Decomposition-based interactive evolutionary optimization guided by multiple search directions compatible with indirect preference information

mgr inż. Michał Tomczyk, dr hab. inż. Miłosz Kadziński

April 24, 2018

Table of contents

Introduction to MOO

- A very simple single-objective optimization problem:

$$\min \quad f(x) = 10 \sum_{i=1,\ldots,n} |x_i - 0.7| \qquad x_i \in [0,1]$$

Introduction to MOO

- A very simple single-objective optimization problem:

$$\min \quad f(x) = 10 \sum_{i=1,\ldots,n} |x_i - 0.7| \qquad x_i \in [0,1]$$

Solution $s = f(x)$ is optimal when: $\forall_{i=1,\ldots,n} x_i = 0.7$.

Introduction to MOO

- A very simple single-objective optimization problem:

$$\min \quad f(x) = 10 \sum_{i=1,\ldots,n} |x_i - 0.7| \qquad x_i \in [0,1]$$

  Solution $s = f(x)$ is optimal when: $\forall_{i=1,\ldots,n} x_i = 0.7$.

- A very simple **multi**-objective optimization problem:

$$\begin{aligned} \min \quad & \boldsymbol{f_1(x)} = 1 - x \\ \min \quad & \boldsymbol{f_2(x)} = x \qquad x \in [0,1] \end{aligned}$$

Introduction to MOO

- A very simple single-objective optimization problem:

$$\min \quad f(x) = 10 \sum_{i=1,\dots,n} |x_i - 0.7| \qquad x_i \in [0,1]$$

  Solution $s = f(x)$ is optimal when: $\forall_{i=1,\dots,n} x_i = 0.7$.

- A very simple **multi**-objective optimization problem:

$$\begin{aligned} \min \quad & \boldsymbol{f_1(x)} = 1 - x \\ \min \quad & \boldsymbol{f_2(x)} = x \qquad x \in [0,1] \end{aligned}$$

  Solution $s = [f_1(x), f_2(x)]$ is **Pareto** optimal for each $x \in [0,1]$.

Introduction to MOO

- A very simple single-objective optimization problem:

$$\min \quad f(x) = 10 \sum_{i=1,\ldots,n} |x_i - 0.7| \qquad x_i \in [0,1]$$

  Solution $s = f(x)$ is optimal when: $\forall_{i=1,\ldots,n} x_i = 0.7$.

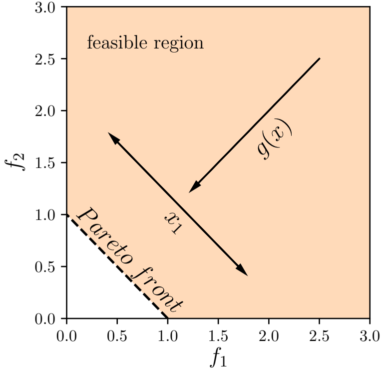
- A very simple **multi**-objective optimization problem:

$$\min \quad \boldsymbol{f_1(x)} = (1 - x_1)(1 + \boldsymbol{g(x)})$$
$$\min \quad \boldsymbol{f_2(x)} = (x_1)(1 + \boldsymbol{g(x)})$$
$$g(x) = 10 \sum_{i=\mathbf{2},\ldots,n} |x_i - 0.7| \qquad x_i \in [0,1]$$

  Solution $s = [f_1(x), f_2(x)]$ is **Pareto** optimal when: $\forall_{i=\mathbf{2},\ldots,N} x_i = 0.7$ and $x_1 \in [0,1]$.

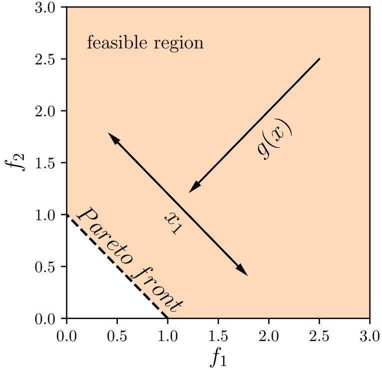
Introduction to MOO

The goal of a Multiple objective optimization problem is to find the best trade-off solutions (Pareto-front).



Introduction to MOO

The goal of a Multiple objective optimization problem is to find ~~the best trade-off solutions (Pareto-front)~~ **a good approximation of Pareto front**.



Reminder on Evolutionary Algorithms

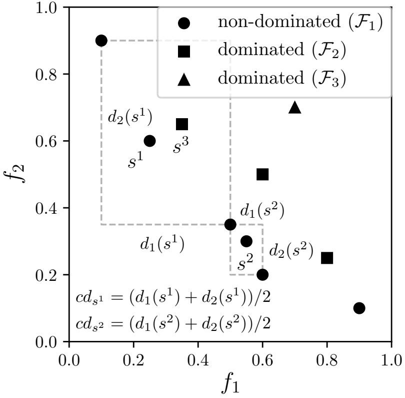
$$\min \quad f(x) = 10 \sum_{i=1,\ldots,n} |x_i - 0.7| \qquad x_i \in [0, 1]$$

Basic scheme of elitist and population-based evolutionary algorithm:

1. Create & evaluate initial population of size N
2. Choose parent solutions
3. Generate & evaluate offspring population of size N
4. Combine current population with offspring
5. Remove N worst solutions
6. Go to (2)

Dominance-based methods

e.g., NSGA-II

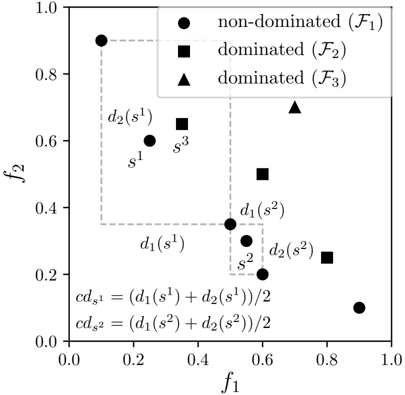


- NSGA-II partitions a population of solutions into nondominated fronts $\mathcal{F}$
- Within each front, the solutions are ranked according to the crowding-distance cd_{s^j}

Partial order over solutions: $s^j \succ_{NSGA-II} s^k \Leftrightarrow (ndf_{s^j} < ndf_{s^k}) \vee (ndf_{s^j} = ndf_{s^k} \wedge cd_{s^j} > cd_{s^k})$

Dominance-based methods

e.g., NSGA-II



- NSGA-II partitions a population of solutions into nondominated fronts $\mathcal{F}$
- Within each front, the solutions are ranked according to the crowding-distance cd_{s^j}

