Surrogate Assisted Interactive Multiobjective Optimization in Energy System Design of Buildings

Pouya Aghaei-Pour, Tobias Rodemann, Jussi Hakanen, Kaisa Miettinen

2021

Surrogate Assisted Interactive Multiobjective Optimization in Energy System Design of Buildings

Pouya Aghaei Pour*[1], Tobias Rodemann†[2], Jussi Hakanen‡[1], and Kaisa Miettinen§[1]

[1]University of Jyvaskyla, Faculty of Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla, Finland
[2]Honda Research Institute Europe, Carl-Legien-Strasse, 30, 63073 Offenbach/Main, Germany

Abstract

In this paper, we develop a novel evolutionary interactive method called interactive K-RVEA, which is suitable for computationally expensive problems. We use surrogate models to replace the original expensive objective functions to reduce the computation time. Typically, in interactive methods, a decision maker provides some preferences iteratively and the optimization algorithm narrows the search according to those preferences. However, working with surrogate models will introduce some inaccuracy to the preferences, and therefore, it would be desirable that the decision maker can work with the solutions that are evaluated with the original objective functions. Therefore, we propose a novel model management strategy to incorporate the decision maker's preferences to select some of the solutions for both updating the surrogate models (to improve their accuracy) and to show them to the decision maker. Moreover, we solve a simulation-based computationally expensive optimization problem by finding an optimal configuration for an energy system of a heterogeneous business building complex. We demonstrate how a decision maker can interact with the method and how the most preferred solution is chosen. Finally, we compare our method with another interactive method, which does not have any model management strategy, and shows how our model management strategy can help the algorithm to follow the decision maker's preferences.

Keywords— Model management, Evolutionary interactive methods, Surrogate-assisted optimization, multiobjective optimization, computationally expensive problems

1 Introduction

Real-world optimization problems often contain multiple conflicting objective functions, and we call them multiobjective optimization problems (MOPs). In MOPs, instead of having one optimal solution, we have many so-called Pareto optimal solutions with different trade-offs. Mathematically, all of the Pareto optimal

*pouya.p.aghaei-pour@student.jyu.fi
†tobias.Rodemann@honda-ri.de
‡jussi.hakanen@jyu.fi
§kaisa.miettinen@jyu.fi

solutions are equally good if no additional information is available since vectors cannot be ordered completely. However, one of the Pareto optimal solutions needs to be selected as the outcome of the optimization process to be implemented. Here, we need an expert known as the decision maker (DM) who knows the properties of the problem and can provide preferences and compare different Pareto optimal solutions.

Based on the literature (see, e.g., [1] and [2]), the DM can participate in solving MOPs in three different ways. In a priori methods, the DM expresses one's preferences before the solution process. In the second category, a posteriori methods, the DM selects the final solution after the method provides a set of solutions representing different trade-offs. In the third category, the DM actively interacts with the algorithm and provides preferences during an iterative solution process. In the literature, the last type is referred to as interactive methods.

By using interactive multiobjective optimization methods that involve a DM's preference information, the DM directs the solution process to the regions that one is interested in. A solution pattern is repeated iteratively, and information is provided to the DM at each iteration, who then needs to provide preferences in order to improve solutions from the current iteration. There are many interactive methods in the literature that use different types of preferences (see, e.g., [1, 3]). Using interactive methods can be beneficial in the process of problem-solving because [1]:

1. The DM learns about the interdependencies between the conflicting objectives and the feasibility of one's preferences.
2. The algorithm focuses on those parts of the objective space that are interesting to the DM.

Moreover, since the DM's understanding of the problem grows during the optimization process, one will have more confidence in the final selection.

There exist several types of methods to solve a MOP (see e.g., [1] and [4]). One of the well-known methods is evolutionary multiobjective optimization (EMO) algorithms. EMO algorithms are population-based a posteriori methods where a set of solutions approximating the actual Pareto optimal solutions, is found [4].

Over the years, EMO algorithms have become popular due to certain advantages. For example, they can provide a set of representative solutions in one run, they can handle different kinds of decision variables [4], and they can be applied to objective functions or constraints that are discontinuous or nondifferentiable. Many EMO algorithms have been proposed (see, e.g, [4, 5]). However, usually, evolutionary algorithms need a considerable number of function evaluations. Recently, some interactive EMO algorithms have been developed, where the DM provides preferences iteratively during the solution process to get a set of solutions that is the most preferable (for reviews, see [6, 7, 8]).

Real-world multiobjective optimization problems may involve functions that do not have any analytic formulation. For instance, when we are dealing with simulation-based problems [9, 10], one only gets output for a given input. Then, in some cases, we can use the output directly as the values of the objective functions, and sometimes some post-processing analysis on the output data is needed to calculate the values of the objective functions. Calculating the output may be time-consuming, and such problems are known as computationally expensive multiobjective optimization problems. EMO algorithms are viable for simulation-based problems since we do not necessarily know the properties of the functions involved, but their need for many function evaluations makes solution processes time-consuming.

In this paper, we focus on finding an optimal configuration for the energy system design of buildings, as formulated in [9]. The usage of local energy production and storage facilities has become increasingly interesting both in terms of energy costs and CO_2 emissions. Facility management is, therefore, looking at how to invest in extensions to the current building energy system optimally. Here a simulator is used that has a time-consuming process to generate the outcome [9].

Even though interactive methods have desirable properties, applying them in computationally expensive problems is not straightforward since the DM must wait for solutions corresponding to one's preferences to be generated, which can take hours. Waiting too long may become exhausting for the DM, and this is why it is desirable to speed up the calculation in such problems. One way to reduce the computing time is to approximate the objective functions by analytic functions. In the literature, this is known as surrogate (meta-model)-assisted optimization (see e.g., [11, 12]).

As far as we know, there has been no attempt to tackle the problem of [9] by any interactive methods. Besides, there are only few interactive evolutionary methods in the literature that are suited for computationally expensive problems. Therefore, we develop an interactive method that is suitable for solving computationally expensive multiobjective optimization problems like [9] to show how it provides decision support for the DM in computationally expensive problems. Moreover, there are some algorithms in the literature that motivated our novel interactive method. The first algorithm is the reference vector guided evolutionary algorithm (RVEA) [13] since it has got good results in similar simulation-based problems like [10]. The second algorithm is the surrogate assisted version of RVEA (K-RVEA) [14] where the Kriging models [15] have been used to reduce the computation time. The final method that inspired us is the interactive version of RVEA [16] where RVEA is modified to be able to incorporate the DM's preferences.

