Realistic Utility Functions Prove Difficult for State-of-the-Art Interactive Multiobjective Optimization Algorithms

Seyed Mahdi Shavarani
Alliance Manchester Business School,
University of Manchester
Manchester, UK
seyedmahdi.shavarani@manchester.ac.uk

Manuel López-Ibáñez
Alliance Manchester Business School,
University of Manchester
Manchester, UK

Joshua Knowles
University of Manchester
Manchester, UK

ABSTRACT

Improvements to the design of interactive Evolutionary Multiobjective Algorithms (iEMOAs) are unlikely without quantitative assessment of their behaviour in realistic settings. Experiments with human decision-makers (DMs) are of limited scope due to the difficulty of isolating individual biases and replicating the experiment with enough subjects, and enough times, to obtain confidence in the results. Simulation studies may help to overcome these issues, but they require the use of realistic simulations of decision-makers. Machine decision-makers (MDMs) provide a way to carry out such simulation studies, however, studies so far have relied on simple utility functions. In this paper, we analyse and compare two state-of-the-art iEMOAs by means of a MDM that uses a sigmoid-shaped utility function. This sigmoid utility function is based on psychologically realistic models from behavioural economics, and replicates several realistic human behaviours. Our findings are that, on a variety of well-known benchmarks with two and three objectives, the two iEMOAs do not consistently recover the most-preferred points. We hope that these findings provide an impetus for more directed design and analysis of future iEMOAs.

CCS CONCEPTS

• **Theory of computation → Evolutionary algorithms**; • **Applied computing → Multi-criterion optimization and decision-making**.

KEYWORDS

Interactive Evolutionary Multi-Objective Optimization, Design of Experiments, Machine Decision Maker

ACM Reference Format:
Seyed Mahdi Shavarani, Manuel López-Ibáñez, and Joshua Knowles. 2021. Realistic Utility Functions Prove Difficult for State-of-the-Art Interactive Multiobjective Optimization Algorithms. In *2021 Genetic and Evolutionary Computation Conference (GECCO '21), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3449639.3459373

GECCO '21, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8350-9/21/07…$15.00
https://doi.org/10.1145/3449639.3459373

1 INTRODUCTION

> "The grandest discoveries of science have been but the rewards of accurate measurement …" (Lord Kelvin)

Evolutionary algorithms are an attractive approach for multi-objective optimization problems due to their heuristic nature, making them generally applicable, and their use of a population allowing the Pareto front (PF) to be approximated straightforwardly in a single run. *Interactive* Multi-Objective Evolutionary Algorithms (iEMOAs) can also exploit information provided by a Decision Maker (DM) to generate only those parts of the PF that are interesting to the DM. These perceived advantages have led to the introduction of many alternative interactive EMOAs in recent years [1, 4, 9, 19]. However, the science of analysing and comparing different iEMOAs still remains problematic due to the influence of the DM's decisions on final results, which makes experiments' conclusions biased, unreliable and unreplicable. When it comes to iEMOAs, few statistical analyses of such algorithms have been published, and they have mostly used a trivial and unrealistic utility function to stand in for a real DM in the experiments. Here, we propose using a realistic sigmoid utility function (UF) inspired by psychological studies. Furthermore, to the best of our knowledge, there has been no direct head-to-head comparison of different iEMOAs in the literature. Our systematic study aims to fill this gap in the literature, paving the way for much deeper experimental analyses of iEMOAs, and for data-driven improvements in the field.

2 BACKGROUND

Many real-life optimization problems have several conflicting objectives to be optimized simultaneously. Given m objective functions that map each decision vector $\mathbf{x}$ in solution space $X \subseteq \mathbb{R}^n$ to $\mathbf{z} = \mathbf{f}(\mathbf{x})$ in objective space, $Z \subseteq \mathbb{R}^m$, the resulting optimization problem has the following general form:[1]

$$\min_{\mathbf{z}\in Z} \quad \mathbf{z} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})) \quad \text{subject to} \quad \mathbf{x} \in X \ . \tag{1}$$

Often the ultimate goal is to find the feasible solution most preferred by the DM. Interactive methods search for this solution using information that is elicited from the DM during the optimization phase. Interactive EMOAs (iEMOAs) differ in interaction pattern, the preference information, the preference modeland the search engine [21]. Broadly speaking, interactive methods can be categorized into ad hoc and non-ad hoc methods [12]. Non ad hoc methods assume the existence of an underlying utility function (UF) that drives

[1]Assuming minimisation of all objectives without loss of generality.

the decisions of the DM [16].[2] Here, we focus on ranking-based non ad hoc methods, where the DM is asked to provide a (partial) ranking of a number of solutions at each interaction. BC-EMOA [1], NEMO [3] and iTDEA [9] fall in this category.

Interactive methods should, if well-designed, facilitate fast convergence to the interesting parts of the Pareto front. However, analysis of such algorithms, to measure progress and direct future development, would ideally be based on studies involving human DMs. This yields a challenging problem as human DMs differ from one another significantly [5]. Zujevs and Eiduks [22] claim that studies using real DMs need more than 65 DMs to perform experiments on iEMOAs. They propose a MDM framework based on minimizing the distance of solutions to an ideal point. However, they do not attempt to simulate any realistic biases or behaviours.

As an alternative to human DMs, most works use simple or unrealistic utility functions (UF), mostly devised by the algorithm's authors, to simulate the behaviour of a DM. In this regard, comparison of different interactive multi-objective optimization algorithms have been performed using linear [9, 15], non-linear (typically quadratic or polynomial functions) [1, 9, 13, 15] and Tchebycheff UFs [9]. Most studies on *evolutionary* algorithms are over-simplified with the use of arbitrary UFs and assume ideal conditions without biases and non-idealities, whereas Stewart [17] suggests biases exist in human decision making.

The studies by Stewart [17, 18] provide perhaps the most comprehensive simulations of DMs in the context of multi-criteria decision-making. Different psychological biases and realistic behaviours gathered from the literature are simulated by modifying a given UF that represents the idealized true preferences of the DM, and these are used to evaluate the ability of MCDM methods to recover true preferred points. Much later, López-Ibáñez and Knowles [10] adapted the Stewart [17] study in proposing a Machine-Decision-Maker (MDM) framework to serve as a laboratory for analysing the performance of interactive algorithms, and analysed BC-EMOA [1] as a proof of concept. However, the MDM used in the analysis [10] relied on the linear and polynomial UFs proposed in the original BC-EMOA to simulate the DM's true preferences.

Inspired by Kahneman and Tversky [8] and Stewart [17, 18], we adopt a non-trivial realistic sigmoid UF. The parameters of this sigmoid UF allow the simulation of various well-known psychological behaviours, such as non-linear utility, asymmetric attitudes towards gains and losses, and high or low compensatory preferences. Compensatory preferences determine how much detriments in one objective value may be compensated by gains in other objectives. The sigmoid UF function also satisfies the accepted requirement of UFs being quasi-concave in the economics literature [11] and it has been used in recent studies to simulate a realistic DM [7, 18].

We aim to perform a comprehensive comparison and analysis of the performance of two well-known iEMOAs, iTDEA and BC-EMOA, using the sigmoid UF. We selected these two particular algorithms due to their similar style of interaction with the DM and yet different internal mechanisms, which makes their comparison of interest.

