Detection of Hidden Objectives and Interactive Objective Reduction

Seyed Mahdi Shavarani¹, Manuel López-Ibáñez¹, and Member, IEEE Richard Allmendinger¹

¹Alliance Manchester Business School, University of Manchester, Manchester, M13 9PL, UK
Email: {seyedmahdi.shavarani, manuel.lopez-ibanez, richard.allmendinger}@manchester.ac.uk

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

Abstract—In multi-objective optimization problems, there might exist hidden objectives that are important to the decision maker but are not being optimized. On the other hand, there might also exist irrelevant objectives that are being optimized but are of less importance to the DM. The question that arises here is whether it is possible to detect and reduce irrelevant objectives without deteriorating the quality of the final results? In fact, when dealing with multi-objective problems, each objective implies a significant cost best avoided if possible. However, existing methods that pertain to the reduction of objectives are computationally intensive and ignore the preferences of the decision maker. In this paper, we propose an approach to exploit the capabilities of interactive evolutionary multi-objective optimization algorithms (EMOAs) and the preference information provided by the decision maker, to detect and eliminate the irrelevant objectives during the optimization process and replace them with hidden ones, if any. The proposed method, which is based on uni-variate feature selection, is computationally effective and can be integrated into any ranking-based interactive EMOA. Using synthetic problems developed in this study, we motivate different scenarios in the experiments and prove the effectiveness of the proposed method in improving the computational cost, the total number of objective evaluations and the quality of the final solutions.

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

I. INTRODUCTION

Interactive Evolutionary Multi-Objective Algorithms (iEMOAs) mostly assume that the objectives being optimized match the ones that interest all potential Decision Makers (DMs), however, this may not be the case. As instance, the DM may not be consulted during the modelling phase or his desires may not be well satisfied during the group decision-making on final objectives to be optimized. As the result, there may exist objectives that are optimized but the DM is not really interested in and on the other hand, there are objectives that the DM cares for but are not being optimized. These inconsistencies may cause the elicited preference information appear 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 modelled as objectives by the algorithm. A real-world example would be the case discussed by Ramos-Pérez et al. [1], where a problem is defined to optimize the food served to school children in terms of not only cost, but also a number of metrics of nutritional value and food variety. The variations in food types is optimized by penalizing repetitions of each food group. Repetitions in each food group may be considered as an independent objective. However, in that study they are merged “a priori” into a single objective. A DM that is interacting with the system may consider a given non-optimal menu (not) appealing but may not be able to justify this decision. Is it because there is a hidden tendency to optimize a certain type of food type? When making decisions, does the DM really considers all nutritious and food variety metrics as assumed by the model? Does the system really need to optimize all of the objectives? Or are there any extra objectives that the DM considers but are not reflected in the model? Another example would be optimizing the behavior of robots [2]. There might exist a set of potential objectives (e.g., average speed, morphology, accuracy, etc.) for how well the robot behaves but, at the end, a human decides which behaviors are better by observing the robot perform tasks. Thus, there might be a disconnection between the set of objectives being optimized and the set of (relevant) objectives that the DM uses in evaluating alternative behaviors.

Under such circumstances, it is desirable to have an algorithm that identifies relevant and irrelevant objectives and updates the optimization model in such a way that only the relevant objectives take part in the optimization process. Removing irrelevant objectives is specifically important in many-objective optimization problems. In some scenarios there is a large number of solution features that DMs may wish to optimise and there is no way but to model the problem as a many-objective optimization problem. However, if not all objectives are relevant to a particular DM, removing those irrelevant objectives will be beneficial to solving many-objective problems where the non-dominance strategy loses its selection pressure as most of the solutions become non-dominated with respect to the current population [3–5]. Even in the case of many-objective algorithms that overcome problems associated
with the dominance-based EMOAs [6, 7], objective reduction
techniques are still beneficial because, first, the running time
increases exponentially with the number of objectives [8] and,
second, the number of non-dominated solutions required for
covering the whole Pareto front increases exponentially with
the number of objectives [9, 10], hence, complicating the
decision-making phase [11].

Given the benefits of reducing the number of objectives,
previous research has considered removing objectives that
are highly correlated to other objectives [11, 12], or their
elimination does not have much of an effect on dominance
relations [5, 13]. Here, we consider an approach for objective
reduction that is different and complementary to previous ones.

Here, for the first time in the literature, we consider the
preference information elicited in interactive EMOAs as an op-
portunity to detect irrelevant objectives during the optimization
and update the model in a way that only relevant objectives, 
i.e. important for the DM, are optimized, while the rest of
the objectives are made inactive. The relevance of objectives
may change from a DM to another. The proposed approach
gives the interactive EMOA the ability to adapt to different
DMs and update objectives that are relevant to the DM who
is interacting with the EMOA. The main contributions of this
study can be summarized as follows:

- A motivation and formal definitions of the problem of
  identifying relevant objectives amongst a set of potential
  objectives in interactive multi-objective optimization.
- Proposal of an approach to detect relevant and hidden
  objectives and to update the set of objectives during the
  optimization to optimize only the relevant ones. The
  approach draws on feature selection methods and can be
  applied to any ranking-based interactive EMOA.
- An approach is proposed to simulate hidden and irrelevant
  objectives which can be applied to any multi-objective
  problem and utility function. This approach is demon-
  strated for DTLZ problems [14] and multi-objective NK-
  landscapes ρMNKs [15].
- A validation of the proposed detection method is per-
  formed which includes:
  i. Problems of varying dimensionality, complexity, and
    Pareto front structure.
  ii. Different utility functions to represent different DMs.
  iii. Different feature selection methods to detect relevant
    objectives, and
  iv. A sensitivity analysis to understand the performance
    impact of key parameters of the proposed approach.