Typically, in surrogate-assisted optimization problems, model management (i.e., how to select solutions to evaluate with a computationally expensive function) is used to improve the accuracy of the surrogate models with updating them. Model management is a very crucial part of surrogate-assisted optimization. For instance, solutions computed by the surrogate functions might deviate substantially from the true values, and it is desirable to find the solutions that are following the DM's preferences when they are evaluated by the original objective functions. A good model management strategy can help the surrogate models to make such selection.

The contributions of this paper are two-fold. First, we develop a novel model management strategy that has a smart selection process, where the solutions, which are generated by the surrogate models, will be examined and the ones that have the highest chance of following the DM's preferences are selected to be shown to the DM and update the surrogate models. The second contribution is to show how model management can help an interactive method to follow the DM's preferences better than when there is no model management involved. In other words, we show that by reserving some of the computational resources that we have available for updating the surrogate models, we can provide several solutions that reflect the DM's preferences well.

The rest of this paper is structured as follows. In Section 2, the energy system design problem is briefly described, along with relevant background information. In Section 3, we present a new interactive method for solving computationally expensive problems. In Section 4, we solve the problem presented in Section 2 with our new interactive method and demonstrate the importance of having a model management strategy with some comparisons. Finally, conclusions are drawn and future research directions mentioned in Section 5.

2 Background

Next, we provide some background about notation and terminology, the energy management problem we consider, and the supporting materials for developing our new interactive method.

2.1 Terminology and Notation

The general form of a multiobjective optimization problem (for minimization) is as follows:

$$\begin{aligned} &\text{minimize } \{f_1(x),\ f_2(x),\ \ldots,\ f_k(x)\} \\ &\text{subject to } x \in S, \end{aligned} \tag{1}$$

where the set S is called the feasible region which is a subset of the decision space $\mathbb{R}^n$. We consider k(≥ 2) objective functions $f_i : S \to \mathbb{R}$. For every feasible decision variable vector x, there is a corresponding objective vector $f(x) = (f_1(x), \ldots, f_k(x))^T$, and $f(S)$ is called the feasible objective region which is a subset of the objective space $\mathbb{R}^k$.

As mentioned earlier in Section 1, usually, the objective functions in problem (1) conflict with each other. Hence, not all the objective functions can achieve their optimal values simultaneously. A feasible solution $x^* \in S$ and the corresponding $f(x^*)$ are called Pareto optimal, if there does not exist another feasible solution $x \in S$ such that $f_i(x) \leq f_i(x^*)$ for all $i = 1, \ldots, k$, and $f_j(x) < f_j(x^*)$ for at least one index j. The set of all Pareto optimal objective vectors is called a Pareto front (PF). A feasible solution $x^* \in S$ and the corresponding $f(x^*)$ are called weakly Pareto optimal, if there does not exist another feasible solution $x \in S$ such that $f_i(x) < f_i(x^*)$ for all $i = 1, \ldots, k$.

Assume that the set $X = \{x^1, \ldots, x^m\}$ is an arbitrary subset of feasible solutions in S, and $F = \{f(x^1), \ldots, f(x^m)\}$ the corresponding objective vectors in the objective space. A solution x^i for $i = 1, \ldots, m$ that satisfies the definition of Pareto optimality within the set X, is called a nondominated solution in X [1]. Note that sometimes in the EMO literature, Pareto optimality and nondominance are regarded as synonyms, but this is a more precise distinction. By definition, a Pareto optimal solution is always nondominated but not necessarily vice versa.

In this paper, we have two important concepts, iteration, and interaction. By an iteration, we mean a fixed number of generations, and in this paper, we update the surrogate models at the end of each iteration. Whenever the DM provides preferences, we call it an interaction, and it happens after a fixed number of iterations. For simplicity, every time we evaluate a decision variable vector with the surrogate models, we refer to it as a surrogate evaluation, and every time we use the original expensive objective functions, we use the term function evaluation.

In the method to be proposed, we use an achievement scalarizing function (ASF) [17] to order nondominated solutions based on a given reference point $\hat{z}$. It consists of aspiration levels $\hat{z}_i$ $(i = 1, \ldots, k)$ provided by the DM. There are different ways to formulate an ASF. Here, we use the following formulation to be minimized:

$$\max_{i=1,\ldots,k} [w_i(f_i(x) - \hat{z}_i)] + \rho \sum_{i=1}^{k} w_i(f_i(x) - \hat{z}_i), \tag{2}$$

where k is the number of objective functions, w is some weighting vector with positive fixed values, and $\rho \sum_{i=1}^{k} w_i(f_i(x) - \hat{z}_i)$ with $\rho > 0$ is the augmentation term to avoid finding weakly Pareto optimal solutions [1].

In this paper, we use an ASF as an indicator of how well a given solution is following the DM's preferences (given as a reference point). The lower the ASF value for a given x, the better it is following the DM's preferences [17].

2.2 Simulation-based Problem Considered

Managers of large buildings are confronted with complex investment decisions concerning possible extensions of the energy system, like photovoltaics, stationary batteries, or heat storage. They have to consider a multitude of objectives, for example, investment and annual operation costs and CO_2 emissions.

Here, we want to find an optimal configuration for an energy system of a heterogeneous business building complex. Because of the complex nature of the problem, it is possible to consider different numbers of objective functions and decision variables. For example, the problem considered in [9] consisted of five objective functions and ten decision variables, and a building simulator based on Modelica [18, 19] was used, which is capable of modeling the most relevant real-world effects. Several EMO algorithms were applied to solve this problem [9]. However, no analysis of the final set of solutions was done to determine the DM's most preferred solution. This can be a difficult task since the DM has to choose a solution from a big pool of solutions with different trade-offs.

We have ten real-valued decision variables (see Appendix for more details) whose values are given to the same simulator that was used in [9] as input. Here, we consider four objective functions:

f_1: minimize initial investment cost (in euros),

f_2: minimize annual operation cost (in euros),

f_3: minimize annual CO_2 emissions (in tons), and

f_4: maximize resilience (in seconds),

where resilience is defined as the time the facility can run without grid power. Here, f_1 is independent of the simulator and it is computationally cheap to calculate $f_1(x)$. On the other hand, the other objective functions are computationally expensive, and we need to post-process the simulator's output to calculate them. More details can be found in [9].

We formulate our multiobjective optimization problem as:

$$\begin{aligned}&\text{minimize } \{f_1(x), f_2(x), f_3(x)\},\\ &\text{maximize } \{f_4(x)\}\\ &\text{subject to } 0 \leq x_i \leq 1, \quad i = 1, ..., 10,\end{aligned} \tag{3}$$

where f_i for $i = 2, ..., 4$ are derived from the output of the simulator and x_i for $i = 1, ..., 10$ are the decision variables which only have box-constraints. In what follows, we consider and solve problem (3).

2.3 Related Work

As we mentioned in the previous section, our method is inspired by RVEA, K-RVEA, and interactive version of RVEA. Here, we provide some background on these algorithms.