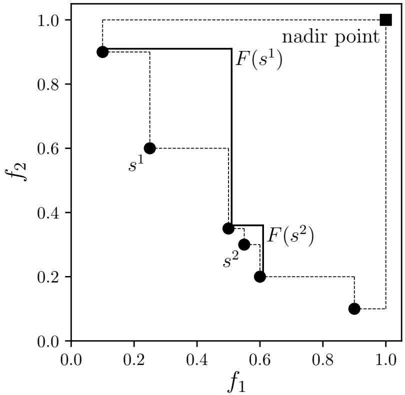
Partial order over solutions: $s^j \succ_{NSGA-II} s^k \Leftrightarrow (ndf_{s^j} < ndf_{s^k}) \vee (ndf_{s^j} = ndf_{s^k} \wedge cd_{s^j} > cd_{s^k})$

Pros & Cons

These algorithms are not suitable for many-objective optimization :(

Indicator-based methods

e.g., IBEA

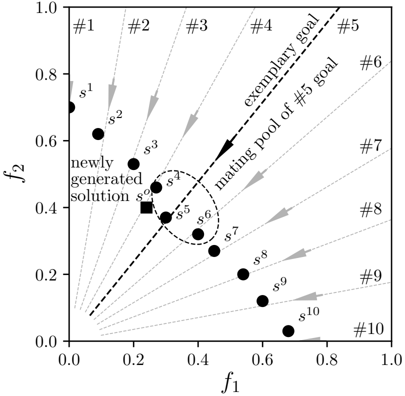


- Indicator is used to asses solution's contribution to the population
- Indicator takes into account diversity and convergence
- The solution that contributes the least should be replaced
- **steady-state (!)**

Pros & Cons
(+) suitable for many-objective optimization
(+) can approximate PFs with complex shapes
(-) computational complexity is poor

Decomposition-based methods

e.g., MOEA/D



- The MOO problem is formulated in terms of several sub-tasks
- Sub-tasks are solved individually though in an interrelated manner
- steady-state (usually)

Pros & Cons

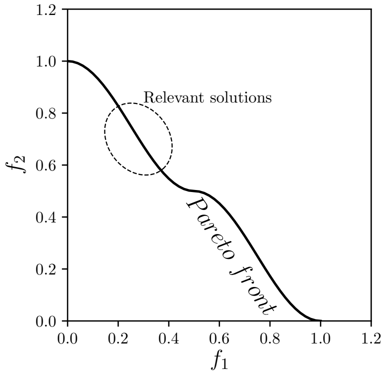
(+) suitable for many-objective optimization
(+) these methods are very effective
(-) convergence strongly depends on selected sub-tasks.

Visualization

NSGA-II vs. MOEA/D

Preference-based EMOAs: Motivation

Incorporation of the DM's preferences into EMOA is oriented toward construction and/or selection of the DM's most preferred solution.



Motivation:

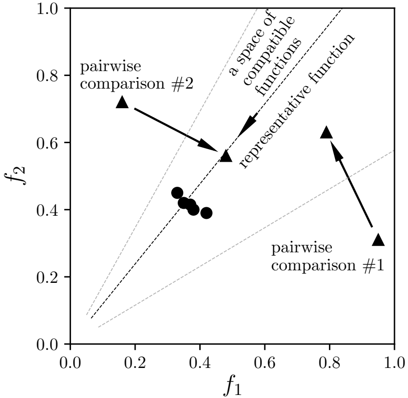
- the preference information can be used to constraint the search space, thereby reducing the complexity of the problem
- the preference information can be used to impose an additional selection pressure, driving population of solutions toward the most relevant region of PF

Attributes of preference-based EMOAs

- **Evolutionary base**: e.g., dominance, indicator, or decomposition
- **Interactiveness**: yes or no (*a priori* and a posteriori approach)
- **Preference model**: e.g., additive value function, L_α-*norm*, Choquet integral, etc.
- **Indirect preference information**: yes or no (parameters of the model)
- **Preference information**: weights, pairwise comparisons, intensities of preferences, etc.
- **Robustness analysis**: yes or no

NEMO-0

NEMO-0 is based on NSGA-II. It replaces crowding-distance with a representative additive value function that is derived from a space of compatible functions.



(+):

- interactive
- indirect preference information: pairwise comparisons

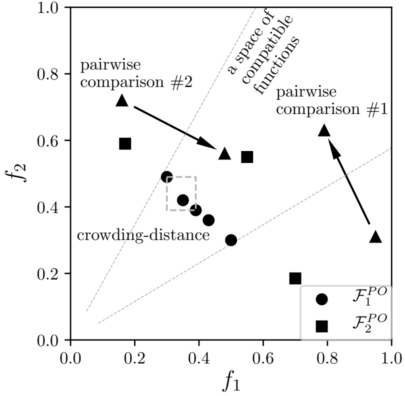
(-):

- no robustness analysis
- dominance-based

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

NEMO-II

NEMO-II is also based on NSGA-II. It replaces fronts of non-dominated solutions with fronts of potentially optimal solutions (NEMO-II uses additive value function to model the DM's judgement policy).



(+):

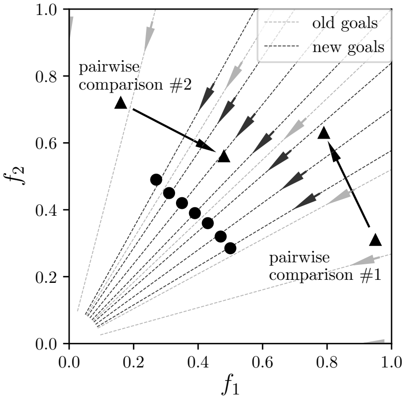
- interactive
- indirect preference information: pairwise comparisons
- robustness analysis

(-):

- dominance-based

IEMO/D

IEMO/D is based on MOEA/D. It explicitly generates a subset of uniformly distributed compatible L_α-*norms* by means of the Monte Carlo simulation. Similarly to MOEA/D, IEMO/D uses such functions to define the search directions in the evolutionary process.



(+):

- interactive
- indirect preference information:
  - pairwise comparisons
  - intensities of preferences
  - pre-order of a subset of solutions
  - best solution form a subset of solutions
- robustness analysis
- decomposition-based

Definition:

L_α-norm as a preference model

$$L_\alpha^w(s^j, z) = \begin{cases} \left(\sum_{i=1,\dots,M} \left(w_i|s_i^j - z_i|\right)^\alpha\right)^{1/\alpha} & \text{for } \alpha < \infty \\ max_{i=1,\dots,M}\{|w_i(s_i^j - z_i)|\} & \text{for } \alpha = \infty \end{cases}$$

$L_\alpha^w(s^j, s^k, z) = L_\alpha^w(s^j, z) - L_\alpha^w(s^k, z)$

$\mathcal{H}$ – a set of holistic indirect judgements

$s^j \succ s^k$ – preference of s^j over s^k

$s^j \succ^{Cl_l} s^k$ – preference intensity

L_α-norm as a preference model

Definition:

$$L^w_\alpha(s^j, z) = \begin{cases} \left(\sum_{i=1,\dots,M} \left(w_i|s^j_i - z_i|\right)^\alpha\right)^{1/\alpha} & \text{for } \alpha < \infty \\ max_{i=1,\dots,M}\{|w_i(s^j_i - z_i)|\} & \text{for } \alpha = \infty \end{cases}$$

