# Robust indicator-based algorithm for interactive evolutionary multiple objective optimization

Michał Tomczyk, Miłosz Kadziński

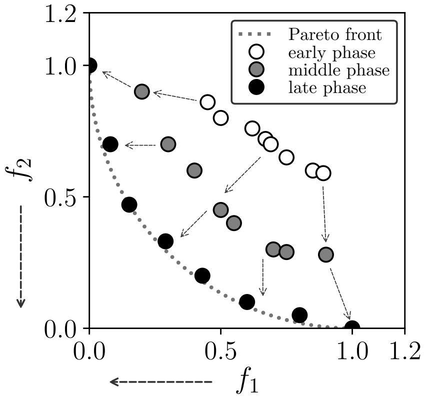
*Laboratory of Intelligent Decision Support Systems*

*Institute of Computing Science*

*Poznań University of Technology*

July 15, 2019

# Evolutionary Multiple-objective Optimization (EMO)

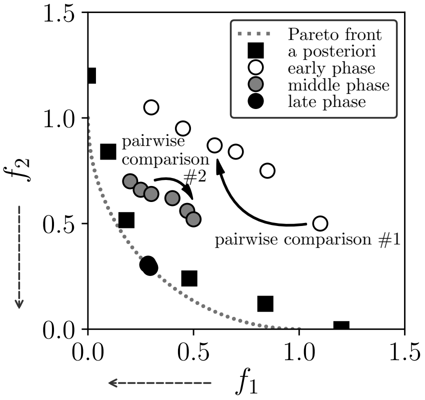



**Evolutionary Algorithms for MOO**

Mimic the process of naturall evolution to solve optimization problems

Advantages of EMO:
– can be applied to problems having complex fitness landscapes
– the computational complexity can be reduced since solutions are optimized in an interrelated manner

# Preference vs. non preference-based EMOAs




**Preference-based EMOAs**

Observation: it is not practical to approximate an entire PF since the DM is interested in finding only relevant solutions to him or her

**Incorporation of DM's preferences**

Preference information can be used to **constrain** the search space, thereby reducing the complexity of the problem

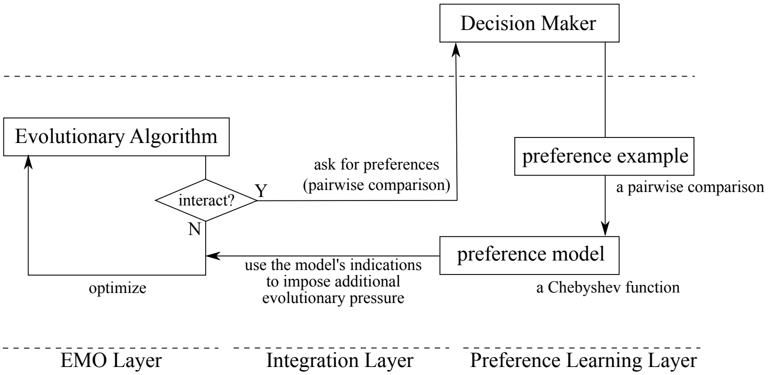
The preference information can be used to impose an additional selection pressure, converging a population of solutions towards **a region of PF being highly preferred to the DM**

# The proposed method – IEMO/I

## Characteristics of the proposed method, called IEMO/I

- ▷ is interactive
- ▷ asks the DM to holistically compare some pairs of solutions
- ▷ employs a relatively simple preference model in a form of a Chebyshef function
- ▷ implements a robustness concern
- ▷ uses an indicator-based evolutionary framework

# Scheme of an interactive EMOA



# Preference modeling in IEMO/I, which

## Preference model

We assume that the DM's value system can be modeled with a Chebyshef function:

$$f^{CF}(s) = \max_{i \in \{1,\ldots,M\}} w_i s_i$$

**Parameters**: weights

## Preference example

The DM is presented a pair of solutions form the current population. (S)he is asked to compare them, i.e., judge which one (s)he prefers more.