RVEA

RVEA [13] is a decomposition-based algorithm which divides the objective space into a number of subspaces using reference vectors. The reference vectors are initially generated so that they are uniformly distributed in the feasible objective space, and they are adjusted within the algorithm based on the structure of the PF. RVEA balances between the diversity of the solutions and the convergence towards Pareto optimality by using an angle penalized distance (APD) scalarization [13] to select solutions from different subspaces for the next generation.

RVEA has three main steps. First, generating a set of uniformly distributed reference vectors to divide the objective space to a number of subspaces. Second, using a heuristic algorithm to find solutions in the created subspaces. Third, assigning the solutions found in the previous step to the reference vectors by using APD and then adjusting the positions of reference vectors based on those solutions.

K-RVEA

As mentioned in Section 1, it takes much time to solve a computationally expensive problem with EMO algorithms. A widely used approach for solving computationally expensive problems is to use surrogate functions to approximate the original ones [11, 12]. A surrogate-assisted version of RVEA called K-RVEA was proposed in [14]. K-RVEA assumes that all the objective functions are computationally expensive, and uses Kriging (also known as Gaussian process regression) as a surrogate model. The main idea of Kriging is to predict the values of a function for a given decision variable vector by generating weighted coefficients of the true values of the function in the neighborhood of the decision variable vector. Typically, the computation time for training the Kriging models in population-based EMO is quite high and there might be a need for a model management strategy to limit the size of the training samples like the one mentioned in [14].

A major difference between K-RVEA and RVEA is that in RVEA, the final population is examined to measure the quality of solutions. However, in K-RVEA, an archive is used to store all the function evaluations, and in the end, the solutions that are stored in the archive are examined to determine the quality of the solutions.

K-RVEA consists of three main steps. First, in the initialization step, a sampling method is used to create a training data set in the decision space. Then, the collected samples are evaluated with the original objective functions, and the data, which is stored in an archive, is used to train a surrogate model for each objective function. Second, RVEA is run with the surrogate models instead of the original objective functions. Third, the surrogate models are updated after a certain number of generations by using both APD and uncertainty information, which is provided by the Kriging models (see [14] for more details).

Interactive RVEA

As mentioned earlier, in interactive methods, the DM guides the algorithm to find one's most preferred solution by providing preference information. There are many types of preferences, for example, reference points, classification, pairwise comparisons, and selecting preferred solutions, see, e.g., [1, 2]. An interactive version of RVEA, to be referred to as iRVEA, was proposed in [16]. In iRVEA, the preference information given by the DM is used to adjust reference vectors $V = \{v^1, \dots, v^m\}$ so that the search focuses on solutions reflecting the preferences. For example, if the DM provides a reference point $\hat{z} = (\hat{z}_1, ..., \hat{z}_k)$, an adjusted reference vector $\bar{v}^i$ is created from v^i by the following formula [16]:

$$\bar{v}^i = \frac{r \cdot v^i + (1-r) \cdot v^c}{||r \cdot v^i + (1-r) \cdot v^c||}, \tag{4}$$

where $i = 1, \dots, k$, $||\hat{z}|| \geq 0$ is the Euclidean norm of the reference point which is used for normalization, and $v_j^c = \frac{\hat{z}_j}{||\hat{z}||}$. If $||\hat{z}|| = 0$, then it means that all the objective functions have the same amount of desirability, and we can use the unit vector as the reference vector. The parameter $r \in (0, 1)$ controls how much the reference vectors are adjusted towards the reference point. If r is close to 1, then the reference point has less effect on the reference vectors, and if it is close to 0, they will get closer to the reference point.

3 Interactive K-RVEA

We selected RVEA as the EMO algorithm that we use in our interactive method (called interactive K-RVEA) because it had reasonable results in similar problems [9, 10]. Moreover, we used Kriging models because they provide uncertainty information that is useful for our model management strategy. Kriging models have been used with a priori EMO algorithms before [14] to approximate the whole PF. However, to the best of our knowledge, they have never been used to incorporate the DM's preferences to focus on particular regions of the objective space. To consider Kriging models when applying interactive methods, we must incorporate DM's preferences in model management, which has some challenges. Here, the main point of our model management strategy is that it improves the ability of the method to follow the preferences with respect to (2).

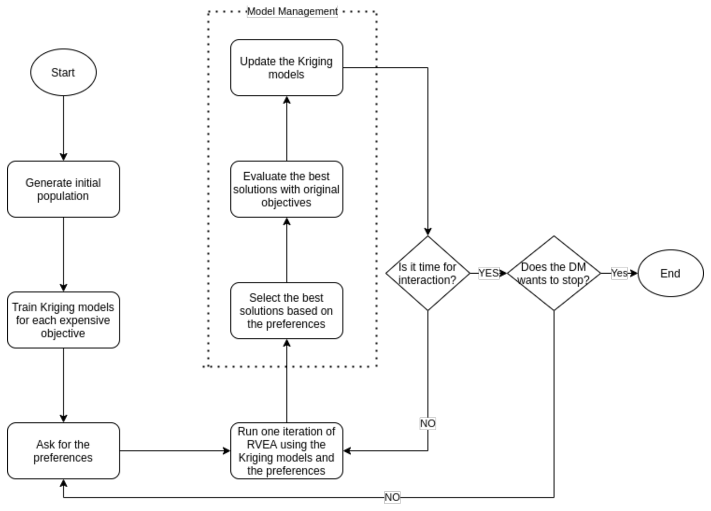
Model Management
Start
Generate initial population
Train Kriging models for each expensive objective
Ask for the preferences
Run one iteration of RVEA using the Kriging models and the preferences
Select the best solutions based on the preferences
Evaluate the best solutions with original objectives
Update the Kriging models
Is it time for interaction?
YES
NO
Does the DM wants to stop?
Yes
NO
End

Figure 1: Flowchart of interactive K-RVEA

Figure 1 presents a flowchart of the main steps of interactive K-RVEA. First, we generate the initial population, evaluate it using the original objective functions, and train a Kriging model for each expensive objective function. Next, the DM provides preferences, and we solve a multiobjective optimization problem (by incorporating the preferences) by replacing original objective functions with the Kriging models. After generating an approximation of a part of the Pareto optimal set reflecting preferences, the accuracy of the Kriging models must be improved to get a better approximation. We propose a model management strategy based on the DM's preferences to update the Kriging models, which is done by selecting solutions that follow the DM's preferences best. The solutions for updating the Kriging models must be evaluated with the original objective functions. Based on how many solutions the DM wants to see at a time, we show to the DM the corresponding number of solutions reflecting the preferences among those evaluated by the original objective functions. Finally, if the DM is satisfied, he/she selects the most preferred solution and the algorithm stops.

