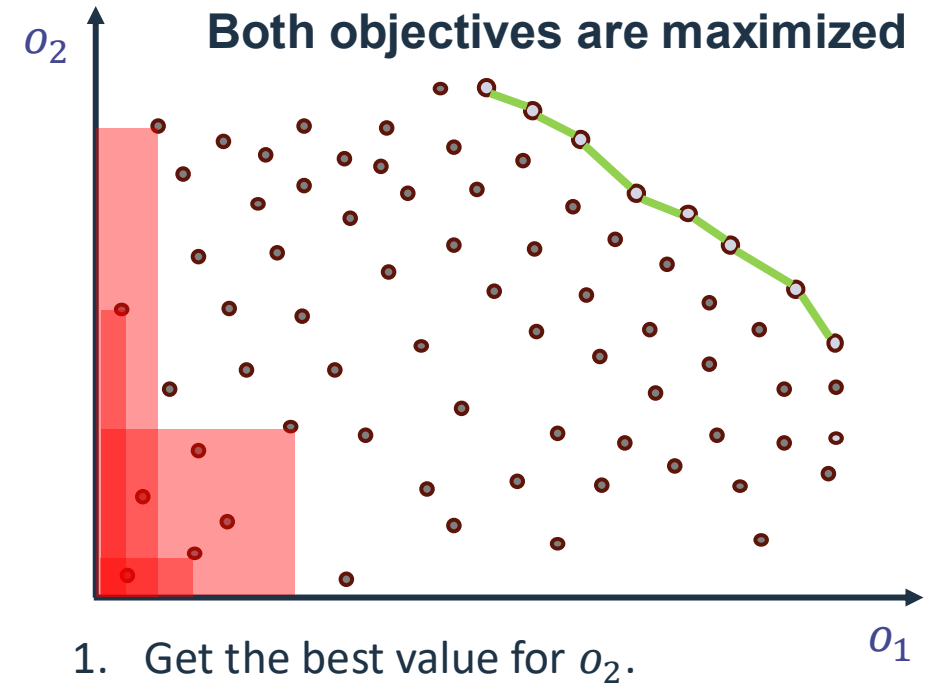
1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.



Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

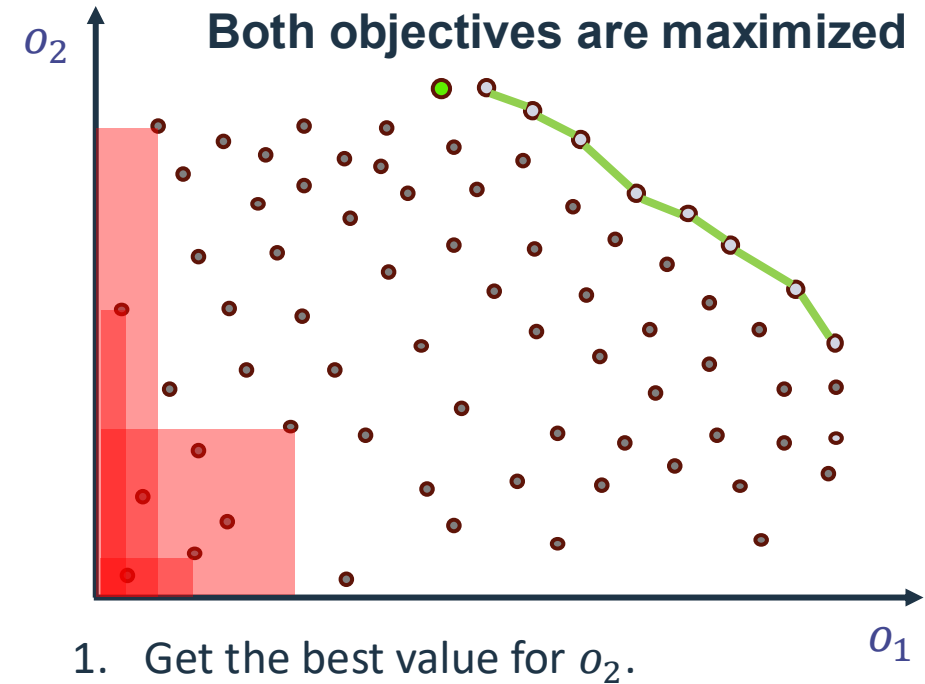
1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.



Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

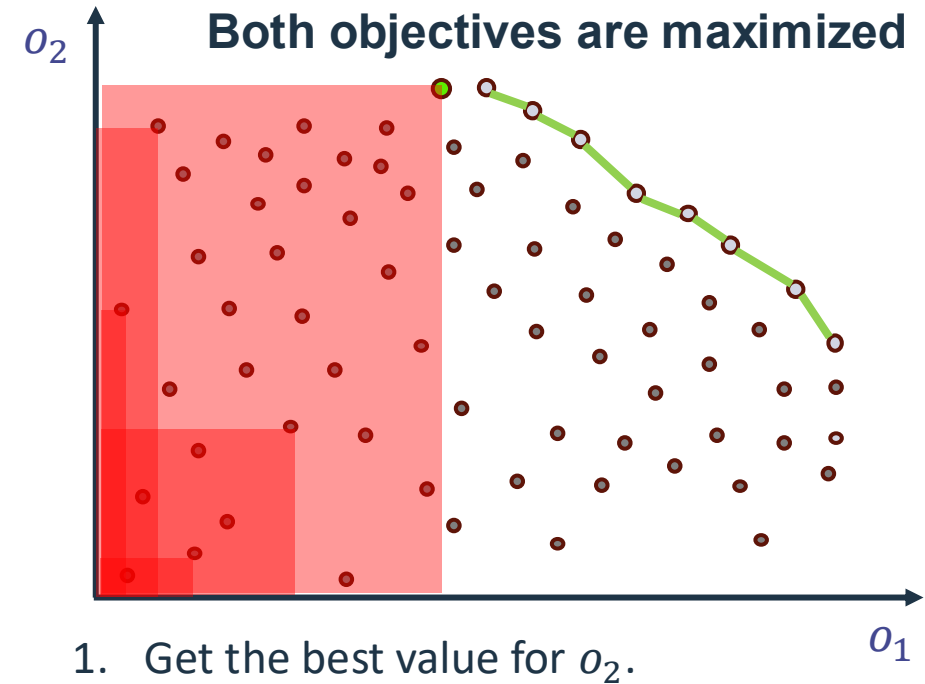
1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.



Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

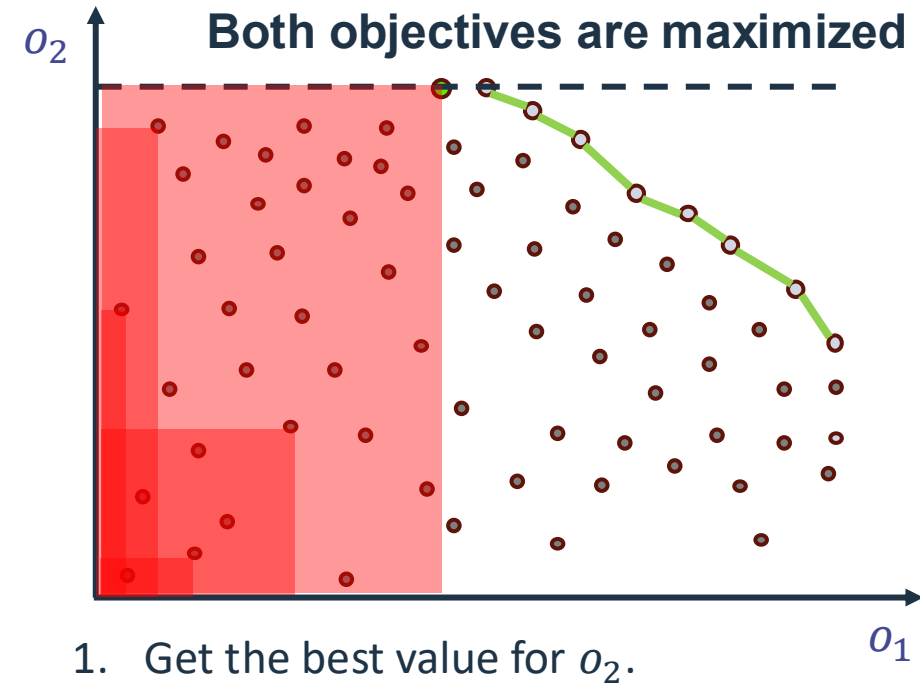
1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.



Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

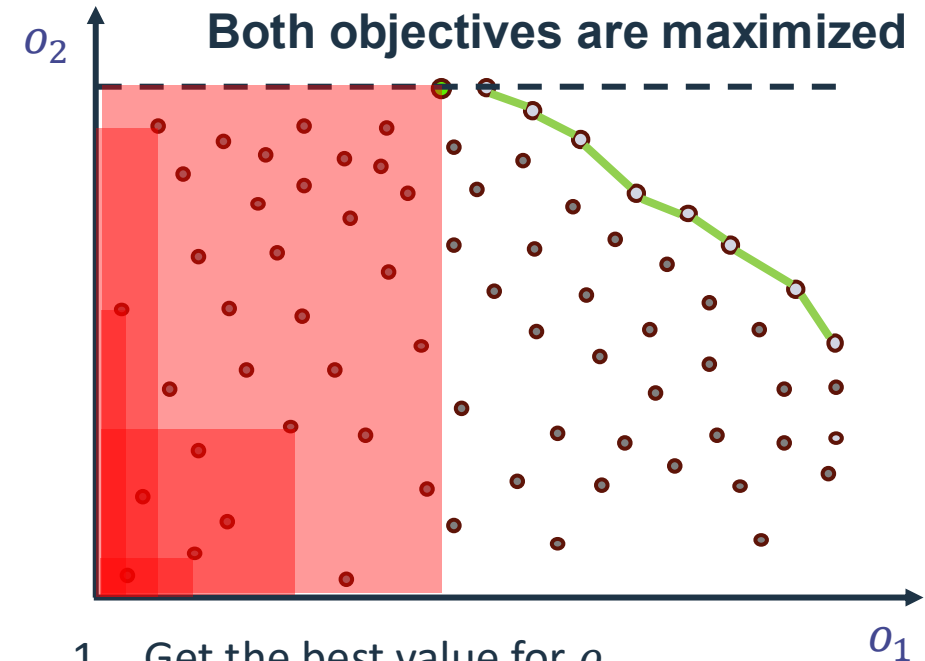
1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.



Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.

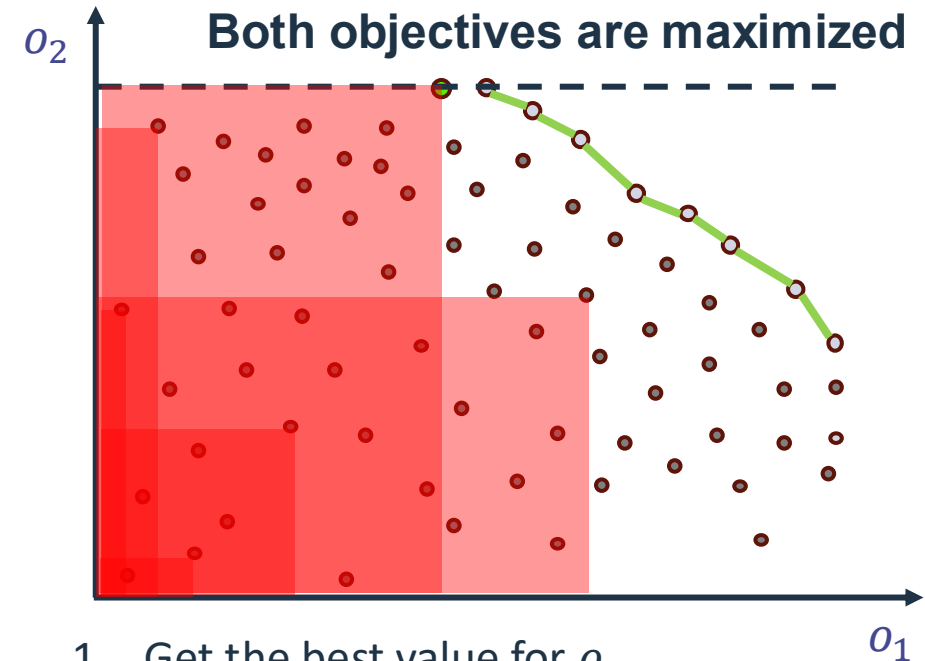


1. Get the best value for o_2 .
2. Do lexicographic optimization o_1, o_2 instead of using SAUGMECON objective function.

Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.

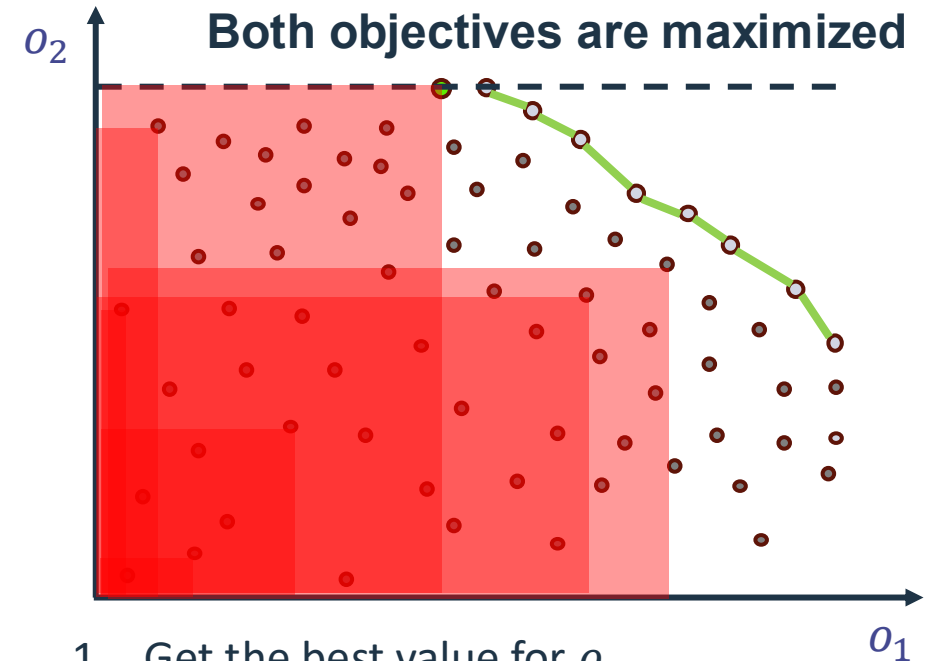


1. Get the best value for o_2 .
2. Do lexicographic optimization o_1, o_2 instead of using SAUGMECON objective function.

Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.