[2]In practice, it may be impossible to determine whether a DM is driven by an UF unknown to them (as assumed by the MDM framework) or not.

Brain-Computer EMOA (BC-EMOA) [1] is based on NSGA-II and uses Support Vector Machine (SVM) to learn the underlying preference model of the DM. In each interaction, a small subset of non-dominated solutions are presented to the DM for pairwise comparison and consequently the solutions are ranked based on these comparisons. Solutions along with their ranks are used for training the SVM. The learned UF replaces the crowding distance of NSGA-II in subsequent generations to break ties between mutually non-dominated solutions. In their paper (ibid.), the DM is simulated as a (linear or quadratic) UF in order to investigate how well the algorithm would approximate the preferred solution.

Interactive Territory Defining Evolutionary Algorithm (iTDEA) [9] maintains two populations, a fixed-size regular population that contains both dominated and non-dominated individuals, and a variable-size archive that only contains non-dominated solutions. In each generation, a single offspring is generated. If it is dominated by the members of the regular population it is discarded, otherwise it replaces a dominated individual or a random one if the new solution does not dominate any other solution. The offspring enters the archive if it is non-dominated and also does not violate the territory of existing individuals. Territory is defined as a region occupied by each individual in the objective space and it controls the density of each region. Solutions in interesting parts of the PF are assigned smaller territory size. As the result, number of solutions in those parts increases, making them more dense. For the territory violation check, the offspring is compared to its closest individual in the archive. In iTDEA, the DM is asked to select the most preferred solution ($\mathbf{z}^{\mathrm{MPS}}$) among several filtered solutions at each interaction. The solutions in the proximity of the best, classified as superior solutions, are given smaller territory, making regions of interest more crowded. The original paper [9] evaluated iTDEA using only linear, quadratic and Tchebycheff UFs.

3 A REALISTIC UTILITY FUNCTION

Most studies on non-ad hoc iEMOAs simulate the behaviour of a human DM by means of a utility function $U(\mathbf{z})\colon \mathbb{R}^m \to [0,1]$ that represents the DM's true preference [1, 9, 14, 20]. To be consistent with typical multi-objective optimization benchmark problems, we assume here that $U(\mathbf{z})$ should be minimized.

Psychological studies by Kahneman and Tversky [8] have established that human DMs often evaluate objective values as gains or losses relative to internal reference levels and their preferences follow an *S*-shaped curve that is concave for gains and convex for losses. Inspired by such studies, Stewart [17] proposed the sigmoid UF shown in Eq. (2) as a more realistic alternative to other UFs used in the simulation of DMs,

$$U(\mathbf{z}) = \sum_{i=1}^{m} w_i u_i(z_i)$$
$$u_i(z_i) = \begin{cases} \dfrac{\lambda_i \cdot (e^{\alpha_i z_i} - 1)}{e^{\alpha_i \tau_i} - 1} & \text{if } 0 \le z_i \le \tau_i \\ \lambda_i + \dfrac{(1-\lambda_i)(1 - e^{-\beta_i (z_i - \tau_i)})}{1 - e^{-\beta_i (1-\tau_i)}} & \text{if } \tau_i < z_i \le 1 \end{cases}, \quad (2)$$

where $w_i \in [0,1]$ is the relative weight of the i^{th} objective function ($\sum_{i=1}^{m} w_i = 1$) and $u_i(z_i)$ is its marginal value function, λ_i is the value of $u_i(z_i)$ at the reference level τ_i, which is the value of objective i where the function changes from concave to convex, i.e., the threshold that separates the values perceived as "gains" or "losses" by the DM.

Different decision making behaviours can be simulated by different combinations of τ_i and λ_i values as described in Table 1. In addition, parameters α_i and β_i control the non-linearity of the function for the i^{th} objective. Figure 1 illustrates the effect of these parameters on the shape of the marginal utility function.

Table 1: Description of different types of DM behaviours simulated by combinations of τ_i and λ_i adapted from [17].

Type	τ_i	λ_i	Description
1	[0.1, 0.4]	[0.1, 0.4]	Limited range of compensation; plus sharp preference threshold
2	[0.1, 0.4]	[0.6, 0.9]	Limited range of compensation.
3	[0.6, 0.9]	[0.1, 0.4]	Mainly compensatory preferences, but with sharp preference threshold.
4	[0.6, 0.9]	[0.6, 0.9]	Mainly compensatory preferences.

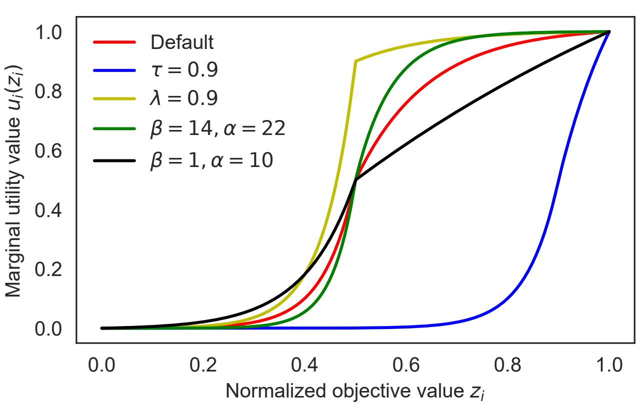

Figure 1: Shape of sigmoid UF with different parameters. In "Default" $\tau = 0.5, \lambda = 0.5, \alpha = 16, \beta = 7$. The effects of changing each parameter from the default value on the shape of the function is illustrated in other curves.

As far as we know, the above UF has never been used as part of a machine decision-maker to evaluate iEMOAs in the literature and we expect it to be significantly harder than linear UFs and produce different behaviours than quadratic UFs, even without simulating any other known DM biases such as noise [10, 17].

For some combinations of sigmoid UF parameters, the utility value of all the PF solutions is squeezed into a narrow interval, which means there is not much difference in utility value between different PF solutions, thus, making it difficult for the algorithms to converge towards the $\mathbf{z}^{\text{MPS}}$. On the other hand, having the $\mathbf{z}^{\text{MPS}}$ on the corners or in the middle of the PF would make it impossible to distinguish between undesired bias towards those particular regions of the objective space and proper convergence to the $\mathbf{z}^{\text{MPS}}$. To address these issues, we also propose a simple method to find the appropriate parameters ($\alpha, \beta, \tau, \lambda$) of the sigmoid UF.

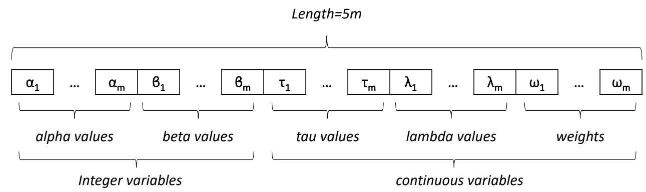

Figure 2: Solution representation (chromosome) used by the EA to find the sigmoid UF parameters that optimize the single objective problem in Eq. (3).