As mentioned earlier, there are only few interactive methods that are suited for computationally expensive problems. In this section, we use Kriging models to reduce the computation time and RVEA as an EMO algorithm to build the basis of a new interactive method called interactive K-RVEA. The main contribution to developing interactive K-RVEA is a model management strategy to incorporate the DM's preferences while using the Kriging models.

We have two main steps in developing interactive K-RVEA. First, we must select the type of preferences that the DM is expected to provide, and second, we must select some of the solutions that are found by using Kriging models in a way that when they are evaluated by the original objective functions, they follow best the DM's preferences (at least they are following the DM's preferences better than other available solutions). For the first task, we mentioned in Section 2 that there exist different ways to express one's preferences for interactive methods. After consulting with experts, who deal with problem (3) regularly, we decided to use a reference point to develop our model management strategy because it is intuitive, and they were comfortable with this kind of preference information. Reference vectors could be adapted based on other types of preference information as done in [16], if so desired.

As for the second step, we have to select the solutions that have the highest chance of following the DM's preferences when they are evaluated with the original objective functions. When a solution is evaluated with the original objective functions, it may have different values than with the surrogate models because surrogates tend to contain some approximation error. Besides, evaluating all the solutions that the Kriging models find is not computationally efficient, especially in cases that some of these solutions are not following the DM's preferences. For example, due to the error of surrogate models, a surrogate evaluation of a given decision variable vector could follow the DM's preferences much better (lower ASF value) than when it is evaluated by the original objective functions. Therefore, these kinds of solutions may not be interesting to the DM, and it is ideal to avoid them. Furthermore, in problems like (3), we usually have a particular budget for the number of function evaluations, and it should be spent carefully on the solutions that have a higher probability of following the DM's preferences.

To increase our chances of selecting the best possible solutions for updating the Kriging models, we use two criteria. First, we use ASF to calculate how close each of the nondominated solutions, which are found by using the Kriging models, are to the DM's reference point. Then, we sort the solutions based on the ASF values, and we select $2 * N_U$ solutions (N_U is the number of solutions to update the Kriging models) that are the closest to the DM's preferences. In other words, we select the solutions that have the lowest values in ASF.

So far, we have selected some solutions which have the lowest ASF value. However, since Kriging models provide uncertainty information, we use this additional information as our second criterion. Typically, when the uncertainty information of generated solutions is available, those which have the highest uncertainty are chosen to improve the accuracy of the Kriging model globally [14]. However, in interactive methods, we are looking to search specific parts of the objective space that the DM has shown interest in. Therefore, after selecting the solutions that have lower ASF value, we select N_U solutions among those that have the lowest uncertainty values to update the Kriging models. By incorporating the DM's preferences in the model management strategy along with the uncertainty information, we increase our chances to select the solutions that are following the DM's preferences, both with the Kriging models and the original objective functions. Algorithm 3 shows the main steps of the interactive K-RVEA algorithm, which are discussed in more detail in the following subsections.

Algorithm 1: Interactive K-RVEA

```
Input: $N_V$ = number of reference vectors; $N_0$ = the population size; $t_{max}$ = maximum number of
       generations; $N_U$ = number of solutions to update the Kriging models with; $FE_{max}$ =
       maximum number of function evaluations; $N_{update}$= number of updates between each
       interaction; $N_S$ = number of solutions to be shown to the DM at each interaction (provided
       by the DM); $\bar{N}_S$ = number of solutions from $N_S$ that are closest to the reference point
       (provided by the DM);
Output: Most preferred solution selected by the DM
/* Initialization * /;
1  Generate initial population of size $N_0$ and store them in $A$. Initialize an empty archive $U$ to store
     the solutions and decision variables for updating the Kriging models. Number of solutions selected
     by ASF, $N_{ASF} = 2 \times N_U$;
2  Evaluate initial population with the original objective functions and store the objective values in $A$
     and set number of function evaluations $FE = N_0$;
3  Train a Kriging model for each expensive objective function by using $A$;
4  Ask the DM if he/she wants to see all the nondominated solutions in A, just the ranges, or continue
     the algorithm without any additional information;
5  Ask DM to provide a reference point;
6  Generate $N_V$ uniformly distributed unit reference vectors $V$ and adjust them using (4) ;
/* outer loop */;
7  while $FE < FE_{max}$ do
       /* middle loop */;
8      Initialize a counter $c_u = 0$ for the number of updates;
9      while $c_u < N_{update}$ do
10         Initialize t = 0;
           /*inner loop */;
11         while $t < t_{max}$ do
12             Run steps 3-6 of Algorithm 1 with Kriging models and update $t = t + 1$;
13         Select $N_{ASF}$ solutions from the final population with the lowest ASF value;
14         Choose $N_U$ solutions with the lowest uncertainty from previous step, evaluate the selected
             solutions with the original objective functions, and store them (and their corresponding
             decision variables) in $A$ and $U$;
15         Re-train the Kriging models by using the samples in $A$;
16         Update $FE = FE + N_U$ and $c_u = c_u + 1$;
17     Select $N_S$ solutions from $U$ with the lowest uncertainty values and show them to the DM;
18     Indicate $\bar{N}_S$ closest solutions based on ASF if the DM wants;
19     if the DM is able to select the most preferred solution, go to step 25;
20     Ask the DM to provide a reference point, adjust $V$ using (4), and set $U = \emptyset$ ;
21 if DM increased $FE_{max}$ then
22     Go to step 7;
23 else
24     Show the nondominated solutions in archive $A$ to the DM, and ask the DM to indicate the most
         preferred solution;
25 END;
```

3.1 Inputs

The first input for interactive K-RVEA is the number of reference vectors N_V. In RVEA, the method called simplex-lattice design method [20] is used to generate a given number of reference vectors. In RVEA, as the number of objective functions increases, the number of reference vectors increases as well. For instance, for a problem with three objective functions, 105 reference vectors were used in [13]. In iRVEA, on the other hand, a lower number of reference vectors was used compared to RVEA [16]. For example, for a problem with five objective functions, only 15 reference vectors were used. The reason for choosing a low number of reference vectors in iRVEA is that there is no model management to select the solutions that the algorithm finds, and all of them are shown to the DM. Therefore, if the number of reference vectors increases, the number of solutions that the DM sees will increase as well, and the cognitive load set on the DM grows.

In interactive K-RVEA, we develop a model management strategy that enables the algorithm to choose the solutions that the DM is most interested in. Here, we are not limited to a low number of reference vectors. In fact, we are more interested in increasing the size of N_V because we will have more solutions to choose from, and the chance of finding solutions that follow the DM's preferences increases. Besides, surrogate evaluations are computationally cheap, and therefore, we do not need to worry about the number of solutions that are found by using the Kriging models.