$L^w_\alpha(s^j, s^k, z) = L^w_\alpha(s^j, z) - L^w_\alpha(s^k, z)$

$\mathcal{H}$ – a set of holistic indirect judgements

$s^j \succ s^k$ – preference of s^j over s^k

$s^j \succ^{Cl_l} s^k$ – preference intensity

Constraints:

normalization: $$\sum_{i=1,\dots,M} w_i = 1$$

Definition:

L_α-norm as a preference model

$$L^w_\alpha(s^j, z) = \begin{cases} \left(\sum_{i=1,\dots,M} \left(w_i|s^j_i - z_i|\right)^\alpha\right)^{1/\alpha} & \text{for } \alpha < \infty \\ max_{i=1,\dots,M}\{|w_i(s^j_i - z_i)|\} & \text{for } \alpha = \infty \end{cases}$$

$L^w_\alpha(s^j, s^k, z) = L^w_\alpha(s^j, z) - L^w_\alpha(s^k, z)$

$\mathcal{H}$ – a set of holistic indirect judgements

$s^j \succ s^k$ – preference of s^j over s^k

$s^j \succ^{Cl_l} s^k$ – preference intensity

Constraints:

normalization: $\sum_{i=1,\dots,M} w_i = 1$

pair-wise comparisons: $\forall_{(s^j \succ s^k) \in \mathcal{H}} L^w_\alpha(s^j, s^k, z) < 0$

rankings of 3 solutions: $\forall_{(s^j \succ s^k \succ s^l) \in \mathcal{H}} (L^w_\alpha(s^j, s^k, z) < 0) \wedge (L^w_\alpha(s^k, s^l, z) < 0)$

picking the best out of 3 solutions: $\forall_{((s^j \succ s^k) \wedge (s^j \succ s^l)) \in \mathcal{H}} (L^w_\alpha(s^j, s^k, z) < 0) \wedge (L^w_\alpha(s^j, s^l, z) < 0)$

preference intensity: $\forall_{(s^j \succ^{Cl_l} s^k) \in \mathcal{H}} (t_l \geq L^w_\alpha(s^k, s^j, z) > t_{l-1})$

Sampling compatible weights of L_α-norms

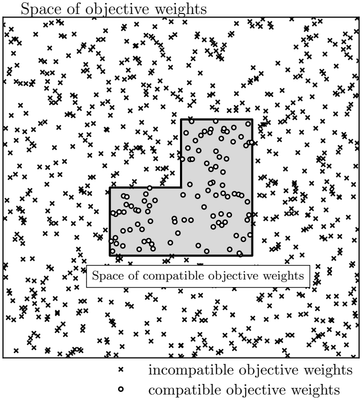
Monte Carlo method: rejection sampling

Parameters:

- Q – number of compatible weight vectors IEMO/D is expected to sample
- T – limit of random samples IEMO/D is allowed to generate ($T \gg L$)
- Z – number of random samples that were generated by IEMO/D
- $\mathcal{L}_\alpha^S$ – set of compatible weight vectors that were found by IEMO/D

Possible scenarios:

- $|\mathcal{L}_\alpha^S| = Q$ and $Z \leq T$: return $|\mathcal{L}^S|$ ✓
- $0 < |\mathcal{L}_\alpha^S| < Q$ and $Z = T$: return $|\mathcal{L}^S|$
- $|\mathcal{L}_\alpha^S| = 0$ and $Z = T$: inconsistency!



Sampling compatible weights of L_α-norms

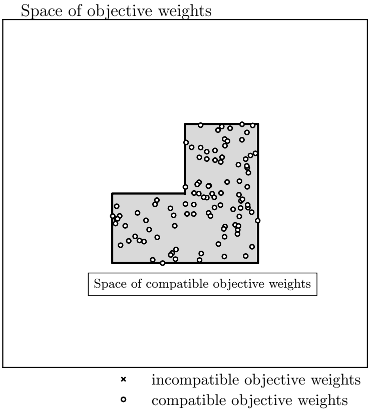
Rejection sampling is not efficient when:

- many objectives are involved
- space of compatible weight vectors is small (e.g., when many holistic judgements are provided)
- objective space is continuous and the solutions of the problem change over time

Sampling compatible weights of L_α-norms

Monte Carlo method: Iteratively Constrained Rejection Sampling (ICRS)

The existing sampling methods, such as Hit-and-Run, are efficient when exploiting the parameter space constrained with a set of linear inequalities.



T. Tervonen, G. van Valkenhoef, N. Basturk, and D. Postmus, "Hit-and- Run enables efficient weight generation for simulation-based multiple criteria decision analysis," European Journal of Operational Research, vol. 224, no. 3, pp. 552-559, 2013.

Sampling compatible weights of L_α-norms

Monte Carlo method: Iteratively Constrained Rejection Sampling (ICRS)

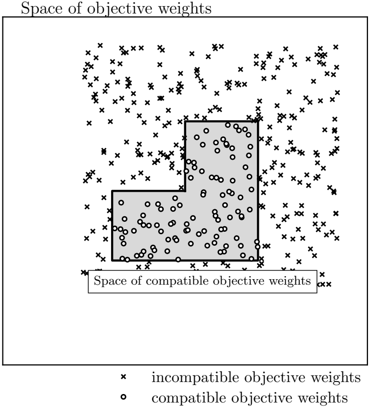
The existing sampling methods, such as Hit-and-Run, are efficient when exploiting the parameter space constrained with a set of linear inequalities.

IEMO/D uses Hit-and-Run to generate candidate samples. The space is constrained with the following set of linear (in)equalities:

$$\text{normalization: } \sum_{i=1,\dots,M} w_i = 1$$

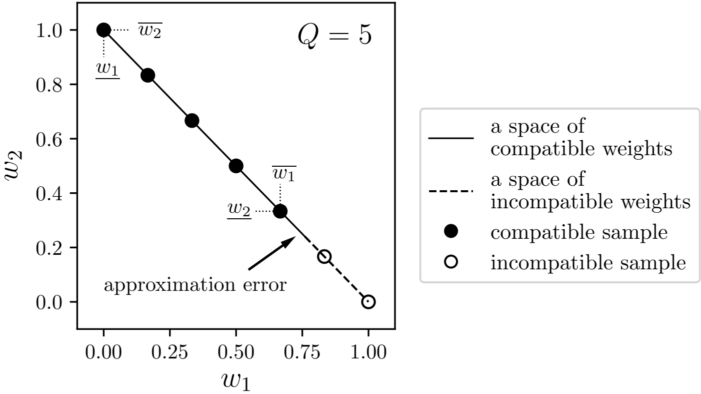
$$\text{bounds for weights: } \forall_{i=1,\dots,M} \quad \underline{w_i} \leq w_i \leq \overline{w_i}$$

IEMO/D systematically approximates the bounds of candidate weights, hence reducing the space of weight vectors to be exploited and decreasing the rejection rate.