The results of the experiments show that our proposed
method is able to quickly replace irrelevant objectives with
relevant ones and significantly improve the quality of final
solutions. Moreover, it is observed that, even in cases where
the utility of final solution is not favored by the proposed
method, the computational effort is highly reduced. The rest
of the paper is organized as follows. The problem is defined
and formulated in Section II. A summarized background
on previous efforts towards objective reduction is given in
Section III. In Section IV the proposed solution method 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 AND FORMULATIONS

Let us consider a many-objective optimization problem
in which we could potentially optimize simultaneously a set
\( \{f_1, \ldots, f_m\} \in F \) of \( m \) objectives:

\[
\begin{align*}
\text{Minimize} & \quad (f_1(x), \ldots, f_m(x)) \\
\text{subject to} & \quad x \in \mathcal{X}
\end{align*}
\]

where each objective \( f_i: \mathcal{X} \rightarrow \mathbb{R} \) depends on a solution vector
\( x = (x_1, \ldots, x_n) \) of \( n \) design (or decision) variables, and \( \mathcal{X} \)
is the feasible decision space.

**Definition II.1** (Domination). A solution \( x \in \mathcal{X} \) dominates
solution \( y \in \mathcal{X} \) if \( f(x) \) is not worse than \( f(y) \) in any objective
and \( f(x) \) is strictly better than \( f(y) \) in at least one objective.

**Definition II.2** (Non-domination). A feasible solution \( x \in \mathcal{X} \)
is said to be non-dominated if there is no solution \( y \in \mathcal{X} \)
that dominates it. Non-dominated solutions are also known
as Pareto optimal solutions.

**Definition II.3** (Pareto front). The projection of the Pareto
solutions on the objective space is known as the Pareto front
(PF).

**Definition II.4** (Potential objectives). All \( m \) objectives in \( F \)
are called potential objectives.

**Definition II.5** (Redundant objectives). An objective is called
redundant if it can be eliminated without changing the set of
Pareto optimal solutions [16, 17]. Saxena et al. [18] 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 DM interacting with the EMOA. Interactive methods can be
classified into ad-hoc and non-ad-hoc methods [19]. Non-ad-
hoc methods assume there exists a utility function (UF) which
derive the DM’s decisions but are unknown to the EMOA [20].
Due to the popularity of UFs in modeling preference models,
vast majority of iEMOAs are non-ad-hoc methods thus we
focus on those in the remainder of the paper.

Without loss of generality, here we assume an unknown
UF \( U: \mathbb{R}^{m_{DM}} \rightarrow [0, 1] \) the value of which depends
mainly on \( F_{DM} \subseteq F \) objectives \( (m_{DM} = |F_{DM}| \leq m = |F|) \).
An instance will be a linear additive utility function where the weights
associated to irrelevant objectives are close to zero.

**Definition II.6** (Irrelevant objectives). An objective is “irrel-
evant” to the DM if it does not appear in the DM’s utility
function. In other words, irrelevant objectives \( F \setminus F_{DM} \)
do not contribute to the utility of solutions.

Interactive methods try to somehow estimate the DM’s UF
by exploiting the preference information that is elicited from
the DM in the interactions that are performed during the
optimization process. The estimated UF is then used in the
next generations of the interactive EMOA to guide the search
towards that part of the Pareto front that is of interest to the
DM. Let us assume that, at some moment of its execution, a
given EMOA only optimizes a subset \( \hat{F} \subseteq F \) (\( \hat{m} = |\hat{F}| \leq m \))
of the potential objectives.

**Definition II.7** (Active objective). An objective is active if it
is optimized by the EMOA. The set of active objectives is
represented by \( \hat{F} \). Inactive objectives (\( F \setminus \hat{F} \)) are either
evaluated but ignored by the EMOA or not evaluated at all,
unless required by the DM during an interaction.

**Definition II.8** (Objective evaluations & computational effi-
ciency). An objective evaluation is defined as a single eval-
uation of any of the objectives corresponding to a solution
(\( f_i(x) \)) and is different than solution evaluation (\( f(x) \)) which
corresponds to evaluation of all of the objectives of a given
solution. In line with our previous definitions, each solution
evaluation is equivalent to \( \hat{m} \) objective evaluations. On the
other hand, computational cost can be defined in a broader
scale inclusive of objective evaluations and other aspects of
the solution procedure such as the cost of non-domination sorting,
which are again affected by the number of active objectives
which participate in sorting and optimization. Thus, we use
the number of active objectives and objective evaluations as an
indicator of computational efficiency; i.e. reducing the number
of active objectives reduces computational costs.

**Definition II.9** (Hidden objectives). An objective \( f_i \in F \), is said to be hidden if \( f_i \in F_{DM} \land f_i \notin \hat{F} \). That is, an objective
that appears in the DM’s UF (relevant), but is not (currently)
active. If \( F_{DM} \subseteq \hat{F} \), then no hidden objectives exist, but the
EMOA is still optimizing some irrelevant objectives. Similarly,
if \( F_{DM} = \hat{F} \), then neither hidden nor irrelevant objectives exist,
and the EMOA is optimizing precisely the objectives that the
DM cares about.

Being able to reveal hidden objectives is beneficial, since
otherwise there is a disconnection between what the algorithm
is optimizing and what the DM cares about. Moreover, the
interactive algorithm may become confused if the interaction
with the DM is consistent with the dominance criterion for the
objectives in \( F_{DM} \) but appears to contradict it for the objectives
in \( \hat{F} \).

Without any prior information about \( F_{DM} \), most EMOAs
would optimize all potential objectives \( F = F \) leading to
a many-objective optimization problem, which is inherently
challenging for EMOAs as discussed above.