$$s^a \succ s^b \rightarrow \max_{i \in \{1,\ldots,M\}} w_i s_i^a < \max_{i \in \{1,\ldots,M\}} w_i s_i^b$$

# Preference modeling in IEMO/I

## Set of constraints

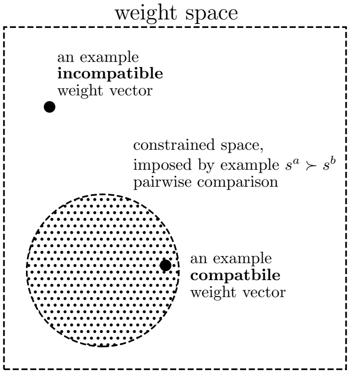
The decision examples can be used to constrain the weight space:

$$\underset{s^a \succ s^b \in \mathcal{H}}{\forall} \max_{i \in \{1,\ldots,M\}} w_i s_i^a < \max_{i \in \{1,\ldots,M\}} w_i s_i^b,$$

$$\sum_{i=1}^{M} w_i = 1,$$

$$\underset{i \in \{1,\ldots,M\}}{\forall} w_i \geq 0.$$

## Compatible model instances



# Preference modeling in IEMO/I

## Set of constraints

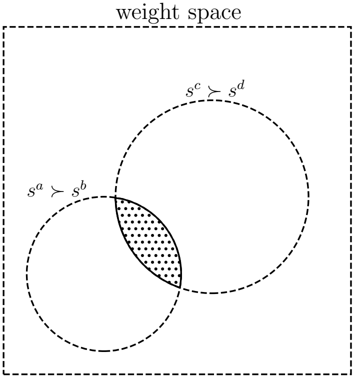
The decision examples can be used to constrain the weight space:

$$\bigforall_{s^a \succ s^b \in \mathcal{H}} \max_{i \in \{1,\dots,M\}} w_i s_i^a < \max_{i \in \{1,\dots,M\}} w_i s_i^b,$$

$$\sum_{i=1}^{M} w_i = 1,$$

$$\bigforall_{i \in \{1,\dots,M\}} w_i \geq 0.$$

## Compatible model instances



# Preference modeling in IEMO/I

## Set of constraints

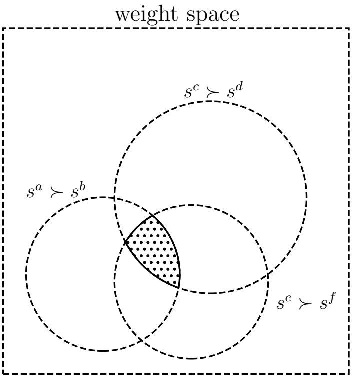
The decision examples can be used to constrain the weight space:

$$\forall_{s^a \succ s^b \in \mathcal{H}} \max_{i \in \{1,\ldots,M\}} w_i s_i^a < \max_{i \in \{1,\ldots,M\}} w_i s_i^b,$$

$$\sum_{i=1}^{M} w_i = 1,$$

$$\forall_{i \in \{1,\ldots,M\}} w_i \geq 0.$$

## Compatible model instances



# Preference modeling in IEMO/I

how different methods exploit a set of compatible model instances?

## Representative model instance

Some methods select only one representative model instance, using some selection policy. For instance, they select the most discriminative model instance:

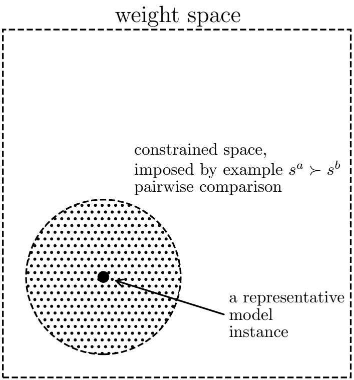
**maximize** $\epsilon$

**subject to:**

$$\forall_{s^a \succ s^b \in \mathcal{H}} \max_{i \in \{1,\ldots,M\}} w_i s_i^a + \epsilon < \max_{i \in \{1,\ldots,M\}} w_i s_i^b,$$

$$\sum_{i=1}^{M} w_i = 1,$$

$$\forall_{i \in \{1,\ldots,M\}} w_i \geq 0.$$

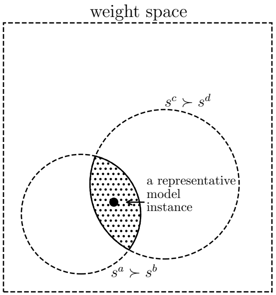


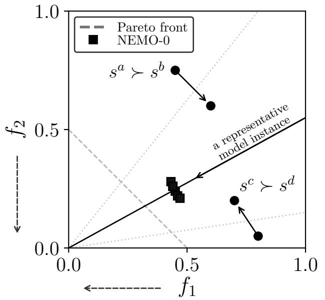
# Preference modeling in IEMO/I

how different methods exploit a set of compatible model instances?

## Example method: NEMO-0[1]

NEMO-0 sorts solutions according to (**primary sorting criterion**) non-dominated sorting and (**secondary sorting criterion**) selected representative model instance.



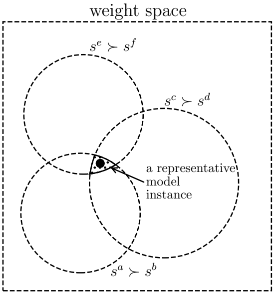

[1] J. Branke, S. Greco, R. Słowiński, and P. Zielniewicz, "Learning valuefunctions in interactive evolutionary multiobjective optimization," IEEE Transactions on Evolutionary Computation, vol. 19, no. 1, pp. 88-102, 2015.

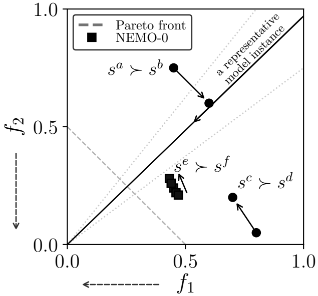
# Preference modeling in IEMO/I

how different methods exploit a set of compatible model instances?

## Example method: NEMO-0[1]

NEMO-0 sorts solutions according to (**primary sorting criterion**) non-dominated sorting and (**secondary sorting criterion**) selected representative model instance.




[1] J. Branke, S. Greco, R. Słowiński, and P. Zielniewicz, "Learning valuefunctions in interactive evolutionary multiobjective optimization," IEEE Transactions on Evolutionary Computation, vol. 19, no. 1, pp. 88-102, 2015.

# Preference modeling in IEMO/I

how different methods exploit a set of compatible model instances?

## Robustness preoccupation

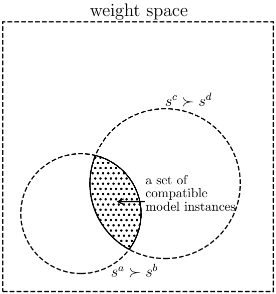
Some methods concern a whole set of compatible model instances. In this regard, they are prudent since they do not neglect any compatible model instance. Furthermore, they approximate a set of Pareto optimal solutions being potentially the most relevant (optimal) to the DM.

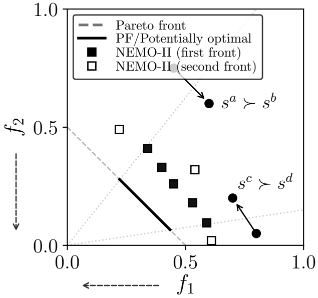
# Preference modeling in IEMO/I

how different methods exploit a set of compatible model instances?

## Example method: NEMO-II[1]

NEMO-II partitions a population into (**primary sorting criterion**) fronts of potential optimality and (**secondary sorting criterion**) sorts according to crowding-distance.