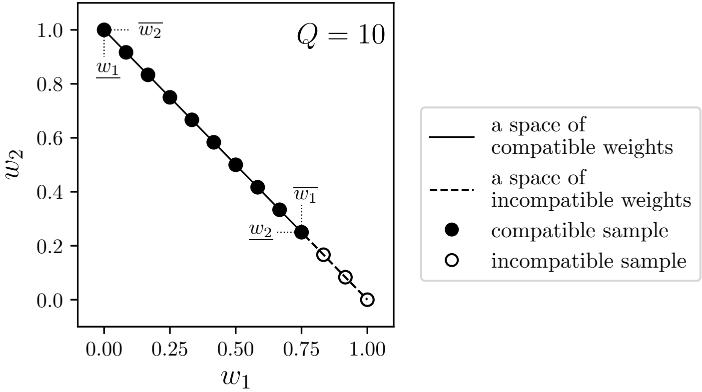
Iteratively constrained rejection sampling

IEMO/D estimates $\underline{w}$ and $\overline{w}$ based on the set of compatible sampled weights $\mathcal{L}_{\alpha}^{S}$



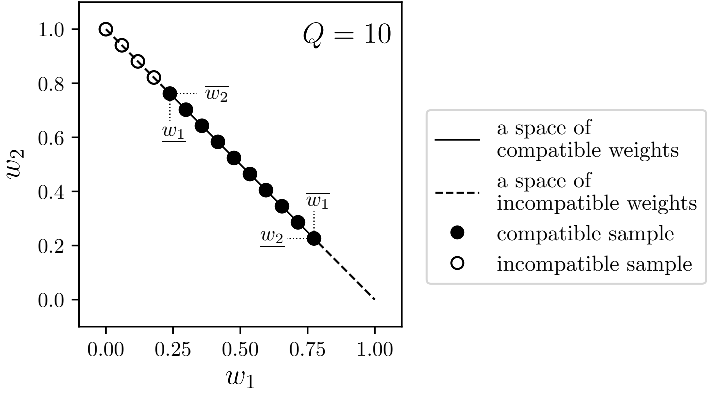
Iteratively constrained rejection sampling

with upsampling

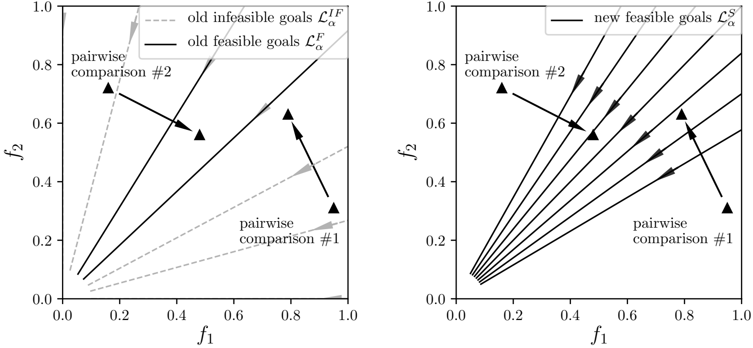


Iteratively constrained rejection sampling

with upsampling (ICRSU)



Prioritizing search directions for the evolutionary search



Prioritizing new search directions: $\mathcal{L}_\alpha^S \succ \mathcal{L}_\alpha^F \succ \mathcal{L}_\alpha^{IF}$
Prioritizing old search directions: $\mathcal{L}_\alpha^F \succ \mathcal{L}_\alpha^S \succ \mathcal{L}_\alpha^{IF}$

They are respected when constructing an updated set of goals for the next generation

A general scheme of the IEMO/D algorithm

Algorithm 1 A general scheme of the IEMO/D algorithm

Input: G – a pre-defined number of generations; N – a size of the population; EI – an elicitation interval

Output: P – population

Generation $\leftarrow 0$

$\mathcal{H} \leftarrow \emptyset$ (or provide preferences *a priori*)

$\mathcal{L}_\alpha^M \leftarrow \emptyset$

$\underline{w}, \overline{w} \leftarrow \overrightarrow{0}, \overrightarrow{1}$

$\mathcal{L}_\alpha^S, \underline{w}^{constr}, \overline{w}^{constr} \leftarrow SampleCompatibleLNorms(\mathcal{H}, Q, T, \underline{w}, \overline{w})$

$\underline{w}, \overline{w} \leftarrow \underline{w}^{constr}, \overline{w}^{constr}$

$\mathcal{L}_\alpha^R \leftarrow PrioritizeSearchDirections(\emptyset, \mathcal{L}_\alpha^S, \mathcal{H})$

$\mathcal{L}_\alpha^M \leftarrow \mathcal{L}_\alpha^R$

$P \leftarrow$ create initial population of size N

MOEA/D: construct neighbourhood w.r.t. P, $\mathcal{L}_\alpha^M$

while Generation $< G$ **do**

  if Time to ask the DM (Generation$\%EI == 0$) **then**

    PI $\leftarrow$ elicit new piece of preference information

    $\mathcal{H} = \mathcal{H} \cup \{\text{PI}\}$

    $\mathcal{L}_\alpha^S, \underline{w}^{constr}, \overline{w}^{constr} \leftarrow SampleFeasibleLNorms(\mathcal{H}, Q, T, \underline{w}, \overline{w})$

    $\underline{w}, \overline{w} \leftarrow \underline{w}^{constr}, \overline{w}^{constr}$

    $\mathcal{L}_\alpha^R \leftarrow PioritizeSearchDirections(\mathcal{L}_\alpha^M, \mathcal{L}_\alpha^S, \mathcal{H})$

    $\mathcal{L}_\alpha^M \leftarrow \mathcal{L}_\alpha^R$

    MOEA/D: construct neighbourhood w.r.t. P, $\mathcal{L}_\alpha^M$

  Generation++

  for L_α^w in $\mathcal{L}_\alpha^M$ **do**

    MOEA/D: update population P w.r.t. $\mathcal{N}(L_\alpha^w)$

Return P

Visualization

NEMO-0 vs. NEMO-II vs. IEMO/D

Experimental evaluation

experimental setting

- The methods were evaluated on the DTLZ1-4 and WFG1-2 benchmarks that involved from $M = 2$ to 5 objectives.
- The number of generations G was set to 300 except for DTLZ3 ($G = 900$) and WFG1 ($G = 1500$), which are more computationally challenging.
- To select the parent solutions, we used a tournament selection of size 5 for NSGA-II and NEMO methods. We used a random selection of a pair of solutions for MOEA/D and IEMO/D.
- To generate offspring, we used a simulated binary crossover (probability of 1.0) with a distribution index of 10.0 and a polynomial mutation with a distribution index of 10.0 and probability of $1/dv$, where dv is a number of decision variables.