The number of generations (t_{max}) and the number of solutions to update Kriging models (N_U) can be set based on the sensitivity analysis in [14]. The number of updates between each interaction (N_{update}) can be set based on how much time it takes to evaluate N_U solutions with original objective functions. Since FE_{max} is based on N_{update} and N_U, we can use the following formulas to calculate an estimation of FE_{max}

$$FE_{int} = N_U * N_{update}, \tag{5}$$

and

$$FE_{max} = N_0 + \mu * FE_{int}, \tag{6}$$

where FE_{int} is the number of function evaluations that we need for one interaction, and μ is the estimation of the number of interactions that the DM wants to have.

3.2 Initialization

Before the DM starts interacting with the algorithm, the Kriging models should be trained with an initial population. The size of the initial population (N_0) should be set based on the type of problem that we are dealing with and the function evaluation budget that we have. Moreover, since the algorithm has no preferences at the beginning, the Kriging models should be trained globally. Therefore, the initial population (P_0) is generated by using a method (e.g., using Latin hypercube sampling [21]). These samples are evaluated by the original objective functions, and then they are stored in the archive A (along with their corresponding decision variables). Then, the samples in A are used to train independent Kriging models for each expensive objective function.

After training the Kriging models, it is time for the DM to set the first reference point. If the DM does not have information about the problem to be confident about her/his preferences, then, we provide three alternatives to the DM. First, to see all the nondominated solutions in the initial population. Second, to see only the ranges of each objective function for the nondominated solutions in the initial population. Third, to proceed without any further information. The purpose of the first two alternatives is to give some idea to the DM of the feasible solutions and speed up the learning process. However, one should note that no optimization has been done in this stage, and this information is not accurate enough to represent the trade-offs between different objective functions. Finally, after the DM provides the first reference point, the reference vectors are adjusted by using (4) to focus on the regions that the DM is interested in.

3.3 Loops

In Algorithm 3, we have three main loops. The inner loop runs RVEA, the middle loop updates the Kriging models after each iteration, and the outer loop interacts with the DM after each interaction.

In the middle and outer loops, we mostly focus on the model management strategy that was mentioned earlier in this section. As it was mentioned earlier, because Kriging models are not completely accurate, it is possible that some of the solutions that are found are not appealing to the DM. In these two loops, we identify and select the solutions which have the highest chance of following the DM's preferences when they are evaluated with the original objective functions. Then, we use the selected solutions to update the Kriging models.

Inner Loop

In the inner loop, we use Kriging models to replace original objective functions. We run RVEA with the Kriging models for a fixed number of generations (t_{max}), and this parameter should be set high enough so that RVEA can perform a sufficient search of the Pareto optimal set.

Middle Loop

In the middle loop, we select the solutions that we want to evaluate with the original objective functions to update the Kriging models. The selected solutions should improve the Kriging models in regions that the DM is interested in. Here, we manage the solutions that are found by the Kriging models in two phases. In the first phase, we select a number of solutions (N_{ASF}) that are following the DM's preferences while using the Kriging models. If the solutions are not close to the DM's preferences even with the Kriging models, then our selection will involve too much randomness, and the model management becomes unstable. In the second phase, we use the uncertainty information that Kriging provides to select the most accurate solutions (solutions with the lowest uncertainty) from the previously selected solutions and store them in U and A to update the Kriging models. Based on our tests, Kriging models can properly approximate the objective functions of problem (3) (see Appendix). However, the surrogate models have inevitably some error and by going through the two phases mentioned, we increase the probability of selecting solutions that are following the DM's preferences.

Outer Loop

Unlike iRVEA, where the number of solutions shown to the DM (N_S) is the same as the number of reference vectors, here N_S is an independent parameter defined by the DM. Once the Kriging models are updated, ASF is used to select N_S solutions from U, and then they are shown to the DM. Then, the DM has the option of separating the best solutions (with respect to (2)) generated in the current iteration visually ($\bar{N}_S$). Next, either the DM decides to finish the solution process by selecting the most preferred solution or set a new reference point to search for more preferred solutions. At the end of this loop, we reset U to the empty set to prepare it for the next interaction. Note that if $N_S > N_U$, then the algorithm cannot provide enough solutions to be shown to the DM, and all the solutions in N_U are shown to the DM.

These three loops keep running until the function evaluation budget runs out, or the DM terminates the algorithm by finding the most preferred solution. In the first case, if the budget of function evaluations runs out and the DM is not satisfied, he/she can either increase FE_{max}, or as the final alternative (step 24), the DM can ask to see all the nondominated solutions that have been generated so far, which are stored in archive A. Then, one can use visualization tools, such as parallel coordinate plots, to study these solutions, or to provide new value to $\bar{N}_S$ to see the closest solutions to the final reference point visually, and then select the most preferred solution from there.

In the next section, we use interactive K-RVEA to solve problem (3). Besides, we show how the model management strategy that we proposed can provide better decision support for the DM by comparing our algorithm with iRVEA.

4 Numerical Results

Here we describe how we can design an energy system for buildings by using interactive K-RVEA. In what follows, we first describe how we set the parameters of interactive K-RVEA, and then how the DM can interact with this algorithm to solve problem (3). We also incorporate visualizations to support the DM in providing preferences and comparing solutions. To show the results, we used the web-based parallel coordinate plots tool [22].

For parameters that are shared between K-RVEA and interactive K-RVEA such as the number of generations before each iteration ($t_{max} = 20$), the number of samples to update the Kriging models with ($N_U = 5$), and the number of reference vectors ($N_V = 109$), we used the same values that have been used when the K-RVEA algorithm was proposed in [14]. Furthermore, determining the number of iterations before each interaction is one of the important parameters. According to private discussions with experts in the domain of problem (3), DMs should not wait more than three minutes before each interaction. Each time we call the simulator, it takes about ten seconds, and since we update the models with five new solutions (c.f. N_U above), each update takes about one minute (including the training time). Consequently, to have at most three minutes waiting time before each interaction, we can update the models three times ($N_{update} = 3$). Based on Step 21 of Algorithm 3, the DM can increase the maximum number of function evaluations (FE_{MAX}) or terminate the algorithm at any time. Here, we need 109 function evaluations to generate the initial population ($FE_{init} = 109$), and based on equation (5), we set $FE_{min} = 15$. Due to the time limitation that we had, we decided to have six interactions ($\mu = 6$), and hence, based on equation (6), we set $FE_{MAX} = 199$.

The number of solutions that the DM wants to see at each interaction (N_S) is the next parameter that must be set. As we mentioned above, we update the Kriging models three times before we ask for a new reference point, and it means that we can show a maximum of 15 solutions to the DM in one interaction. Here, the DM decided to see all of the solutions that interactive K-RVEA finds in each interaction ($N_S = 15$).

