This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2023.3269348

Detecting Hidden and Irrelevant Objectives in Interactive Multi-Objective Optimization

Seyed Mahdi Shavarani, Manuel López-Ibáñez, *Member, IEEE*, and Richard Allmendinger, *Senior Member, IEEE*

Alliance Manchester Business School, University of Manchester, Manchester, M13 9PL, UK
Email: {seyedmahdi.shavarani,manuel.lopez-ibanez, richard.allmendinger}@manchester.ac.uk

***Abstract*—Evolutionary multi-objective optimization algorithms (EMOAs) typically assume that all objectives that are relevant to the decision-maker (DM) are optimized by the EMOA. In some scenarios, however, there are *irrelevant* objectives that are optimized by the EMOA but ignored by the DM, as well as, *hidden* objectives that the DM considers when judging the utility of solutions but are not optimized. This discrepancy between the EMOA and the DM's preferences may impede the search for the most-preferred solution and waste resources evaluating irrelevant objectives. Research on objective reduction has focused so far on the structure of the problem and correlations between objectives and neglected the role of the DM. We formally define here the concepts of irrelevant and hidden objectives and propose methods for detecting them, based on uni-variate feature selection and recursive feature elimination, that use the preferences already elicited when a DM interacts with a ranking-based interactive EMOA (iEMOA). We incorporate the detection methods into an iEMOA capable of dynamically switching the objectives being optimized. Our experiments show that this approach can efficiently identify which objectives are relevant to the DM and reduce the number of objectives being optimized, while keeping and often improving the utility, according to the DM, of the best solution found.**

***Index Terms*—Interactive Multi-Objective Optimization, Hidden Objectives, Irrelevant Objectives, Machine Learning, Dimension Reduction, Feature Selection**

I. Introduction

In many real-world optimization problems, there are tens of numerical features of a candidate solution that could, in principle, be optimized by means of an Evolutionary Multi-Objective Algorithm (EMOA) and it is often tempting to model as many objectives as possible [1]. However, the runtime of various steps within an EMOA increases with the number of objectives [2, 3] and, in the case of hypervolume-based methods, this increase is exponential [4]. Moreover, the fraction of solutions that are Pareto-optimal increases exponentially with the number of objectives [5, 6], which complicates the *a posteriori* decision-making phase [7].

Previous research on objective reduction considered removing objectives that are highly correlated to other objectives [1, 7] or do not significantly alter the dominance relations among solutions [8, 9]. However, regardless of the structure of the problem, some of the objectives may not be relevant to the DM and they can be removed from the optimization model. Interactive EMOAs (iEMOAs) [10] iteratively elicit and exploit preference information of a decision maker (DM) to guide the optimizer towards preferred solutions. It is possible to exploit the elicited information to identify and remove objectives that are not relevant to the DM.

Such a scenario can happen if this particular DM was not consulted during the modeling phase or her preferences changed during the optimization due to learning [11] and "preference drift" [12]. Thus, there may be objectives that are optimized by the iEMOA but are *irrelevant* to the DM. In other cases, the DM's preferences may depend on both the value of the objectives being optimized and the value of other numerical features that are measured by the system and observed by the DM but are not optimized. Such features are said to be *hidden* from the optimizer and would lead to results that are not satisfactory to the DM if not considered by the optimizer [13, 14]. Stewart [15] discusses the concept of "unmodelled" criteria, which appear in the DM's internal utility function but are missing from the preference model.

This discrepancy may make the elicited preferences seem non-rational, e.g. when a solution that is dominated with respect to the modelled objectives is preferred by the DM over a non-dominated one because the former is better than the latter with respect to features that are not optimized as objectives. Let us consider a simplified example inspired by the real-world problem discussed by Ramos-Pérez et al. [16] of planning school lunches in terms of not only cost, but also a number of metrics of nutritional value and food variety. Imagine two candidate menu plans with three features $\mathbf{z}_1 = (7, 4, 2)$ and $\mathbf{z}_2 = (7, 3, 6)$, where each feature is total cost, the total amount of calcium, and the variety of vegetables, respectively. Further assume that the iEMOA was designed to optimize only the first two features as objectives, thus, $\mathbf{z}_1$ dominates $\mathbf{z}_2$, i.e., $\mathbf{z}_1$ is not worse in cost and is better in amount of Calcium than $\mathbf{z}_2$. However, a DM (e.g., the nutritionist of a particular school who is aware that children in this school already have a diet rich in Calcium outside school but struggle to eat their vegetables) may not look at the second objective (Calcium) and instead wishes to maximize the third feature (vegetables variety), thus prefer $\mathbf{z}_2$ over $\mathbf{z}_1$.

One may argue that the system should allow the DM to specify which features must be optimized. But in practice a DM may only realize the relevance of a feature during the optimization process [12]. Moreover, the DM may not be conscious of the particular features that are guiding her decisions, e.g., the nutritionist may directly compare the composition of the menus instead of looking at any summary metrics provided by the iEMOA. In some problems a DM may be able to judge solutions according to qualitative aspects (e.g., the quality of the behavior of a robot performing a task) without being aware that there exist numerical features (e.g., the number of turns or minimum distance to walls) that could act as "proxy" objectives [17] to that aspect.

In the above scenarios, it would be desirable if an iEMOA could (1) detect the discrepancy between the objectives being optimized and the features that influence the preferences provided by the DM at each interaction; and (2) dynamically select the features that are optimized after each interaction. In the case of *irrelevant* objectives, removing them from the optimization phase increases the efficiency of any EMOA (as fewer objectives are optimized) and may also help to find better solutions [9].

To the best of our knowledge, for the first time in the literature we consider the preference information collected by an iEMOA when interacting with the DM as an opportunity to detect irrelevant objectives and remove them during the optimization. In addition, our proposal is also able to detect numerical features that are measured but not optimized by the iEMOA before the interaction, but are correlated with the DM's preferences. We propose that, after each interaction, the iEMOA dynamically activates the optimization of such *hidden* objectives, thus adapting the search to the preferences of the DM. By removing irrelevant objectives and optimizing hidden ones, an iEMOA is able to adapt to a diverse range of DMs and preference changes during optimization without requiring the simultaneous optimization of every potential objective.

Our main contributions can be summarized as follows:

- The formal definition of *irrelevant* and *hidden* objectives in interactive multi-objective optimization.
- A method to detect irrelevant and hidden objectives from the ranking information provided by a DM and to dynamically update the set of objectives after each interaction. The approach draws on feature selection methods and can be applied to any ranking-based iEMOA.
- A benchmarking approach that simulates irrelevant and hidden objectives using classical multi-objective problems. This approach is demonstrated for DTLZ problems [18] and multi-objective NK-landscapes ρMNKs [19].
- The empirical validation of the proposed detection method using:
  i. Problems of varying dimensionality, complexity, and Pareto front structure.
  ii. Different utility functions that simulate different DMs.
  iii. Different feature selection methods to detect relevant objectives.
- A sensitivity analysis to understand the performance impact of key parameters of the proposed approach.

Experimental results show that the proposed method can almost always replace irrelevant objectives with relevant ones quickly and significantly improve the utility of the solutions found.

The rest of the paper is organized as follows. Several fundamental concepts on which this work is based are defined in Section II. A summarized background on previous efforts towards objective reduction is given in Section III. In Section IV, the proposed method and several variants of it are elaborated in detail. The experimental setup is laid out in Section V. The results of the experiments are discussed in Section VI. Finally Section VII provides conclusions and future research directions.

II. Definitions

Let us consider an optimization problem with n decision variables, where, given a solution vector $\mathbf{x} = (x_1, \ldots, x_n)$ from the feasible decision space $\mathcal{X}$, we can compute a set $F = \{f_1, \ldots, f_m\}$ of m numerical *features*, $f_i \colon \mathcal{X} \to \mathbb{R}$. In principle, all these features could be optimized as objectives. In the following, we assume minimization without loss of generality.

Definition II.1 (Potential objectives)**.** All m features in F are called *potential* objectives.

Let us assume as well that, either due to decisions made during the modeling phase or efficiency reasons, only a subset $\hat{F} \subseteq F$ ($\hat{m} = |\hat{F}|$) of the potential objectives must be minimized as optimization objectives, resulting in the following multi-objective optimization problem:

$$\begin{aligned} \text{Minimize} \quad & \left(f_1(\mathbf{x}), \ldots, f_{\hat{m}}(\mathbf{x})\right) \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X} \end{aligned} \tag{1}$$

where $f_i \in \hat{F}$ are the objectives minimized by the optimization method, while $F \setminus \hat{F}$ are not.

Definition II.2 (Active objective)**.** An objective is *active* if it must be optimized by the optimization method. The set of active objectives is denoted by $\hat{F}$. Inactive objectives ($F \setminus \hat{F}$) are either evaluated but ignored by the optimization method or not evaluated at all.

In the EMOA literature, computational cost is often measured in terms of *solution evaluations*, where each *solution evaluation* usually means the evaluation of all its (active) objectives. Here we will consider *objective evaluations* instead because different solutions may be evaluated for different subsets of of objectives, and these subsets may also vary in cardinality.

Definition II.3 (Objective evaluation)**.** The evaluation of any of the objectives f_i corresponding to a solution $\mathbf{x}$ is counted as one objective evaluation. Thus, the cost of a solution evaluation is $\hat{m}$ objective evaluations.

Although most EMOAs assume that the set of active of objectives is decided in the modeling phase and remains constant, it is possible to change the set of active objectives during the optimization by choosing any subset of potential objectives, as we will show later. When solving the above problem in terms of Pareto optimality, an EMOA only considers active objectives.

Definition II.4 (Dominance and non-dominance [20])**.** Given two solutions $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, we say that $\mathbf{x}$ *dominates* $\mathbf{y}$ if the former is not worse than the latter in any objective and it is strictly better in at least one, i.e., $\forall f_i \in \hat{F}$, $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ and $\exists f_j \in \hat{F}$, $f_j(\mathbf{x}) < f_j(\mathbf{y})$. When $\mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\mathbf{y})$ and $\mathbf{x}$ does not dominate $\mathbf{y}$ nor $\mathbf{y}$ dominates $\mathbf{x}$, we say that they are (mutually) *non-dominated*.