From these 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. While there are studies on 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

A brief review of the literature on the reduction of redundant
objectives is presented here. These methods are applied either
“a priori” to facilitate the optimization process or “a posteriori”
to assist the DM in decision making. Presented studies either
consider correlations among objectives or their importance in
preserving the dominance relations among solutions. We have
categorized existing methods accordingly in such a way that
makes it easy to compare similar studies and also discuss their
limitations. We acknowledge there are reduction methods that
attend to the dimensions of the decision space, such as those
discussed in [21–23], as well as those that focus on constraint
reduction [24, 25]. However, they fall out of the scope of this
research as our focus is on the objective space. Many of these
studies use the term dimension reduction for the same concept.
However, to avoid confusion with methods that reduce the
number of decision variables [21], we use the term “objective
reduction”.

1) Subset selection & preserving Pareto front solutions:
Gal and Leberling [16] proposed one of the earliest approaches
that considered preserving the PF. Although the method was
later extended by Agrell [17], they both make exhaustive
assumptions about the problem structure that are impossible to
validate for real-life problems. Brockhoff and Zitzler [5, 13]
consider objective reduction while preserving the PF. They
introduce two minimum objective subset selection (MOSS)
methods: \( \delta \)MOSS finds a minimal set of objectives with
maximum error of \( \delta \) and \( k \)-EMOSS finds a subset with fixed
size \( k \) having the minimum error. Error here is a measure
of change in the set of PF solutions. The performance of these
methods is compared with PCA-based methods in [26] and
the results indicated the competitiveness of the methods. In
[27], the authors use their minimal set approach online in
simple indicator-based evolutionary algorithm (SIBEA) [28].
As MOSS problems are computationally expensive, Guilén-
Gosálbez [29] proposed a linear programming model for \( k\)
-EMOSS problem, which was demonstrated to have acceptable
performance.

López Jaimes et al. [30] use an unsupervised feature selection
technique [31] for preserving the most conflicting objectives
and identifying the subset of non-redundant objectives. In
their approach, neighborhoods of objective functions are
formed based on correlation of some non-dominated solutions.
Then an objective is selected from each compact neighborhood
while the rest are dropped. Later, this technique was incor-
porated into NSGA-II [32] to construct an online reduction
method [33].

Singh et al. [10] focus on the Pareto corner points to identify
which objectives are sufficient to reproduce the PF. Some
drawbacks of their approach are: not capturing the whole
PF, losing objectives that are contributing somewhere on the
PF away from corner points and possible overestimation of
dimensions.

2) Recombining non-conflicting objectives: Freitas et al.
[34] used the harmonic level, introduced in [13, 35–37], to
identify objectives that can be merged into a new compound
scalarized objective with minimal effect on the PF. As
opposed to conflict, there is “harmony” among objectives when
improvement of either of them does not lead to deterioration
of the other [35–37]. When there is harmony between two
objectives, they can be merged without any loss in quality of
the results. Similar to other objective reduction methods, the DM preferences are not considered in the reduction process in [34]. However, the DM can decide on the final number of objectives “a priori”. Similarly, de Freitas et al. [38] use aggregation trees to identify harmonious objectives that could be aggregated into single scalarized objectives with minimal loss.

3) PCA & Correlation among objectives: One of the first methods that relied on linear PCA to spot redundant objectives was by Deb and Saxena [39] that was implemented within the NSGA-II framework. Linear PCA relies on correlation among objectives to detect conflicting objectives, thus it does not guarantee that the dominance relation is preserved [11]. Furthermore, linear PCA does not provide a good performance for non-linear data, is limited by the type of the data it can handle and it might fail identify all non-conflicting objectives properly, but in local regions where solutions are close it may still be used for determination of conflicts and harmony between objectives [40]. To address these issues, two more methods were introduced in [41], which used correntropy PCA [42] and maximum variance unfolding as was previously used in [43, 44]. PCA was also used in [45], however the method was validated on a problem with 4 objectives only. PCA was also used inline with NSGA-II in [46, 47] among others, to handle redundant objectives in non-linear contexts. PCA-based methods are applied on non-dominated solutions and thus their performance relies heavily on proper approximation of the PF [10].

In their study, Costa and Oliveira [48] have demonstrated that objectives that are deemed redundant by PCA may be “informative”, i.e., contain trade-off information that would be lost if omitted. They suggest using PCA coupled with BIPILOT [49] representation to confirm if a PCA-identified redundant objective is not “informative”. The applicability of the method is limited to the problems with only one independent objective.

4) Visualization: Some studies suggest identification of redundant objectives through visualization [50–53]. However, most of these studies aim to facilitate intuitive visualization of the solutions rather than the optimization process itself and are limited in the number of objectives they can handle. Thus, they use objective reduction methods such as PCA to enable the visualization and hence bring new insights to the DM.

IV. METHODS

As laid out, existing approaches for objective reduction suffer from dependency on non-dominated solutions, expensive computations, elimination of informative objectives, offline reduction, and not exploiting the DM’s preferences. In this section, we illustrate a method that is able to use the DM’s preferences, elicited during optimization by an interactive method, to identify irrelevant objectives as well as hidden ones dynamically. The experimental study will show that the proposed method is able to efficiently reduce the number of objectives being optimized by the optimizer, hence dedicating more computational resources to the optimization of relevant objectives only.

In a nutshell, our proposal works as follows. At some point during the run of the EMOA, the method 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. Feature selection applied to the rankings and the objective values identifies 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 EMOA then continues the search using the new set of active objectives. In what follows the method is described in detail.

A. Feature Selection

We explore two feature selection methods in this study: Uni-variate feature selection and Recursive Feature Elimination (RFE). This gives us the opportunity to investigate and compare the results with respect to different feature selection methods. Hereafter, feature and objective are used interchangeably in this context.