As mentioned in Section 2, in problem (3), calculating the outcome of the first objective function (initial investment cost) is not computationally expensive. Therefore, we use Kriging models only for the other three objective functions. Note that based on discussions with real DMs, one of the authors (TR) provided feedback on presented solutions similar to what we would expect from a real DM.

4.1 Interactive Solution Process

To get started, we generated the initial population randomly and trained Kriging models for expensive objective functions. Then, the DM was asked to provide the first reference point. To support the DM in providing the first reference point, interactive K-RVEA has different options (c.f. step 4). First, he asked to visually see nondominated solutions of the initial population (see Figure 2). Note that the solutions provided in Figure 2 are nondominated solutions from the random initial population, which have not yet been optimized, and they can only give a rough idea of feasible solutions. In addition to the visualization, the DM can naturally always see the numerical values of the selected solutions (N_S) in the form of a table at each interaction. However, in this paper, we only show the parallel coordinate plots during the interactive solution process for compactness. Note that the figures in this section have different scales so that the changes between the solutions can be better seen.

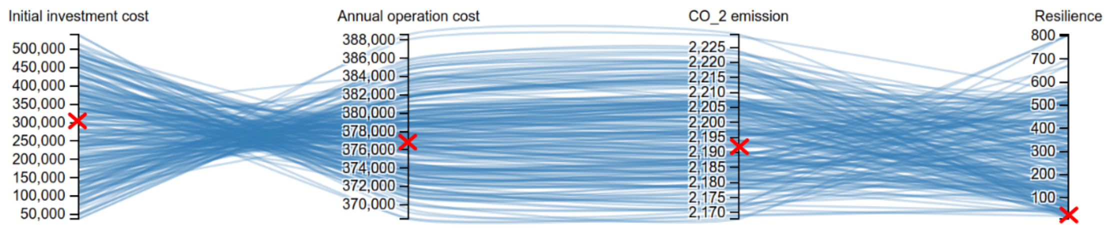

Figure 2: The nondominated solutions in the initial population. Red crosses are the aspiration levels forming the first reference point.

Here, based on the objective functions' ranges shown in Figure 2, and the prior knowledge that f_1 and f_3 (initial investment cost and CO_2 emission) are regarded as the most important objective functions, the DM sets $RP_1 = (298806, 377430, 2194, 28)$ as the first reference point since he believes it is a good compromise for f_1. Components of the reference point are indicated by red crosses in Figure 2. Based on the solutions that were generated after providing RP_1 (see Figure 3), the DM provides $RP_2 = (47950, 382509, 2215, 12)$ as the second reference point because the values of f_1 for the generated solutions are all in this range and he also wants to improve the trade-offs between f_1 and the rest of objective functions.The corresponding aspiration levels are depicted in Figure 3 with red crosses and the previous aspiration levels with orange dots.

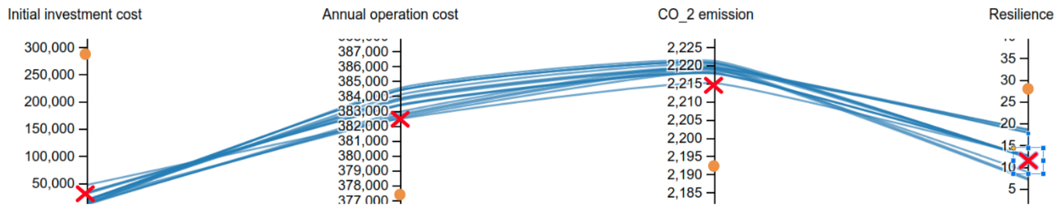

Figure 3: Solutions after the first interaction of interactive K-RVEA. The orange dots are the aspiration levels forming the first reference point, and the red crosses are the aspiration levels forming the second reference point.

Next, the solutions in Figure 4 were generated and presented to the DM. This time, the generated solutions are well spread at around RP_2. However, the trade-offs between f_1 and the rest of the objective functions still are not satisfying. The DM decides not to make a significant change in the reference point to continue searching this region of objective space. He chooses $RP_3 = (37192, 382426, 2219, 152)$ as the third reference point (denoted by red crosses in Figure 4) because based on the generated solutions he knows such a solution is achievable, and it is quite cheaper (it has smaller value for f_1) than RP_2 and it only produces a little more CO_2 than RP_2.

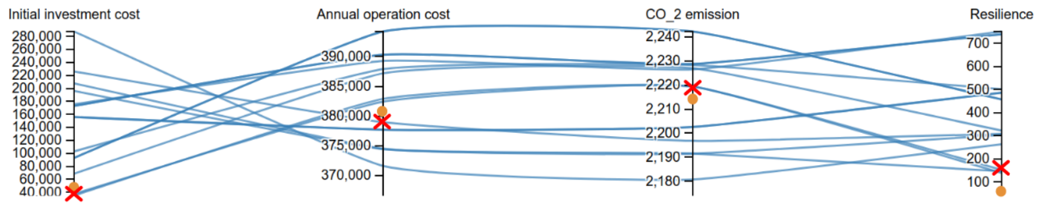

Figure 4: Solutions after the second interaction of interactive K-RVEA. The orange dots are the aspiration levels forming the second reference point, and the red crosses are the aspiration levels forming the third reference point.

Figure 5 shows the solution set that was generated after the third interaction. Now, the DM finds out that the aspiration level for f_1 in RP_2 and RP_3 is too small, and therefore, the trade-offs cannot improve significantly. As for the fourth reference point, the DM makes a compromise and sets $RP_4 = (156067, 377696, 2202, 500)$ to find a more balanced solution.

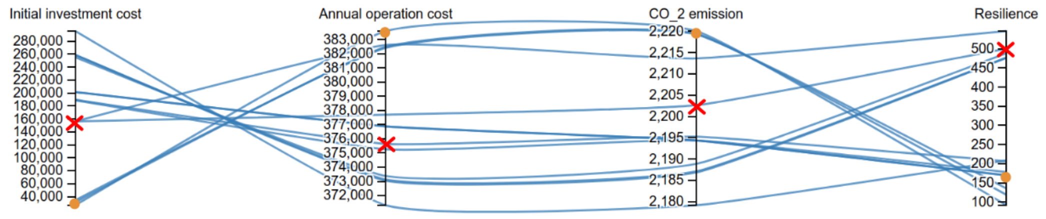

Figure 5: Solutions after the third interaction of interactive K-RVEA. The orange dots are the aspiration levels forming the third reference point, and the red crosses are the aspiration levels forming the third reference point.

Figure 6 shows the the results corresponding to RP_4. Here, the DM was satisfied with the trade-offs and selects $(149886, 380764, 2211, 561)$ as the most preferred solution since it has the same trade-offs as RP_4 but with lower value for f_1.