Definition II.5 (Pareto optimal [20])**.** A feasible solution $\mathbf{x} \in \mathcal{X}$ is called Pareto optimal if there is no $\mathbf{y} \in \mathcal{X}$ that dominates it. The set of (mutually non-dominated) Pareto optimal solutions is the *Pareto set*.

Definition II.6 (Pareto front [20])**.** The image of the Pareto set on the objective space defined by $\hat{F}$ is known as the Pareto front (PF).

Definition II.7 (Redundant objectives [21])**.** An objective is called *redundant* if it can be removed from the set of active objectives without changing the set of Pareto optimal solutions. Saxena et al. [22] extend this definition to include objectives that are not conflicting with a non-redundant objective.

The above definitions are independent of the preferences of a human DM interacting with an EMOA. In the case of interactive EMOAs (iEMOAs), the DM provides preference information, e.g., by ranking a subset of solutions, to guide the algorithm towards the DM's most preferred solution. Let us assume the DM can observe the value of all potential objectives when comparing solutions. For reasons explained in the introduction, there may exist a discrepancy between the active objectives being optimized by the iEMOA and the objectives considered by the DM when comparing solutions.

We can formally define this discrepancy in the case of non-ad-hoc interactive methods, which assume there exists a utility function (UF) guiding the DM's decisions but unknown to the iEMOA [23–27]. Ad-hoc methods assume that no such UF exists [23]. Due to the popularity of UFs in modeling preferences, the vast majority of iEMOAs are non-ad-hoc methods, thus we focus on them in the remainder of the paper. Without loss of generality, we assume an UF of the form $U\colon \mathbb{R}^m \to \mathbb{R}$, whose input is the vector-valued function $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ with components being the set F of potential objectives. Although U receives as input the value of all potential objective functions, it may not use all those values to calculate its output.

Definition II.8 (Irrelevant objectives)**.** An objective $f_i \in F$ is called *irrelevant* if its value does not affect the value of the DM's UF. That is, any two solutions $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ with the same value in all potential objectives except f_i should also have the same utility value, i.e., $f_j(\mathbf{x}) = f_j(\mathbf{y}), \forall f_j \in F \setminus \{f_i\} \Rightarrow U(\mathbf{f}(\mathbf{x})) = U(\mathbf{f}(\mathbf{y}))$.

Hereafter, $F_{\text{DM}} \subseteq F$ denotes the set of objective functions relevant to the DM, thus the set of irrelevant objectives is given by $F \setminus F_{\text{DM}}$.

Definition II.9 (Hidden objectives)**.** An objective $f_i \in F$ is *hidden* if it is relevant but not (currently) active, i.e., $f_i \in F_{\text{DM}} \wedge f_i \notin \hat{F}$.

Hidden objectives may confuse the iEMOA, since the interaction with the DM may be consistent with the dominance criterion for the objectives in F_{DM} but not for the objectives in $\hat{F}$. If $F_{\text{DM}} \subset \hat{F}$, then no hidden objectives exist, but the iEMOA is optimizing some irrelevant objectives, which makes the problem more challenging for the iEMOA and is wasteful if the evaluation of those objectives is expensive. Similarly, if $F_{\text{DM}} = \hat{F}$, then neither hidden nor irrelevant objectives exist, and the iEMOA is optimizing precisely the objectives that the DM cares about.

In the rest of the paper, when considering benchmark problems and known UFs, we will assume for simplicity that irrelevant objectives are not a (trivial) function of relevant ones nor vice versa, so that the set of relevant objectives F_{DM}, and, hence, irrelevant and hidden ones, can be inferred from the definition of the UF. In practice, the DM's UF is unknown and, in the case of black-box optimization, we may not know whether an objective is a function of other objectives, thus an objective is considered irrelevant if its value does not seem to influence the DM's decisions.

From the above definitions, it can be concluded that while *redundant* objectives are determined based on the structure of the problem, *irrelevant* and *hidden* objectives are defined from the DM's perspective. An irrelevant objective may be redundant or not, however, a redundant objective cannot be relevant unless the DM's preferences are somehow inconsistent with Pareto optimality. On the other hand, a redundant objective may *appear to be* relevant if it is correlated with a relevant objective. While there are studies on the detection and elimination of redundant objectives, which we review in the next section, there is no prior research on the identification of irrelevant and hidden objectives to the best of our knowledge. Our focus here is to fill this gap and we propose a method to tackle it in Section IV.

III. Background and Literature Review

We have carried out a thorough literature review of methods for reducing the number of objectives, which we briefly summarize here. Many of these studies use the term *dimension* reduction to refer to the same concept. However, to avoid confusion with methods that reduce the number of decision variables [28], we use the term "objective reduction".

Most of the studies on objective reduction focus on selecting *a priori* a subset of objectives to facilitate the optimization process while preserving Pareto optimal solutions as much as possible. Early proposals [21, 29] make strict assumptions about the problem structure that are impossible to meet for real-life problems. Recent approaches [6, 8] sacrifice to exactly identify the correct subset of objectives and capture the entire PF in order to increase applicability.

Other approaches identify similar objectives and recombine them into a single one. For example, harmonic levels [30] and aggregation trees [31] are used to identify harmonious objectives (improvement of one objective does not lead to deterioration of the others). In particular, aggregation trees are used *a posteriori* for facilitating the decision-making phase. Similarly, principal component analysis (PCA) has been used

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2023.3269348

to identify correlated objectives that may be combined into a single objective *a posteriori* to facilitate decision-making [32] or during the optimization process to increase computational efficiency [33]. However, Costa and Oliveira [34] have shown that objectives that are deemed redundant by PCA may be "informative", i.e., contain trade-off information that would be lost if omitted.

Finally, projection methods map all objectives into two or three dimensions for visualization [35]. These methods aim to help decision-making *a posteriori* (after optimization), however, they do not help the optimization process itself and do not consider the DM's preferences.

Interactive methods iteratively elicit the DM's preferences to direct the search toward preferred regions of the PF [36, 37]. However, it is possible to use the provided information to identify objectives that are relevant to the DM, a task that has not been achieved in existing iEMOAs. Generally, preference elicitation and interaction style can take two forms [38].Direct preference elicitation requires the DM to identify some parameters of the preference model directly, which can be in the form of the reference point (aspiration level/goal) (see, e.g., PBEA [39], WASF-GA [40]), reservation levels [41], and weights (see, e.g., R-NSGAII [42]), among others. On the other hand, in indirect approaches, the DM is required to provide some holistic judgments, which tend to be less demanding, mainly in the form of exemplary decisions. When indirect approaches are used, the DM is not required to have prior knowledge of solution space and the optimization algorithm [43]. Indirect queries can be in the form of pairwise comparisons [44, 45, 45–49], selecting the best among a small subset of solutions [27, 50, 51], accepting or rejecting a presented trade-off [52], or ordering a subset of solutions [53]. Generally, iEMOAs that require the DM to rank a subset of solutions are known as ranking-based iEMOAs. In this research, we propose an approach that can use the solutions that are ranked by the DM in ranking-based iEMOAs to identify relevant objectives and update the set of active objectives accordingly.

IV. Methods

As described above, existing approaches for objective reduction are mainly concerned with removing redundant or correlated objectives without interacting with a DM. In this section, we propose a method that is able to use the DM's preferences, elicited when interacting with an iEMOA, and equip the iEMOA with the ability to identify irrelevant objectives as well as hidden ones, and switch them dynamically during the optimization process. In practice, we have found that it is relatively easy and efficient to extend iEMOAs with this capability, as switching objectives can only happen after an interaction and the number of interactions is always much smaller than the number of generations of the iEMOA.

In a nutshell, our proposal works as follows. At some point during its execution, the iEMOA interacts with the DM by showing the value of all potential objectives of a selected subset of solutions and asking the DM to rank the solutions according to her preferences. Feature selection, applied to the rankings and the objective values, is used to identify which objectives have the most significant effect on the ranking. The method uses this information to possibly activate currently inactive objectives and/or deactivate currently active ones. The iEMOA then continues its execution using not only the ranking information provided but possibly a new set of active objectives. In what follows, we describe our proposal in detail.

A. Feature Selection

We explore two feature selection methods in this study to understand the relevance of this algorithmic component: Uni-variate feature selection and recursive feature elimination (RFE). Hereafter, feature and objective are used interchangeably in this context.

1) Uni-variate Feature Selection: We propose the application of F-test[1] uni-variate feature selection for identifying the most relevant features. The F-test assumes that the data is normally distributed and p-values may be unreliable for large deviations from normality. If the normality assumption is not valid, then the alternative approach is to use the non-parametric mutual information [54, 55], which measures the dependency between two random variables. Our explanation focuses on the F-test but it is easily extended to the mutual information-based test. Preliminary experiments have shown that the normality assumption is valid for the problems considered in our study, as verified by the D'Agostino-Pearson test [56], and the F-test provides slightly better accuracy. These experiments also indicate that the accuracy of the method is acceptable even with a small sample size and improves significantly when the training set increases to 10 and 15. The details of these experiments are provided in Appendix II.

In uni-variate methods, each feature is considered independently and any correlation between features is ignored [57]. Let T be the set of solutions presented to the DM at an interaction, where $\mathbf{z}_j \in T$ is the vector of objective values of the j^{th} solution presented to the DM, and z_{ji} denotes the value of its i^{th} objective out of the m potential objectives ($f_i \in F$). The DM ranks the solutions according to her own preferences (smaller rank values are more preferred). The vector of rankings is given by $\mathbf{r}$, where r_j is the rank corresponding to $\mathbf{z}_j \in T$. There is no restriction on the rankings and two solutions may have the same rank.

The procedure for F-test uni-variate feature selection can be described as follows (for a more detailed introduction to F-test feature selection please refer to [58]):

Step 1: The correlation ρ_i between each objective (feature) i and $\mathbf{r}$ is computed as

$$\rho_i = \sum_{j=1}^{|T|} \frac{(z_{ji} - \bar{z_{.i}}) \cdot (r_j - \bar{\mathbf{r}})}{S_{z_{.i}} S_{\mathbf{r}}} \,, \tag{2}$$

where $\bar{z_{.i}}$ and $S_{z_{.i}}$ are the mean and standard deviation of the i^{th} objective value over all solutions in T, respectively, and $\bar{\mathbf{r}}$ and $S_{\mathbf{r}}$ are the same for the vector of rankings.

