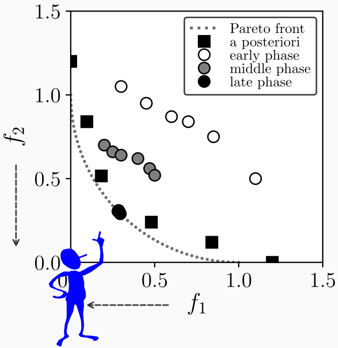
Interactive evolutionary multiple objective optimization algorithm using a fast calculation of holistic acceptabilities

Michał Tomczyk **Miłosz Kadziński**

Institute of Computing Science
Poznan University of Technology, Poland

michal.tomczyk@cs.put.poznan.pl
www.cs.put.poznan.pl/mtomczyk/

Research scope



EMO
MCDA

Keywords:

- Multiple Objective Optimization
- Evolutionary Algorithms
- **Preference Learning**

Preference learning:

1. It is a cooperation between the algorithm and the DM where one participant **interactively** learns from the other.
2. The DM's preferences are inferred via **preference disaggregation** – deriving a global model from some incomplete preferential structures, e.g., **pairwise comparisons**.

Reminder on IEMO/D and EMOSOR

- **EMOSOR** and **IEMOD/D** use a functional preference model to **represent the DM's preferences mathematically –** L-norms:

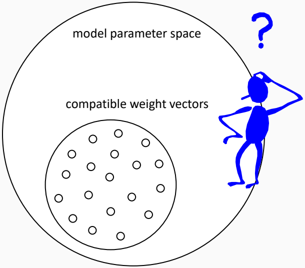
$$L_{\alpha}^{w}(s) = \begin{cases} \left[\sum_{i=1}^{M} (w_i s_i)^{\alpha}\right]^{1/\alpha} & \text{for } \alpha < \infty, \\ max_{1,\ldots,M\{w_i s_i\}} & \text{for } \alpha = \infty. \end{cases}$$

- α – compensation level – is provided *a priori*. Weight vector **is uknown**.
- Interactivelly provided **pairwise comparisons** are used to **constrain** the model parameter space:

$$\bigvee_{s^j \succ s^k \epsilon H} L_{\alpha}^{w}(s^j) < L_{\alpha}^{w}(s^k) \rightarrow w \text{ is compatible (feasible)}$$

- **A fine representation** of the compatible weight vectors is used to assess solutions in the population **consistently with the DM's preferences**.

How to build recommendations consistent with the DM's preferences?



IEMOD/D: *M. K. Tomczyk and M. Kadziński. Decomposition-based interactive evolutionary algorithm for multiple objective optimization. IEEE Transactions on Evolutionary Computation 24, 2 (2020), 320–334*

EMOSOR: *M. K. Tomczyk and M. Kadziński. EMOSOR: Evolutionary multiple objective optimization guided by interactive stochastic ordinal regression. Computers & Operations Research 108, 2019, 134–154*

Reminder on IEMO/D and EMOSOR

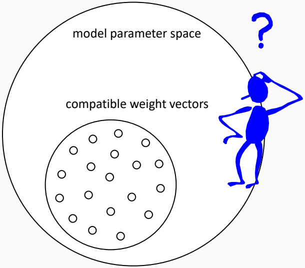
ROBUST ORDINAL REGRESSION – extreme results of the analysis; preservative, but imposes lower evolutionary pressure. **Examples:**

- **Necessary Preference –** one solution is considered preferred than another if it attains a better score for each compatible preference model instance.
- **Potential Optimality –** a solution is considered potentially optimal when it attains the best score in the solution set for at least one compatible preference model instance.

STOCHASTIC ORDINAL REGRESSION – results derived by aggregating potential outcomes imposed by each compatible preference model instance; there is a risk (controlled) of making mistakes, but allows better differentiating between solutions. **Examples:**

- **Pairwise Winning Index –** the probability that one solution is better than another, estimated by using each compatible preference model instance.
- **Rank Acceptability Index** – the probability that a solution attains j-th rank in the population, estimated by using each compatible preference model instance.

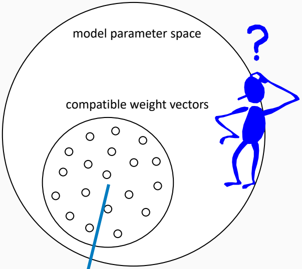
How to build recommendations consistent with the DM's preferences?