1) Uni-variate Feature Selection: We propose the application of F-test uni-variate feature selection for identifying the most relevant features. In uni-variate methods, each feature is considered independently and any correlation between features is ignored [54]. The uni-variate detection method applied here estimates the degree of linear dependency between two random variables. Let T be the set of solutions presented to the DM at an interaction, where \( f_j \in T \) is the \( j^{th} \) vector of objective values, and \( f_{ji} \) denotes the value of its \( j^{th} \) objective out of the \( m \) potential objectives. The DM ranks the solutions according to her own preference of the objective values. The vector of rankings is given by \( r \), where \( r_j \) is the rank corresponding to \( f_j \in T \). 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 reference on F-test feature selection see [55]):

Step 1: The correlation \( \rho_i \) between each objective (feature) \( i \) and \( r \) is computed as:

\[
\rho_i = \sum_{j=1}^{T} \frac{(f_{ji} - \bar{f}_i) \cdot (r_j - \bar{r})}{\text{Var}(f_i) \cdot \text{Var}(r)}
\]  

(2)

where \( \bar{f}_i \) and \( \text{Var}(f_i) \) are, respectively, the mean and variance of the \( i^{th} \) objective over all solutions in \( T \); and \( \bar{r} \) and \( \text{Var}(r) \) are the same for the vector of rankings.

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

\[
F_i = \frac{\rho_i}{1 - \rho_i} \cdot (|T| - 2)
\]  

(3)

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 is calculated by any statistical software.

Step 4: Features with lower \( p \)-value are selected. Number of selected features can be either fixed \( k = \hat{m} \) or variable. In the latter case, objectives with \( p \)-values less
Algorithm 1: Uni-Variate Feature Selection

Input:
\[ F \]: Set of all potential objectives
\[ T \]: Set of ranked objective vectors
\[ r \]: vector of ranks

Either
- \( k < |F| \) (for fixed number of objectives) or
- \( \tau \) (for variable number of objectives)

Output: \( \hat{F} \): Selected objectives

1) Fixed number of objectives

- Step 1: Calculate \( p_i \) using Eq. (2)
- Step 2: Calculate \( F_i \) using Eq. (3)
- Step 3: Calculate \( p_i \) (p-value) from \( F_i \)

2) Variable number of objectives then

\[ \hat{F} \leftarrow k \text{ objectives with lowest } p\text{-values} \]

else

\[ \hat{F} \leftarrow \{ f_i \in F \mid p_i < \tau \} \]

return \( \hat{F} \)

Algorithm 2: Recursive Feature Elimination

Input:
\[ F \]: Set of all potential objectives
\[ T \]: Set of ranked solutions
\[ r \]: vector of ranks

Either
- \( k < |F| \) (for fixed number of objectives) or
- \( \tau \) (for variable number of objectives)

Output: \( \hat{F} \): Selected objectives

1) Fixed number of objectives

- Step 1: \( M \leftarrow \text{Build_Model}(T, r, \hat{F}) \)
- Step 2: \( f_j \leftarrow \arg \min_{f_i \in \hat{F}} \phi(f_i) \)
- if \( \phi(f_j) > \tau \) or \( |\hat{F}| = k \) or \( |\hat{F}| = 2 \) then
  break

else

- Step 3: \( \hat{F} \leftarrow \hat{F} \setminus f_j \)

return \( \hat{F} \)

than a predetermined threshold \( \tau \) are selected. 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.

2) Recursive Feature Elimination: RFE is different than uni-variate feature selection in that, here, first logistic regression is used to build a model based on all the features to predict the rankings. In the next step, the feature with the minimum contribution to the constructed model is excluded from the selected subset [56].

The two steps are repeated until a preset number of objectives \( k \) is selected (fixed number of objectives) or the minimum contribution of the remaining objectives is greater than a threshold \( \tau \) (variable number of objectives). There are several ways to measure contribution to the model (\( \phi \)) such as the coefficient of the objective in the model or the importance of the objective. Here, we use the latter to be consistent with uni-variate feature selection. The importance of each feature is calculated based on the drop in the accuracy of the model when that feature is eliminated. The algorithm for RFE is depicted in Algorithm 2.

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.

2) Variable number of objectives: We select the subset of objectives which meet a predetermined threshold \( \tau \). In case of uni-variate feature-selection, the lower the value of \( \tau \), the lower would be the number of objectives with acceptable \( p\)-values. If there is only one objective with a \( p\)-value lower than \( \tau \), the two objectives with lowest \( p\)-values are selected instead. For RFE, in each step an objective is selected only when its contribution to the model is greater than \( \tau \). Thus, higher values of \( \tau \) correspond to less number of active objectives. 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:

- \( k\)-HD: Uni-variate feature selection with fixed number of objectives
- \( \tau\)-HD: Uni-variate feature selection with variable number of objectives
- \( k\)-HDR: RFE with fixed number of objectives
- \( \tau\)-HDR: RFE with variable number of objectives

The proposed methods can be applied to any ranking-based interactive EMOA 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 [57] with our proposed method to show how the method 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

As discussed, our proposed HD/HDR method can be applied to any ranking based interactive method. In this study, BCEMOA is selected and equipped with detection of hidden objectives. BCEMOA [57] is an interactive EMOA based on NSGA-II. The algorithm starts with a population of randomly generated solutions (pop). The population is evolved with NSGA-II for \( N_{ev} \) generations. Next, at each interaction step, the best \( N_{solutions} \) solutions are selected from the evolved population and ranked by the DM. The data along their ranks are then used to train a Support Vector Machine (SVM) model to learn a utility function (\( U_{svm} \)). The evaluations of the learned
UF 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.

Similar to the original BCEOMA, the BCEMOA-HD algorithm, proposed here, starts with a set of active objectives $\hat{F}$. All inactive objectives not in $\hat{F}$ do not need to be evaluated during the optimization and do not participate in dominance ranking and evolution of the population. However, during the interactions, all objectives in $\hat{F}$ 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 identify relevant objectives and update $\hat{F}$. Consequently, the population should also be updated to be evaluated for $f_i \in \hat{F}$. SVM is also used to learn $U_{SVM}$ based on active objectives in updated $\hat{F}$ and their rankings. An overview of BCEMOA-HD is shown in Algorithm 3.