Whenever different is not explicitly stated, we assumed that:

- the DM was asked to compare pairwise solutions from the current population;
- the number of preference elicitation iterations was limited to a realistic level of 12, and hence $EI = G/12$;
- α in the preference model used by IEMO/D as well as α^{DM} employed for simulating a decision model $L^{w^{DM}}_{\alpha^{DM}}$ of an artificial DM were set to 5;
- z was set to a utopian point.

Experimental evaluation

experimental setting

For each test problem, we simulated 100 artificial DMs with the randomly selected weight vectors w^{DM} incorporated into $L_{\alpha^{DM}}^{w^{DM}}$. For each run with a unique artificial DM, we reported the following measures for the compared algorithms:

- BRSD: a relative score difference of the best constructed solution $s^j \in P$ to the optimal solution $s_{w^{DM}}^{opt}$ according to the DM's model: $BRSD(P, L_{\alpha^{DM}}^{w^{DM}}) =$

$$min_{s^j \in P}\{L_{\alpha^{DM}}^{w^{DM}}(s^j, s_{w^{DM}}^{opt}, z)/L_{\alpha^{DM}}^{w^{DM}}(s_{w^{DM}}^{opt}, z)\}; \quad (1)$$

- ARSD: an average relative score difference of all constructed solutions to the optimal solution $s_{w^{DM}}^{opt}$ according to the DM's model: $ARSD(P, L_{\alpha^{DM}}^{w^{DM}}) =$

$$\left(\sum_{s^j \in P} L_{\alpha^{DM}}^{w^{DM}}(s^j, s_{w^{DM}}^{opt}, z)/L_{\alpha^{DM}}^{w^{DM}}(s_{w^{DM}}^{opt}, z)\right) \Big/ |P|. \quad (2)$$

To aggregate the results from the experimental runs involving different artificial DMs, we computed the mean values of BRSD and ARSD along with the standard deviations (StD) as well as with the averaged ranks $\overline{R}$ attained by different algorithms.

Impact of evolutionary basis on evolutionary search

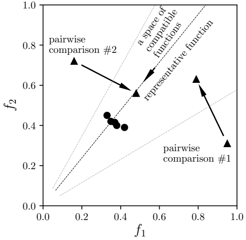


Figure: NEMO-0

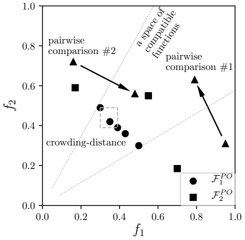


Figure: NEMO-II

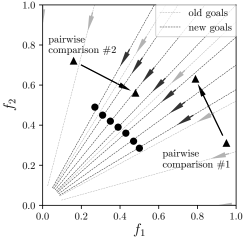


Figure: IEMO/D

Since NEMO-I and NEMO-II use a preference model in form of an additive value function U, to reliably compare the impact of the evolutionary base, we developed a counterpart of NEMO, called NEMO-L_α, which employs the L_α-norm to represent the DM's preferences.

Impact of evolutionary basis on evolutionary search

Since incorporation of the L_α-norm instead of U makes the underlying constraints non-linear, we restrict operations performed by NEMO-L_α to an explicitly defined set $\mathcal{L}_\alpha^S$ of instances of the L_α-norm sampled with *ICRSU* rather than a set of infinitely many compatible value functions delimited by the mathematical constraints.

The exploitation of $\mathcal{L}_\alpha^S$ by different variants of NEMO-L_α is performed in the following way:

- NEMO-0-L_α: a single representative (MDVF) L_α-norm is selected from $\mathcal{L}_\alpha^S$ as follows:

$$L_\alpha^w = argmax_{L_\alpha^w \in \mathcal{L}_\alpha^S} \{min\{L_\alpha^w(s^k, s^j, z) : (s^j \succ s^k) \in \mathcal{H}\}\};$$

- NEMO-II-L_α: verification of the potential optimality of solution s^j that is involved in the computation of $\mathcal{F}^{PO}$ is performed as follows:

$$\exists_{L_\alpha^w \in \mathcal{L}_\alpha^s} \forall_{s^k \in P,\ s^k \neq s^j} L(s^j, s^k, z) < 0 \rightarrow s^j \text{ is potentially optimal.}$$

Convergence plots

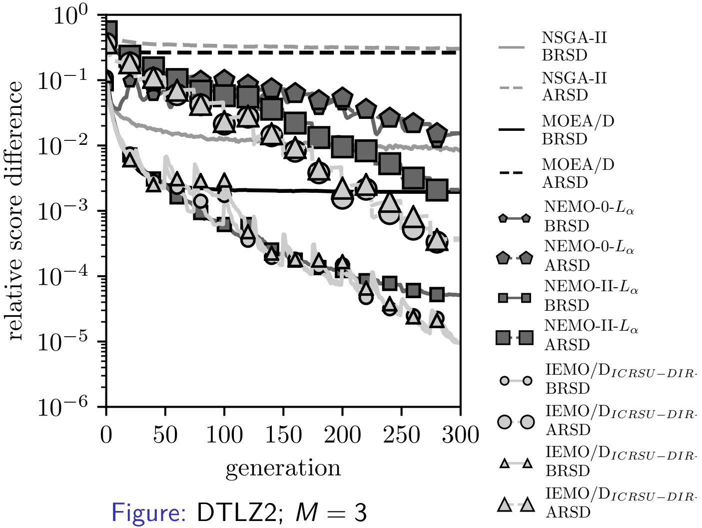


Figure: DTLZ2; $M = 3$

Convergence plots

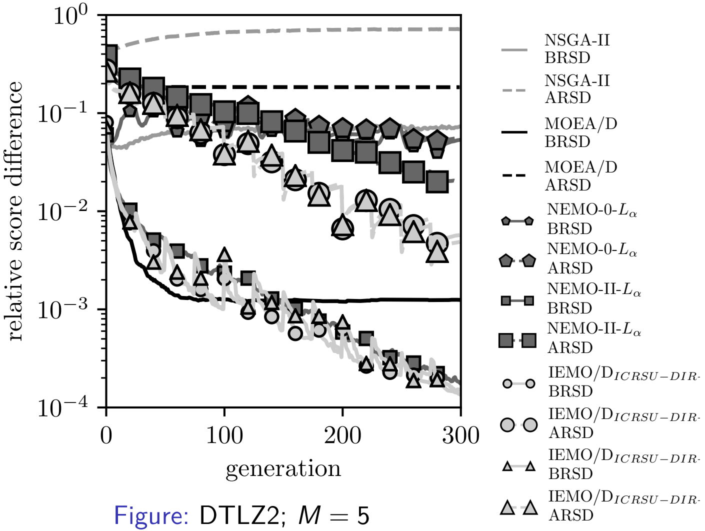


Figure: DTLZ2; $M = 5$

Numerical results: DTLZ2

Average BRSD (first row) and ARSD (second row) for the populations generated in the last iteration by six algorithms for the DTLZ2 test problem with $M = 2, 3, 4$, and 5 objectives. Average ranks $\overline{R}$ attained by the algorithms according to either BRSD or ARSD.