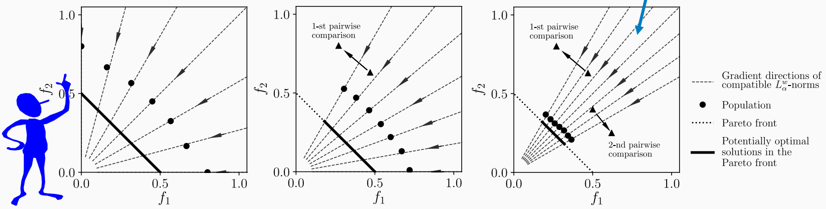


Reminder on IEMO/D

IEMO/D

- IEMO/D is based on **MOEA/D algorithm.**
- Interactive, based on pairwise comparisons of solutions
- Represents the DM's by using a fine representation of the space of compatible L-norms.
- After each preference elicitation, this set is incorporated into the decomposition-based framework as optimization goals.



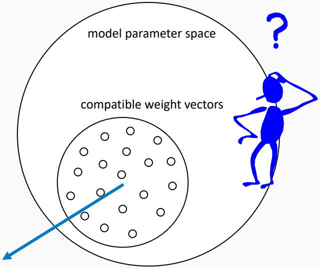


***IEMO/D:** M. K. Tomczyk and M. Kadziński. Decomposition-based interactive evolutionary algorithm for multiple objective optimization. IEEE Transactions on Evolutionary Computation 24, 2 (2020), 320–334*

Reminder on EMOSOR

EMOSOR

- EMOSOR is based on **NSGA-II algorithm.**
- Interactive, based on pairwise comparisons of solutions
- Represents the DM's by using a fine representation of the space of compatible L-norms.
- The representative set is used to assess solutions in the population consistently with the DM's preferences.



Particularly good results were reported by EMOSOR when assessing solutions according to their holistic acceptabilities:

$$HA(s^j, P) = \sum_{r=i}^{N} \alpha_i \, RAI(s^j, r, P)$$

Rank Acceptability Index

Weighting Scheme, here we consider the inverse scheme: $1/r$

EMOSOR: *M. K. Tomczyk and M. Kadziński. EMOSOR: Evolutionary multiple objective optimization guided by interactive stochastic ordinal regression. Computers & Operations Research 108, 2019, 134–154*

iMOEA-HA = the best from IEMO/D and EMOSOR

IEMO/D

- Efficient evolutionary-framework ✓ :
  - *decomposition-based / steady-state*
  - *restricted mating pool*
- Low-computational complexity ✓
- Identifies potentially optimal solutions (**robust ordinal regression**) ✗

EMOSOR

- Less efficient evolutionary-framework: ✗
  - Fronts-based / generational
  - Non-restricted mating pool
- High-computational complexity ✗
- Assesses solutions according to holistic acceptabilities (**stochastic ordinal regression**) ✓

iMOEA-HA

- Efficient evolutionary-framework ✓ :
  - *quasi decomposition-based (restricted mating pool);*
  - *front-based + steady-state (non-dominated fronts + holistic acceptabilities)*
- **Considerably lower computational complexity when compared to EMOSOR** ✓
- Assesses solutions according to holistic acceptabilities (stochastic ordinal regression) ✓

Reminder on EMOSOR

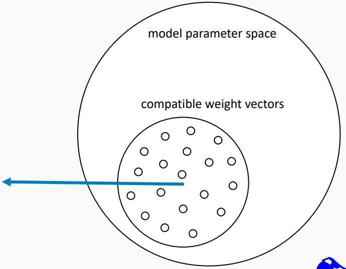
Using the set of compatible L-norms, we can identify the potential ranks a solution may attain:

	1	2	3	4	5	...	N
$L_\alpha^{1,w}$	s^4	s^2	s^N	s^7	s^9	...	s^5
$L_\alpha^{2,w}$	s^1	s^2	s^3	s^5	s^4	...	s^8
$L_\alpha^{3,w}$	s^2	s^1	s^7	s^5	s^4	...	s^9
$L_\alpha^{4,w}$	s^4	s^2	s^1	s^N	s^3	...	s^5
$L_\alpha^{5,w}$	s^4	s^1	s^N	s^2	s^7	...	s^5
...	...	...	...	...	...	...	...
$L_\alpha^{G,w}$	s^2	s^N	s^1	s^4	s^3	...	s^5

$$HA(s^j, P) = \sum_{r=i}^{N} 1/r \; RAI(s^j, r, P)$$

Then, we may compute solutions' holistic acceptabilities – the bigger the score, the better fitness:

	s^1	s^2	s^3	s^4	s^5	...	s^N
HA	0.89	0.76	0.54	0.43	0.32	...	0.02



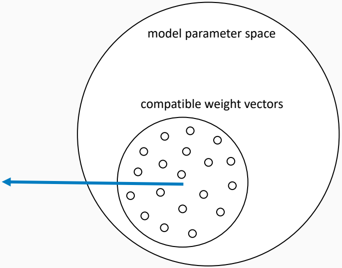
Estimating holistic acceptabilities is computationally demanding

iMOEA-HA: fast calculation of holistic acceptabilities

Observation: worse ranks may not contribute to the overall HA score relevantly. Therefore, to reduce the computational burden, only K-first ($K \ll N$) ranks may be involved in HA-score estimation.

$K \ll N$

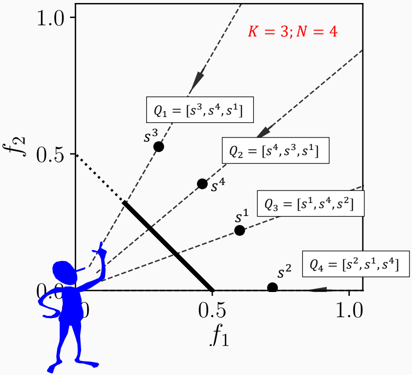
	1	2	3	4	5	...	N
$L_\alpha^{1,w}$	s^4	s^2	s^N	s^7	s^9	...	s^5
$L_\alpha^{2,w}$	s^1	s^2	s^3	s^5	s^4	...	s^8
$L_\alpha^{3,w}$	s^2	s^1	s^7	s^5	s^4	...	s^9
$L_\alpha^{4,w}$	s^4	s^2	s^1	s^N	s^3	...	s^5
$L_\alpha^{5,w}$	s^4	s^1	s^N	s^2	s^7	...	s^5
...	...	...	...	...	...	...	...
$L_\alpha^{N,w}$	s^2	s^N	s^1	s^4	s^3	...	s^5



The estimation accuracy depends on the queue limit K

iMOEA-HA: fast calculation of holistic acceptabilities

Solution: use the maintained compatible model instances as queues of a limited size (K) **employed to sort solutions locally**. The associated function is used as **a sorting criterion**. If K is relativelly small, the queues can be implemented using the insertion-sort procedure.



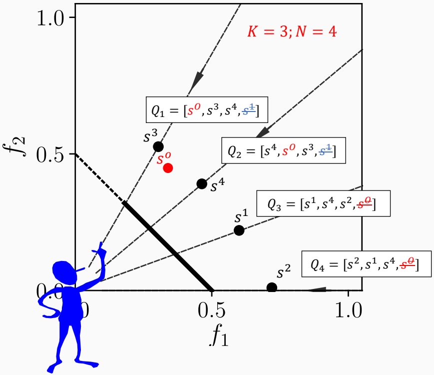
- At the cost of increased memory complexity, the **computational complexity is reduced**.
- The possible ranks a solution main attain are dynamically updated and stored so that HA-score estimation can be performed quickly.
- iMOEA-HA is run in a steady-state mode. Therefore it implements two procedures: **insertion and deletion for updating queues**.

Auxiliary data structure for storing ranks

s^1: [1,1,2] → $HA =$ 0.54
s^2: [1,0,1] → $HA =$ 0.33
s^3: [1,1,0] → $HA =$ 0.38
s^4: [1,2,1] → $HA =$ 0.58

iMOEA-HA: fast calculation of holistic acceptabilities

Solution: use the maintained compatible model instances as queues of a limited size (K) **employed to sort solutions locally**. The associated function is used as **a sorting criterion**. If K is relativelly small, the queues can be implemented using the insertion-sort procedure.