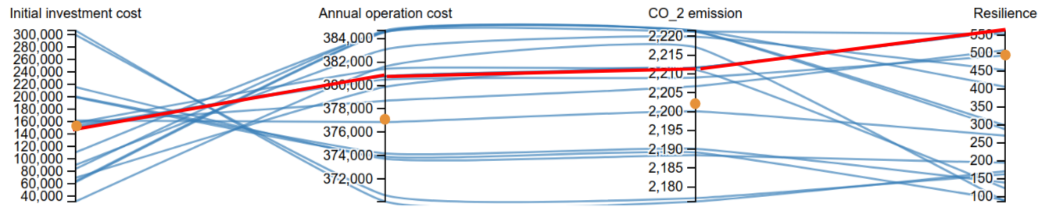

Figure 6: Solutions after the fourth interaction of interactive K-RVEA. The orange dots are the aspiration levels forming the fourth reference point, and the red line is the most preferred solution selected by the DM.

4.2 Performance Evaluation

As mentioned earlier, in this paper, we show how using model management in surrogate models can incorporate DM's preferences in an interactive method to get satisfactory solutions. To show the importance of model management strategies used in this paper, we applied iRVEA with the Kriging models as objective functions and compared the results. However, comparing interactive methods is not a trivial task in the field of multiobjective optimization, and there is no widely accepted way for this.

We used the same reference points (RP_1, RP_2, RP_3, and RP_4) that were used in interactive K-RVEA. Note that interactive K-RVEA used 60 function evaluations to update the Kriging models, and since iRVEA does not update them, we increased the size of the initial population by 60 to have the same number of function evaluations as interactive K-RVEA. Next, we evaluated the final solutions that iRVEA generated with the original objective functions and present the nondominated ones in Figure 7. The final set of solutions generated by iRVEA are more scatter than interactive K-RVEA around the final reference point (RP_3) in Figure 5. Finally, the DM chooses $(367142, 380138, 2273, 45)$ as the final solution since it has the best compromise between the objective functions.

None of the final solutions dominate each other. However, the final solution for interactive K-RVEA has better values than iRVEA for f_1, f_3, and f_4 objective function and only slightly worst value for f_2.

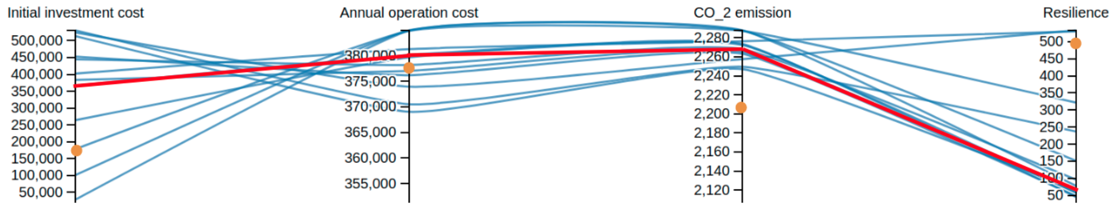

Figure 7: The final solutions of iRVEA. The orange dots are the aspiration levels forming the fourth reference point, and the red line is the most preferred solution selected by the DM.

To compare interactive K-RVEA and iRVEA in terms of following the DM's preferences, we ran both algorithms with the same configuration ten times and used three different ways (ASF, domination and R-metric [23]) to evaluate their performance. Experiments were run on a laptop with core i7 CPU, using 16 GB of RAM, and the running OS was Linux (Ubuntu).

Computation Time

In Table 1, we present the total computation times for both algorithms without considering the decision making time. Interactive K-RVEA and iRVEA included the same number of function evaluations. However, interactive K-RVEA had the model management, where Kriging models were updated. On the other hand, iRVEA used all the function evaluations for the initial population and the solution process only used the surrogate evaluations. Therefore, the computation time for interactive K-RVEA was a bit higher than for iRVEA.

As far as waiting time is concerned, we updated the Kriging models iteratively in interactive K-RVEA. On the other hand, there was no update for iRVEA, and therefore, the waiting time of iRVEA was shorter. However, the waiting time for both methods was under three minutes, that met the DM's time limitation.

ASF

We recorded the ASF values for the final set of solutions (see Table 2) to measure how close they were to the final reference point. In all of the independent runs, interactive K-RVEA had lower ASF values than iRVEA,

Table 1: Average of computation time of interactive K-RVEA and iRVEA between interactions (in seconds). The best results are highlighted in boldface.

Algorithm	min	mean	max
Interactive K-RVEA	520	552	575
iRVEA	**511**	**535**	**567**

which means that interactive K-RVEA had a better convergence towards DM's preferences than iRVEA.

Table 2: ASF values for the 10 independent runs with interactive K-RVEA and iRVEA. The best results are highlighted.

	Best	Mean	Worst
Interactive K-RVEA	**0.41**	**0.53**	**0.59**
iRVEA	0.71	0.77	0.82

Domination

Here, we checked to see if iRVEA solutions dominate the final set of solutions generated by interactive K-RVEA. In all ten runs, none of the solutions provided by iRVEA dominated any of the solutions that were generated by interactive K-RVEA. However, this was not the case when we checked the inverse situation. In other words, in all ten runs, we could find at least one solution generated by iRVEA that was dominated by one or multiple solutions that interactive K-RVEA generated. In Table 3, we show how many of the final solutions of iRVEA were dominated by the final solutions of interactive K-RVEA for ten independent runs.

Moreover, we merged all the solutions generated in the ten independent runs for both methods and checked how many nondominated solutions were generated with each method. Furthermore, iRVEA had 108 nondominated solutions and dominated only seven solutions generated by interactive K-RVEA. On the other hand, interactive K-RVEA had 117 nondominated solutions and dominated 31 solutions that were generated by iRVEA. The number of nondominated solutions generated by interactive K-RVEA is still more significant than iRVEA, which shows that the model management strategy used in interactive K-RVEA helps the method provide more nondominated solutions than iRVEA.

Table 3: iRVEA final solutions that are dominated by interactive K-RVEA.

	Number of dominated solutions by the other algorithm		
	Best	Mean	Worst
Interactive K-RVEA	**0**	**0**	**0**
iRVEA	7	4.7	2

As we showed in Figure 7 and Table 2, the solutions generated by iRVEA were more scattered than by interactive K-RVEA, which means that interactive K-RVEA followed the DM's preferences better than iRVEA. Besides, when the DM interacts with interactive K-RVEA, all the solutions that he works with are evaluated with the original objective functions, but when the DM interacts with iRVEA, the solutions are evaluated by the Kriging models. Hence, the DM cannot be sure that when the final set of solutions (generated with iRVEA) is evaluated with the original objective functions, it will follow the DM's preferences and before (when it was evaluated with surrogate functions).