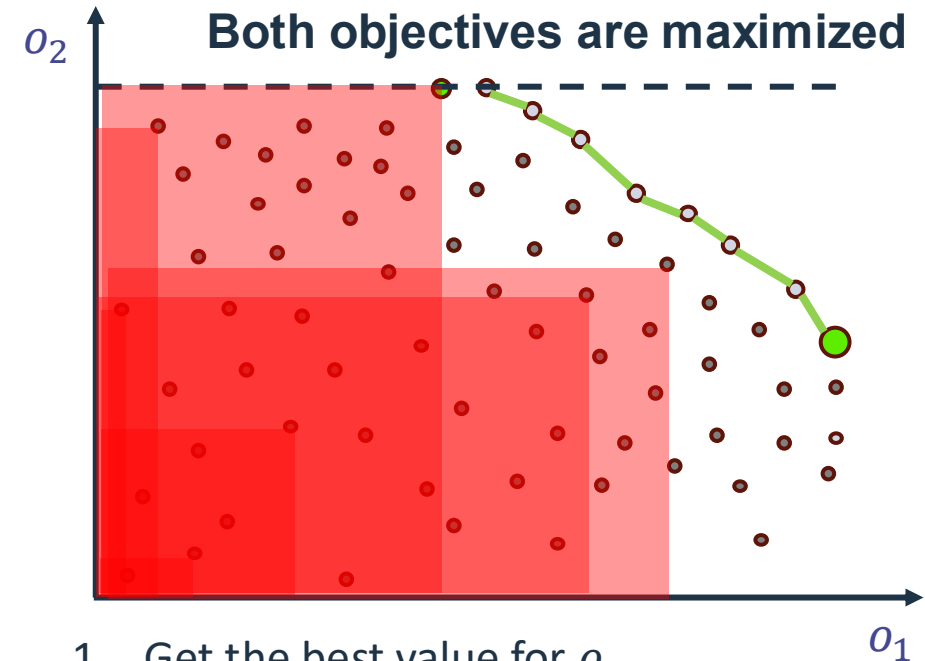


1. Get the best value for o_2 .
2. Do lexicographic optimization o_1, o_2 instead of using SAUGMECON objective function.

Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.

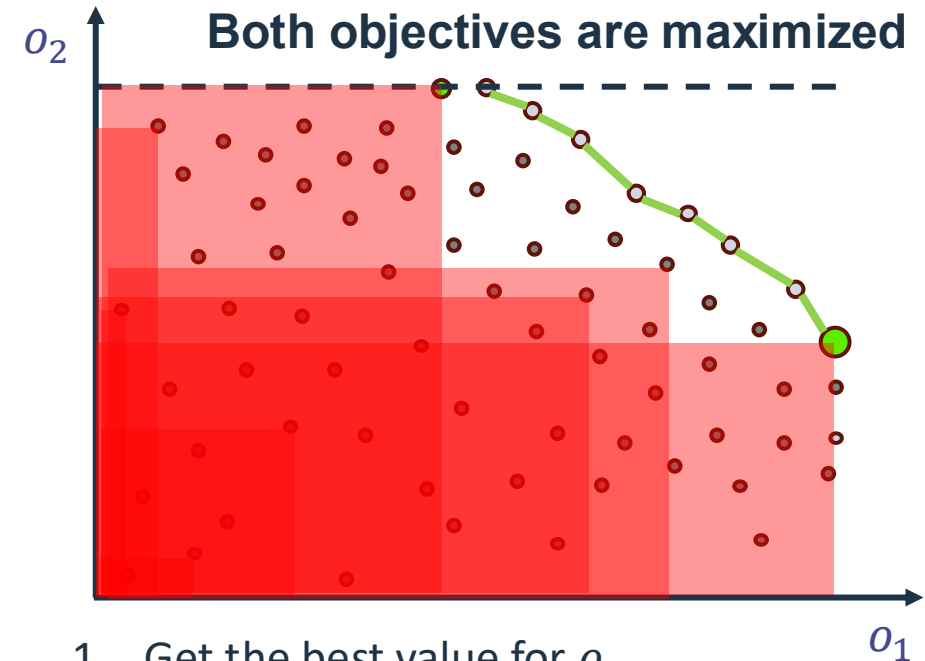


1. Get the best value for o_2 .
2. Do lexicographic optimization o_1, o_2 instead of using SAUGMECON objective function.

Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.

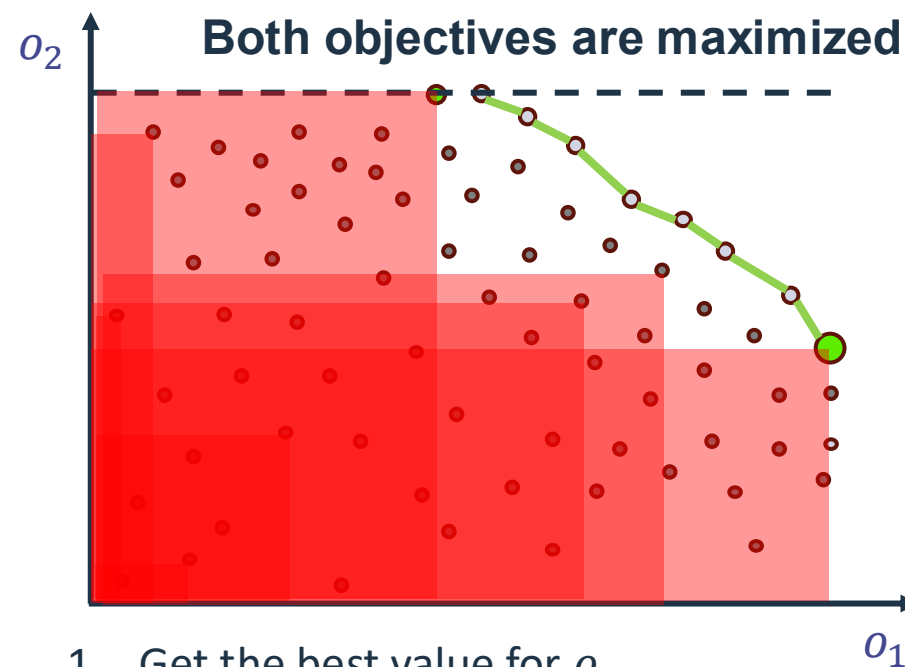


1. Get the best value for o_2 .
2. Do lexicographic optimization o_1, o_2 instead of using SAUGMECON objective function.

Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.

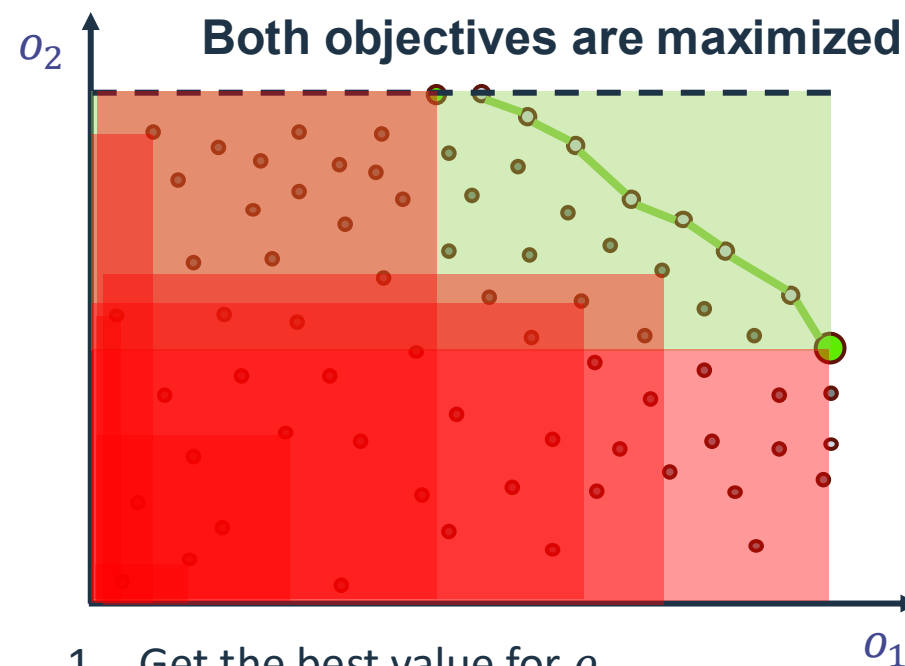


1. Get the best value for o_2 .
2. Do lexicographic optimization o_1, o_2 instead of using SAUGMECON objective function.
3. Explore the objective space by constraining o_2 .

Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.

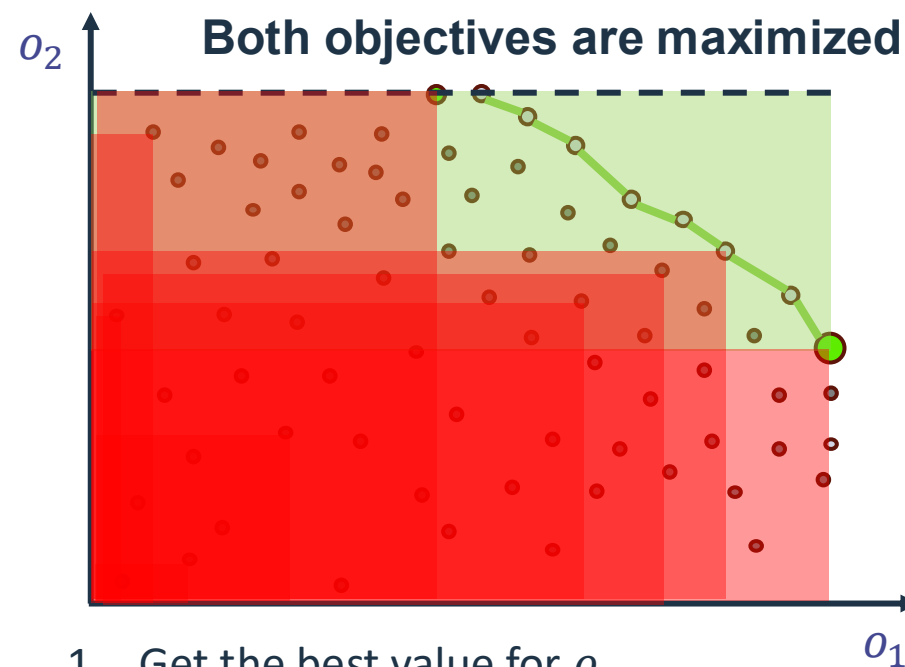


1. Get the best value for o_2 .
2. Do lexicographic optimization o_1, o_2 instead of using SAUGMECON objective function.
3. Explore the objective space by constraining o_2 .

Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.

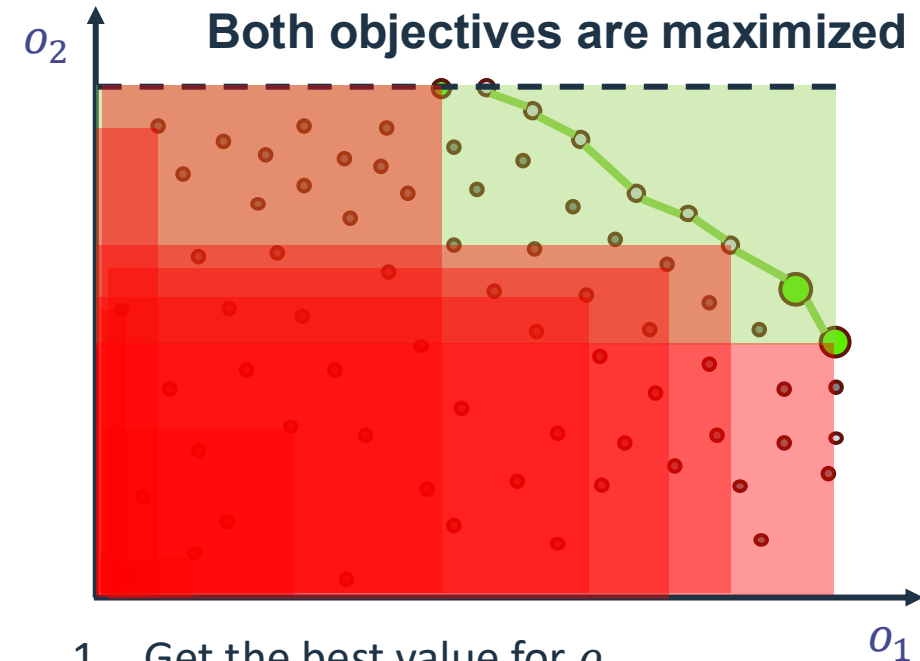


1. Get the best value for o_2 .
2. Do lexicographic optimization o_1, o_2 instead of using SAUGMECON objective function.
3. Explore the objective space by constraining o_2 .

Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.

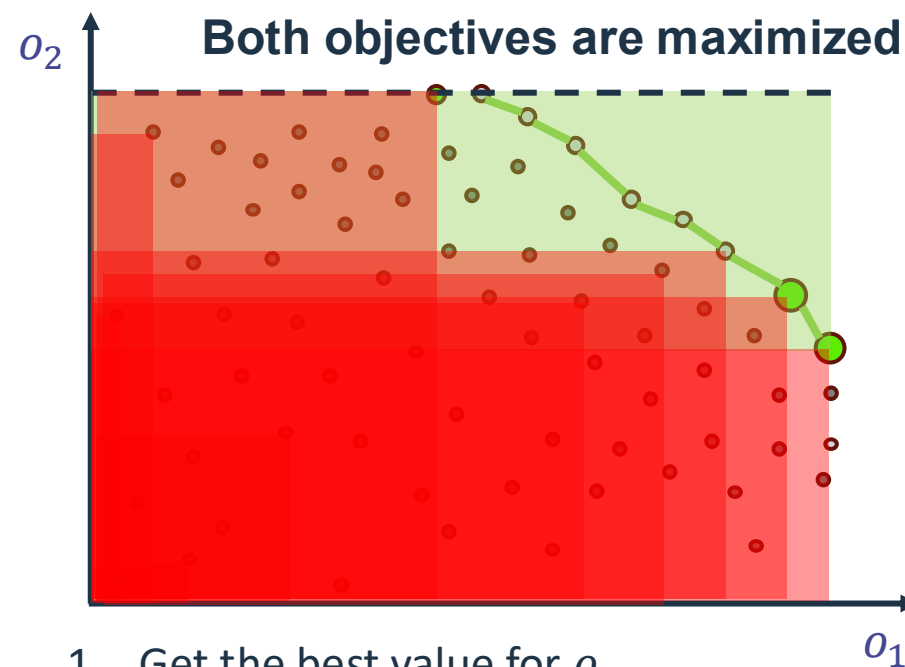


1. Get the best value for o_2 .
2. Do lexicographic optimization o_1, o_2 instead of using SAUGMECON objective function.
3. Explore the objective space by constraining o_2 .

Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.

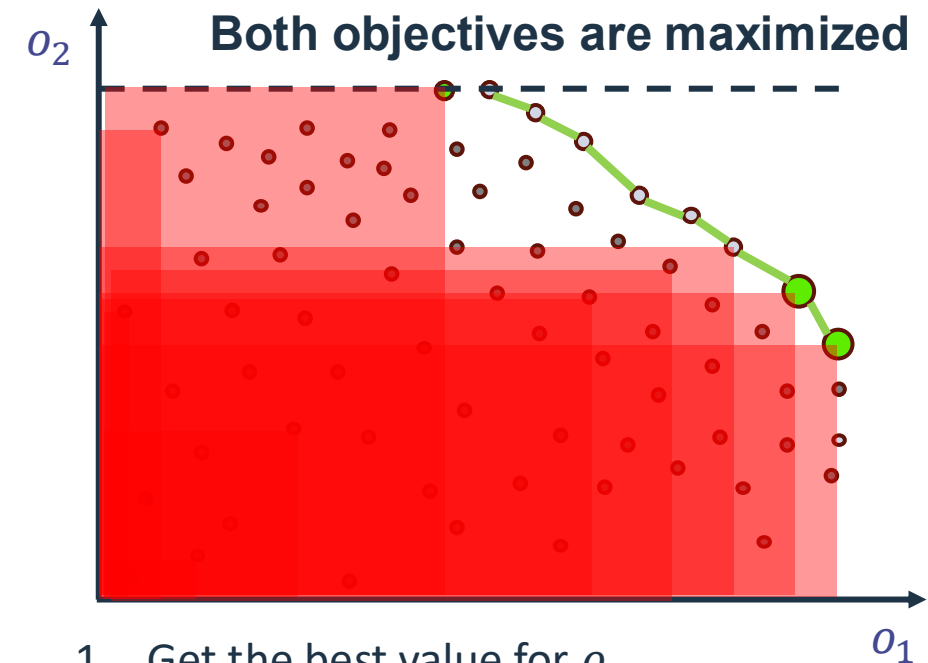


1. Get the best value for o_2 .
2. Do lexicographic optimization o_1, o_2 instead of using SAUGMECON objective function.
3. Explore the objective space by constraining o_2 .

Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.

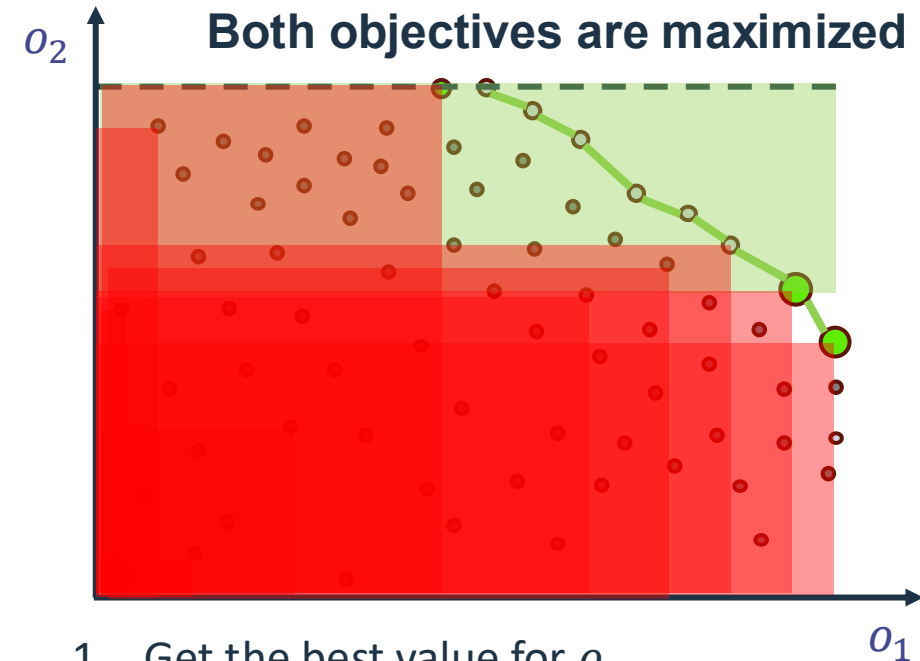


1. Get the best value for o_2 .
2. Do lexicographic optimization o_1, o_2 instead of using SAUGMECON objective function.
3. Explore the objective space by constraining o_2 .

Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.

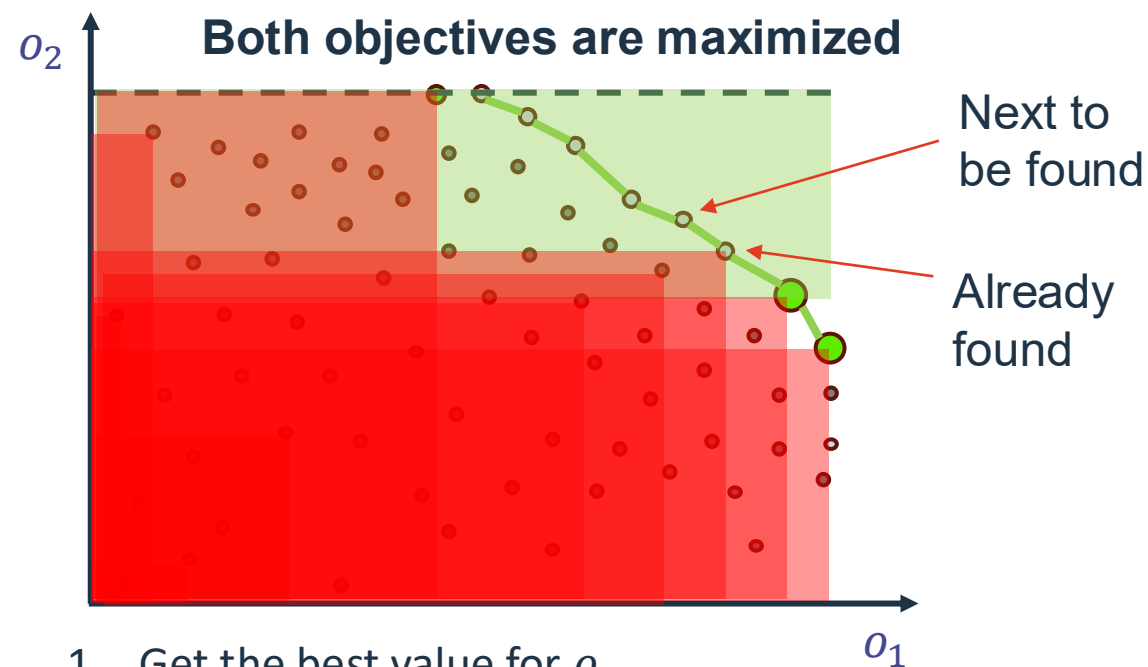


1. Get the best value for o_2 .
2. Do lexicographic optimization o_1, o_2 instead of using SAUGMECON objective function.
3. Explore the objective space by constraining o_2 .

Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.



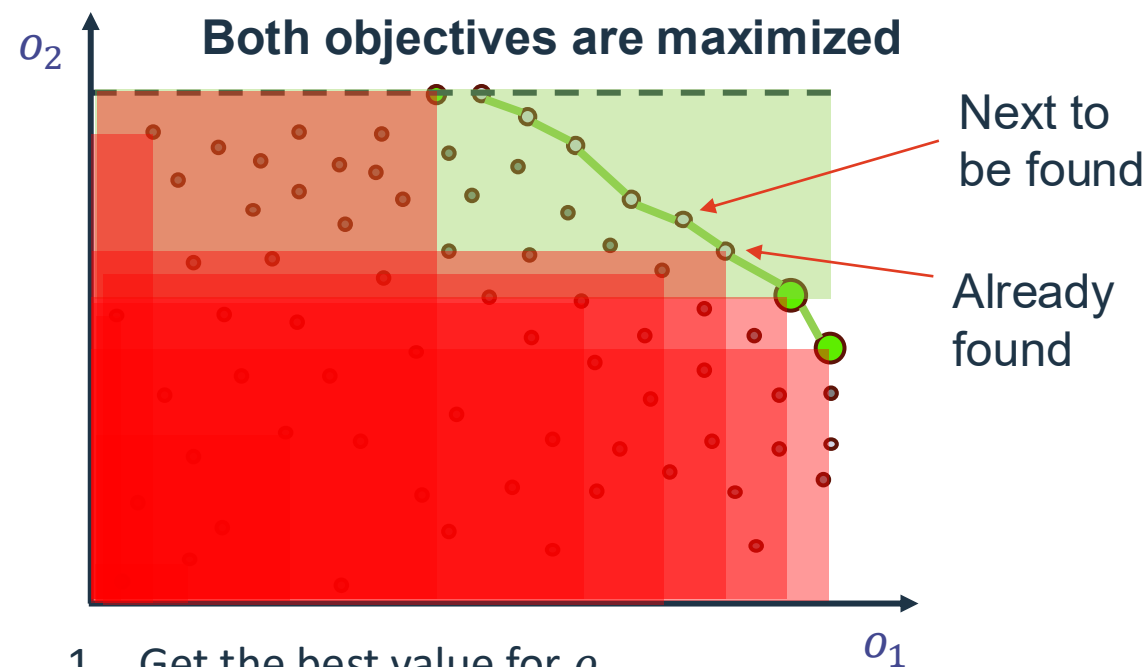
1. Get the best value for o_2 .
2. Do lexicographic optimization o_1, o_2 instead of using SAUGMECON objective function.
3. Explore the objective space by constraining o_2 .