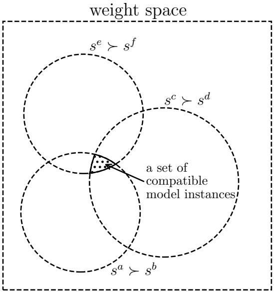

[1] J. Branke, S. Greco, R. Słowiński, and P. Zielniewicz, "Learning valuefunctions in interactive evolutionary multiobjective optimization," IEEE Transactions on Evolutionary Computation, vol. 19, no. 1, pp. 88-102, 2015.

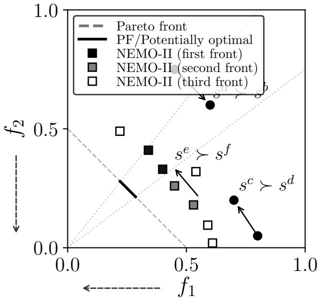
# Preference modeling in IEMO/I

how different methods exploit a set of compatible model instances?

## Example method: NEMO-II[1]

NEMO-II partitions a population into (**primary sorting criterion**) fronts of potential optimality and (**secondary sorting criterion**) sorts according to crowding-distance.




[1] J. Branke, S. Greco, R. Słowiński, and P. Zielniewicz, "Learning valuefunctions in interactive evolutionary multiobjective optimization," IEEE Transactions on Evolutionary Computation, vol. 19, no. 1, pp. 88-102, 2015.

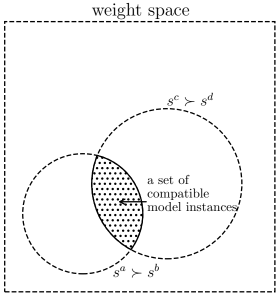
# Preference modeling in IEMO/I

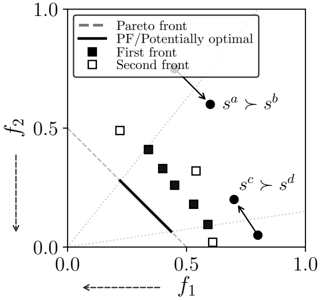
## Comparison of IEMO/I and NEMO methods

| | NEMO-0 | NEMO-II | IEMO/I |
|---|---|---|---|
| Interactive | Yes ✓ | | |
| Preference information | Pairwise comparisons ✓ | | |
| Preference model | Additive value function | | Chebyshef function (CF) |
| Robustness preoccupation | No ✗ | Yes ✓ | Yes ✓ |
| Evolutionary base | Dominance-based ✗ | | Indicator-based ✓ |

# Comparison of IEMO/I and NEMO-II

| | NEMO-II | IEMO/I |
|---|---|---|
| Evolutionary scheme | generational | steady-state |
| Secondary sorting criterion | crowding-distance | (fast) hypervolume [2] |
| Verification of potential optimallity | solving LP | Monte Carlo simulation |




[2] J. Bader and E. Zitzler. 2011. "HypE: An Algorithm for Fast Hypervolume-based Many-objective Optimization". Evolutionary Computation 19, 1 (2011), 45-76.

# Experimental setting

## Evaluated methods

| *A posteriori methods* | NSGA-II | HypE |
|---|---|---|
| Non-robust methods | NEMO-0 | ECC-MRW [3] |
| Robust methods | NEMO-II | IEMO/I |

## Benchmark problems

- WFG test suite
- The number of objectives $M$ ranged from 2 to 5

## Evolutionary setting

- The population size was made dependent on $M$
- The number of generations was set according to the difficulty of underlying problem
- To generate offspring, we used SBX and PM operators

[3] M. Kadziński, M. K. Tomczyk, and R. Słowiński. 2018. "Interactive Cone Contraction for Evolutionary Mutliple Objective Optimization. In Advances in Data Analysis with Computational Intelligence Methods: Dedicated to Professor Jacek Żurada, A. E. Gawęda, J. Kacprzyk, L. Rutkowski, and G. G. Yen (Eds.). Springer International Publishing, Cham, 293-309."