		NSGA-II			MOEA/D			NEMO-0-L_α			NEMO-II-L_α			IEMO/D $ICRSU-DIR-NEW$			IEMO/D $ICRSU-DIR-OLD$			ρ
	M	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	
DTLZ2	2	19.98	17.69	5.80	9.86	25.59	4.82	17.36	84.63	4.11	0.00	0.00	1.25	0.00	0.01	2.46	0.01	0.02	2.56	4
		3397.76	1695.04	5.44	3428.82	1605.07	5.56	22.57	91.76	3.37	0.22	0.80	2.22	0.03	0.16	2.08	0.04	0.10	2.33	
	3	83.79	55.32	5.65	19.44	34.71	4.45	152.48	303.45	4.69	0.51	4.10	2.28	0.10	0.19	1.98	0.09	0.28	1.95	4
		3032.94	1013.15	5.84	2634.82	938.98	5.14	154.98	307.68	3.21	21.48	65.93	2.55	3.59	8.79	2.15	3.84	7.60	2.11	
	4	481.77	321.81	5.61	21.73	32.48	3.80	415.98	618.40	4.78	56.88	137.35	3.23	0.53	0.89	1.82	0.62	1.36	1.76	4
		5874.40	1530.44	6.00	2136.56	492.12	4.97	428.44	619.57	3.32	110.37	183.83	2.62	17.92	30.29	2.09	17.75	29.96	2.00	
	5	729.81	466.71	5.72	12.49	19.33	3.85	533.14	624.33	5.05	1.79	2.76	2.88	1.51	4.09	1.81	1.36	4.18	1.69	4
		7174.19	1418.86	6.00	1836.97	356.47	4.92	561.61	638.50	3.30	203.87	219.59	3.07	57.19	106.82	1.84	48.80	64.28	1.87	

Numerical results: WFG2

Average BRSD (first row) and ARSD (second row) for the populations generated in the last iteration by six algorithms for the WFG2 test problem with $M = 2, 3, 4,$ and 5 objectives. Average ranks $\overline{R}$ attained by the algorithms according to either BRSD or ARSD.

		NSGA-II			MOEA/D			NEMO-0-L_α			NEMO-II-L_α			IEMO/D *ICRSU – DIR – NEW*			IEMO/D *ICRSU – DIR – OLD*			ρ
	M	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	
WFG2	2	1.82	3.46	4.59	3.05	6.20	4.07	1.77	4.08	2.52	1.68	3.96	2.28	2.49	5.98	3.89	1.86	3.79	3.65	2
		27.46	12.08	5.71	23.14	11.06	5.28	2.42	4.77	2.47	2.18	4.70	2.07	2.58	5.98	2.78	1.91	3.78	2.69	
	3	2.17	3.37	5.08	2.62	4.43	4.39	2.73	4.37	3.42	1.35	3.41	1.91	2.46	4.41	3.16	2.60	4.66	3.04	2
		18.54	6.53	5.88	13.06	5.17	5.02	2.85	4.44	2.69	1.49	3.38	2.15	2.47	4.41	2.69	2.63	4.64	2.57	
	4	1.43	1.61	4.23	1.72	2.58	4.32	2.78	3.84	4.28	0.47	1.39	2.14	2.10	3.09	3.00	1.82	3.02	3.03	2
		17.20	3.40	5.99	8.43	2.27	4.90	2.88	3.83	3.04	0.81	1.35	2.21	2.16	3.05	2.45	1.91	2.97	2.41	
	5	0.84	0.54	4.15	2.09	2.96	4.47	2.11	3.59	4.28	0.50	1.64	2.17	1.78	2.77	3.09	1.64	2.91	2.84	2
		15.00	3.75	5.99	6.42	2.31	4.86	2.40	3.82	2.86	1.05	1.58	2.38	1.93	2.69	2.52	1.83	2.83	2.39	

Numerical results: Averaged Ranks

Average ranks for the populations generated in the last iteration by six algorithms for all test problems with $M = 2, 3, 4,$ and 5 objectives.

		NSGA-II		MOEA/D		NEMO-0-L_α		NEMO-II-L_α		IEMO/D *ICRSU – DIR – NEW*		IEMO/D *ICRSU – DIR – OLD*	
	M	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD
Average ranks ($\overline{R}$)	2	5.48	0.45	4.13	0.75	3.43	0.66	1.90	0.74	3.09	0.56	2.97	0.51
		5.72	0.21	5.19	0.35	2.98	0.40	2.08	0.53	2.53	0.40	2.50	0.39
	3	5.64	0.36	4.26	0.40	4.33	0.49	2.34	0.58	2.21	0.50	2.23	0.42
		5.88	0.15	5.01	0.24	3.28	0.33	2.58	0.44	2.11	0.36	2.14	0.27
	4	5.47	0.66	3.96	0.47	4.61	0.34	2.95	0.73	2.03	0.49	1.99	0.51
		5.97	0.07	4.82	0.30	3.28	0.21	2.91	0.58	2.07	0.31	1.97	0.24
	5	5.53	0.70	3.84	0.60	4.55	0.41	3.14	0.85	2.01	0.55	1.94	0.45
		6.00	0.00	4.72	0.37	3.03	0.42	3.24	0.62	1.97	0.33	2.05	0.27
	All	5.53	1.22	4.05	0.98	4.23	1.07	2.58	0.98	2.33	0.80	2.28	0.75
		5.89	1.19	4.93	1.04	3.14	0.72	2.70	0.85	2.17	0.58	2.16	0.55

Visualization of convergence

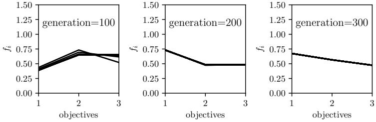


Figure: NEMO-0-L_α

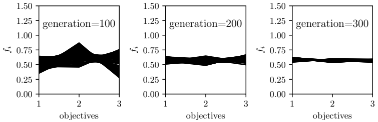


Figure: NEMO-II-L_α

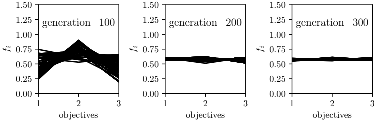


Figure: IEMO/D (prioritizing: new)

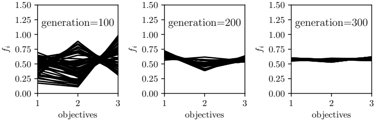


Figure: IEMO/D (prioritizing: old)

Solutions constructed by different methods in last generation for DTLZ2 with $M = 3$. (each connected black line corresponds to a single solution). The decision making policy was simulated by assuming equal weights of objectives. The optimal solution is $s^{opt3} \approx [0.57, 0.57, 0.57]$