Contribution: PAUGMECON a new multi-objective CP algorithm

Improvements:

1. Add to the Pareto constraint the **intermediate** solutions.
2. Use the Pareto constraint from **preprocessing**, when individually **optimizing** the $p - 1$ objectives.

PAUGMECON implements 3 more mechanisms, check the paper.



1. Get the best value for o_2 .
2. Do lexicographic optimization o_1, o_2 instead of using SAUGMECON objective function.
3. Explore the objective space by constraining o_2 .

Experimental results

Experiments

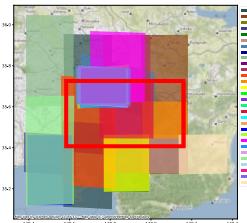
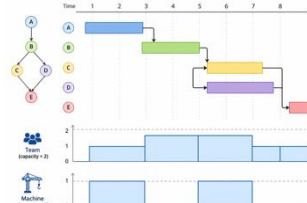
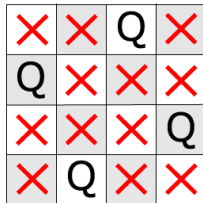
CP solver: Choco 5.0.0. Search strategy: dom/wdeg.

Two “toy” problems from 2 to 5 objectives. Timeout 1.5h.

- Knapsack: 2 – 5 objectives. 110 instances.
- N-Queens: 2 – 5 objectives. 160 instances.

Two “real-world” problems from 2 to 3 objectives. Timeout 3h.

- Resource-constrained project scheduling problem (RCPSP): 2 objectives. 70 instances.
- Satellite image mosaic selection problem (SIMS)¹: 2 and 3 objectives. 48 instances.



Experiments

Performance indicators:

- Time, when the Pareto front was found for almost all instances.
- Otherwise, we measure convergence and diversity:
 - Hypervolume (HV).
 - Inverted generational distance with modified distance (IGD⁺).

Knapsack – time

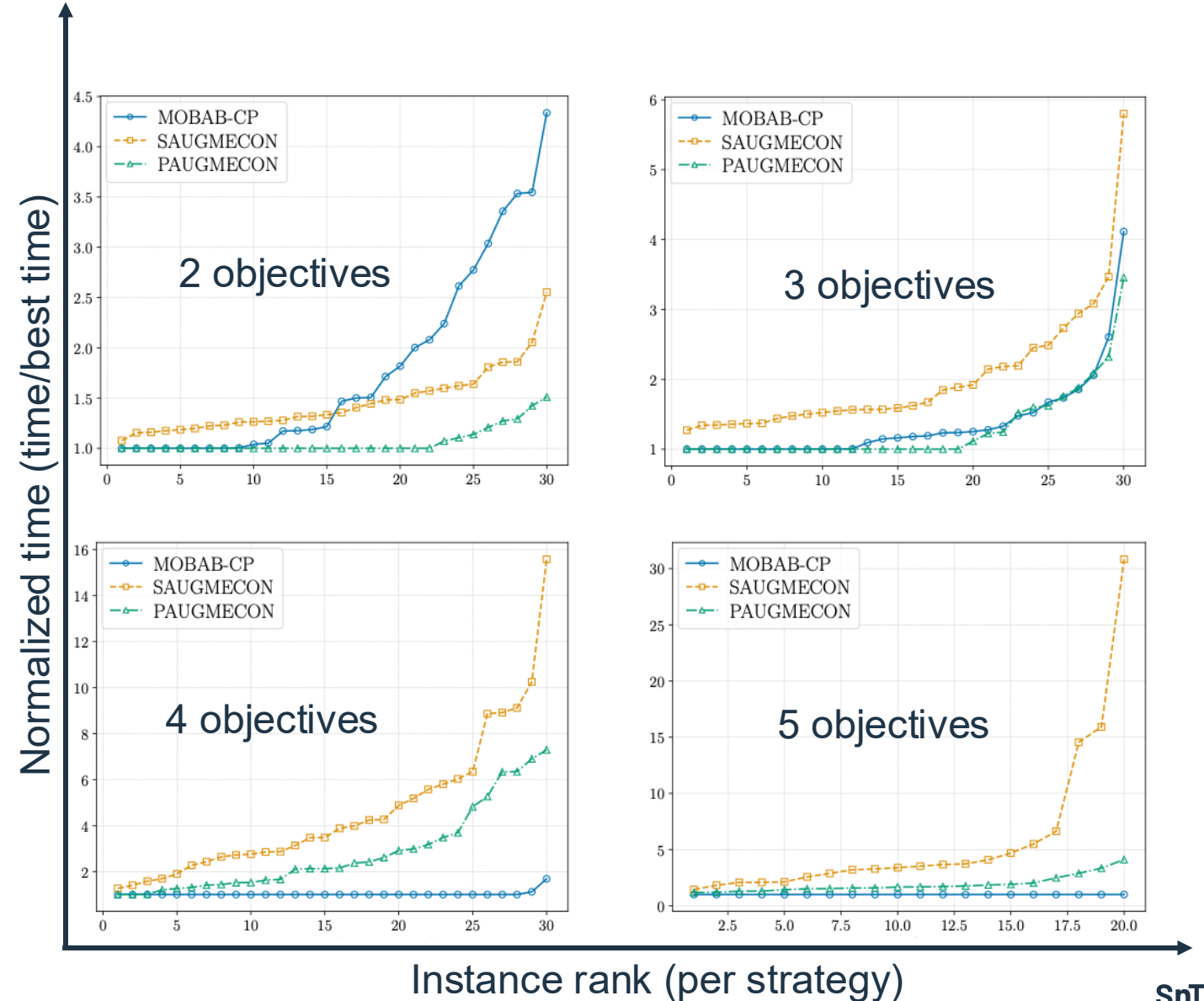
Each curve is sorted independently from best relative time to worst.

Lower curves are better.

- **PAUGMECON** best for 2 and 3 objectives. Better than the adapted algorithms!
- **MOBAB-CP** best for 4 and 5.

Intuition:

Number of restarts may increase with the number of objectives.