As described above, compared to the original BCEOMA we have modified the algorithm in lines 10 and 11, where feature selection is deployed and the set of active objectives is updated. Another modification was applied to the original BCEOMA 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, the DM when asked to compare pairs of objective vectors actually does not need to be evaluated for $f_i \in \hat{F}$. Consequently, the population should also be updated to be evaluated for $f_i \in \hat{F}$. SVM is also used to learn $U_{SVM}$ based on active objectives in updated $\hat{F}$ and their rankings. An overview of BCEMOA-HD is shown in Algorithm 3.

As described above, compared to the original BCEOMA we have modified the algorithm in lines 10 and 11, where feature selection is deployed and the set of active objectives is updated. Another modification was applied to the original BCEOMA 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, the population should also be updated to be evaluated for $f_i \in \hat{F}$. SVM is also used to learn $U_{SVM}$ based on active objectives in updated $\hat{F}$ and their rankings. An overview of BCEMOA-HD is shown in Algorithm 3.

As described above, compared to the original BCEOMA we have modified the algorithm in lines 10 and 11, where feature selection is deployed and the set of active objectives is updated. Another modification was applied to the original BCEOMA 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 almost zero, which would result in the feature selection considering that objective as irrelevant and replacing it with a different one. To address this issue and to preserve the elitism, 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.

V. Experimental Setup

To evaluate the effectiveness of our proposed method, we design a set of experiments that can comprehensively cover different aspects of the problem of identifying hidden and irrelevant objectives. To do so, we propose a possible way of simulating hidden and irrelevant objectives. Two sets of well-known benchmark problems are selected from the literature. Each problem features various difficulties to different parts of the solution method including convergence towards most preferred point and performance in highly rugged landscapes. In what follows, a detailed description of the design of the experiments is laid out.

In the experiments with variable number of objectives we seek to investigate how the method performs for objective reduction purposes. Thus, for these set of experiments all objectives are active from the start of the run ($\hat{F} = F$), while for fixed number of objectives only designated objectives are active ($\hat{F} \subset F$) at the beginning of the optimization.

A. Synthetic problems with hidden 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 a binary vector $d \in \{0, 1\}^m$ that specifies which objectives are considered by the optimizer, i.e., $d_i = 1$ iff $f_i \in \hat{F} \subset F$, where $f_i$ indicates the $i$th objective function. That is, given a solution $x$, whose objective vector is $f(x) = (f_1(x), \ldots, f_m(x))$, the optimizer only considers $\hat{f}(x) = f(x) \odot d$, where $\odot$ denotes the element-wise product of two vectors. The optimizer is able to change the set of active objectives by changing the vector $d$.

On the other hand, feature selection methods and the DM have access to $f(x)$, that is, the unknown UF used by the DM when asked to compare pairs of objective vectors actually evaluates $U(f(x))$.

B. Benchmark Problems

We consider two well-known numerical and binary benchmark problems, namely, multi-objective NK landscape problems with correlation between objectives ($\rho$MNK) [15] and DTLZ problems [14] 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.

$\rho$MNK problems are used to analyse the effects of correlation among objectives and smoothness of the landscape on

| Algorithm 3: BCEMOA-HD |
|-----------------------|
| **Input:**          |
| $N_{int}$: Total number of interactions |
| $N_{tr}^2$: 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 |
| **Output:** The most preferred solution |
| $T \leftarrow \{\}$ |
| $r \leftarrow \{\}$ |
| $pop \leftarrow$ NSGA-II for $gen_1$ generations (pop) |
| for $1$ to $N_{int}$ do |
| $T_i \leftarrow$ select $N_{tr}^2$ solutions |
| Evaluate solutions in $T_i$ for all objectives in $F$ |
| $r_i \leftarrow$ DM.ranks($T_i$) |
| $T \leftarrow T_i \cup T$ |
| $r \leftarrow r_i \cup r$ |
| $\hat{F} \leftarrow$ feature_selection($T$, $r$, $F$, $\tau$ / $k$) |
| Evaluate $pop$ for $f_i \in \hat{F}$ |
| $U_{SVM} \leftarrow$ train_SVM($T$ with objectives $\in \hat{F}$, $r$) |
| $Crowding\_Distance \leftarrow U_{SVM}$ |
| $pop \leftarrow$ NSGA-II for $gen_i$ generations, only considering $\hat{F}$ |
| return Best (first) solution in the pop ranked by non-domination sorting and $U_{SVM}$ |
the performance of the proposed method. We consider $\rho$ MNK 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 $\rho \geq -1/(m-1)$ [15] 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 $\rho$ MNK problems.

We use DTLZ1, DTLZ2, DTLZ7 from DTLZ test suit, which were also used in the experiments on BCEMOA by its authors [57] and also in [27] for objective reduction. DTLZ1 contains $11^k - 1$ local Pareto-optimal fronts, and each of them can attract the evolutionary algorithm. Thus, it can be used to test the ability of the algorithm to deal with multiple local attractors. DTLZ2 investigates the performance of the algorithm when dealing with many objective proximities and a quality of the solutions in regard with proximity to true PF. 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 [14], the decision space dimension ($n$) is $m+4$ for DTLZ1, $m+9$ for DTLZ2 and $m+19$ for DTLZ7. In 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 [27] 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 by the authors of BCEMOA [57]. Please note that this modification is not needed for DTLZ7 because it does not collapse to a single point.