Step 2: The F-statistic for each objective is computed as

$$\mathcal{F}_i = \frac{\rho_i}{1-\rho_i} \cdot (|T| - 2) \,. \tag{3}$$

[1]The name of the test is unrelated to the set F of potential objectives.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2023.3269348

Table I

A NUMERICAL INSTANCE OF UNI-VARIATE FEATURE SELECTION. $N_{\text{EXA}} = 5$ SOLUTIONS ARE RANKED ($\mathbf{r}$) BY THE DM ON A PROBLEM WITH $m = 4$ OBJECTIVES. THE p-VALUE OF EACH OBJECTIVE IS CALCULATED AS EXPLAINED IN THE ALGORITHM 1. $k = 2$ OBJECTIVES WITH MINIMUM p-VALUES ARE ACTIVATED (f_1 AND f_4 IN BOLD-FACE) AND THE EMOA CONTINUES THE OPTIMIZATION WITH THE NEW SET OF ACTIVE OBJECTIVES.

$N_{\text{exa}} = 5$	f_1	f_2	f_3	f_4	$\mathbf{r}$
1	0.71	0.63	0.37	0.45	*4*
2	0.65	0.08	0.89	0.40	*3*
3	0.51	0.64	0.75	0.12	*1*
4	0.65	0.84	0.79	0.31	*2*
5	0.95	0.32	0.86	0.82	*5*
p-values	**0.03**	0.45	0.82	**0.01**	

Here, the F-statistic is a notion of how well an objective can explain the rankings provided by the DM.

Step 3: The p-values corresponding to each F-statistic can be calculated by any statistical software.

Step 4: Features with lower p-value are selected. Number of selected features can be either fixed to a given value k ($2 \leq k < |F|$) or variable. In the latter case, objectives with p-values less than a predetermined threshold $\tau \in (0, 1]$ are selected (or at least two objectives). These two variants are explained in Section IV-B.

The lower the p-value, the better is the corresponding objective function in explaining the DM's rankings. The pseudo-code of uni-variate feature selection is illustrated in Algorithm 1.

To illustrate the procedure after a given interaction, we outline an example in Table I. We assume a fixed number of objectives ($k = 2$) is considered here. $N_{\text{exa}} = 5$ solutions are evaluated by the DM as indicated by $\mathbf{r}$. The p-value of each objective is obtained as explained by uni-variate feature selection. Regardless of the state of the algorithm and set of active objectives before this interaction, the two objectives f_1 and f_4 are activated after this interaction, and other objectives (f_2 and f_3) are deactivated.

2) Recursive Feature Elimination: RFE is different from uni-variate feature selection in that it first uses logistic regression to build a model based on all the features to predict the rankings, and then excludes from the selected subset the feature with the minimum contribution to the regression model [59]. There are several ways to measure the contribution of the i^{th} objective to the regressed model, and here to be consistent with uni-variate variant, we use the statistical significance level (p-value), $\phi(f_i) \in [0, 1]$, for the objective's coefficient in the regressed model, which can be calculated with any statistical software. In the next iteration, the model is built again using the pruned set of objectives. The process is repeated until the size of the pruned set is equal to k in the case of fixed number of objectives. For the variable number of objectives, the pruning stops when the remaining objectives are all significant. The detection method using RFE is depicted in Algorithm 2.

Algorithm 1: Uni-Variate Feature Selection

Input:
F : Set of all potential objectives (features)
T : Set of objective vectors ranked by the DM
$\mathbf{r}$: Vector of ranks
and either
- $k \in [2, |F|) \subset \mathbb{N}$ (for fixed number of objectives) or
- $\tau \in (0, 1] \subset \mathbb{R}$ (for variable number of objectives)

for $i \leftarrow 1$ **to** $|F|$ **do**
  Step 1: Calculate ρ_i using Eq. (2)
  Step 2: Calculate $\mathcal{F}_i$ using Eq. (3)
  Step 3: Calculate p_i (p-value) from $\mathcal{F}_i$
if *Fixed number of objectives* **then**
  $\hat{F} \leftarrow k$ objectives from F with lowest p-value
else
  $\hat{F} \leftarrow \{f_i \in F \mid p_i < \tau\}$
  if $|\hat{F}| < 2$ **then**
    $\hat{F} \leftarrow 2$ objectives from F with lowest p-value
return $\hat{F}$ *(selected objectives)*

Algorithm 2: Recursive Feature Elimination

Input:
F : Set of all potential objectives (features)
T : Set of ranked solutions
$\mathbf{r}$: Vector of ranks
and either
- $k \in [2, |F|) \subset \mathbb{N}$ (for fixed number of objectives) or
- $\tau \in (0, 1] \subset \mathbb{R}$ (for variable number of objectives)

$\hat{F} \leftarrow F$
while $|\hat{F}| > 2$ **do**
  Step 1: $M \leftarrow$ `Build_Model`(T, $\mathbf{r}$, $\hat{F}$)
  Step 2: $f_j \leftarrow \arg\max_{f_i \in \hat{F}} \phi(f_i)$
  if *Fixed number of objectives* **then**
    if $|\hat{F}| = k$ **then**
      break
  else if $\phi(f_j) < \tau$ **then**
    break
  Step 3: $\hat{F} \leftarrow \hat{F} \setminus f_j$
return $\hat{F}$ *(selected objectives)*

B. Fixed versus Variable Number of Active Objectives

The number of features (active objectives) selected can be defined in different ways. Here we explore the following two alternatives:

1) Fixed Number of Objectives (k): The optimization starts with k active objectives and this number is kept constant throughout the optimization process such that activating an inactive objective implies deactivating an active one. The benefit of this approach is that the iEMOA only needs to handle a specific number of objectives, which is simpler than handling a variable number of objectives. The downside is that some relevant objectives may remain hidden if the number of

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2023.3269348

relevant objectives $|F_{\text{DM}}|$ is larger than k. Thus, the goal is to identify the k most relevant objectives for the DM out of all potential objectives.

2) Variable Number of Objectives: We select the subset of objectives that meets a predetermined threshold τ. The lower the value of τ, the lower would be the number of objectives with acceptable p-values. If there is only one objective with a p-value lower than τ, then the two objectives with lowest p-values are selected instead. For RFE, the process stops if the size of the pruned set of objectives is reached 2.

Having two feature selection methods and two approaches with fixed and variable number of objectives as explained above, we have four total variations defined as follows:

1) *k-HD*: Uni-variate feature selection with fixed number of objectives
2) *τ-HD*: Uni-variate feature selection with variable number of objectives
3) *k-HDR*: RFE with fixed number of objectives
4) *τ-HDR*: RFE with variable number of objectives

The convention used for naming the variants can be described as follows; the prefix k indicates the variants with fixed number of objectives and the prefix τ specifies the variable number of objectives. HD indicates that Hidden/irrelevant objective Detection is on. The suffix R is added when the recursive feature elimination is used. The proposed methods can be applied to any ranking-based iEMOA for objective reduction and/or detection of hidden objectives in order to find the objectives that are relevant to the DM. Here, we will focus on extending BCEMOA [44] with our proposed method. We chose BCEMOA due to its popularity and availability of the source code. The proposed method however, can be integrated with any ranking-based algorithm. In what follows, the modified BCEMOA, here called BCEMOA-HD, is explained in detail.

C. BCEMOA-HD

BCEMOA [44] is an iEMOA based on NSGA-II. It starts with a population of randomly generated solutions (*pop*), and the population is evolved with NSGA-II for gen_1 generations. Next, at each interaction step, the best N_{exa} solutions are selected from the evolved population, all potential objectives are evaluated and presented to the DM for ranking. The objective vectors and their ranks are then used to train a support vector machine (SVM) model to learn a utility function (U_{SVM}). The learned U_{SVM} replaces the crowding distance in the next generations. Further interactions with the DM provide additional samples to re-train the SVM model and improve the predictions of the learned utility function.

Similar to the original BCEOMA, the BCEMOA-HD algorithm, proposed here, starts with a set of active objectives $\hat{F}$. All inactive objectives ($F \setminus \hat{F}$) do not need to be evaluated during the optimization and do not participate in dominance ranking and evolution of the population. At each interaction with the DM, all objectives in F are evaluated for the solutions that are presented to the DM. Immediately after each interaction, the feature selection method described in Section IV-A is applied to the objective vectors and their rankings to

Algorithm 3: BCEMOA-HD

Input:
N_{int} : Total number of interactions
N_{exa} : Number of training examples per interaction
pop : Population of solutions
gen_1 : Generations before first interaction
gen_i : Generations between two interactions
F : Set of potential objectives
$\hat{F}$: Set of active objectives
and either
- $k < |F|$ (for fixed number of objectives) or
- τ (for variable number of objectives)

1 $T \leftarrow \emptyset$, $r \leftarrow \emptyset$
2 *pop* $\leftarrow$ run NSGA-II for gen_1 generations optimizing only $\hat{F}$
3 **for** 1 **to** N_{int} **do**
4 $\quad T_i \leftarrow$ select N_{exa} solutions
5 $\quad$ Evaluate solutions in T_i for all objectives in F
6 $\quad r_i \leftarrow \texttt{DM\_ranks}(T_i)$
7 $\quad T \leftarrow T \cup T_i$
8 $\quad \mathbf{r} \leftarrow \mathbf{r} \cup r_i$
9 $\quad \hat{F} \leftarrow \texttt{feature\_selection}(F, T, \mathbf{r}, k \textbf{ or } \tau)$
10 $\quad$ Evaluate *pop* for $f_i \in \hat{F}$
11 $\quad U_{\text{SVM}} \leftarrow \texttt{train\_SVM}(\{z_{ji} \mid \mathbf{z}_j \in T \land f_i \in \hat{F}\}, \mathbf{r})$
12 $\quad \texttt{Crowding\_Distance} \leftarrow U_{\text{SVM}}$
13 $\quad$ *pop* $\leftarrow$ run NSGA-II for gen_i generations optimizing only $\hat{F}$
14 **return** *Best* $\mathbf{x} \in$ *pop ranked first by non-dominated sorting and then* U_{SVM} *considering only* $\hat{F}$

identify relevant objectives and update $\hat{F}$. Consequently, the population may need to be updated by evaluating any objective $f_i \in \hat{F}$ that has become active. SVM is also used to learn U_{SVM} based on active objectives in the updated $\hat{F}$ and their rankings. An overview of BCEMOA-HD is shown in Algorithm 3.