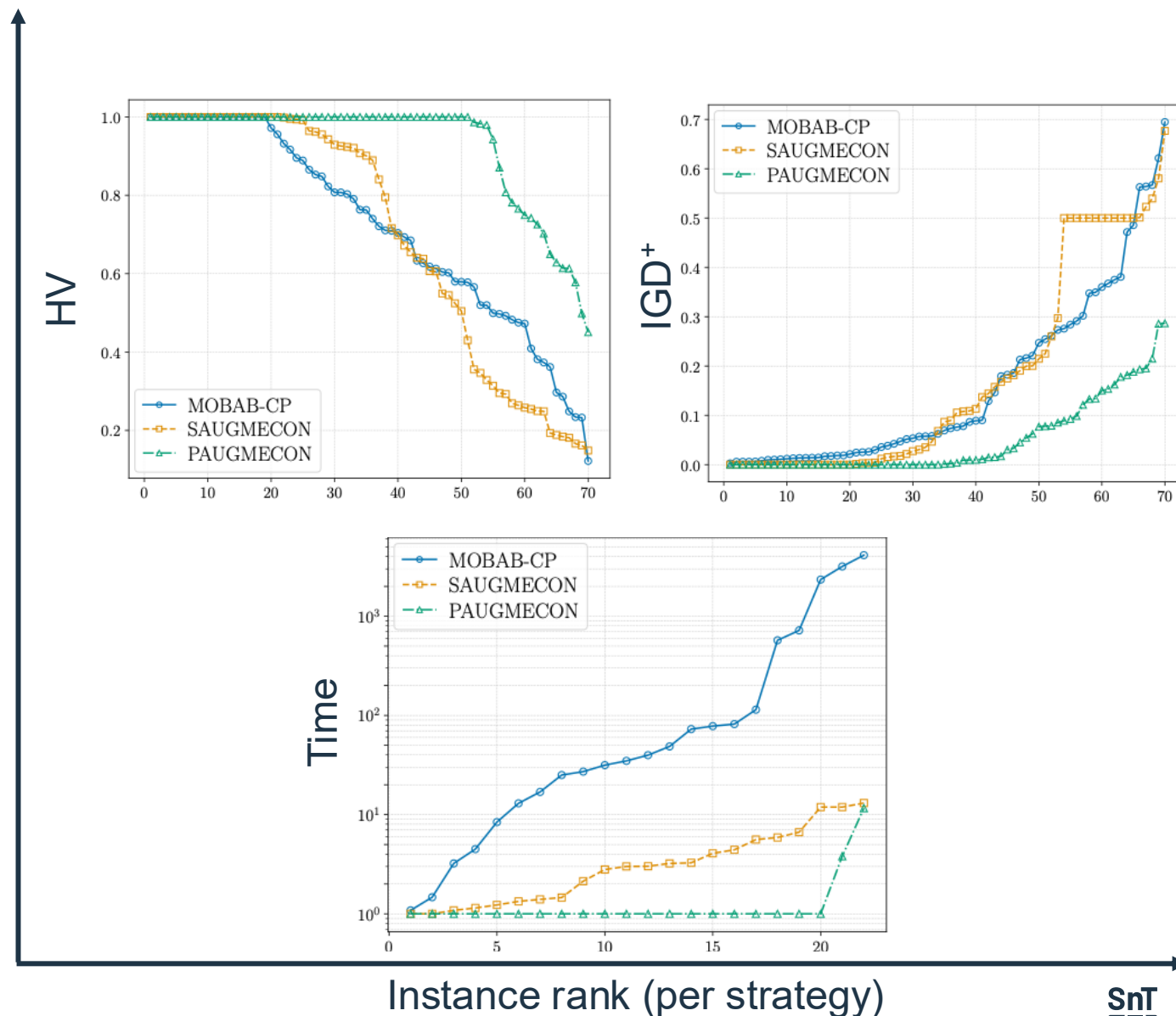
RCPSP

Instances with 2 objectives.

HV: Higher curves are better.

IGD⁺: Lower curves are better.

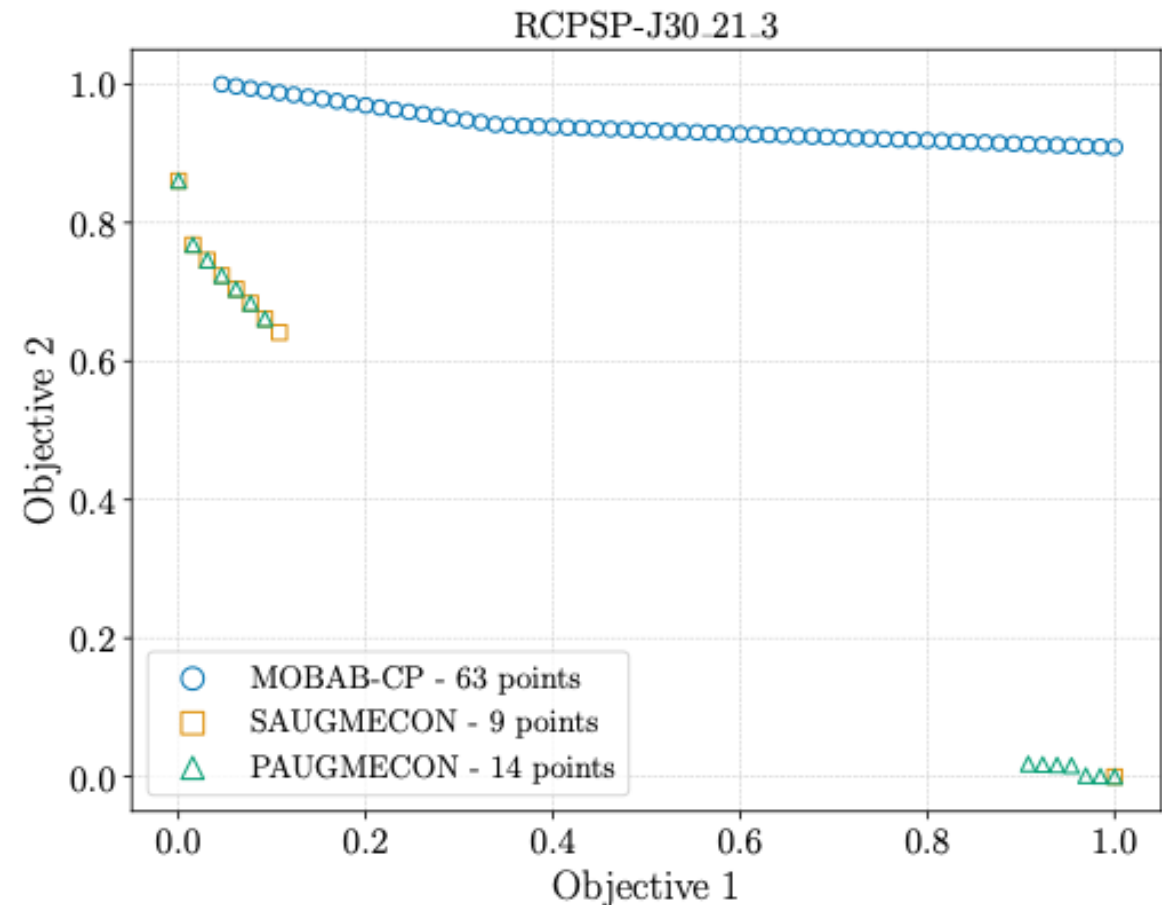
- **PAUGMECON** best for all metrics! Solved more instances 22 / 70.
 - Faster than SAUGMECON.
- **MOBAB-CP** worst for IGD⁺. Could not complete any instance.



RCPSP intuition

- **MOBAB-CP.** Solutions close in the search tree may not dominate solutions in the current front.
 - Finds many solutions that do not improve the current front.
- The other algorithms find better-quality solutions.

Both objectives are minimized



Summary of experiments

Evaluation considering **HV**, **IGD⁺** and **time**

Problem	Best algorithm	Worst algorithm
MUKP 2,3	PAUGMECON	MOBAB-CP (2 obj); SAUGMECON (3 obj)
MUKP 4,5	MOBAB-CP	SAUGMECON
N-Queens	MOBAB-CP	SAUGMECON
RCPSP	PAUGMECON	SAUGMECON (HV); MOBAB-CP (IGD ⁺)
SIMS	PAUGMECON	MOBAB-CP (HV); SAUGMECON (IGD ⁺)

Take away: PAUGMECON is a robust choice for bi- and tri-objective problems. It is the only algorithm that was **never the worst**. It is the best in **bi-objective** and **tri-objective** settings, except for **N-Queens**, where MOBAB-CP was the best.

Conclusion & perspectives

Conclusion

- We have proposed a **new** multi-objective **algorithm** in CP: **PAUGMECON**, which combines SAUGMECON with the Pareto global constraint to reduce redundant search. It was the **best overall algorithm** for **bi-objective** and **tri-objective** problems.