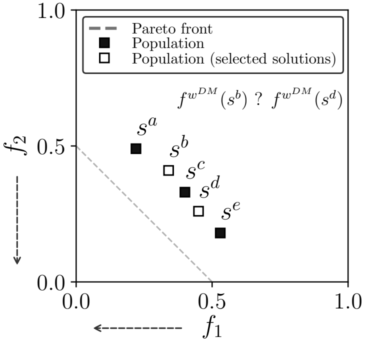
## Simulating DM's answers

▷ We modeled the DM's value system either with:

- a Chebyshef function ($f^{w^{DM}}_{CF}$) or
- a Weighted Sum ($f^{w^{DM}}_{WS}$)

▷ The preference elicitation was performed 10 times at regular intervals.

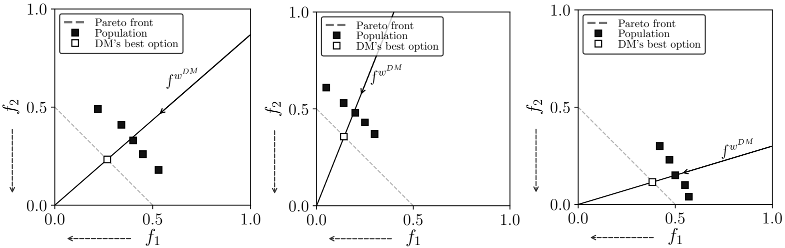
▷ During preference elicitation, two non-dominated solutions were selected from the current population and compared as imposed by the underlying preference function of the artificial DM



# Experimental setting

## Evaluation strategy

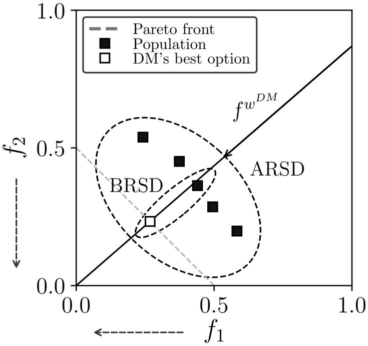
▷ Each method was run 100 times (for each scenario, i.e., benchmark problem, number of objectives, etc.), each time interacting with a different artificial DM



# Experimental setting

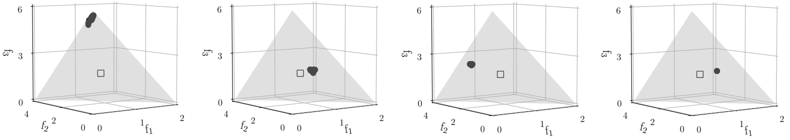
## Evaluation strategy

▷ To assess the performance of the method, we computed relative score differences between generated solutions and the best option to the DM.



## Comparison of robust and non-robust method

Population constructed by **ECC-MRW** applied to WFG3 with $M = 3$. The DM's value system was modelled with $f_{CF}^{w^{DM}} = [1/3, 1/3, 1/3]$.



(a) $\boldsymbol{generation} = \mathbf{50}$ (after 1 interaction)

(b) $\boldsymbol{generation} = \mathbf{150}$ (after 3 interactions)

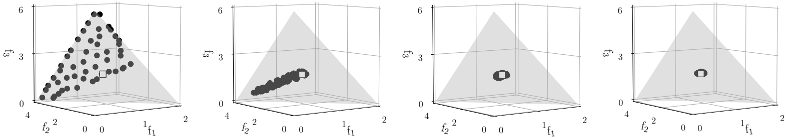
(c) $\boldsymbol{generation} = \mathbf{250}$ (after 5 interactions)

(d) $\boldsymbol{generation} = \mathbf{500}$ (after 10 interactions)

ECC-MRW (non-robust) explored different regions in objective space, instead of converging towards the preferred region...