As described above, compared to the original BCEMOA, we have modified the algorithm in Lines 9 and 10, where feature selection is deployed and the set of active objectives is updated, respectively. A further modification applied to the original BCEMOA is in the selection of the best solutions presented to the DM. When there is no variance in the values of some objective, for example, because its values are near-optimal, then their correlation with the rankings provided by the DM is undefined, and their p-value will be set to 1. As a result, the feature selection will deactivate the objective and replace it with an irrelevant objective that might have a higher correlation by chance. To preserve the elitism and avoid losing the DM's desired solution, the solution that was ranked best by the DM in the last interaction is always included in the next set of solutions presented to the DM by BCEMOA-HD. This way, we make sure we will not lose the DM's most desired solution so far and the utility of the selected solution does not decrease.

V. Experimental Setup

To evaluate the effectiveness of the proposed method, we design a set of experiments that cover different aspects of the problem of identifying hidden and irrelevant objectives. In the experiments with variable number of objectives, we seek to investigate how the methods perform for objective reduction purposes, thus all objectives are active from the start of the run ($\hat{F} = F$). In the case of fixed number of objectives, only specific objectives are active ($\hat{F} \subset F$) at the start.

Although this research is motivated by real-world scenarios, it would be very difficult to analyze the methods using complex real-world problems and human DMs. Instead we use well-known benchmarking problems from the literature, which we extend to simulate hidden and irrelevant objectives, and we simulate a human DM using various utility functions.

In what follows, a detailed description of the design of the experiments is laid out.

A. Simulation of Active and Inactive Objectives

We create synthetic problems that feature irrelevant and hidden objectives by extending existing benchmark problems as follows: Given a problem with $m = |F|$ potential objectives, we extend it with $\mathbf{d} \subseteq [1, m]$, an ordered index set of active objectives such that $i < j \to d_i < d_j$, specifying which objectives are active (optimized), i.e., $i \in \mathbf{d}$ iff $f_i \in \hat{F} \subseteq F$, where f_i is the i^{th} potential objective function. That is, given a solution $\mathbf{x}$, whose objective vector is $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$, the optimizer only considers $\hat{\mathbf{f}}(\mathbf{x}) = (f_{d_1}(\mathbf{x}), \ldots, f_{d_{\hat{m}}}(\mathbf{x}))$ and is able to change the set of active objectives by changing the indices in $\mathbf{d}$.

On the other hand, feature selection methods and the DM have access to $\mathbf{f}(\mathbf{x})$. In particular, when asked to rank a solution $\mathbf{x}$, the simulated DM evaluates $U(\mathbf{f}(\mathbf{x}))$, where U is the utility function that measures the DM's preferences, and U simulates irrelevant objectives by disregarding those components of $\mathbf{f}$, as we explain below.

The above technique can be applied to any multi-objective optimization problem. We describe next the underlying benchmark problems used in our experiments.

B. Underlying Benchmark Problems

We applied the above technique to two well-known numerical and binary benchmark problems, namely, multi-objective NK landscape problems with correlation between objectives (ρMNK) [19] and DTLZ problems [18] with $m \in \{4, 10, 20\}$ objectives. Problems with $m = 4$ help us to better understand and investigate the dynamics of the proposed methods, while larger number of objectives allows us to evaluate the efficiency of the feature selection with variable number of objectives in many-objective problems.

ρMNK problems allow us to analyse the effects of correlation among objectives and smoothness of the landscape on the performance of the proposed method. We consider ρMNK instances with different values of correlation among objectives $\rho \in \{-0.25, 0, 0.25, 0.5, 0.75, 0.9\}$, taking into account the restriction that $\rho \geq -1/(m-1)$ [19] and different values of parameter K, which controls the smoothness of the landscape, namely, $K \in \{1, 4, 6, 8\}$ for problems with 4 objectives and $K \in \{1, 5, 10, 15\}$ for many objective problems, considering the constraint $K < n$. The greater the value of K, the more rugged is the fitness landscape. The value of n is kept fixed at 10 for problems with $m = 4$, 20 for problems with $m = 10$, and 30 for problems with $m = 20$ for ρMNK problems.

From the DTLZ test suite, we focus on DTLZ1, DTLZ2 and DTLZ7, which were also used in the experiments of the original BCEMOA [44] and also in [60] for objective reduction. DTLZ1 contains $11^k - 1$ local Pareto-optimal fronts. Thus, it can be used to test the ability of the algorithm to deal with multiple local attractors. DTLZ2 has a concave Pareto front. Finally, DTLZ7 has 2^{m-1} disconnected Pareto-optimal regions in the objective space and is used to check the diversity of the solutions and the performance of the algorithm in disconnected feasible space.

As suggested in [18], the decision space dimension (n) is $m+4$ for DTLZ1, $m+9$ for DTLZ2 and $m+19$ for DTLZ7. With DTLZ problems, optimizing a subset of objectives will optimize the rest of the objectives as well. To make the problem more challenging and also to avoid collapsing the PF to one point when projected to $k < m$ objectives, we follow [60] and map x_i to $x_i/2 + 0.25$, $i = 1, \ldots, n$, for DTLZ2 and bound x_i within $[0.25, 0.75]$ for DTLZ1, which is also suggested in [44]. Please note that this modification is not needed for DTLZ7 as it does not collapse to a single point.

C. Machine Decision Maker (MDM)

We adhere to the MDM framework introduced in [61] and simulate the DM's preferences with an utility function (UF) that explicitly expresses which objectives are relevant, so that we can assess the effectiveness of the methods proposed here for identifying relevant objectives.

We define $\mathbf{c}$ as the ordered index set of relevant objectives such that $i < j \to c_i < c_j$, and consider the following quadratic UFs that were proposed in experiments on the original BCEMOA [44]:

$$\text{UF1}(\mathbf{f}) = 0.28f_{c_1}^2 + 0.38f_{c_2}^2 + 0.29f_{c_1}f_{c_2} + 0.05f_{c_1} \quad (4)$$

$$\text{UF2}(\mathbf{f}) = 0.6f_{c_1}^2 + 0.05f_{c_1}f_{c_2} + 0.23f_{c_1} + 0.38f_{c_2} \quad (5)$$

$$\text{UF3}(\mathbf{f}) = 0.44f_{c_1}^2 + 0.14f_{c_2}^2 + 0.09f_{c_1}f_{c_2} + 0.33f_{c_1} \quad (6)$$

In addition, we consider the following Tchebychef UF

$$U_{\text{tch}}(\mathbf{f}) = \max_{i \in \mathbf{c}} w_i |f_i - f_i^*| \quad (7)$$

with $\mathbf{0}$ as the ideal point $\mathbf{f}^*$. The weights w_i of irrelevant objectives ($i \notin \mathbf{c}$) are set to zero while the weights of relevant objectives were manually selected for each problem such that the most preferred solution is away from the corner points as far as possible.[2]

In the UFs above, the DM only considers $F_{\text{DM}} = \{f_i | i \in \mathbf{c}\}$, while other objectives are irrelevant. Our simulation strategy assumes that relevant objectives are not strongly correlated with irrelevant ones. Otherwise, an objective that does not explicitly appear in the definition of a particular UF may still

[2] The weights used for each problem are given in the Appendix III.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2023.3269348

be identified as relevant by the proposed methods. We explain how we selected relevant objectives for each problem in the next section.

Although the UFs in Eqs. (4)–(7) are designed for minimization, we reverse and scale all utility values in the experiments to the range $[0, 1]$, such that 1 corresponds to the best utility value and 0 to the worst one, for consistency with multi-attribute utility theory [62].

D. Selecting Relevant Objectives

Projection of the PF on lower dimensions might make it collapse to a single point, for some problems. This is true for DTLZ problems even when the problem is bounded [60]. Thus, careful consideration should be given when using these problems to simulate active and inactive objectives. For instance, the PF of DTLZ7 collapses to a point if the first two objectives are active and the rest are inactive. After a careful examination, here the first and fourth objectives are selected as relevant for DTLZ problems ($\mathbf{c} = \{1, 4\}$). In the case of ρMNK problems the first two objectives are selected ($\mathbf{c} = \{1, 2\}$). For ρMNK problem with four objectives, having $F_{\text{DM}} = \{f_1, f_2\}$ and given an initial $\mathbf{d} = \{2, 4\}$, we can see that f_1 is a hidden objective (relevant but not optimized), f_2 is both relevant and optimized, f_3 is irrelevant and not optimized, and f_4 is irrelevant and optimized.

E. Evaluation of the Results

The experiments are performed in three different modes to enable the assessment of the proposed algorithms:

1) *Golden* mode: No interaction is done in this mode and the algorithm directly accesses the true UF of the DM instead of learning a UF. Moreover, only relevant objectives are optimised from the start to end. This is the ideal scenario.
2) *Only learning* mode: This mode corresponds to the original BCEMOA without any detection of hidden objectives. The algorithm does not have access to the DM's UF and instead a UF is learned from the rankings provided by the MDM, i.e. at each interaction the MDM uses its true UF to rank solutions. Predictions from the learned UF are used to rank non-dominated solutions, replacing the crowding distance in NSGA-II. The algorithm still uses non-dominated sorting as the first criteria to rank solutions. Both non-dominated sorting and the learned UF only consider the set of active objectives $\hat{\mathbf{f}}(x)$. The set of active objectives never changes, that is, $\mathbf{d}$ remains constant throughout the run.
3) *Learning + detection* mode: This is our proposed BCEMOA-HD that performs detection of hidden objectives and is able to modify the set of active objectives. Within this mode, we test 4 variants of the HD method (see Section IV-B): k-*HD*, τ-*HD*, k-*HDR*, τ-*HDR*. Similar to the *Only learning* mode, the optimization algorithm relies on non-dominated sorting and an UF that is learned based on $\hat{\mathbf{f}}(x)$, and not on the MDM's true UF. However, in this mode, $\mathbf{d}$ and subsequently $\hat{\mathbf{f}}$ may change after each interaction with the ultimate goal of converging to the objectives that are actually relevant for the DM (F_{DM}).