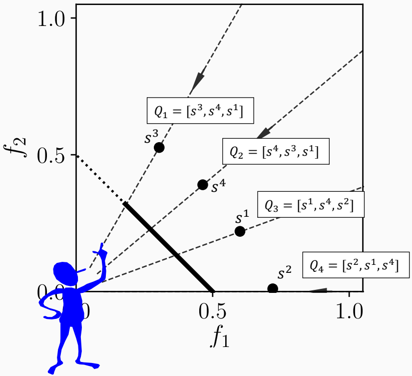
- At the cost of increased memory complexity, the **computational complexity is reduced**.
- The possible ranks a solution main attain are dynamically updated and stored so that HA-score estimation can be performed quickly.
- iMOEA-HA is run in a steady-state mode. Therefore it implements two procedures: **insertion and deletion for updating queues**.

Auxiliary data structure for storing ranks

s^1: $[1,1,0] \rightarrow HA = 0.38$
s^2: $[1,0,1] \rightarrow HA = 0.33$
s^3: $[0,1,1] \rightarrow HA = \mathbf{0.21}$
s^4: $[1,1,2] \rightarrow HA = 0.54$
s^o: $[1,1,0] \rightarrow HA = 0.38$

Candidate for removal, if s^3 is in the last non-dominated front, it will be removed from the population.

iMOEA-HA: summary



iMOEA-HA – characteristics:

- interactive, based on pairwise comparisons, represents the DM's preferences as a set of compatible L-norms.
- is run in a steady-state mode
- sorts solutions according to two criteria:
  1. non-dominated fronts (fast calculation)
  2. HA-scores (fast calculation)
- in the study, we considered two selection procedures:
  - **(nonrestricted)** a regular tournament selection
  - **(restricted)** two random solution from a randomly selected queue

Experimental setting

- **Evolutionary setting:** follows the standards in the literature on EMO
- **Decision-making layer:**

Interactions: triggered 10 times during a single run, evenly distributed

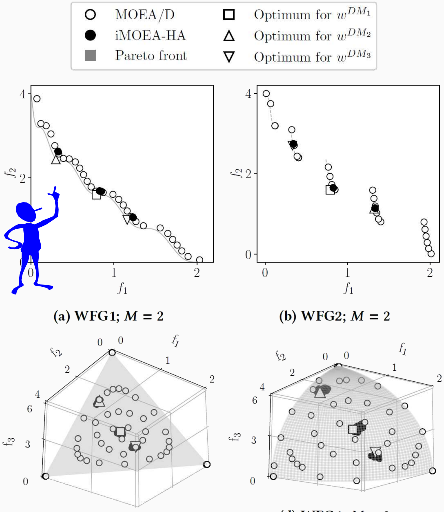
Simulating the DM's answers: the DM's value system was modeled using an L-norm

Comprehensiveness & reliability of the experiments: for each setting, the run was repeated 100 times, each time involving a different artificial DM (these were pre-generated by generating uniformly distributed weight vectors).

Performance evaluation: solutions in the population were compared against the optimal solution identified in advance using exact or heuristic methods. Specifically, we computed the Best/Average Relative Score Differences (BRSD/ARSD) between the most favored (average for all solutions) and the optimum, where scores were assessed using the artificial DM's internal function.

Statistics: mean, standard deviation, average rank

Visualization of convergence



(a) WFG1; $M = 2$ (b) WFG2; $M = 2$

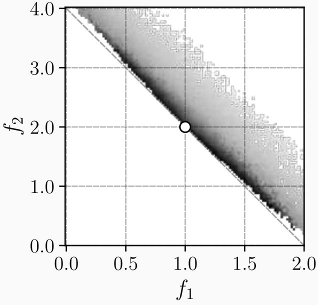
(c) WFG3; $M = 3$ (d) WFG4; $M = 3$

Populations constructed by MOEA/D and iMOEA-HA (here, with tournament selection; K=10) run for different DMs, applied to different benchmark problems (the A-parameters in WFG3 were set to 1 to make the PF non-degenerated).