The algorithm works as follows. First, given the PF of a particular problem, or a very good approximation thereof, a point of interest on the approximated PF is chosen as the desired preferred point $\check{\mathbf{z}}$. In our experiments, $\check{\mathbf{z}}$ was set to $(\mathbf{z}^{\text{nadir}} - \mathbf{z}^{\text{ideal}})/3$, where $\mathbf{z}^{\text{nadir}}$ and $\mathbf{z}^{\text{ideal}}$ are defined with respect to the approximated PF. As this point may be infeasible, the following single-objective optimization problem is solved to find the parameters of the sigmoid UF (Eq. 2) for all objectives i ($\alpha_i, \beta_i, \tau_i, \lambda_i$ and w_i) in a way to make the $\mathbf{z}^{\text{MPS}}$ as close as possible to $\check{\mathbf{z}}$:[3]

$$\begin{aligned} \min \quad & \delta = \sum_{i=1}^{m} |\check{\mathbf{z}}_i - \mathbf{z}_i^{\text{MPS}}| \\ \text{subject to:} \quad & \\ & \mathbf{z}^{\text{MPS}} = \arg\min_{\mathbf{z} \in PF} U(\mathbf{z}; \alpha, \beta, \tau, \lambda, \mathbf{w}) \\ & \alpha_i > \beta_i \quad \forall i = 1, \ldots, m \\ & \textstyle\sum_{i=1}^{m} w_i = 1 \\ & \alpha_i, \beta_i \in \mathbb{Z}^+; \tau_i, \lambda_i, w_i \in [0, 1] \qquad \forall i = 1, \ldots, m \end{aligned} \tag{3}$$

By further restricting the ranges for these decision variables, one can simulate various DM behaviors as laid out in Table 1. The decision vector of this problem has a length of $5m$ as depicted in Figure 2. τ, λ and $\mathbf{w}$ are defined as continuous variables, while α and β values are integers. To calculate the fitness of each decision vector, the resulting sigmoid UF is applied to all objective vectors in the PF. Then the Manhattan distance (for speed) between the point on the PF having the minimum UF ($\mathbf{z}^{\text{MPS}}$) and $\check{\mathbf{z}}$ is considered as the fitness of the solution. The problem of minimizing the distance between $\mathbf{z}^{\text{MPS}}$ and $\check{\mathbf{z}}$ can be optimized by any single-objective optimizer. For simplicity, here we use the Simple Genetic Algorithm implemented by the Pygmo library [2].

4 STATISTICAL STUDY OF TWO IEMOAS USING REALISTIC UFS

4.1 Experimental setup

4.1.1 Benchmark problems. We use benchmark problems DTLZ1, DTLZ2 and DTLZ7 [6] with 2 and 3 objectives, which were the problems used in both original works [1, 9].[4] Having few objectives would help us to better understand and investigate the problem and also enables the visualization of UFs. As suggested [6], the solution

[3] Here we deviate from the original formulation of Stewart, which assumes non-compensatory preferences for substantially poor values of z_i. An equivalent formulation would use $\alpha_i < \beta_i$ due to the minimization of U.

[4] It is not always obvious how the parameters of these algorithms should be set for new problems, as some parameter values in iTDEA are set specifically for each problem.

space dimension (n) is $m+4$ for DTLZ1, $m+9$ for DTLZ2 and $m+19$ for DTLZ7, m being the number of objectives.

4.1.2 Machine DM parameter settings. We adopt the machine DM (MDM) proposed in [10], replacing the linear and quadratic UFs used there by the sigmoid UF explained above (Eq. 2). Since the sigmoid UF already has several parameters that simulate various human behaviours, we do not enable any additional biases in the MDM to simplify the analysis here.

For the ranges of the UF parameter, we consider first the four possible combinations of intervals for τ_i and λ_i shown in Table 1. Together with the 5 benchmark problems, this results in 20 configurations of UF parameters. In addition, we consider two levels of non-linearity, namely, a low level of $\alpha_i \in [10, 16]$ and $\beta_i \in [1, 7]$ and a high level with $\alpha_i \in [16, 22]$ and $\beta_i \in [8, 14]$. We also constrain α_i and β_i to be integers. The resulting values of the UF parameters are shown in Tables 2 and 3.

4.1.3 BC-EMOA and iTDEA parameter settings. All parameters of BC-EMOA and iTDEA are set as suggested in the original papers [1, 9]. The initial and final territory for iTDEA are set to 0.1 and 0.00001, respectively, which was one of the alternatives suggested by its original authors. For the number of solutions presented to the DM in each interaction, the authors of iTDEA suggest $2m$, while the authors of BC-EMOA performed tests with 3 to 50 solutions. Considering the number of pairwise comparisons that a DM is asked to perform at each interaction, values larger than 5 would impose a high burden on the DM. Thus, we set this number to 5. BCEMOA asks the DM to make pairwise comparisons, which are then translated into a ranking. iTDEA ask the DM to select the best solution, which also needs to compare solutions pairwise. Thus the preference elicitation in both algorithms boils down to ranking. Population size in all experiments is 100. For the purposes of a fair comparison, the number of fitness evaluations is set to 80 000 for both algorithms. Since iTDEA evaluates only one solution per generation, and BC-EMOA creates a completely new population, 80 000 evaluations correspond to 80 000 generations in iTDEA and 800 generations in BC-EMOA. Finally, we perform experiments with 2, 4 and 6 interactions.

Each experiment is repeated 40 times with different random seeds. The algorithms and the Machine DM are implemented in Python 3.7.6, using the NSGA-II and benchmarks implementation provided by the Pygmo library 2.16.0 [2] and the SVM module provided by scikit-learn 0.23.1 (http://scikit-learn.org/). For our implementation of iTDEA in Python, we studied the C++ implementation provided by its original authors at https://bitbucket.org/ibrahimkarahan/itdea.

4.2 Results and Discussion

As a first step, we plot the solutions obtained by the iEMOAs on the bi-objective benchmark problems, together with the underlying UFs. For reasons of space, we focus on the results after 6 interactions. Each row in Figures 3 and 4 shows results for one benchmark problem, while each column corresponds to one of the DM types (Table 1). A first observation is that the different UF parameters lead to very different UF landscapes. Another general observation is that iTDEA very often fails to get close to the $\mathbf{z}^{\text{MPS}}$, especially for DTLZ7 (bottom row). BC-EMOA, on the other hand, is sometimes able to return solutions close to the most-preferred one, however, in many runs it returns solutions that are very far away from it, often in the opposite extreme of the PF, as can be seen in Tests 1, 3, 8 and 15 (Figure 3). These results seem to concur with our intuition that iEMOAs may struggle with the sigmoid UF.

We further evaluate the performance of the iEMOAs in two different ways. First, we plot the utility value of the final solution returned by each of the 40 independent repetitions of each experiment. These values are shown in Figures 5 and 6. In these figures, each row corresponds to one DM type (as defined in Table 1) and each column relates to one benchmark problem.

Second, following the original studies [1, 9], we measure the approximation error of the utility of the solution returned by the algorithms $U(\mathbf{z})$ to the utility of the most-preferred solution $U(\mathbf{z}^{\text{MPS}})$, calculated as

$$(U(\mathbf{z}) - U(\mathbf{z}^{\text{MPS}}))/(U(\mathbf{z}^{\text{W}}) - U(\mathbf{z}^{\text{MPS}})) \ , \quad (4)$$

where $U(\mathbf{z}^{\text{W}})$ is the utility of the worst PF solution. Thus, lower approximation error is desired. An approximation error greater than one indicates the returned solution is not on the PF. Mean and standard deviation of the approximation errors are shown in Table 4. For conciseness, we only show results for 2 and 6 interactions.