Having these three modes makes it possible to evaluate the performance of the proposed method compared to the original BCEMOA and to the best solution that is achieved under an ideal scenario in *Golden* mode. The criteria for evaluation of the performance is the true utility value of the final solution returned by the algorithm. For *Learning + detection*, we also record the active objectives after each interaction to investigate how good the proposed method performs in detecting the relevant objectives during an optimization run. This allows us to measure the number of objective evaluations (see Definition II.3). Understanding the relationship between utility and the computational/resource effort invested is of particular relevance to problems where objective evaluations are expensive, time-consuming and/or resource-intense [63]. In such problems, identifying high-utility solutions with as few as possible objective evaluations is preferred (or even needed in order to avoid premature termination of the optimization process due to a lack of resources). As a side effect, fewer objective evaluations means a reduced level of complexity, e.g. in case objectives are heterogeneous [64], and a reduced cognitive load on the DM as ranking is done considering fewer objectives comparisons in total.

Parameter Settings: All variants of BCEMOA use the parameter settings proposed in the original paper [44], including the parameters of the SVM learning model. In particular, the total number of generations is 500 and $N_{\text{exa}} = 5$ solutions are shown to the DM at each interaction. Within BCEMOA, NSGA-II uses a population size is 100 and creates 100 new solutions at each generation. NSGA-II runs for $gen_1 = 200$ generations before the first interaction and there are $gen_i = 30$ generations between subsequent interactions. The total number of generations after the last interaction is calculated as $500 - gen_1 - gen_i(N_{\text{int}} - 1)$. Thus, changing the number of interactions (N_{int}) would not alter total number of generations. We run experiments with 1, 3 and 6 interactions for DTLZ problems. For ρMNK problems, we only consider 6 interactions and, instead, we investigate the effect of different levels of correlation (ρ) and ruggedness (K). Each algorithmic run was repeated 40 times with different random seeds.

Implementations: The algorithms, machine DM and ρMNK problems are implemented in Python 3.7.6. The implementations of NSGA-II within BCEMOA and DTLZ benchmarks are provided by the Pygmo library 2.16.0 [65], the univariate feature-selection and RFE implementations are based on Scikit-learn 0.23.1 (http://scikit-learn.org/). To motivate further research, we make our code [66] publicly available.

VI. Experimental Results

The interactive methods are designed to help the DM reach a satisfying solution and, thus, the utility value of the final solution returned by the algorithm is used to evaluate the performance of the interactive methods [24]. The utility values are normalized to the $[0, 1]$ interval in all the results. In this section, we focus on the most important findings and some figures are omitted to save space. The complete set of results and figures can be found in Appendix I.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2023.3269348

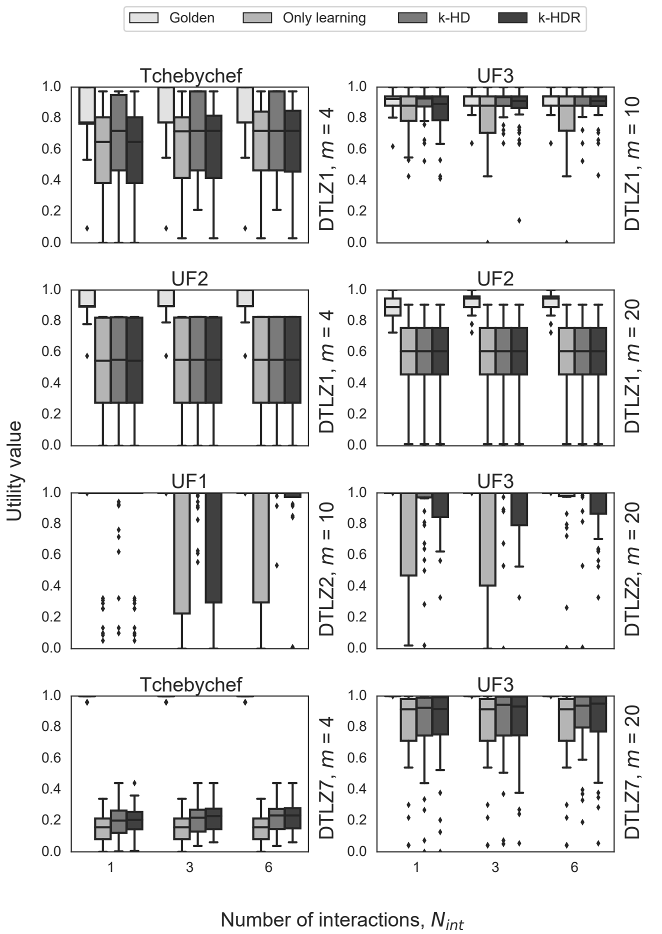

Figure 1. Comparison of the performance of different modes for DTLZ problems. The number of active objectives is fixed at $k = 2$. The vertical axis is the utility value (larger is better). The horizontal axis indicates the number of interactions.

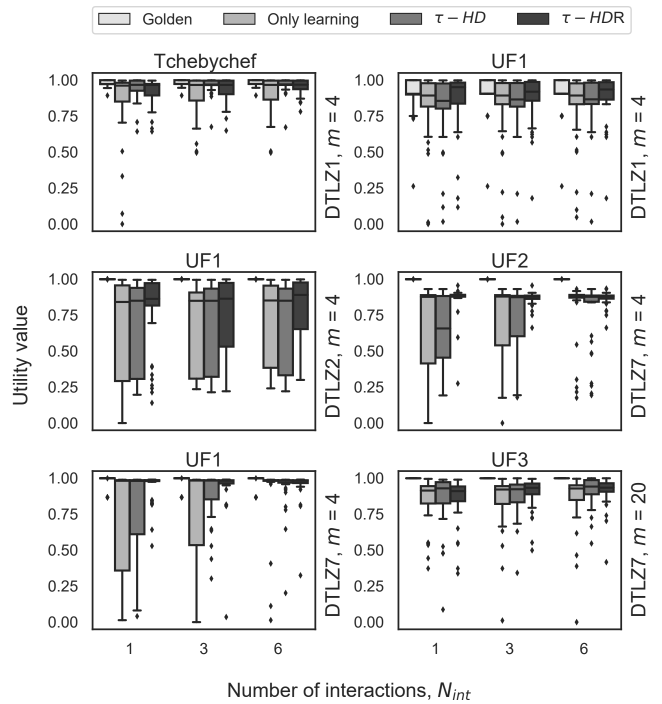

Figure 2. Comparison of the performance of different modes for DTLZ problems. The number of active objectives is not fixed ($\tau = 0.02$) and detection mode is used as an objective reduction technique. The vertical axis is the utility value (larger is better). The horizontal axis indicates the number of interactions.

A. DTLZ Problems with Fixed Number of Objectives

The results of experiments on DTLZ problems with a fixed number of active objectives are illustrated in Figure 1. When using BCEMOA-HD with DTLZ1 problem with $m = 4$ or $m = 20$ and fixed number of active objectives, almost no improvement is observed in utility value compared to *Only learning* when comparing means, although the BCEMOA-HD manages to find better solutions in some instances. However, when $m = 10$, the performance of k*-HD* and k*-HDR* are significantly better than *Only learning* and almost as good as *Golden* mode except for UF2, which still shows no improvement.

For DTLZ2, improvements in the performance can be seen when BCEMOA-HD is used in the case of $m = 10$ and $m = 20$ together with UF1 and UF2. Another important observation is the better performance of k*-HDR* with more interactions, although it fails to get as good as k*-HD*.

For DTLZ7, there are slight improvements when detection methods (k*-HD*, k*-HDR*) are used. Complete list of figures can be found in the Appendix I.

In general, we observe that the proposed methods can significantly improve the utility value of the final solutions with respect to *Only learning* for the UFs tested. In some cases where it fails to do so, the utility value is not deteriorated by the method.

B. DTLZ Problems with Variable Number of Objectives

In this set of experiments, the effectiveness of the BCEMOA-HD is investigated with regard to objective reduction capabilities and thus the number of active objectives is not fixed. Thus, the execution of the algorithms start with all the objectives being active. The key results of these experiments are illustrated in Figure 2. For DTLZ1 with $m = 4$, the τ*-HD* and τ*-HDR* perform better than the *Only learning* mode on Tchebychef UF, while for other UFs they have almost the same performance in terms of the utility value. With $m = 10$ and $m = 20$, τ*-HD* and τ*-HDR* perform as well as the *Golden* mode while τ*-HDR* is slightly outperformed by τ*-HD*. Results for DTLZ2 are identical to those of DTLZ1, i.e., τ*-HD* and τ*-HDR* perform as well as *Golden* mode and outperform *Only learning* mode with $m = 10$ and $m = 20$. For $m = 4$ τ*-HD* and τ*-HDR* outperform *Only learning* when UF3 is used, but cannot perform as well as *Golden* mode. In general, the superiority of τ*-HD* and τ*-HDR* compared to *Only learning* becomes more prevalent with a higher number of objectives. Although in some cases the algorithms return solutions with similar utility, we will show in Section VI-D3 that the detection method results in significant computational savings.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2023.3269348

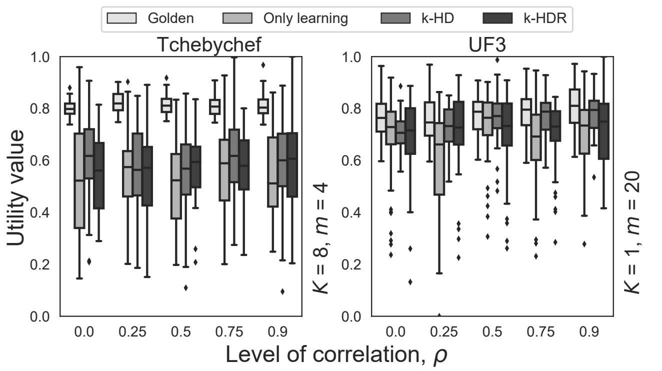

Figure 3. Comparison of the performance of different modes for ρMNK problems. The number of active objectives is fixed to $k = 2$. The vertical axis is the utility value (larger is better). The horizontal axis indicates the ρ values.