## Comparison of robust and non-robust method

Population constructed by **IEMO/I** applied to WFG3 with $M = 3$. The DM's value system was modelled with $f_{CF}^{w^{DM}} = [1/3, 1/3, 1/3]$.




(a) $\boldsymbol{generation} = \mathbf{50}$ (after 1 interaction)

(b) $\boldsymbol{generation} = \mathbf{150}$ (after 3 interactions)

(c) $\boldsymbol{generation} = \mathbf{250}$ (after 5 interactions)

(d) $\boldsymbol{generation} = \mathbf{500}$ (after 10 interactions)

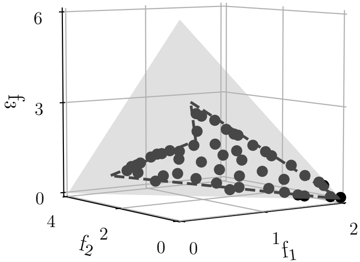
...while IEMO/I (robust) progressively constrained search space, constructing highly preferred solutions

# Experimental evaluation

## Verification of performance o IEMO/I

We assumed that the DM would judge:

- ▷ $[1.0, 0.0, 0.0] \succ [0.0, 2.0, 3.0]$ in the $1^{st}$ generation;
- ▷ $[0.0, 2.0, 0.0] \succ [0.0, 0.0, 6.0]$ in the $151^{th}$ generation;
- ▷ $[0.0, 0.0, 3.0] \succ [0.0, 4.0, 0.0]$ in the $251^{th}$ generation;

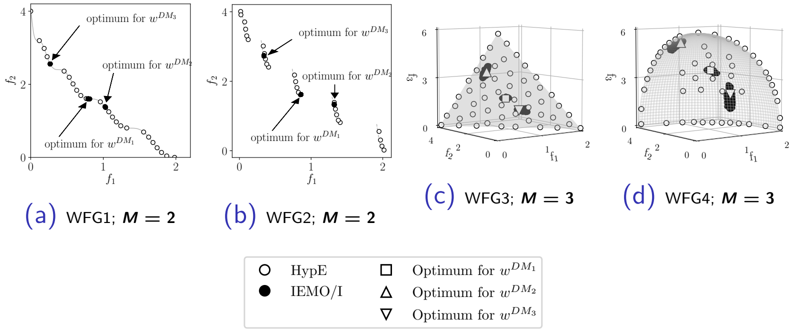



It constrains the weight space in the following way (according to the Chebyshev function): $[w_2 > w_1 \vee w_3 > w_1] \wedge [w_3 > 0.5w_2] \wedge [w_2 > 0.5w_3]$.

IEMO/I constructed a fine approximation of the region in objective space imposed by the constraints on the weight space

## Comparison with HypE

Populations constructed by IEMO/I and HypE applied to different benchmark problems.



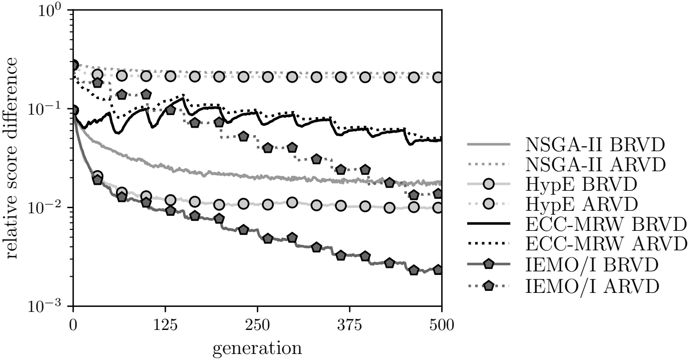

(a) WFG1; $M = 2$ (b) WFG2; $M = 2$ (c) WFG3; $M = 3$ (d) WFG4; $M = 3$

IEMO/I was able to find DM's preferred option, outperforming in this regard *a posteriori* method – HypE