C. Machine Decision Maker (MDM)

We adhere to the MDM framework introduced in [58] and simulate the DM’s preference with a UF that implicitly expresses which objectives are relevant. We define $c$ as the ordered index set of relevant objectives such that $i < j \rightarrow c_i < c_j$, and consider the following quadratic UFs that were proposed in experiments on the original BCEMOA [57]:

\begin{align}
UF1(f) &= 0.28f_{c_1}^2 + 0.38f_{c_2}^2 + 0.29f_{c_1}f_{c_2} + 0.05f_{c_1} \\
UF2(f) &= 0.6f_{c_1}^2 + 0.05f_{c_1}f_{c_2} + 0.23f_{c_1} + 0.38f_{c_2} \\
UF3(f) &= 0.44f_{c_1}^2 + 0.14f_{c_2}^2 + 0.09f_{c_1}f_{c_2} + 0.33f_{c_1}f_{c_2}.
\end{align}

In addition, we consider the following Tchebychef UF:

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

with 0 as ideal point $f^{*}$. The weights $w_i$ of irrelevant objectives ($i \notin c$) are set to zero while the weights of relevant objectives are selected by trial and error in a way that the most preferred solution is away from the corner points as far as possible. As shown in the utilities above, in all cases, the DM only considers $F_{DM} = \{f_i | i \in c\}$, while other objectives are irrelevant. We explain how we selected relevant objectives for each problem in the next section.

Since the DTLZ objectives are to be minimized and to preserve consistency, we assume minimization for utility values as well; i.e. solutions with lower utility value are preferred by the DM.

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 [27]. Thus, the extended problem and its PF should be given a careful examination before proceeding to experiments. For instance, the PF of DTLZ7 would collapse 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 ($c = \{1, 4\}$). In the case of $\rho$ MNK problems the first two objectives are selected ($c = \{1, 2\}$). For $\rho$ MNK problem with four objectives, having $F_{DM} = \{f_1, f_2\}$ and given an initial $d = (0, 1, 0, 1)$, 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. The projection of the PF and the UF contours are depicted in Figure 1 for DTLZ problems and selected relevant objectives. The location of the most preferred solution is indicated by a triangle and the worst solution with a rectangle. The graphs for UF1 and UF3 are identical to UF2 on DTLZ1 and hence omitted to save space. The same is true for DTLZ2.

E. Evaluation of the results

The experiments are performed in three different modes to enable the assessment of the 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 pairwise comparisons provided by the MDM at each interaction; i.e. at each interaction the MDM uses true UF to rank solutions. Predictions from the learned utility 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 utility only consider the set of active objectives $f(x)$. The set of active objectives never changes, that is, $d$ 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: $k$-HD, $\tau$-HD, $k$-HDR, $\tau$-HDR. Similar to the Only Learning mode, the optimization algorithm relies on non-dominated sorting and an UF that is learned based
solutions at each generation. The total number of generations is 500 and $N^{\text{eva}} = 5$ solutions are shown to the DM at each interaction.

Within BCEMOA, NSGA-II runs for $\text{gen}_1 = 200$ generations before the first interaction and there are $\text{gen}_1 = 30$ generations between subsequent interactions. The total number of generations after the last interaction is calculated by $500 - \text{gen}_1 - \text{gen}_2 (N_{\text{int}} - 1)$. Thus, changing the number of interactions ($N_{\text{int}}$) would not alter total number of generations. We run experiments with 1, 3 and 6 interactions for DTLZ problems. For $\rho\text{MNK}$ problems, we only consider 6 interactions and, instead, we investigate the effect of different levels of correlation ($\rho$) and ruggedness ($K$). Each test was repeated 40 times with different random seed.

2) Implementations: The algorithms, machine DM and $\rho\text{MNK}$ 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 [59], the univariate feature-selection and RFE implementations are based on Scikit-learn 0.23.1 (http://scikit-learn.org/).

VI. EXPERIMENTAL RESULTS & DISCUSSION

We have considered 3 DTLZ benchmark problems and each of them are considered with 3 different values for number of interactions. The $\rho\text{MNK}$ problems are considered with 6 values of $\rho$ and 4 values of $K$. Considering 3 different number of objectives, in total we have investigated 99 problems, each of which are considered with 4 utility functions. Each experiment is repeated 40 times with different random seeds and the average values are reported. Comparison of different modes is done with regard to the utility value of the final solution returned by the algorithm averaged over 40 runs. In this section we focus on the most important findings. Figures that do not include key findings are not presented to save space. However, the complete set of results and figures can be found in the Appendix.

A. DTLZ problems with fixed number of objectives

The results of experiments on DTLZ problems with fixed number of active objectives is illustrated in Fig. 2. When using online detection of hidden objectives with DTLZ1 problem with $m = 4$ or $m = 20$ and fixed number of active objectives, almost no improvement is observed in terms of the utility value compared to Only learning. However, when $m = 10$ the performance of $k$-HD and $k$-HDR is significantly better than Only learning and almost as good as Golden mode. The only exception is when UF2 is used.

For DTLZ2, improvements in the performance can be seen when detection of hidden objectives is active in the case of $m = 10$ and $m = 20$ when utility functions UF1 and UF2 are used. Another important observation is 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. In general, it is observed that the proposed method can significantly improve the utility value of

Figure 1. PF of the selected problems with first and fourth objectives being active depicted over the contour lines of the different UFs. The position of the worst and best solutions is marked respectively with a rectangle and a triangle.
the final solutions. Although in some cases 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 proposed detection method is investigated with regard to objective reduction capabilities and thus the number of active objectives is not fixed. Furthermore, the experiments start with all the objectives being active. The key results of these experiments are illustrated in Figure 3. For DTLZ1 with \( m = 4 \), the \( \tau \)-HD and \( \tau \)-HDR perform better than the only learning mode on Tchebychef UF, while for other UFs they have almost the same performance. With \( m = 10 \) and \( m = 20 \), \( \tau \)-HD and \( \tau \)-HDR performs as good the Golden mode while \( \tau \) is slightly outperformed by \( \tau \)-HD. Results for DTLZ2, are identical to those of DTLZ1; i.e. \( \tau \)-HD and \( \tau \)-HDR still being as good as Golden mode and outperform Only learning mode with \( m = 10 \) and \( m = 20 \). For \( m = 4 \) the performance of \( \tau \)-HD and \( \tau \)-HDR outperforms only learning when UF3 is used, but cannot perform as good as Golden mode.