Visualization of convergence

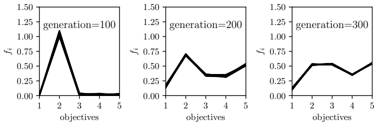


Figure: NEMO-0-L_α

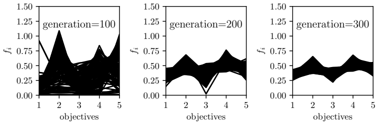


Figure: NEMO-II-L_α

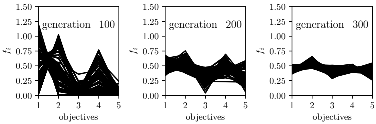


Figure: IEMO/D (prioritizing: new)

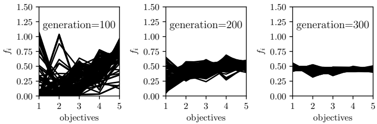
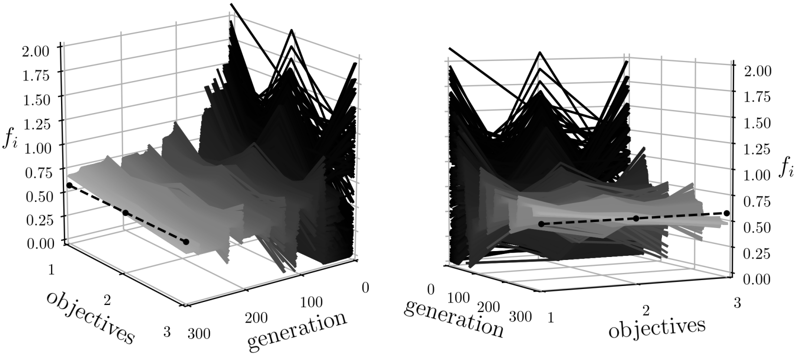


Figure: IEMO/D (prioritizing: old)

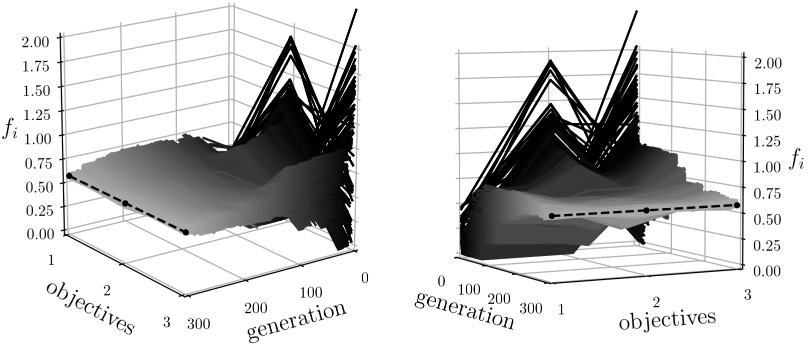
Solutions constructed by different methods in last generation for DTLZ2 with $M = 5$. (each connected black line corresponds to a single solution). The decision making policy was simulated by assuming equal weights of objectives. The optimal solution is $s^{opt5} \approx [0.44, 0.44, 0.44, 0.44, 0.44]$.

Visualization of convergence (3D)



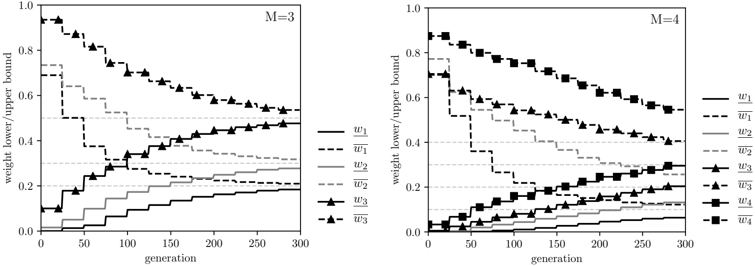
Solutions constructed by NEMO-0$-L_\alpha$ over generations for DTLZ2 with $M = 3$ (each connected black line corresponds to a single solution). The decision making policy was simulated by assuming equal weights of objectives. The optimal solutions are $s^{opt3} \approx [0.57, 0.57, 0.57]$ and $s^{opt5} \approx [0.44, 0.44, 0.44, 0.44, 0.44]$ for $M = 3$ and $M = 5$, respectively. A black-gray gradient is used to distinguish subsequent generations, and optimal solution is marked with a dashed line.

Visualization of convergence (3D)



Solutions constructed by IEMO/D$_{ICRSU-DIR-OLD}$ over generations for DTLZ2 with $M = 3$ (each connected black line corresponds to a single solution). The decision making policy was simulated by assuming equal weights of objectives. The optimal solutions are $s^{opt3} \approx [0.57, 0.57, 0.57]$ and $s^{opt5} \approx [0.44, 0.44, 0.44, 0.44, 0.44]$ for $M = 3$ and $M = 5$, respectively. A black-gray gradient is used to distinguish subsequent generations, and optimal solution is marked with a dashed line.

Visualization of convergence



Bounds of objectives weights ($\underline{w}$ and $\overline{w}$) approximated with ICRSU throughout 300 generations for DTLZ2 with $M = 3$ or $M = 4$ objectives, averaged over 100 runs.

Comparison with NEMO-0 and NEMO-II

Average ARSD and attained ranks $\overline{R}$ for the populations constructed in the final generation by IEMOD$_{ICRSU-DIR-OLD}$ and NEMO using different preference models for various artificial DM's models $L_{\alpha^{DM}}$ ($\rho = 3$).

	DTLZ2 (DTLZ2-C in case $\alpha^{DM} = 1$)						WFG1					
	$M = 3$			$M = 5$			$M = 3$			$M = 5$		
	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$
	$\alpha^{DM} = 1 \Rightarrow$ Linear additive value function											
NEMO-0-G	43.71	73.94	5.45	138.05	116.18	5.66	53.61	41.12	6.72	91.98	38.98	6.80
NEMO-0-PL	31.52	55.74	5.04	112.60	90.49	5.23	21.70	41.43	4.13	29.38	48.92	4.32
NEMO-0-L	20.45	40.08	4.18	77.70	90.63	4.20	10.73	23.95	3.54	33.80	79.62	4.00
NEMO-II-G	384.22	81.48	8.77	743.33	53.09	8.78	257.17	52.99	8.57	175.23	30.90	8.41
NEMO-II-PL	338.78	91.33	8.20	673.22	82.46	8.20	248.88	54.29	8.41	176.89	33.41	8.45
NEMO-II-L	1.72	2.55	2.86	40.30	22.37	3.78	1.09	1.46	1.33	10.71	9.31	3.55
IEMO/D$_{ICRSU-DIR-OLD}$, $\alpha = \infty$	12.02	16.48	4.90	64.20	54.89	4.50	22.66	21.62	5.58	16.20	10.42	4.67
IEMO/D$_{ICRSU-DIR-OLD}$, $\alpha = 5$	7.61	15.05	3.78	36.35	39.43	3.03	6.15	6.13	3.56	8.62	5.00	3.39
IEMO/D$_{ICRSU-DIR-OLD}$, $\alpha = 1$	0.66	1.45	1.82	13.18	11.24	1.62	6.48	6.59	3.16	1.99	2.36	1.41

Comparison with NEMO-0 and NEMO-II

Average ARSD and attained ranks $\overline{R}$ for the populations constructed in the final generation by IEMOD$_{ICRSU-DIR-OLD}$ and NEMO using different preference models for various artificial DM's models $L_{\alpha^{DM}}$ ($\rho = 3$).