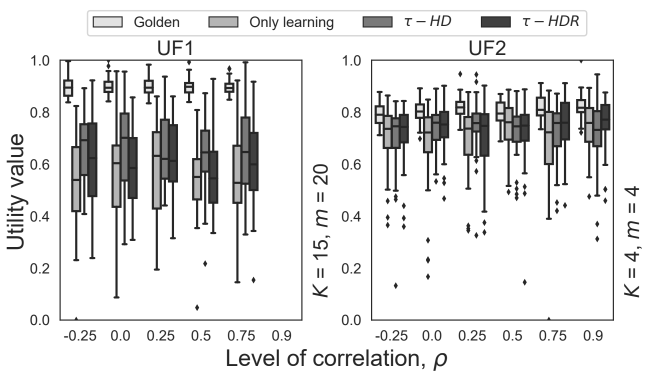

Figure 4. Comparison of the performance of different modes for ρMNK problems. The number of active objectives is not fixed ($\tau = 0.02$) and detection mode is used as an objective reduction technique. The vertical axis is the utility value (larger is better). The horizontal axis indicates the ρ values.

C. ρMNK Problems

The results of experiments on ρMNK problems with a fixed and variable number of objectives are depicted in Figures 3 and 4, respectively. In most of these experiments, the proposed methods outperform the *Only learning* mode. In the case of the fixed number of objectives, in 30% of the cases this better performance of the proposed method is significant (based on the Wilcoxon signed-rank test with Holm's adjustment for multiple comparisons, p-value < 0.05). For the variable number of objectives, the significant observations increase to 50%. The detailed comparison of different modes related to Figs 3 and 4 can be found in the Appendix. In the next section, we will further analyze the results and show that the relevant objectives are indeed detected, which means that a similar solution utility is reached in fewer objective evaluations and the proposed method can improve the computational efficiency of the interactive methods.

D. Further Analysis

1) Anytime Behavior within Each Run: Figure 5 illustrates the change in the utility value of the best solution gained after each interaction in a single run of the algorithm averaged over 40 runs. We observe that all interactions lead to some improvement in the utility of the best solution found, but the improvements become smaller with subsequent interactions.

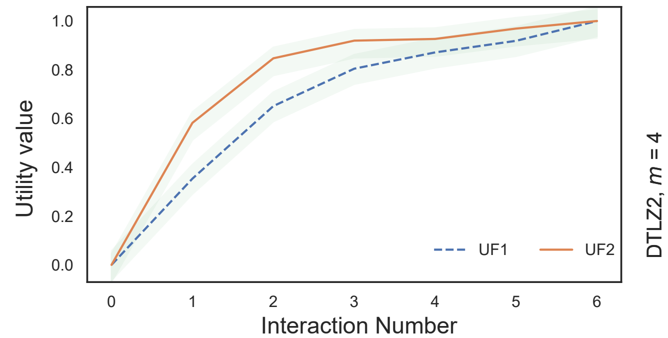

Figure 5. Utility of the best-so-far solution within a single run after each interaction on DTLZ2 problem when using τ-*HD* ($\tau = 0.02$). The lines show the mean value over 40 runs and the shaded area shows the 95% confidence interval around the mean. The results for other test problems are similar.

2) Power of Detection of Relevant Set of Objectives: In terms of the power of the detection, a heatmap plot is provided in Figure 6. The plot illustrates the number of times the relevant and irrelevant objectives are activated by τ-*HD* across all experiments on ρMNK problems with 10 objectives. The x-axis shows interactions within a single run. Interaction 0 refers to the state of the algorithm before the first interaction, when all objectives are active. The y-axis shows the index of all potential objectives. Each cell in the heatmap indicates the number of times the potential objective shown in the y-axis was active after the interaction shown in the x-axis. It can be observed that after the first interaction most of the objectives are deactivated while the first and second objectives, which are the relevant ones, are kept active. It can be easily verified that the τ-*HD* converges fast towards the relevant objectives. Another observation is that after the 6$^{\text{th}}$ interaction, almost all objectives become active, although to a lesser degree compared to the relevant ones. This observation is explained as follows: When the relevant objectives are optimized to their near-optimal value with respect to the DM's UF, the values of these objectives will be nearly constant in the solutions presented to the DM. During feature selection, the correlation for such objectives would be undefined (Eq. 2), the F-statistic will be set to 0.0 and the p-value to 1.0, thus the objectives would be identified as irrelevant and replaced with inactive ones that, by chance, show some correlation with the rankings of the DM.

3) Analysis of the Threshold Parameter (τ) and Computational Efficiency: As an important parameter of τ-*HD* and τ-*HDR*, τ indirectly controls the number of active objectives; thus, careful examination should be given in determining its value. To inspect the effect of parameter τ, the DTLZ problems with $m = 20$ objectives are solved with different values of τ. The results for other problems are similar and hence not discussed here. Setting $m = 20$ provides a better illustration of the efficiency of the proposed method in reducing the computational requirements and objective evaluations. When $\tau = 1$, all objectives have p-value less than the threshold and,

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2023.3269348

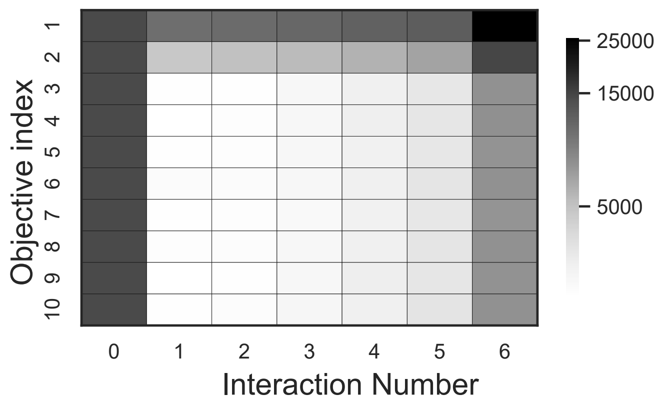

Figure 6. Convergence of the τ-*HD* ($\tau = 0.02$) towards relevant objectives ($c = \{1, 2\}$) through interactions on problem ρMNK with 10 objectives. The x-axis indicates the interaction number and the y-axis indicates the index of the objective functions. The darker color, the higher is the number of the times the objective has been activated. The relevant objectives are found after the first interaction in the majority of the experiments.

thus, all of them are active; this means no objective reduction is performed and the mode is identical to *Only learning*. The results in Figure 7 show that the performance of the τ-*HD* improves with lower values of τ on DTLZ problems.

On the other hand, reducing the τ value would reduce the number of active objectives (up to a minimum of 2 active objectives). A lower number of active objectives increases the efficiency of an EMOA and, by avoiding the evaluation of inactive objectives, possibly translates to savings in computational and economical costs. To illustrate these potential savings, Figure 8 shows the changes in the number of active objectives after each interaction, averaged over 40 runs. The shaded areas show the 95% confidence interval around the mean. It can be clearly verified that after the first interaction, the number of active objectives experiences a steep decrease. As expected, when $\tau = 1$, no objective reduction is performed.

Another important criteria is the ratio of evaluations of relevant objectives to those of irrelevant ones. Since inactive objective are not evaluated during optimization, we measure the total number of objective evaluations (Def. II.3) only for active objectives and we observe that τ-*HD* effectively reduces this total number. For instance, when $\tau = 1$ (equivalent to *Only learning* mode), an experiment on DTLZ1 with UF3, uses $600,000$ objective evaluations and only 10% of these evaluations pertain to relevant objectives. However, when τ-*HD* is used, only $45,000$ objective evaluations are done of which $30,000$ (67%) are dedicated to relevant objectives. In general, objective evaluations are reduced by up to 80% compared to *Only learning* when τ-*HD* or τ-*HDR* is used. These savings could used to run the optimizer for longer, leading to improved solutions (although we do not explore this possibility here).

VII. Conclusion and Future Work

This study has considered multi-objective problems that are solved by means of iEMOAs and where only an unknown subset of all the potential objectives are of relevance to the DM. In this context, we provided formal definitions of irrelevant, hidden and active objectives that complement the definition of redundant objectives already studied in the literature. We propose here a detection method that may be incorporated into any ranking-based iEMOA to identify irrelevant and hidden objectives. Furthermore, we show that an iEMOA able to dynamically change the active objectives can use this method to find solutions with higher utility for the DM in fewer objective evaluations. In addition, for the purpose of benchmarking, we propose a methodology for the simulation of irrelevant, hidden, and active objectives.

Two variants of the method with a fixed and variable number of active objectives were studied. The results show that the variant with the variable number of objectives is useful for dimension reduction purposes, reducing the number of active objectives even after the first interaction. This eliminates unnecessary evaluations of irrelevant objectives, thus saving computational effort, and improves the utility of the final solution returned by the iEMOA. The variant with a fixed number of active objectives has shown to be able to both remove irrelevant objectives and activate hidden ones. We also explored the application of recursive feature selection. However, the results indicate that there is no gain in using this method over the uni-variate feature selection. Comparing the results achieved for different test problems, we observed that the improvements in the final utility value are more significant for DTLZ problems. However, savings with regard to objective evaluations are achieved for both test problems. These savings may be most beneficial in problems where objective evaluations are expensive in terms of computational time, economical cost, or physical resources. We showed experimentally that the value of τ affects the number of active objectives and can be used as a tool to control this aspect. We considered four different UFs to simulate DMs with different preferences. Future studies should consider other UFs, such as the Sigmoid UF [67].

There is scope to obtain further improvements. We observed that, in some experiments, once the relevant objectives have reached near-optimal values with respect to the DM's UF, the proposed methods may replace active relevant objectives with irrelevant ones. Thus, it would desirable to introduce a mechanism that avoids such a behavior. Future studies should also consider DM simulations of learning and preference drift and how our proposed detection method can cooperate with an iEMOA to detect and adapt to such changes. Our proposal relies on uni-variate feature selection based on the correlation between objectives and DM's rankings. Considering nonlinear regression in feature selection would be a subject worth studying. In the case of BCEMOA, the learned SVM model could be used to identify relevant objectives. However, our proposal here is more general and does not require that the iEMOA uses a specific learning model.