For DTLZ7 with \( m = 4 \) all algorithms perform as good as Golden mode which suggests the problem may be too easy for the algorithms and optimizing any subset of objectives may lead to optimization of other objectives as well [27]. However, with higher number of objectives, they fail to perform as good as the Golden mode and \( \tau \)-HD and \( \tau \)-HDR outperform Only learning.

C. \( \rho \)MNK problems

The results of experiments on \( \rho \)MNK problems with fixed number of objectives and variable number of objectives are depicted in Figures 4 and 5, respectively. In all of these experiments, detection method fails to make significant improvements in terms of the utility value compared to Only learning mode. However, there are significant savings in terms of computational cost when \( \tau \)-HD and \( \tau \)-HDR are used. In the next Section, this aspect will be discussed in full.

D. Sensitivity analysis and Discussions

1) Power of Detection of True Set of Objectives: In terms of the power of the detection, a heatmap plot is provided in Figure 6. The plot illustrates the frequency of the times
the true set of objectives are activated by the detection of hidden objectives across all experiments on \( \rho \)MNK problems with 10 objectives when \( \tau \)-HD is on. The x-axis indicates the number of interaction within a single run and the y-axis (rows) pertain to different objectives numbered from 1 to 10. Interaction 0 refers to the state of the algorithm before the first interaction, when all objectives are active. At the first interaction all objectives are active. Then immediately after the first interaction most of the objectives are dropped and first and second objectives that are relevant and included in the utility functions, are kept active. It can be easily verified that the \( \tau \)-HD converges fast towards the true set of objectives. Another surprising observation is that after the 5th interaction, almost all objectives become active, although to a lesser degree comparing with the relevant ones. This observation can be justified as follows: When detection of hidden objectives is enabled, the selected objectives might be optimized to near-optimal values and then become fixed in the next generations. The uni-variate correlation of these objectives having fixed values with ranking vector would diminish to almost 0. As a result, the objectives would be identified as irrelevant and substituted with new ones whether or not they are important to the DM.

2) Analysis of Performance and Threshold \( \tau \): As an important parameter of \( \tau \)-HD and \( \tau \)-HDR, \( \tau \) has a direct effect on the number of active objectives and those that are removed. Thus, careful examination should be given in determining this parameter. To inspect the effect of parameter \( \tau \), which indirectly controls the number of active objectives, the DTLZ problems with \( m = 20 \) objectives are solved for different values of \( \tau \). The results for other problems are similar and hence not discussed here. Setting \( m = 20 \) would provide for 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, 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 \( \tau \)-HD improves with lower values of \( \tau \) on DTLZ problems. On the other hand, reducing the \( \tau \) value would reduce the number of active objectives. As explained earlier, in the case where there are fewer than 2 objectives that pass the selection criteria, the two objectives with least \( p \)-values are selected.

Figure 8 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. All changes in the objectives results in improvement of the utility value after each interaction. The changes become smaller as the algorithm converges towards a good solution.

The changes in the number of active objectives after each interaction, averaged over 40 runs, are depicted in Figure 9 where the shaded areas show the 95% confidence interval around the mean. It can clearly be 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.

Lower number of active objectives translates to lower computational costs. However, another important criteria to consider is the ratio of evaluations of relevant objectives to those of irrelevant ones.

\( \tau \)-HD effectively reduces the number of objective evaluations and most of objective evaluations pertain to relevant objectives. As defined, whenever we refer to objective evaluations, we measure the total number of individual objective evaluations. Thus, the number of objective evaluations depend on the number of active objectives. For instance, when \( \tau = 1 \) (equivalent to Only learning mode), an experiment on DTLZ1 with UF3 and \( \tau = 0.2 \), uses 600,000 objective evaluations after the first interaction (before the first interaction everything is identical, thus no comparison is made) and only 10% of these evaluations pertain to relevant objectives. However, when \( \tau \)-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 \( \tau \)-HD or \( \tau \)-HDR is used.

These results prove the effectiveness of the \( \tau \)-HDR in terms of both performance with regard to the quality of the final results and computational efficiency.
VII. Conclusion and Future Work

This study has considered interactive multi-objective problems where only an unknown subset of all the defined 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. Consequently, simulation of irrelevant, hidden, and active objectives are discussed and an efficient method is proposed to deal with these objectives.

Two variants of the method with variable and fixed number of active objectives were studied. The results show that the variant with variable number of objectives can be considered for dimension reduction purposes, reducing efficiently the number of active objectives even after the first interaction; this eliminates unnecessary evaluations of irrelevant objectives and improves computational time and quality of the final solution returned by the algorithm. The variant with fixed number of active objectives also manages to detect and switch to relevant objectives. We also explored the application of recursive feature selection to investigate if further improvements could be achieved. 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 problem suits, it can be observed that improvement in the utility value of the final solutions are more significant for DTLZ problems. However, achievements with regard to saving in objective evaluations and computational costs is similar for both test problems. The benefits of this achievement is highlighted where expensive objective evaluations are avoided. The effect of correlation among objectives was investigated and proved to be insignificant in tests with ρMNK problems.