Looking at overall performance, the results with 6 interactions (Table 4) show that iTDEA performs best in tests 11 and 16 (DTLZ1 with two objectives, DM types 3 and 4), whereas BC-EMOA performs best (with both 2 and 6 interactions) in tests 4, 9, and 19 (DTLZ2 with 3 objectives), for high non-linearity, and tests 5, 10, 20 (DTLZ7 with two objectives) and 18, for low non-linearity. Thus, it seems the performance of BC-EMOA is acceptable on problem DTLZ2 with 3 objectives and DTLZ7 in all cases except DM type 3.

Overall, BC-EMOA produces the worse results on DTLZ1 with 3 objectives in all tests, while tests on DTLZ2 and DTLZ7 with 2 and 3 objectives seem easy for this algorithm, except for DTLZ7 with DM type 3. The boxplots (Figs. 5 and 6) corroborate these observations and also clearly show that the variance of the results produced by iTDEA is much larger, while the results of BC-EMOA are more consistent, even though they are not always better. In particular, the results of iTDEA are less consistent in DTLZ2 and DTLZ7, whereas BC-EMOA is more consistent on those two problems.

Looking at the effect of the number of interactions, we can confidently say that BC-EMOA does not perform better when increasing the number of interactions from 2 to 6 as can be verified by comparing the respective columns in Table 4. The improvements that are observed (e.g., tests 13, 14, and 19, for high non-linearity and tests 4, 13 and 14, for low non-linearity) can be attributed to extreme outliers (very poor runs) that happen more frequently with few interactions, as shown in the corresponding boxplots of those tests (Figures 5 and 6). The lack of improvement with higher number of interactions and the frequency of such extreme outliers suggests that BC-EMOA sometimes gets stuck in regions with poor utility values and it is not able to escape from them.

The performance of iTDEA typically gets better with more interactions, which can be seen by comparing columns 2 and 6 for iTDEA (Table 4), with a few exceptions (tests 5 and 14 for high non-linearity and 10 and 14, for low non-linearity), where the performance of iTDEA is anyway very poor.

Table 2: Parameter values of the sigmoid UF for tests with low values of α and β. Type correspond to the DM types in Table 1. n: dimension of the problem. m: dimension of the objective space. $U(\mathbf{z}^{\mathbf{MPS}})$: utility of $\mathbf{z}^{\mathbf{MPS}}$, $U(\mathbf{z}^{\mathbf{w}})$: utility of the worst PF solution. $\alpha, \beta, \tau, \lambda$ and w are the parameters of UF as laid out in Equation 2.

Test	Prob	n	m	Type	$U(\mathbf{z}^{\text{MPS}})$	$U(\mathbf{z}^{w})$	τ	α	β	λ	w
1	DTLZ1	6	2	1	0.33	0.49	0.33, 0.36	11, 15	1, 1	0.18, 0.11	0.48, 0.52
2	DTLZ1	7	3	1	0.09	0.38	0.38, 0.34, 0.37	13, 16, 16	1, 1, 1	0.21, 0.13, 0.22	0.4, 0.3, 0.29
3	DTLZ2	11	2	1	0.47	0.70	0.39, 0.4	15, 16	1, 5	0.12, 0.1	0.52, 0.48
4	DTLZ2	12	3	1	0.30	0.54	0.4, 0.39, 0.38	16, 16, 12	7, 3, 1	0.11, 0.1, 0.12	0.32, 0.33, 0.35
5	DTLZ7	21	2	1	0.12	0.91	0.38, 0.28	16, 16	5, 4	0.12, 0.33	0.12, 0.88
6	DTLZ1	6	2	2	0.46	0.73	0.4, 0.4	11, 16	1, 7	0.6, 0.6	0.53, 0.47
7	DTLZ1	7	3	2	0.27	0.54	0.39, 0.38, 0.38	16, 14, 15	1, 3, 1	0.78, 0.66, 0.67	0.33, 0.34, 0.33
8	DTLZ2	11	2	2	0.39	0.89	0.4, 0.37	11, 16	2, 1	0.62, 0.81	0.39, 0.61
9	DTLZ2	12	3	2	0.13	0.93	0.22, 0.28, 0.15	12, 15, 15	4, 2, 4	0.73, 0.77, 0.66	0.58, 0.29, 0.13
10	DTLZ7	21	2	2	0.48	0.91	0.21, 0.37	12, 14	4, 1	0.68, 0.65	0.49, 0.51
11	DTLZ1	6	2	3	0.01	0.55	0.83, 0.65	15, 12	5, 4	0.16, 0.29	0.42, 0.58
12	DTLZ1	7	3	3	0.00	0.34	0.89, 0.69, 0.85	11, 12, 11	7, 5, 2	0.36, 0.17, 0.25	0.44, 0.14, 0.41
13	DTLZ2	11	2	3	0.26	0.69	0.62, 0.6	16, 11	5, 3	0.1, 0.37	0.27, 0.73
14	DTLZ2	12	3	3	0.01	0.56	0.65, 0.67, 0.84	10, 12, 15	4, 5, 4	0.28, 0.38, 0.11	0.35, 0.59, 0.06
15	DTLZ7	21	2	3	0.01	0.81	0.89, 0.61	14, 10	7, 1	0.11, 0.38	0.13, 0.87
16	DTLZ1	6	2	4	0.00	0.76	0.88, 0.82	16, 11	4, 3	0.6, 0.69	0.19, 0.81
17	DTLZ1	7	3	4	0.02	0.45	0.63, 0.61, 0.61	13, 12, 13	7, 1, 1	0.75, 0.82, 0.89	0.45, 0.25, 0.3
18	DTLZ2	11	2	4	0.35	0.88	0.6, 0.66	12, 15	2, 2	0.8, 0.9	0.36, 0.64
19	DTLZ2	12	3	4	0.06	0.72	0.65, 0.63, 0.87	10, 10, 12	3, 2, 2	0.78, 0.71, 0.65	0.48, 0.45, 0.07
20	DTLZ7	21	2	4	0.13	0.88	0.61, 0.61	16, 10	5, 3	0.85, 0.79	0.13, 0.87

Table 3: Parameter values of the UF for tests with large α and β. For the meaning of the columns see the caption of Table 2.