References

[1] A. Sinha, D. K. Saxena, K. Deb, and A. Tiwari, "Using objective reduction and interactive procedure to handle many-objective optimization problems," *Applied Soft Computing*, vol. 13, no. 1, pp. 415–427, 2013.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2023.3269348

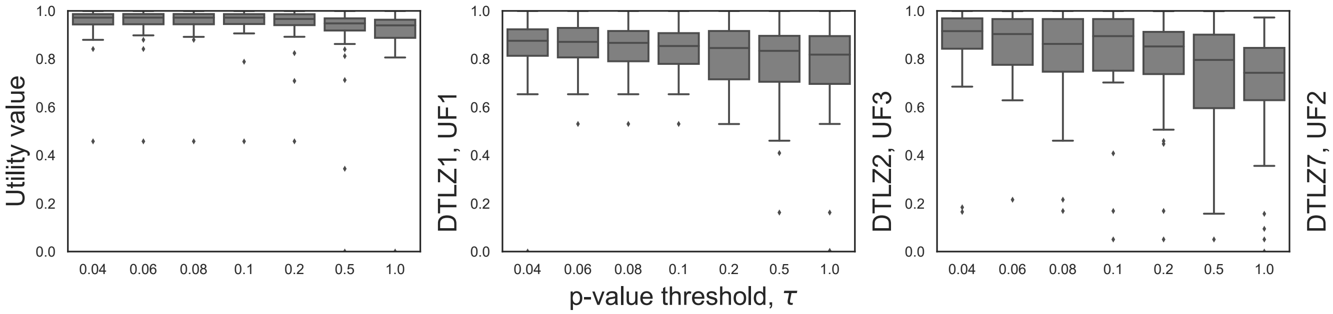

Figure 7. Performance analysis of τ-*HDR* with different values of τ for DTLZ problems with $m = 20$. The number of interactions in all runs is 6.

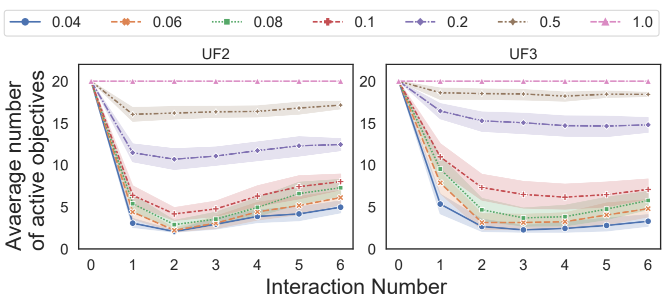

Figure 8. Number of active objectives within a single run after each interaction for different values of τ in τ-*HDR* mode on DTLZ1 (left figure) and DTLZ2 (right figure) test problems with 20 objectives. The x-axis indicates the interaction number and the y-axis is the number of active objectives after each interaction averaged over 40 runs. The shaded areas show the 95% confidence interval around the mean. The results for other test problems are identical.

[2] M. T. Jensen, "Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 5, pp. 503–515, 2003.

[3] R. Allmendinger, A. Jaszkiewicz, A. Liefooghe, and C. Tammer, "What if we increase the number of objectives? Theoretical and empirical implications for many-objective combinatorial optimization," *Computers & Operations Research*, vol. 145, p. 105857, 2022.

[4] N. Beume, C. M. Fonseca, M. López-Ibáñez, L. Paquete, and J. Vahrenhold, "On the complexity of computing the hypervolume indicator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1075–1082, 2009.

[5] R. C. Purshouse and P. J. Fleming, "Evolutionary many-objective optimisation: an exploratory analysis," in *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'03)*. Piscataway, NJ: IEEE Press, Dec. 2003, pp. 2066–2073.

[6] H. K. Singh, A. Isaacs, and T. Ray, "A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 4, pp. 539–556, 2011.

[7] D. Brockhoff and E. Zitzler, "Objective reduction in evolutionary multiobjective optimization: Theory and applications," *Evolutionary Computation*, vol. 17, no. 2, pp. 135–166, 2009.

[8] ——, "Dimensionality reduction in multiobjective optimization: The minimum objective subset problem," in *Operations Research Proceedings 2006*, K.-H. Waldmann and U. M. Stocker, Eds. Berlin/Heidelberg: Springer, 2007, pp. 423–429.

[9] ——, "Are all objectives necessary? On dimensionality reduction in evolutionary multiobjective optimization," in *Parallel Problem Solving from Nature – PPSN IX*, ser. Lecture Notes in Computer Science, T. P. Runarsson, H.-G. Beyer, E. K. Burke, J.-J. Merelo, D. Whitley, and X. Yao, Eds., vol. 4193. Heidelberg: Springer, 2006, pp. 533–542.

[10] A. Jaszkiewicz and J. Branke, "Interactive multiobjective evolutionary algorithms," in *Multiobjective Optimization: Interactive and Evolutionary Approaches*, ser. Lecture Notes in Computer Science, J. Branke, K. Deb, K. Miettinen, and R. Słowiński, Eds. Heidelberg: Springer, 2008, vol. 5252, pp. 179–193.

[11] P. Campigotto and A. Passerini, "Adapting to a realistic decision maker: experiments towards a reactive multi-objective optimizer," in *Learning and Intelligent Optimization, 4th International Conference, LION 4*, ser. Lecture Notes in Computer Science, C. Blum and R. Battiti, Eds. Heidelberg: Springer, 2010, vol. 6073, pp. 338–341.

[12] P. Pu and L. Chen, "User-involved preference elicitation for product search and recommender systems," *AI Magazine*, vol. 29, no. 4, p. 93, Dec. 2008.

[13] R. Battiti and P. Campigotto, "Reactive search optimization: Learning while optimizing. an experiment in interactive multi-objective optimization," in *Proceedings of MIC 2009, the 8th Metaheuristics International Conference*, M. Caserta and S. Voß, Eds. Hamburg, Germany: University of Hamburg, 2010.

[14] A. Kryston, M. F. Müller, G. Penny, D. Bolster, J. L. Tank, and M. S. Mondal, "Addressing climate uncertainty and incomplete information in transboundary river treaties: A scenario-neutral dimensionality reduction approach," *Journal of Hydrology*, vol. 612, p. 128004, 2022.

[15] T. J. Stewart, "Robustness of additive value function methods in MCDM," *Journal of Multi-Criteria Decision Analysis*, vol. 5, no. 4, pp. 301–309, 1996.

[16] J.-M. Ramos-Pérez, G. Miranda, E. Segredo, C. León, and C. Rodríguez-León, "Application of multi-objective evolutionary algorithms for planning healthy and balanced school lunches," *Mathematics*, vol. 9, no. 1, p. 80, Dec. 2021.

[17] V. Trianni and M. López-Ibáñez, "Advantages of task-specific multi-objective optimisation in evolutionary robotics," *PLoS One*, vol. 10, no. 8, p. e0136406, 2015.

[18] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization*, ser. Advanced Information and Knowledge Processing, A. Abraham, L. Jain, and R. Goldberg, Eds. London, UK: Springer, Jan. 2005, pp. 105–145.

[19] S. Verel, A. Liefooghe, L. Jourdan, and C. Dhaenens, "On the structure of multiobjective combinatorial search space: MNK-landscapes with correlated objectives," *European Journal of Operational Research*, vol. 227, no. 2, pp. 331–342, 2013.

[20] E. Zitzler, J. D. Knowles, and L. Thiele, "Quality assessment of Pareto set approximations," in *Multiobjective Optimization: Interactive and Evolutionary Approaches*, ser. Lecture Notes in Computer Science, J. Branke, K. Deb, K. Miettinen, and R. Słowiński, Eds. Heidelberg: Springer, 2008, vol. 5252, pp. 373–404.

[21] T. Gal and H. Leberling, "Redundant objective functions in linear vector maximum problems and their determination," *European Journal of Operational Research*, vol. 1, no. 3, pp. 176–184, 1977.

[22] D. K. Saxena, T. Ray, K. Deb, and A. Tiwari, "Constrained many-objective optimization: A way forward," in *2009 IEEE Congress on Evolutionary Computation*, 2009, pp. 545–552.

[23] R. E. Steuer and L. Gardiner, "On the computational testing of procedures for interactive multiple objective linear programming," in *Operations Research*, G. Fandel and H. Gehring, Eds. Berlin/Heidelberg: Springer, 1991, pp. 121–131.

[24] B. Afsar, K. Miettinen, and F. Ruiz, "Assessing the performance of interactive multiobjective optimization methods: A survey," *ACM Computing Surveys*, vol. 54, no. 4, 2021.

[25] J. W. Fowler, E. S. Gel, M. Köksalan, P. Korhonen, J. L. Marquis, and J. Wallenius, "Interactive evolutionary multi-objective optimization for quasi-concave preference functions," *European Journal of Operational Research*, vol. 206, no. 2, pp. 417–425, 2010.

[26] L. R. Pedro and R. H. C. Takahashi, "Decision-maker preference modeling in interactive multiobjective optimization," in *Evolutionary Multi-criterion Optimization, EMO 2013*, ser. Lecture Notes in Computer Science, R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, and J. Shaw, Eds. Heidelberg: Springer, 2013, vol. 7811, pp. 811–824.

[27] M. Köksalan and İ. Karahan, "An interactive territory defining evolutionary algorithm: iTDEA," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 702–722, Oct. 2010.

[28] R. Allmendinger and J. D. Knowles, "Evolutionary optimization on problems subject to changes of variables," in *Parallel Problem Solving from Nature, PPSN XI*, ser. Lecture Notes in Computer Science, R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, Eds. Heidelberg: Springer, 2010, vol. 6238, pp. 151–160.

[29] P. J. Agrell, "On redundancy in multi criteria decision making," *European Journal of Operational Research*, vol. 98, no. 3, pp. 571–586, 1997.

[30] A. R. R. Freitas, P. J. Fleming, and F. G. Guimarães, "A non-parametric harmony-based objective reduction method for many-objective optimization," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE Press, 2013, pp. 651–656.

[31] A. R. R. de Freitas, P. J. Fleming, and F. G. Guimarães, "Aggregation trees for visualization and dimension reduction in many-objective optimization," *Information Sciences*, vol. 298, pp. 288–314, 2015.