Perspectives

- **New exact CP algorithms** – Combine ideas from and ILP with CP-specific mechanisms, in particular the Pareto global constraint. Empirical results suggest that this combination is promising.
- **PAUGMECON** – Improve it for more than 3 objectives. Study implementation in ILP solvers.



Interdisciplinary Centre
for Security, Reliability
and Trust

Contact



Manuel Combarro Simón

Postdoctoral researcher
manuel.combarrosimon@uni.lu

Connect with us



SnT, Interdisciplinary Centre for
Security, Reliability and Trust

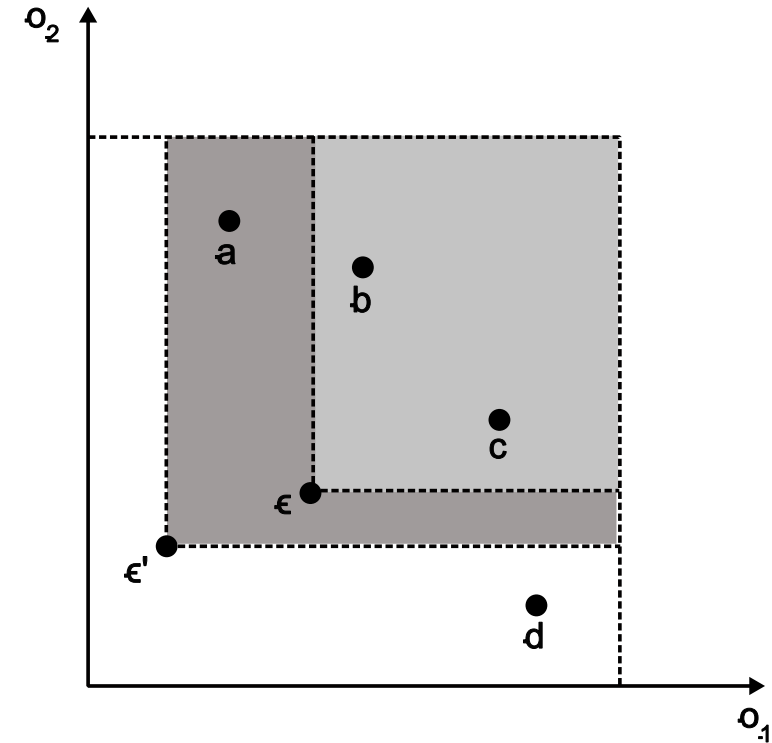
Scan here to read the [paper](#)



PAUGMECON: additional improvements

Non-dominated solutions so far $A = \{a, b, c, d\}$

Additional improvements:

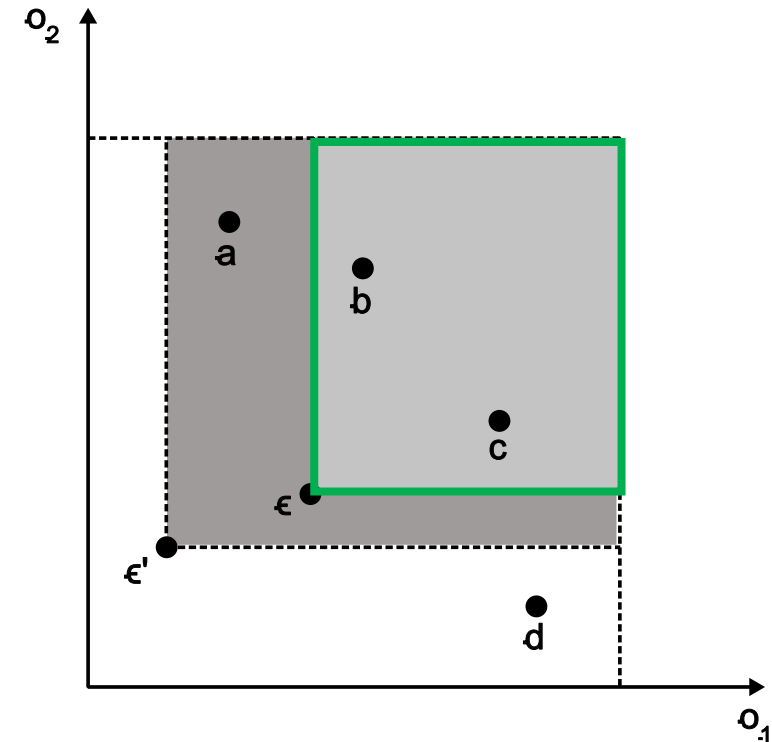


PAUGMECON: additional improvements

Non-dominated solutions so far $A = \{a, b, c, d\}$

Additional improvements:

3. Reduce Pareto constraint overhead by using just the necessary subset of the solutions. $A = \{a, b, c, d\} \rightarrow A' = \{b, c\}$.

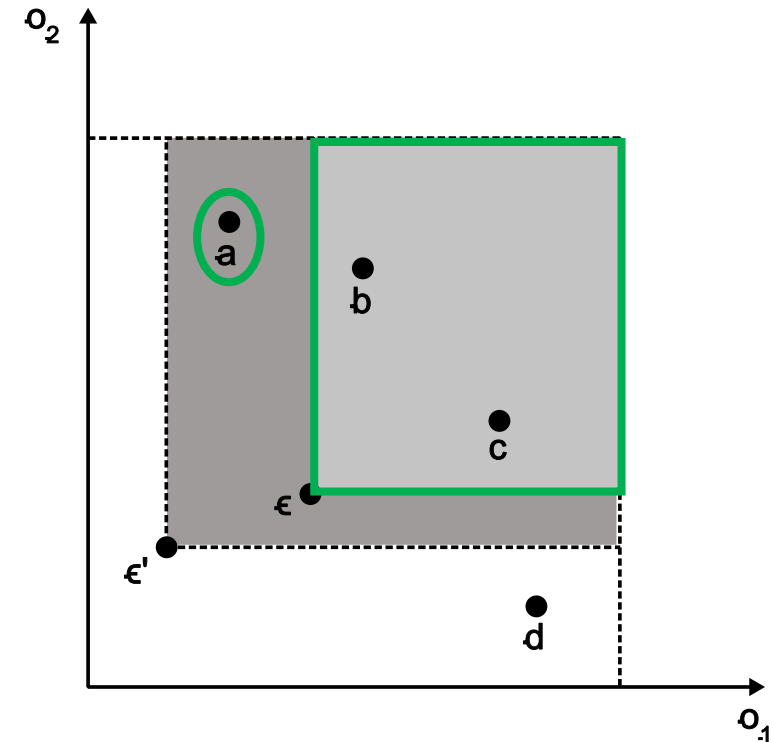


PAUGMECON: additional improvements

Non-dominated solutions so far $A = \{a, b, c, d\}$

Additional improvements:

3. Reduce Pareto constraint overhead by using just the necessary subset of the solutions. $A = \{a, b, c, d\} \rightarrow A' = \{b, c\}$.
4. Derive an upper bound for objective p (main objective).
 ϵ' solved before than ϵ .



PAUGMECON: additional improvements

Non-dominated solutions so far $A = \{a, b, c, d\}$

Additional improvements:

3. Reduce Pareto constraint overhead by using just the necessary subset of the solutions. $A = \{a, b, c, d\} \rightarrow A' = \{b, c\}$.
4. Derive an upper bound for objective p (main objective).
 ϵ' solved before than ϵ .
5. Reuse of information from previous solutions considering the Pareto constraint.
 $A = \{b, c, d\}$, solve ϵ' returns a and $a(o_3) < b(o_3), c(o_3)$
 b and c are associated with ϵ' .
 ϵ will return either b or c , can be skipped.