# Experimental evaluation

## Comparative study: Robust vs. non-robust vs. *a posteriori* methods

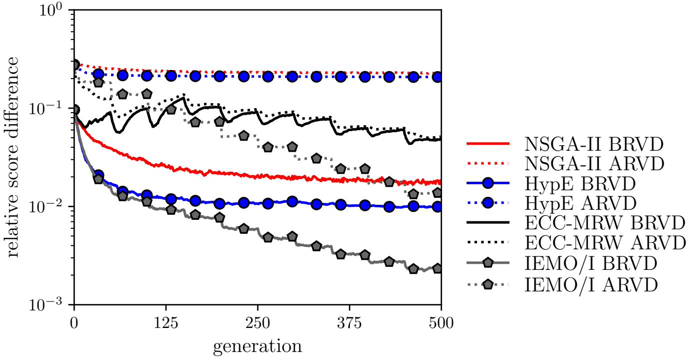
Average BRSD and ARSD throughout 500 generations for different algorithms applied to WFG3 with $M = 3$. **The artificial DM was modeled with a Chebyshev function.**



# Experimental evaluation

## Comparative study: Robust vs. non-robust vs. *a posteriori* methods

Average BRSD and ARSD throughout 500 generations for different algorithms applied to WFG3 with $M = 3$. **The artificial DM was modeled with a Chebyshev function.**

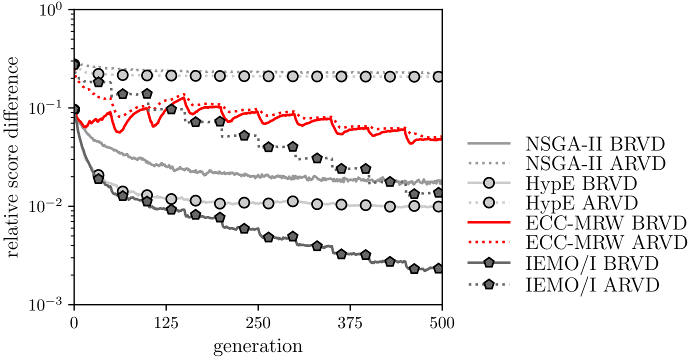



NSGA-II and HypE attained poor ARSD. When compared to IEMO/I, BRSD attained by these methods is also not satisfactory.

# Experimental evaluation

## Comparative study: Robust vs. non-robust vs. *a posteriori* methods

Average BRSD and ARSD throughout 500 generations for different algorithms applied to WFG3 with $M = 3$. **The artificial DM was modeled with a Chebyshev function.**

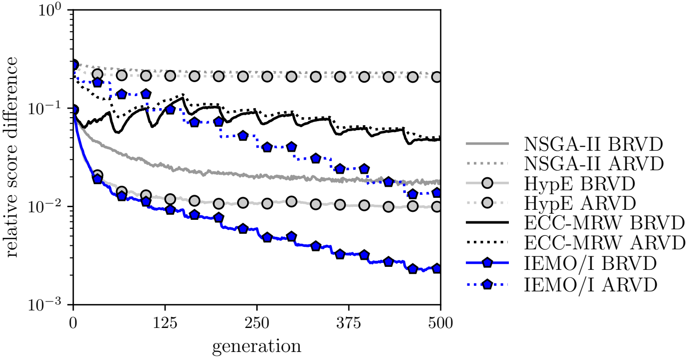



ECC-MRW (non-robust method) could not focus the search on DM's preferred region in the objective space.