Test	Prob	n	m	Type	$U(\mathbf{z}^{\text{MPS}})$	$U(\mathbf{z}^{w})$	τ	α	β	λ	w
1	DTLZ1	6	2	1	0.48	0.69	0.38, 0.4	19, 19	10, 13	0.21, 0.14	0.5, 0.5
2	DTLZ1	7	3	1	0.09	0.51	0.33, 0.38, 0.37	16, 16, 22	8, 8, 10	0.11, 0.33, 0.22	0.35, 0.3, 0.35
3	DTLZ2	11	2	1	0.46	0.99	0.34, 0.37	19, 17	12, 14	0.11, 0.13	0.46, 0.54
4	DTLZ2	12	3	1	0.07	0.99	0.23, 0.18, 0.19	19, 19, 20	13, 14, 11	0.21, 0.25, 0.15	0.64, 0.29, 0.07
5	DTLZ7	21	2	1	0.49	0.99	0.3, 0.31	16, 18	9, 14	0.31, 0.37	0.49, 0.51
6	DTLZ1	6	2	2	0.50	0.81	0.4, 0.4	18, 22	8, 12	0.6, 0.6	0.5, 0.5
7	DTLZ1	7	3	2	0.17	0.60	0.4, 0.39, 0.39	17, 22, 19	11, 12, 8	0.71, 0.86, 0.64	0.34, 0.3, 0.36
8	DTLZ2	11	2	2	0.45	1.00	0.13, 0.18	16, 16	13, 14	0.64, 0.74	0.45, 0.55
9	DTLZ2	12	3	2	0.21	0.99	0.36, 0.14, 0.16	17, 21, 22	8, 14, 8	0.9, 0.86, 0.7	0.32, 0.47, 0.21
10	DTLZ7	21	2	2	0.35	0.99	0.39, 0.4	16, 21	11, 10	0.71, 0.87	0.35, 0.65
11	DTLZ1	6	2	3	0.00	0.55	0.89, 0.63	17, 16	14, 8	0.36, 0.35	0.44, 0.56
12	DTLZ1	7	3	3	0.00	0.45	0.85, 0.74, 0.69	16, 19, 21	13, 8, 11	0.26, 0.23, 0.32	0.51, 0.34, 0.15
13	DTLZ2	11	2	3	0.41	0.67	0.6, 0.63	20, 16	14, 11	0.25, 0.39	0.41, 0.59
14	DTLZ2	12	3	3	0.00	0.62	0.61, 0.61, 0.9	16, 16, 19	9, 8, 11	0.36, 0.17, 0.12	0.31, 0.63, 0.06
15	DTLZ7	21	2	3	0.01	0.88	0.9, 0.6	20, 16	14, 13	0.12, 0.38	0.12, 0.88
16	DTLZ1	6	2	4	0.01	0.84	0.79, 0.62	18, 17	12, 14	0.72, 0.87	0.16, 0.84
17	DTLZ1	7	3	4	0.00	0.49	0.65, 0.63, 0.68	20, 17, 19	10, 12, 14	0.61, 0.77, 0.73	0.4, 0.12, 0.49
18	DTLZ2	11	2	4	0.27	0.87	0.62, 0.67	16, 21	11, 13	0.85, 0.76	0.27, 0.73
19	DTLZ2	12	3	4	0.02	0.90	0.61, 0.61, 0.9	16, 16, 21	8, 12, 14	0.89, 0.85, 0.66	0.47, 0.48, 0.04
20	DTLZ7	21	2	4	0.10	0.95	0.61, 0.61	22, 16	14, 12	0.88, 0.82	0.1, 0.9

Somewhat surprisingly, there is no clear effect of the non-linearity of the UF on the performance of the algorithms. In particular, non-linearity seems to have little effect for BC-EMOA, as can be seen by comparing the two sides of Table 4 or the boxplots in Fig. 5 versus Fig. 6. The only clear exceptions are tests 1 (top-left plot in Figs. 5 and 6), 4 and 6, where BC-EMOA performs clearly worse with low non-linearity, which is rather unexpected. As for iTDEA, its results sometimes get better and sometimes worse when comparing the

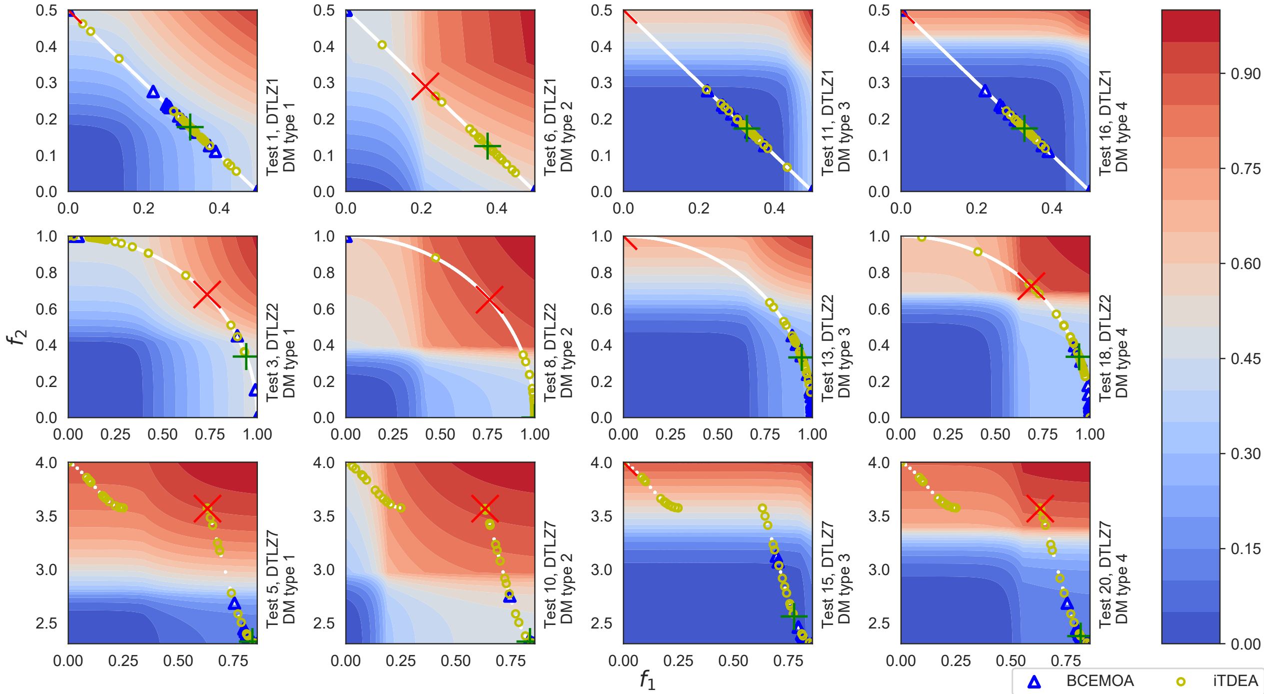

Figure 3: Distribution of final solutions returned by 40 runs of both algorithms with 6 interactions on 2-objective benchmark problems. The contour lines show the value of sigmoid UFs (with low values of α and β). The PF approximation is shown as white points. The solutions with worst and best utility values on the PF are depicted by red and green markers, respectively.