	DTLZ2						WFG1					
	$M = 3$			$M = 5$			$M = 3$			$M = 5$		
	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$
	$\alpha^{DM} = 5$											
NEMO-0-G	23.68	59.42	3.54	39.09	46.07	4.30	30.26	25.07	5.61	54.98	33.51	6.14
NEMO-0-PL	57.22	100.74	4.28	64.04	85.91	4.41	51.74	85.19	5.17	67.11	82.83	5.74
NEMO-0-L	139.31	179.04	6.01	91.15	125.80	4.67	60.03	87.14	5.46	54.25	78.45	5.48
NEMO-II-G	357.41	111.81	8.34	723.87	158.64	8.40	211.76	46.66	8.49	128.56	28.18	8.12
NEMO-II-PL	313.27	94.61	7.70	729.63	159.53	8.60	197.49	40.74	8.08	128.30	28.88	8.20
NEMO-II-L	141.28	168.15	5.74	100.33	117.05	5.51	21.98	37.69	3.95	14.79	24.22	3.72
IEMO/D$_{ICRSU-DIR-OLD}$, $\alpha = \infty$	4.51	22.30	2.82	12.03	17.26	3.27	5.85	5.19	3.31	8.10	5.10	3.36
IEMO/D$_{ICRSU-DIR-OLD}$, $\alpha = 5$	0.38	0.76	1.44	4.88	6.43	2.38	1.10	1.20	1.31	4.53	3.65	2.22
IEMO/D$_{ICRSU-DIR-OLD}$, $\alpha = 1$	98.78	138.88	5.13	37.98	68.71	3.46	21.53	47.87	3.62	5.05	5.77	2.02

Comparison with NEMO-0 and NEMO-II

Average ARSD and attained ranks $\overline{R}$ for the populations constructed in the final generation by $\text{IEMOD}_{ICRSU-DIR-OLD}$ and NEMO using different preference models for various artificial DM's models $L_{\alpha^{DM}}$ $(\rho = 3)$.

	DTLZ2						WFG1					
	$M=3$			$M=5$			$M=3$			$M=5$		
	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$
	$\alpha^{DM} = \infty \Rightarrow$ Chebyshev function											
NEMO-0-G	39.54	73.05	3.37	38.07	46.45	3.86	35.17	27.01	5.34	53.88	33.67	6.20
NEMO-0-PL	77.90	99.86	4.44	76.54	82.78	4.69	44.14	49.72	4.93	44.75	77.76	4.96
NEMO-0-L	134.57	159.08	5.80	118.87	142.29	4.88	103.86	125.98	6.15	54.96	92.02	5.15
NEMO-II-G	358.01	99.13	8.14	722.15	163.11	8.45	217.91	44.34	8.49	131.12	28.85	8.34
NEMO-II-PL	326.87	85.83	7.75	731.57	168.22	8.53	202.69	37.81	8.06	129.45	29.24	8.20
NEMO-II-L	169.08	153.53	6.11	107.60	118.23	5.34	36.87	57.25	4.17	21.36	33.32	4.01
$\text{IEMO/D}_{ICRSU-DIR-OLD}$, $\alpha = \infty$	5.83	5.78	2.25	16.61	12.06	3.37	5.56	3.84	2.41	8.29	4.14	3.22
$\text{IEMO/D}_{ICRSU-DIR-OLD}$, $\alpha = 5$	4.55	5.65	1.85	10.52	9.86	2.42	4.16	4.84	1.74	5.83	3.86	2.51
$\text{IEMO/D}_{ICRSU-DIR-OLD}$, $\alpha = 1$	127.59	149.39	5.29	45.15	71.21	3.46	21.19	38.42	3.71	7.75	7.35	2.41

Impact of inconsistency on evolutionary search

Average ARSD for the populations constructed in the final generation by IEMO/D$_{ICRSU-DIR-OLD}$ using different preference models L_α with various artificial DM's models $L_{\alpha^{DM}}$ ($\rho = 3$), for DTLZ2 with $M = 3$. Average ranks $\overline{R}$ attained by different variants of IEMO/D according to ARSD.

	$\alpha^{DM} = 3$			$\alpha^{DM} = 5$			$\alpha^{DM} = 7$			$\alpha^{DM} = 9$			$\alpha^{DM} = 11$			$\alpha^{DM} = \infty$		
	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$
α	DTLZ2, $M = 3$, $\rho = 5$																	
3	19.28	62.16	2.10	61.66	177.22	2.75	152.96	339.20	3.07	239.25	542.87	3.41	258.32	706.80	3.16	825.06	1286.61	3.78
5	30.60	110.60	2.46	38.37	75.95	2.58	79.91	201.41	2.92	122.84	226.27	3.03	175.61	373.60	3.11	492.78	745.92	3.28
7	60.67	214.12	3.22	63.91	162.45	3.11	53.31	116.39	2.68	63.93	140.34	2.53	131.30	354.92	2.96	484.06	754.29	3.21
9	57.97	150.97	3.44	112.83	603.11	3.03	63.52	115.68	2.94	85.43	146.15	3.01	144.29	353.12	2.87	447.89	565.29	3.04
11	90.53	201.65	3.78	112.14	277.69	3.53	138.82	305.26	3.39	114.04	191.17	3.02	104.23	165.14	2.90	539.89	619.96	3.58
∞	440.85	1538.99	5.23	269.28	387.37	5.05	553.47	1144.60	4.97	485.38	966.04	4.88	325.50	459.27	4.82	692.48	700.58	4.11

Conclusions

- We proposed an interactive evolutionary multiple objective optimization algorithm implementing the paradigm of decomposition, called IEMO/D.
- IEMO/D accepts indirect preference information (e.g., pairwise comparisons), hence not forcing the DM to specify directly the values of preference model parameters.
- IEMO/D generates a set uniformly distributed instances of L_α-norms that are compatible with the DM's pieces of preferences. This process involves the Monte Carlo simulation based on a suitably adapted rejection sampling method.

Conclusions

- Our experimental results proved that both an evolutionary mechanism and a robustness preoccupation had a strong impact on the results of the interactive optimization. In fact, IEMO/D outperformed:
  - NEMO-0(-L_α) which guides the search by means of an arbitrarily selected single compatible preference model instance,
  - as well as its robust counterpart NEMO-II (-L_α).

  The advantage of IEMO/D over NEMO-0-L_α and NEMO-II-L_α is more evident when more objectives are involved.
- We demonstrated that the results are vastly improved when IEMO/D employs L_α-norm was highly consistent with the DM's judgement policy.