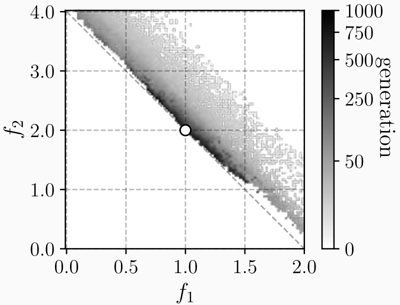
- $\boldsymbol{M} = \mathbf{2}$:
  - $w^{DM_1} = [0.5, 0.5]$
  - $w^{DM_2} = [0.3, 0.7]$
  - $w^{DM_3} = [0.8, 0.2]$
- $\boldsymbol{M} = \mathbf{3}$:
  - $w^{DM_1} = \left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right]$
  - $w^{DM_2} = [0.7, 0.2, 0.1]$
  - $w^{DM_3} = [0.2, 0.3, 0.5]$

iMOEA-HA can successfully converge towards different DM's optima

Performance evaluation for different queue limits



$K = 1$



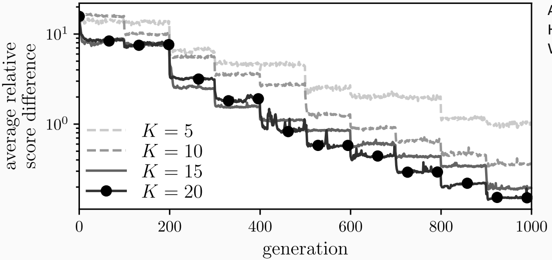
$K = 20$

ViPEMO plots for iMOEA-HA with (a) $K = 1$ and (b) $K = 20$ run for WFG3 with $M = 2$ and $w^{DM} = [0.5, 0.5]$. The DM's most preferred option is marked with a white dot.

The greater the K, the more accurate the HA-score estimation, and therefore population convergences faster towards the DM's most relevant solution

ViPEMO: *M. Kadziński, M. K. Tomczyk, and R. Słowiński, Preference-based cone contraction algorithms for interactive evolutionary multiple objective optimization. Swarm and Evolutionary Computation 52, 100602, 2020.*

Performance evaluation for different queue limits



ARSD (averaged across all runs) for iMOEA-HA with different queue sizes K applied to WFG4 with $M = 3$ objectives.

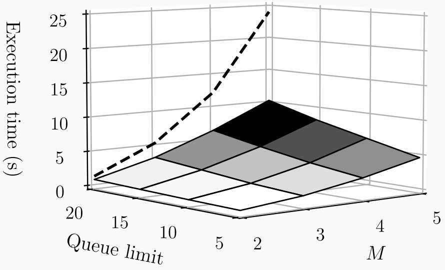
Average ranks attained by iMOEA-HA with different queue limits K for all test problems considered jointly.

K	$M = 2$	$M = 3$	$M = 4$	$M = 5$
5	2.72	2.73	2.75	2.84
10	2.50	2.52	2.54	2.55
15	2.45	2.42	2.39	2.37
20	2.32	2.33	2.32	2.24

Performance improvement

The greater the K, the more accurate the HA-score estimation, and therefore population convergences faster towards the DM's most relevant solution

Comparison of iMOEA-HA and EMOSOR execution times



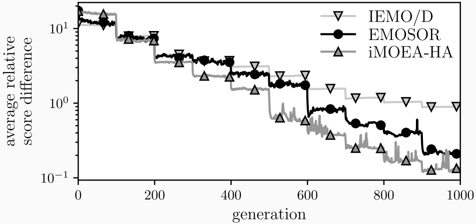
Execution times for iMOEA-HA (surface plot) and EMOSOR (dashed line).

- iMOEA-HA performs significantly faster than its main competitor, EMOSOR
- iMOEA-HA with the queue limit at least of K=15 proved competitive to EMOSOR. Therefore, this variant was employed in the following experiments.