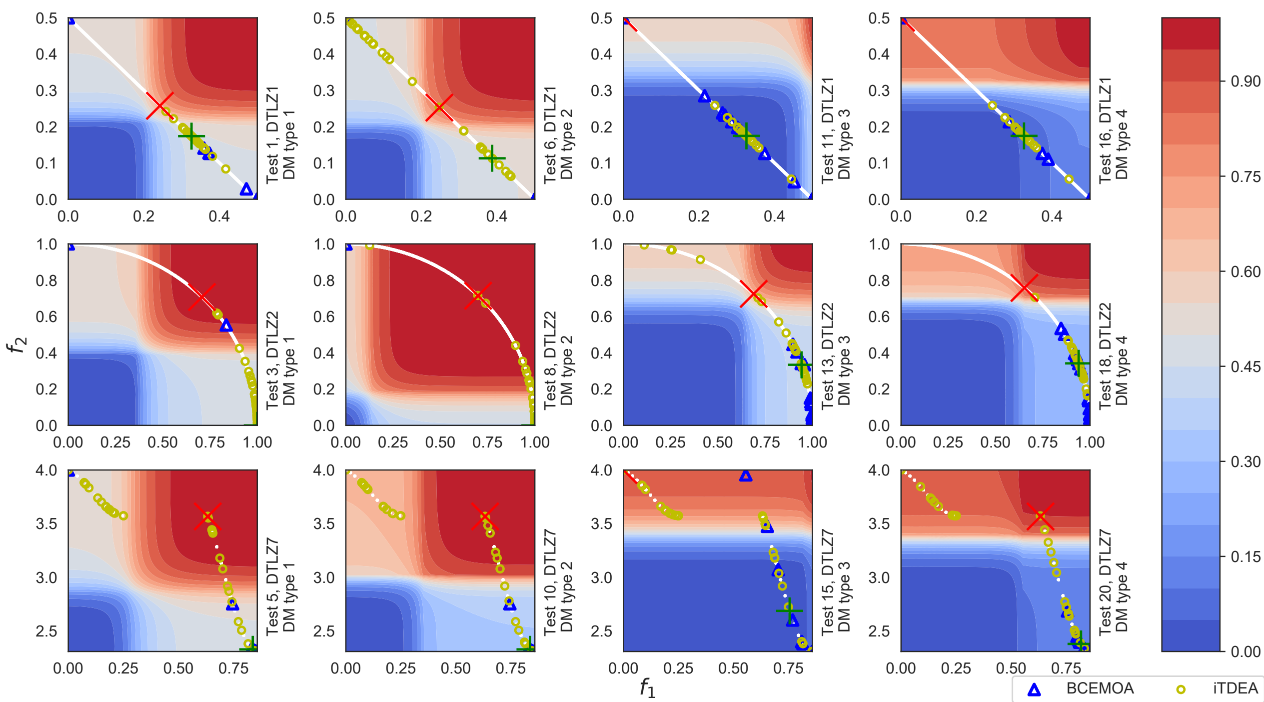

Figure 4: Distribution of final results for tests with high values of α and β. See caption of Figure 3 above for details.

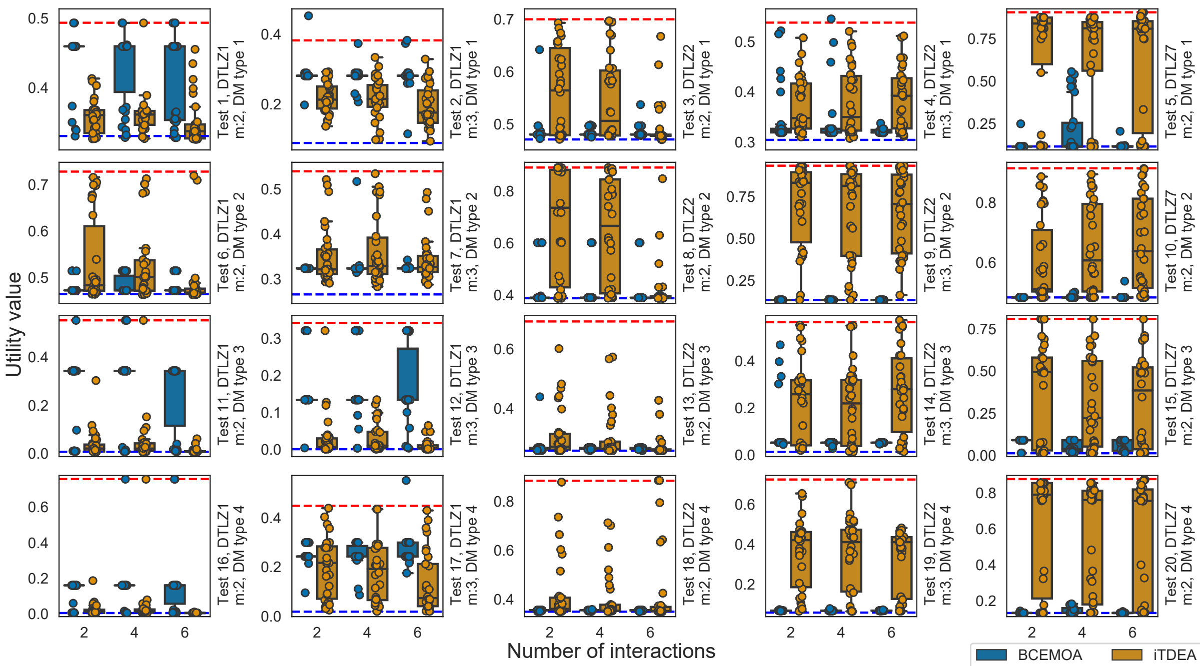

Figure 5: Comparison of the performance of the two algorithms for different tests with low α and β values of the sigmoid UF. The utility values are averaged over 40 runs. The plots include results for 2, 4 and 6 interactions. Red and blue dash-lines indicate the worst and best utility values for PF solutions.

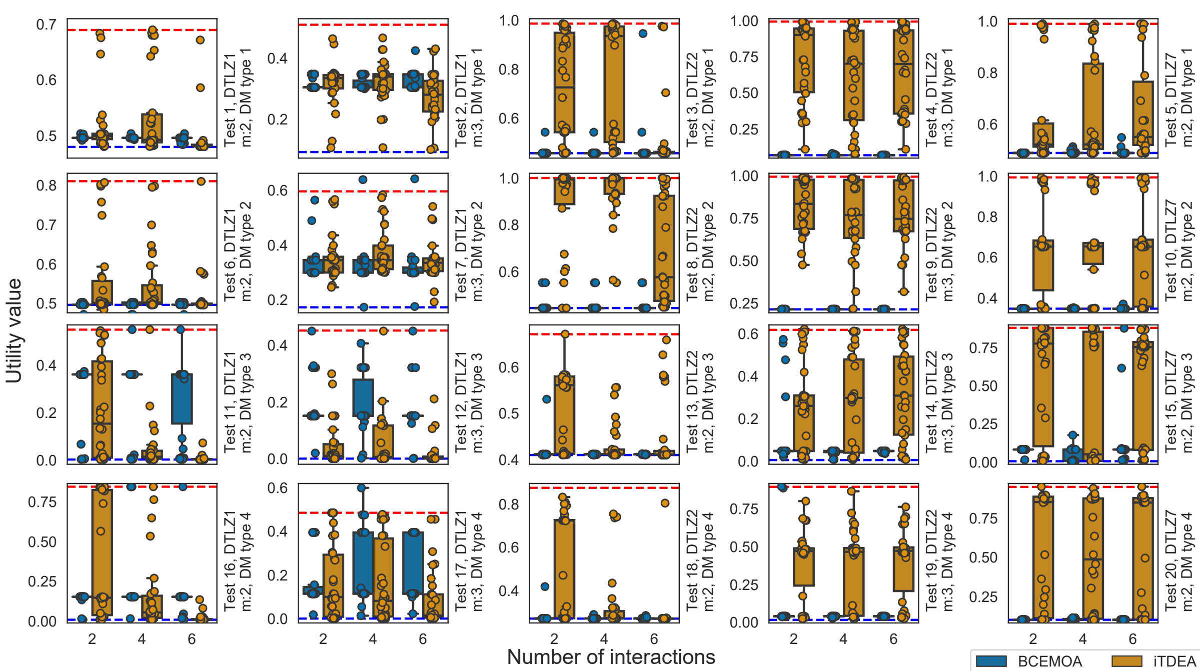

Figure 6: Comparison of the performance of the two algorithms for different tests with high values of α and β. See caption of Figure 5 above for details.