A sensitivity analysis was performed to scrutinize different aspects of the problem, effect of key parameters of the algorithms on final results, and to provide further insight into the proposed method. 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. The proposed feature selection method only considers linear relations between objectives and DM’s rankings. Considering nonlinear regression in feature selection would be a subject worth to study, although it adds to the computational effort. In the case of BCEMOA, one can use the SVM’s support vector in order to identify most important objectives. However, the intention of this study was to target ranking-based algorithms in general rather than a specific one. We considered 4 different UFs to capture MDMs with different preferences. Future study could expand on this and consider a range of further UFs, such as a Sigmoid UF and others [60]. For some experiments, we observed that once the relevant objectives are optimized, the proposed methods try to explore optimizing other inactive objectives in the hope to find an even better utility. In future studies, it would be desirable to introduce a stopping mechanism to avoid such excess search and thus increase computational efficiency/reduce resource usage.

REFERENCES

[1] 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, 2021.
D. Brockhoff and E. Zitzler, “All are 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, J.-J. Merelo, D. Whitley, and X. Yao, Eds., vol. 4193. Heidelberg: Springer, 2006, pp. 533–542.

E. Zitzler and S. Künzli, “Indicator-based selection in multiobjective search,” in *Parallel Problem Solving from Nature — PPSN VIII*, ser. Lecture Notes in Computer Science, X. Yao et al., Eds. Heidelberg: Springer, 2004, vol. 3242, pp. 832–842.

N. Beume, B. Naujoks, and M. T. M. Emmerich, “SMS-EMOA: Multiobjective selection based on dominated hypervolume,” *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.

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.

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)*. Piscatawy, NJ: IEEE Press, Dec. 2003, pp. 2066–2073.

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.

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

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.

D. Brockhoff and E. Zitzler, “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.

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.

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.

T. Gal and H. Leberling, “Reduced objective functions in linear vector maximum problems and their determination,” *European Journal of Operational Research*, vol. 1, no. 3, pp. 176–184, 1977.

P. J. Agrell, “On reducibility in multi criteria decision making,” *European Journal of Operational Research*, vol. 98, no. 3, pp. 571–586, 1997.

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.

R. E. Steuer, *Multiple Criteria Optimization: Theory, Computation and Application*, ser. Wiley Series in Probability and Mathematical Statistics. New York: John Wiley & Sons, 1989.

R. E. Steuer and L. Gardner, “On the computational testing of procedures for interactive multiple objective linear programming,” in *Operations Research*, G. Fandel and H. Gehrung, Eds. Berlin/Heidelberg: Springer, 1991, pp. 121–131.

R. Allmendinger and J. D. Knowles, “Evolutionary optimization on problems with subjective 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.
[41] D. K. Saxena and K. Deb, “Non-linear dimensionality reduction procedures for certain large-dimensional multi-objective optimization problems: Employing correntropy and a novel maximum variance unfolding,” in *Evolutionary Multi-Criterion Optimization*, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 772–787.

[42] J. Xu, P. P. Pokharel, A. R. C. Paiva, and J. C. Principe, “Nonlinear component analysis based on correntropy,” in *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2006*. IEEE, 2006, pp. 1851–1855.

[43] K. Q. Weinberger, F. Sha, and L. K. Saul, “Learning a kernel matrix for nonlinear dimensionality reduction,” in *Proceedings of the 21st International Conference on Machine Learning, ICML 2004*. C. E. Brodley, Ed. New York, NY: ACM Press, 2004.

[44] K. Q. Weinberger and L. K. Saul, “An introduction to nonlinear dimensionality reduction by maximum variance unfolding,” in *Proceedings of the 21st National Conference on Artificial Intelligence*, A. Cohn, Ed., vol. 6. AAAI Press/MIT Press, Menlo Park, CA, 2006, pp. 1683–1686.

[45] T. Goel, R. Vaidyanathan, R. T. Haftka, W. Shyy, N. V. Queipo, and K. Tucker, “Response surface approximation of Pareto optimal front in multi-objective optimization,” *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 4, pp. 879–893, 2007.

[46] C. Pozo, R. Ruiz-Femenia, J. Caballero, G. Guillén-Gosálbez, and L. Jiménez, “On the use of principal component analysis for reducing the number of environmental objectives in multi-objective optimization: Application to the design of chemical supply chains,” *Chemical Engineering Science*, vol. 69, no. 1, pp. 146–158, 2012.

[47] 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.

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

[49] K. R. GABRIEL, “The biplot graphic display of matrices with application to principal component analysis,” *Biometrika*, vol. 58, no. 3, pp. 453–467, 12 1971.

[50] S. Obayashi and D. Sasaki, “Visualization and data mining of Pareto solutions using self-organizing map,” in *Evolutionary Multi-criterion Optimization, EMO 2003*, ser. Lecture Notes in Computer Science, C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, Eds., vol. 2632. Heidelberg: Springer, 2003, pp. 796–809.

[51] C. A. Mattson and A. Messac, “Concept selection using s-Pareto frontiers,” *AIAA Journal*, vol. 41, no. 6, pp. 1190–1198, 2003.

[52] M. Koppen and K. Yoshida, “Visualization of Pareto-sets in evolutionary multi-objective optimization,” in *7th International Conference on Hybrid Intelligent Systems (HIS 2007)*. IEEE, 2007, pp. 156–161.

[53] 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.

[54] 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.

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

[56] 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.

[57] 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.

[58] 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.

[59] F. Biscani, D. Izzo, and C. H. Yam, “A global optimisation toolbox for massively parallel engineering optimisation,” *Arxiv preprint arXiv:1004.3824*, 2010.

[60] 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.

[61] *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2007, 25-28 September 2007, Singapore*. Piscataway, NJ: IEEE Press, 2007.

[62] *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2008, June 1-6, 2008, Hong Kong, China*. Piscataway, NJ: IEEE Press, 2008.

[63] C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, Eds., *Evolutionary Multi-Criterion Optimization, Second International Conference, EMO 2003, Fano, Portugal, April 2003: proceedings*, ser. Lecture Notes in Computer Science. Heidelberg: Springer, 2003, vol. 2632.