Comparison with EMOSOR and IEMO/D

		EMOSOR			iMOEA-HA			IEMO/D		
	M	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$	Mean	StD	$\overline{R}$
WFG1	2	1.161	9.702	1.87	1.532	13.366	1.89	1.345	11.505	2.24
	3	2.061	17.386	2.15	0.751	4.996	1.65	1.165	6.722	2.20
	4	1.366	5.079	1.87	0.984	2.229	1.63	1.873	6.771	2.50
	5	2.131	3.753	1.76	1.763	3.676	1.61	3.387	5.329	2.63
WFG2	2	0.702	3.092	2.02	0.362	2.101	2.33	0.305	1.285	1.65
	3	1.465	4.526	1.94	0.797	2.371	1.93	0.970	3.146	2.13
	4	1.418	2.674	1.75	1.840	4.605	1.94	2.684	5.532	2.31
	5	1.675	3.181	1.54	4.750	10.018	2.15	4.146	9.539	2.31
WFG3	2	0.053	0.373	1.72	0.304	2.846	1.93	0.147	1.146	2.35
	3	0.280	2.393	1.92	0.124	0.967	1.73	0.168	1.193	2.35
	4	0.087	0.146	1.81	0.073	0.097	1.77	0.109	0.128	2.42
	5	0.177	0.218	1.81	0.167	0.170	1.89	0.210	0.262	2.30
WFG4	2	0.191	1.203	1.97	0.265	2.597	1.99	0.009	0.025	2.04
	3	0.214	1.021	1.74	0.196	0.946	1.91	0.898	5.479	2.35
	4	0.834	1.963	1.77	0.831	2.402	1.78	1.901	5.490	2.45
	5	2.005	6.424	1.69	2.153	4.565	1.78	4.018	9.315	2.53
WFG5	2	2.608	24.164	1.84	2.659	24.168	2.02	2.654	24.167	2.14
	3	1.253	9.948	1.74	1.284	9.847	2.01	1.497	9.497	2.25
	4	1.567	7.130	1.73	1.728	6.661	1.87	3.340	18.096	2.40
	5	2.752	6.426	1.75	3.860	15.540	1.83	5.303	14.081	2.42
WFG6	2	0.738	6.916	2.03	0.793	7.355	1.61	0.516	4.541	2.36
	3	0.726	5.815	1.76	0.720	5.458	1.98	1.897	16.235	2.26
	4	0.992	3.426	1.74	1.033	3.256	1.78	2.025	5.889	2.48
	5	2.076	4.608	1.78	2.154	5.275	1.68	5.274	13.632	2.54
WFG7	2	0.032	0.286	2.00	0.004	0.011	1.93	0.008	0.025	2.07
	3	0.204	0.540	1.77	0.189	0.683	1.97	1.408	11.635	2.26
	4	0.906	3.094	1.88	1.025	4.547	1.78	1.751	6.310	2.34
	5	2.198	8.386	1.59	2.711	14.009	1.80	4.365	15.640	2.61
WFG8	2	2.882	24.084	1.86	1.324	11.668	1.96	1.387	11.653	2.18
	3	1.885	16.130	1.80	0.398	1.328	1.97	0.746	4.112	2.23
	4	1.173	4.341	1.77	1.319	5.035	1.84	1.997	5.949	2.39
	5	2.722	9.233	1.67	4.287	14.921	1.77	4.927	13.807	2.56
WFG9	2	1.740	15.647	1.66	2.599	23.345	1.96	1.900	16.614	2.38
	3	1.365	10.855	1.72	0.958	6.461	1.87	3.107	27.098	2.41
	4	2.093	11.713	1.72	2.193	9.124	1.80	2.639	10.404	2.48
	5	3.096	9.908	1.64	3.515	10.770	1.80	6.241	17.072	2.56

Average ARSD and respective average ranks R attained by different algorithms applied to WFG1–9 problems with $M = 2 - 5$ objectives.



ARSD (averaged across all runs) for different algorithms applied to WFG4 with $M = 3$ objectives.

iMOEA-HA with K=15 performs similar to EMOSOR, but significantly better than IEMO/D. Given that iMOEA-HA performs much faster than EMOSOR, it can be considered a better algorithm.

Conclusions and avenues for future research

Conclusions

- We introduced a novel preference-based iMOEA-HA algorithm implementing the paradigm of preference learning
- iMOEA-HA is based on the up-to-date concepts in EMO and MCDA, i.e., it uses an efficient evolutionary framework and is based on stochastic ordinal regression
- iMOEA-HA introduces a fast procedure for calculating holistic acceptabilities
- iMOEA-HA performs significantly better than IEMO/D and no worse than EMOSOR, i.e., its two predecessors, but performing much faster than the latter algorithm

Avenues for future research

- we will further enhance the proposed algorithms for maintaining queues by replacing the insertion sort procedure with hybrid approaches
- we will investigate the performance of iMOAE-HA when the DM's preference information is imprecise or when (s)he acts irrationally
- we will apply the proposed algorithm to real-world problems such as portfolio optimization

Thank you for your attention!

michal.tomczyk@cs.put.poznan.pl