Table 4: Approximation error (Eq. 4) statistics for BC-EMOA and iTDEA. The mean (and standard deviation in parenthesis) over 40 independent runs is shown for each test. Minimum mean errors in each row are highlighted in bold font. Differences in performance of the algorithms are significant (Wilcoxon test, p-value < 0.05) unless indicated by an asterisk (*).

				High values of α and β				Low values of α and β			
			DM	2 interactions		6 interactions		2 interactions		6 interactions	
Test	**Problem**	m	**type**	BC-EMOA	iTDEA	BC-EMOA	iTDEA	BC-EMOA	iTDEA	BC-EMOA	iTDEA
1	DTLZ1	2	1	0.079(0.01)*	0.173(0.255)*	0.086(0.071)	**0.057(0.186)**	0.69(0.276)	0.167(0.15)	0.56(0.361)	0.115(0.211)
2	DTLZ1	3	1	0.531(0.043)*	0.537(0.17)*	0.555(0.067)	0.433(0.184)	0.669(0.12)*	0.44(0.147)*	0.658(0.138)	**0.361(0.205)**
3	DTLZ2	2	1	**0.006(0.03)**	0.495(0.404)	0.036(0.17)	0.087(0.257)	0.069(0.13)	0.394(0.357)	0.057(0.049)	0.104(0.191)
4	DTLZ2	3	1	**0.0(0.0)**	0.709(0.295)	**0.0(0.0)**	0.629(0.313)	0.175(0.242)	0.302(0.235)	0.074(0.018)	0.35(0.261)
5	DTLZ7	2	1	**0.0(0.0)**	0.275(0.391)	0.006(0.023)	0.328(0.384)	0.006(0.031)	0.699(0.392)	0.005(0.021)	0.647(0.411)
6	DTLZ1	2	2	**0.004(0.006)**	0.187(0.334)	0.006(0.008)*	0.064(0.193)*	0.047(0.049)*	0.267(0.356)*	0.057(0.06)	0.086(0.237)
7	DTLZ1	3	2	0.364(0.14)*	0.424(0.19)*	0.341(0.163)	0.397(0.164)	**0.212(0.001)***	0.309(0.233)*	0.215(0.013)*	0.281(0.181)*
8	DTLZ2	2	2	**0.013(0.049)**	0.793(0.344)	**0.013(0.049)**	0.43(0.402)	0.047(0.128)*	0.585(0.429)*	0.046(0.128)	0.065(0.188)
9	DTLZ2	3	2	**0.0(0.0)**	0.786(0.198)	**0.0(0.0)**	0.71(0.235)	**0.0(0.0)**	0.706(0.305)	**0.0(0.0)**	0.657(0.295)
10	DTLZ7	2	2	**0.0(0.0)**	0.467(0.345)	0.001(0.006)	0.449(0.361)	0.001(0.0)	0.277(0.325)	0.004(0.023)	0.426(0.369)
11	DTLZ1	2	3	0.53(0.258)*	0.383(0.379)*	0.5(0.293)	**0.007(0.024)**	0.511(0.255)	0.057(0.104)	0.455(0.271)	0.009(0.021)
12	DTLZ1	3	3	0.41(0.192)	0.102(0.176)	0.382(0.2)	0.039(0.102)	0.451(0.207)	0.09(0.186)	0.488(0.299)	**0.027(0.046)**
13	DTLZ2	2	3	0.02(0.083)	0.403(0.318)	**0.005(0.002)***	0.153(0.298)*	0.027(0.074)	0.129(0.2)	0.015(0.007)	0.031(0.086)
14	DTLZ2	3	3	0.161(0.252)	0.416(0.326)	**0.067(0.005)**	0.505(0.351)	0.149(0.207)	0.399(0.309)	0.07(0.003)	0.469(0.331)
15	DTLZ7	2	3	0.085(0.014)	0.647(0.418)	0.126(0.202)	0.6(0.406)	0.091(0.024)	0.447(0.371)	**0.063(0.04)**	0.381(0.337)
16	DTLZ1	2	4	0.161(0.043)*	0.402(0.432)*	0.18(0.164)	0.011(0.031)	0.161(0.085)	0.025(0.046)	0.206(0.233)	**0.001(0.002)**
17	DTLZ1	3	4	0.367(0.254)*	0.339(0.35)*	0.42(0.285)	**0.177(0.272)**	0.534(0.087)	0.416(0.287)	0.571(0.145)*	0.276(0.287)*
18	DTLZ2	2	4	0.009(0.045)	0.465(0.387)	**0.002(0.005)**	0.03(0.161)	0.01(0.003)*	0.142(0.25)*	0.01(0.003)	0.141(0.306)
19	DTLZ2	3	4	0.092(0.246)	0.433(0.26)	0.027(0.001)	0.437(0.259)	**0.015(0.002)**	0.438(0.26)	**0.015(0.001)**	0.396(0.237)
20	DTLZ7	2	4	**0.0(0.0)**	0.538(0.432)	**0.0(0.0)**	0.5(0.445)	0.001(0.003)	0.637(0.42)	0.001(0.002)	0.537(0.438)

same test with low versus high non-linearity. Hence, no general pattern can be observed.

Lastly, the layout of Figures 5 and 6 allows us to observe that the behaviour of the iEMOAs changes across DM types for the same problem (i.e., across rows for the same column), thus the type of DM does have an impact on the behaviour of the algorithms. However, it is not possible to see a clear pattern of the DM type that repeats across problems (i.e., across columns for the same row), thus each DM type seems to have a different effect for each problem.

5 CONCLUSION

The quantitative assessment of iEMOAs is challenging because of the dual problems that EMOAs are stochastic, necessitating repeated experiments, and that human DMs add further variability which needs careful controlling. More importantly, human DMs are very difficult to work with in large studies because they suffer from fatigue, making them behave differently to how they might behave in a realistic single-shot environment. These challenges have motivated researchers to simulate the interaction with the human DM by replacing them with a utility function. The Machine DM framework introduced in [10] goes further, enabling various realistic behaviours of DMs to be simulated. However, to the best of our knowledge, a systematic statistical comparison of different iEMOAs has not been attempted until this paper. While all the previous statistical studies on the performance of iEMOAs have used linear or arbitrary non-linear UFs to simulate a real DM in their experiments, we propose here to use a sigmoidal UF that can be easily tuned to mimic different decision-making attitudes and behaviours. The sigmoidal UF has been widely accepted as realistic in the MCDM [17] and behavioural economics literature [8].