# Experimental evaluation

## Comparative study: Robust vs. non-robust vs. *a posteriori* methods

Average BRSD and ARSD throughout 500 generations for different algorithms applied to WFG3 with $M = 3$. **The artificial DM was modeled with a Chebyshev function.**

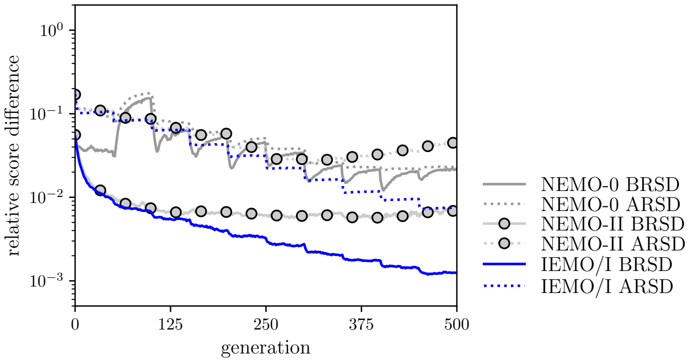



IEMO/I (robust method) attained the best BRSD and ARSD, outperforming in this regard the non-robust ECC-MRW and *a posteriori* methods.

# Experimental evaluation

## Comparative study: NEMO methods vs. IEMO/I

Average ARSD and BRSD throughout 500 generations for different algorithms applied to WFG4 with $M = 3$. **The artificial DM was modeled with a Chebyshev function.**

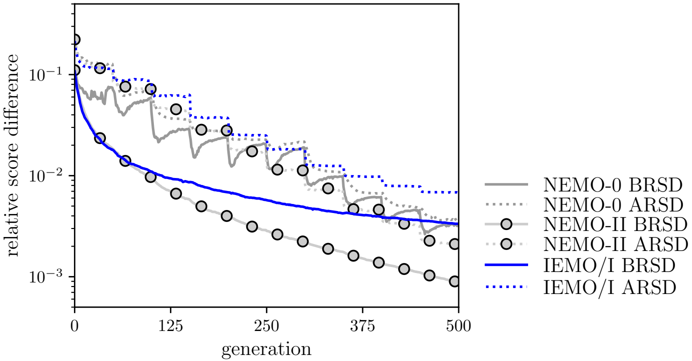



IEMO/I outperformed NEMO methods, which are based on the diminance principle and an additive value function.

# Experimental evaluation

## Comparative study: NEMO methods vs. IEMO/I

Average ARSD and BRSD throughout 500 generations for different algorithms applied to **cWFG4** with $M = 3$. **The artificial DM was modeled with a weighted sum.**




IEMO/I was outperformed by NEMO-II due to incorporating a preference model being inconsistent with the DM's decision policy.

# Conclusions

## We propose a novel preference-based EMOA, IEMO/I, which

- ▷ is interactive ✓
- ▷ asks the DM to holistically compare some pairs of solutions ✓
- ▷ employs a simple preference model in a form of a Chebyshef function ✓
- ▷ implements a robustness concern ✓
- ▷ uses an indicator-based evolutionary framework ✓

## The conducted experiments prove that:

- ▷ IEMO/I can find a highly preferred region of the PF with a limited number of interactions,
- ▷ IEMO/I outperforms some selected state-of-the-art methods which also incorporate pairwise comparisons, but select only one compatible model instance to promote solutions during evolutionary search
- ▷ the performance of an interactive EMOA can be improved, when the assumed preference model aligns with the DM's value system

# Conclusions

## We propose a novel preference-based EMOA, IEMO/I, which

- ▷ is interactive ✓
- ▷ asks the DM to holistically compare some pairs of solutions ✓
- ▷ employs a simple preference model in a form of a Chebyshef function ✓
- ▷ implements a robustness concern ✓
- ▷ uses an indicator-based evolutionary framework ✓

## Related papers


- ▷ **Decomposition-based method:** M. Tomczyk, M. Kadziński, "Decomposition-based interactive evolutionary algorithm for multiple objective optimization," IEEE Transactions on Evolutionary Computation, 2019, (accepted for publication), Doi:10.1109/TEVC.2019.2915767.
- ▷ **Stochastic approach:** M. Tomczyk, M. Kadziński, "EMOSOR: Evolutionary multiple objective optimization guided by interactive stochastic ordinal regression," Computers & Operations Research 108, 134-154, 2019.