[32] R. J. Lygoe, M. Cary, and P. J. Fleming, "A many-objective optimisation decision-making process applied to automotive diesel engine calibration," in *Simulated Evolution and Learning*, K. Deb, A. Bhattacharya, N. Chakraborti, P. Chakroborty, S. Das, J. Dutta, S. K. Gupta, A. Jain, V. Aggarwal, J. Branke, S. J. Louis, and K. C. Tan, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 638–646.

[33] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: Linear and nonlinear algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 1, pp. 77–99, 2013.

[34] L. Costa and P. Oliveira, "Biplots in offline multiobjective reduction," in *IEEE Congress on Evolutionary Computation*, 2010, pp. 1–8.

[35] J. E. Fieldsend and R. M. Everson, "Visualising high-dimensional Pareto relationships in two-dimensional scatterplots," in *Evolutionary Multi-criterion Optimization, EMO 2013*, ser. Lecture Notes in Computer Science, R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, and J. Shaw, Eds. Heidelberg: Springer, 2013, vol. 7811, pp. 558–572.

[36] B. Xin, L. Chen, J. Chen, H. Ishibuchi, K. Hirota, and B. Liu, "Interactive multiobjective optimization: A review of the state-of-the-art," *IEEE Access*, vol. 6, pp. 41 256–41 279, 2018.

[37] J. Branke, K. Deb, K. Miettinen, and R. Słowiński, Eds., *Multiobjective Optimization: Interactive and Evolutionary Approaches*, ser. Lecture Notes in Computer Science. Heidelberg: Springer, 2008, vol. 5252.

[38] M. K. Tomczyk and M. Kadziński, "On the elicitation of indirect preferences in interactive evolutionary multiple objective optimization," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, ser. GECCO '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 569–577.

[39] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina, "A preference-based evolutionary algorithm for multi-objective optimization," *Evolutionary Computation*, vol. 17, no. 3, pp. 411–436, 2009.

[40] A. B. Ruiz, M. Luque, K. Miettinen, and R. Saborido, "An interactive evolutionary multiobjective optimization method: Interactive WASF-GA," in *Evolutionary Multi-criterion Optimization, EMO 2015 Part II*, ser. Lecture Notes in Computer Science, A. Gaspar-Cunha, C. H. Antunes, and C. A. Coello Coello, Eds. Heidelberg: Springer, 2015, vol. 9019, pp. 249–263.

[41] S. González-Gallardo, R. Saborido, A. B. Ruiz, and M. Luque, "Preference-based evolutionary multiobjective optimization through the use of reservation and aspiration points," *IEEE Access*, vol. 9, pp. 108 861–108 872, 2021.

[42] K. Deb and J. Sundar, "Reference point based multi-objective optimization using evolutionary algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2006*, M. Cattolico *et al.*, Eds. New York, NY: ACM Press, 2006, pp. 635–642.

[43] E. Jacquet-Lagreze and Y. Siskos, "Preference disaggregation: 20 years of mcda experience," *European Journal of Operational Research*, vol. 130, no. 2, pp. 233–245, 2001.

[44] R. Battiti and A. Passerini, "Brain-computer evolutionary multiobjective optimization: A genetic algorithm adapting to the decision maker," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 671–687, 2010.

[45] J. Branke, S. Corrente, S. Greco, R. Słowiński, and P. Zielniewicz, "Using Choquet integral as preference model in interactive evolutionary multiobjective optimization," *European Journal of Operational Research*, vol. 250, no. 3, pp. 884–901, 2016.

[46] M. K. Tomczyk and M. Kadziński, "Robust indicator-based algorithm for interactive evolutionary multiple objective optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2019*, M. López-Ibáñez, A. Auger, and T. Stützle, Eds. New York, NY: ACM Press, 2019, pp. 629–637.

[47] N. Benabbou, C. Leroy, and T. Lust, "An interactive regret-based genetic algorithm for solving multi-objective combinatorial optimization problems," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 03, pp. 2335–2342, 4 2020.

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

[49] J. Branke, S. Greco, R. Słowiński, and P. Zielniewicz, "Interactive evolutionary multiobjective optimization driven by robust ordinal regression," *Bulletin of the Polish Academy of Sciences: Technical Sciences*, vol. 58, no. 3, pp. 347–358, 2010.

[50] J. W. Fowler, E. S. Gel, M. M. Köksalan, P. Korhonen, J. L. Marquis, and J. Wallenius, "Interactive evolutionary multi-objective optimization for quasi-concave preference functions," *European Journal of Operational Research*, vol. 206, no. 2, pp. 417–425, 2010.

[51] M. K. Tomczyk and M. Kadziński, "Decomposition-based co-evolutionary algorithm for interactive multiple objective optimization," *Information Sciences*, vol. 549, pp. 178–199, 2021.

[52] S. Zionts and J. Wallenius, "An interactive multiple objective linear programming method for a class of underlying nonlinear utility functions," *Management Science*, vol. 29, no. 5, pp. 519–529, 1983.

[53] A. Sinha, P. Malo, and M. Kallio, "Convex preference cone-based approach for many objective optimization problems," *Computers & Operations Research*, vol. 95, pp. 1–11, 2018.

[54] A. Kraskov, H. Stögbauer, and P. Grassberger, "Erratum: Estimating mutual information [phys. rev. e 69, 066138 (2004)]," *Phys. Rev. E*, vol. 83, p. 019903, Jan 2011.

[55] B. C. Ross, "Mutual information between discrete and continuous data sets," *PLoS One*, vol. 9, no. 2, pp. 1–5, Feb. 2014.

[56] R. D'Agostino and E. S. Pearson, "Tests for departure from normality. empirical results for the distributions of b_2 and $\sqrt{b_1}$," *Biometrika*, vol. 60, no. 3, pp. 613–622, Dec. 1973.

[57] A. Destrero, S. Mosci, C. Mol, A. Verri, and F. Odone, "Feature selection for high-dimensional data," *Computational Management Science*, vol. 6, pp. 25–40, 2009.

[58] G. W. Heiman, *Understanding Research Methods and Statistics: An Integrated Introduction for Psychology*, 2nd ed. Wadsworth Publishing Company, August 2000.

[59] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1, pp. 389–422, 2002.

[60] D. Brockhoff and E. Zitzler, "Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods," in *Proceedings of the 2007 Congress on Evolutionary Computation (CEC 2007)*. Piscataway, NJ: IEEE Press, 2007, pp. 2086–2093.

[61] M. López-Ibáñez and J. D. Knowles, "Machine decision makers as a laboratory for interactive EMO," in *Evolutionary Multi-criterion Optimization, EMO 2015 Part II*, ser. Lecture Notes in Computer Science, A. Gaspar-Cunha, C. H. Antunes, and C. A. Coello Coello, Eds. Heidelberg: Springer, 2015, vol. 9019, pp. 295–309.

[62] R. Keeney and H. Raiffa, *Decisions with multiple objectives: Preferences and value tradeoffs*. Cambridge University Press, 1993.

[63] R. Allmendinger, "Tuning evolutionary search for closed-loop optimization," Ph.D. dissertation, The University of Manchester, UK, Jan. 2012.

[64] R. Allmendinger and J. Knowles, "Heterogeneous objectives: State-of-the-art and future research," *arXiv preprint arXiv:2103.15546*, 2021.

[65] F. Biscani, D. Izzo, and C. H. Yam, "A global optimisation toolbox for massively parallel engineering optimisation," in *Astrodynamics Tools and Techniques (ICATT 2010), 4th International Conference on*, 2010.

[66] S. M. Shavarani, M. López-Ibáñez, and R. Allmendinger, "Detecting Hidden and Irrelevant Objectives in Interactive Multi-Objective Optimization: Supplementary Material," https://doi.org/10.5281/zenodo.7538986, Jan. 2023.

[67] S. M. Shavarani, M. López-Ibáñez, and J. D. Knowles, "Realistic utility functions prove difficult for state-of-the-art interactive multiobjective optimization algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2021*, F. Chicano and K. Krawiec, Eds. New York, NY: ACM Press, 2021, pp. 457–465.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2023.3269348

Mahdi Shavarani received his Ph.D. degree in Industrial Engineering from Eastern Mediterranean University in 2018. He is currently a Ph.D. Candidate in Multi-Objective Optimization and Multi-Criteria Decision-Making at the University of Manchester, where he has been working extensively on interactive evolutionary multi-objective optimization algorithms. He is a member of the International Society on Multiple Criteria Decision Making and Decision and Cognitive Science Research Center at Alliance Manchester Business School. His research interests lie in operations research, particularly optimization algorithms, logistics, production systems, and applications of machine learning in OR.

Manuel López-Ibáñez (M'10) received his Ph.D. degree from Edinburgh Napier University, Edinburgh, U.K., in 2009. He is a Senior Lecturer (Associate Professor) at the Alliance Manchester Business School, University of Manchester, U.K. Between 2020-2022, he was also a Senior Distinguished Researcher at the University of Málaga, Spain. He has published more than 90 papers in international peer-reviewed journals and conferences. He is a member of the IEEE Task Force on Automated Algorithm Design, Configuration and Selection, an elected member of the ACM SIGEVO Executive Board and the Business Committee of the Genetic and Evolutionary Computation Conference (GECCO), Editor-in-Chief of ACM Transactions on Evolutionary Learning and Optimization and Associate Editor of the Evolutionary Computation journal. His research interests include the automatic configuration and design of optimization algorithms including multi-objective and interactive optimization algorithms for continuous, black-box, and combinatorial problems.

Richard Allmendinger (SM'23, M'14) received a Diplom in Business Engineering from Karlsruhe Institute of Technology, and the Ph.D. degree in Computer Science from the University of Manchester (UoM), UK. He is currently a Professor of Applied AI at UoM, Fellow of the Alan Turing Institute, and Advisor for a dedicated AI fund (called *fund*:AI) for the North of UK and managed by River Capital. Previously, he was Postdoctoral Research Associate at the Biochemical Engineering Department, University College London. His research interests include multi-objective, dynamic, safe, and expensive optimization and machine learning. He is a Vice-Chair of the IEEE CIS Bioinformatics and Bioengineering (BB) Technical Committee, Co-Founder of the IEEE CIS Task Force on Optimization Methods in BB, and on the Editorial Board of several international journals.