The experiment detailed here strongly suggests that at least some state-of-the-art iEMOAs do not perform as expected or desired under this more realistic UF. In particular, each of the iEMOAs evaluated show different but similarly underwhelming behaviors: iTDEA results lack consistency, whereas BC-EMOA sometimes gets stuck in poor utility regions. The various types of DM behaviour that can be expressed by this UF do have an influence on the results, yet no clear pattern has emerged from our experiments. On the other hand, and somewhat surprisingly, the two levels of non-linearity tested did not seem to have a clear effect.

Although the experiments performed here may appear particularly challenging, this study is just a first step towards the statistical analysis of iEMOAs under realistic DM interaction scenarios. Real DMs are even more complex and manifest additional biases and other non-idealities [17]. By adopting the machine DM framework [10], we plan to extend our comparison to assess the impact of such non-idealities. This study opens the door to further comparisons of iEMOAs under such realistic (and challenging) conditions, hopefully motivating the proposal of new algorithms able to overcome these challenges. To motivate further research, we make our code publicly available at doi:10.5281/zenodo.4501313 .

ACKNOWLEDGMENTS

M. López-Ibáñez is a "Beatriz Galindo" Senior Distinguished Researcher (BEAGAL 18/00053) funded by the Ministry of Science and Innovation of the Spanish Government.

REFERENCES

[1] R. Battiti and A. Passerini. 2010. Brain-Computer Evolutionary Multiobjective Optimization: A Genetic Algorithm Adapting to the Decision Maker. *IEEE Transactions on Evolutionary Computation* 14, 5 (2010), 671–687. https://doi.org/10.1109/TEVC.2010.2058118

[2] F. Biscani, D. Izzo, and C. H. Yam. 2010. A Global Optimisation Toolbox for Massively Parallel Engineering Optimisation. *Arxiv preprint arXiv:1004.3824* (2010). http://arxiv.org/abs/1004.3824

[3] J. Branke, S. Corrente, S. Greco, R. Słowiński, and P. Zielniewicz. 2016. Using Choquet integral as preference model in interactive evolutionary multiobjective optimization. *European Journal of Operational Research* 250, 3 (2016), 884–901. https://doi.org/10.1016/j.ejor.2015.10.027

[4] J. Branke, S. Greco, R. Słowiński, and P. Zielniewicz. 2010. Interactive evolutionary multiobjective optimization driven by robust ordinal regression. *Bulletin of the Polish Academy of Sciences: Technical Sciences* 58, 3 (2010), 347–358. https://doi.org/10.2478/v10175-010-0033-3

[5] J. T. Buchanan. 1994. An experimental evaluation of interactive MCDM methods and the decision making process. *Journal of the Operational Research Society* 45, 9 (1994), 1050–1059.

[6] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. 2005. Scalable Test Problems for Evolutionary Multiobjective Optimization. In *Evolutionary Multiobjective Optimization*, A. Abraham et al. (Eds.). Springer, London, UK, 105–145.

[7] Z. Gong, N. Zhang, and F. Chiclana. 2018. The optimization ordering model for intuitionistic fuzzy preference relations with utility functions. *Knowledge Based Systems* 162 (2018), 174–184. https://doi.org/10.1016/j.knosys.2018.07.012

[8] D. Kahneman and A. Tversky. 1979. Prospect theory: An analysis of decision under risk. *Econometrica* 47, 2 (1979), 263–291. https://doi.org/10.2307/1914185

[9] M. Köksalan and İ. Karahan. 2010. An Interactive Territory Defining Evolutionary Algorithm: iTDEA. *IEEE Transactions on Evolutionary Computation* 14, 5 (Oct. 2010), 702–722. https://doi.org/10.1109/TEVC.2010.2070070

[10] M. López-Ibáñez and J. D. Knowles. 2015. Machine Decision Makers as a Laboratory for Interactive EMO. In *Evolutionary Multi-criterion Optimization, EMO 2015 Part II*, A. Gaspar-Cunha et al. (Eds.). Lecture Notes in Computer Science, Vol. 9019. Springer, Heidelberg, Germany, 295–309. https://doi.org/10.1007/978-3-319-15892-1_20

[11] A. Mas-Colell, M. D. Whinston, and J. R. Green. 1995. *Microeconomic theory*. Vol. 1. Oxford university press, New York, New York.

[12] K. Miettinen. 1999. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers. 298 pages.

[13] J. Mote, D. L. Olson, and M. A. Venkataramanan. 1988. A comparative multiobjective programming study. *Mathematical and Computer Modelling* 10, 10 (1988), 719–729. https://doi.org/10.1016/0895-7177(88)90085-4

[14] V. Ojalehto, D. Podkopaev, and K. Miettinen. 2016. Towards Automatic Testing of Reference Point Based Interactive Methods. In *Proceedings of PPSN XIV, 14th International Conference on Parallel Problem Solving from Nature*, J. Handl et al. (Eds.). Lecture Notes in Computer Science, Vol. 9921. Springer, Heidelberg, Germany, 483–492. https://doi.org/10.1007/978-3-319-45823-6_45

[15] G. R. Reeves and J. J. Gonzalez. 1989. A comparison of two interactive MCDM procedures. *European Journal of Operational Research* 41, 2 (1989), 203–209. https://doi.org/10.1016/0377-2217(89)90385-8

[16] R. E. Steuer and L. Gardiner. 1991. On the Computational Testing of Procedures for Interactive Multiple Objective Linear Programming. In *Operations Research*, G. Fandel and H. Gehring (Eds.). Springer, Berlin, Heidelberg, 121–131. https://doi.org/10.1007/978-3-642-76537-7_8

[17] T. J. Stewart. 1996. Robustness of Additive Value Function Methods in MCDM. *Journal of Multi-Criteria Decision Analysis* 5, 4 (1996), 301–309.

[18] T. J. Stewart. 2005. Goal programming and cognitive biases in decision-making. *Journal of the Operational Research Society* 56, 10 (2005), 1166–1175. https://doi.org/10.1057/palgrave.jors.2601948

[19] M. K. Tomczyk and M. Kadzinski. 2019. Decomposition-based interactive evolutionary algorithm for multiple objective optimization. *IEEE Transactions on Evolutionary Computation* 24, 2 (2019), 320–334. https://doi.org/10.1109/TEVC.2019.2915767

[20] M. K. Tomczyk and M. Kadziński. 2021. Decomposition-based co-evolutionary algorithm for interactive multiple objective optimization. *Information Sciences* 549 (2021), 178–199. https://doi.org/10.1016/j.ins.2020.11.030

[21] B. Xin, L. Chen, J. Chen, H. Ishibuchi, K. Hirota, and B. Liu. 2018. Interactive Multiobjective Optimization: A Review of the State-of-the-Art. *IEEE Access* 6 (2018), 41256–41279. https://doi.org/10.1109/ACCESS.2018.2856832

[22] A. Zujevs and J. Eiduks. 2011. New decision maker model for multiobjective optimization interactive methods. In *17th International Conference on Information and Software Technologies, Kaunas, Lithuania*. 51–58.