R-metric

Finally, we used a well-known R-metric indicator, which evaluates the quality of a set of solutions with respect to a reference point. Originally, R-metric was developed for a priori methods to compare different sets of solutions, but since it includes a reference point, we apply it for the final set of solutions of interactive K-RVEA and iRVEA. To compare two sets of solutions, R-metric takes four main steps. First, we remove the common solutions between the two sets. Second, based on the closeness of solutions to DM's reference point (Δ), we remove some of the solutions that do not represent the region of interest in the objective space. Third, we transfer the solutions into a virtual position concerning the reference point using ASF, and finally, we use an indicator like hypervolume to evaluate the quality of the solutions. For details, see [23].

For the second step of R-metric, we must set a value for Δ. Initially, the value of Δ is set as an arbitrary number in [23]. However, since there does not exist a widely accepted way to set this value, we decided to analyze the results with three different values of Δ with respect to the last reference point (RP_4), and create a vector for Δ, representing separate exploration rates for each objective function. Here, we add 10, 15, and 20 percent to the aspiration levels of RP_4 to create the vector Δ. Note that we remove the solutions that are exceeding Δ in at least one objective function. We calculated the R-metric by using the hypervolume indicator for each method's ten independent runs, normalized the hypervolume values, and present the results in Table 4. Moreover, a pairwise two-tailed t-test [24] was conducted between the two interactive methods for the R-metric results. The significance level of our testing was set at %5. In Table 4, $\uparrow$ indicates that the statistical significance of the pairwise comparison between interactive K-RVEA and iRVEA is significant in favor of interactive K-RVEA.

Table 4: Results of R-metric for interactive K-RVEA and iRVEA. The best results are highlighted.

	Interactive K-RVEA				iRVEA		
Δ	Best	Mean	Worst		Best	Mean	Worst
$1.10 * RP_4$	**0.81**	**0.71**	**0.65**	$\uparrow$	0.23	0.11	0.00
$1.15 * RP_4$	**0.88**	**0.79**	**0.72**	$\uparrow$	0.28	0.19	0.12
$1.20 * RP_4$	**0.98**	**0.90**	**0.82**	$\uparrow$	0.42	0.35	0.27

As it is shown in Table 4, interactive K-RVEA is performing better than iRVEA. Table 4 shows that for the first value of Δ, iRVEA might generate zero solutions (for the worst case), which means none of the solutions generated by iRVEA were in the region determined by Δ. Moreover, for the first and second values of Δ, interactive K-RVEA is getting much higher R-metric values than iRVEA, which shows that more solutions are generated by interactive K-RVEA that are concentrating on the regions around RP_4. In addition, for the third value of Δ, iRVEA's performance gets much better than the previous values of Δ, which is in line with the fact that solutions are generated with iRVEA are more scattered than interactive K-RVEA. However, interactive K-RVEA is still obtaining much higher R-metric values than iRVEA. We did not continue with higher values of Δ since we wanted to analyze how each method can generate solutions close to the DM's reference point, and based on the results above, interactive K-RVEA is doing a better job than iRVEA.

5 Conclusions

In this paper, we developed a novel evolutionary interactive multiobjective optimization method, called interactive K-RVEA, that is suitable for real-world computationally expensive problems. As integral elements of the new method, we chose the RVEA algorithm as our optimizer and the Kriging models as surrogate models. We developed a novel model management strategy that incorporates the DM's preferences (reference point in our case) in the Kriging models.

We demonstrated the performance of the developed method by solving a computationally expensive simulation-based problem where our goal was to find an optimal configuration for an energy system of a heterogeneous business building complex, and we were able to generate a reasonable solution, which had better values than the final reference point provided by the DM except for the second objective. We demonstrated how the decision maker can interact with the method and how the most preferred solution is chosen. Then, we compared the results produced by interactive RVEA that has no model management strategy. We ran both algorithms for ten independent runs and considered three different performance indicators (achievement scalarizing function, domination, and R-metric). We showed the importance of having a model management strategy for computationally expensive problems. Besides, we demonstrated that interactive K-RVEA followed the decision maker's preferences better than interactive RVEA. Thanks to interactive K-RVEA, very good results were generated without spending too much of computational resources.

In this paper, we fixed the values of most of the parameters in interactive K-RVEA, and developing an adaptive method to change these values during optimization is one of our future research directions. Another possible future research topic is to address different types of preferences and make interactive K-RVEA compatible with them. Here, the challenge is how to develop a model management strategy that can use different preferences and incorporate them within the surrogate models.

Acknowledgment

This work was partly supported by Honda Research Institute Europe. This research was partly supported by the Academy of Finland (grant no 311877) and is related to the thematic research area DEMO (Decision Analytics utilizing Causal Models and Multiobjective Optimization, jyu.fi/demo) of the University of Jyväskylä. Interactive K-RVEA will be implemented in the open-source software DESDEO (https://desdeo.it.jyu.fi/) funded by the Academy of Finland.

Appendix

In this Appendix, first, we describe the decision variables of the simulation-based problem (3). Then, we show the performance of Kriging models for problem (3).

Decision variables

The simulator uses decision variables to calculate parameters of four different investment options. Then, based on the parameters and the investment options, we can calculate the objective functions values (f_2, f_3 and f_4). The investment options are as follows:

1. A photovoltaic (PV) system on the building roof or carport.
2. An extension of the internal heat storage.
3. A stationary battery.
4. Optimization of the operation of co-generator for heat and power (CHP).

The first three decision variables are related to the PV system: x_1 is the inclination angle, x_2 is the orientation angle, and x_3 is the peak output power of the PV system. The next two decision variables, x_4 and x_5, control the stationary battery's capacity and the maximum charging/discharging power. The minimum and maximum battery state of charge are maintained by x_6. Then, the battery has a charging and discharging threshold connected to the next two decision variables x_7 and x_8. Next, the decision variable x_9 is used to calculate the size of the heat storage. Finally, the CHP generator will only turn on if the ambient temperature is below a certain level, which defines the final decision variable x_{10}. For a full explanation of the decision variables, see [9].

Surrogate models

Here, we show that Kriging models are suitable for problem (3) with different initial population sizes. We tested different Kriging models (with different kernels) along with five other well known surrogate models [12, 25]. First, we used Kriging with normal, radial basis function (RBF), rational quadratic (RQ), exponential sine squared (ESS) and Matern kernels. Second, we used support vector regression (SVR) [26] with linear, RBF and polynomial kernels. Last, we used random forest [27] and Bayesian surrogates [28]. We tested these surrogates with different training sample sizes. The first sample size was set as 35 based on [9]. Then, we doubled the sample size. According to [29, 30], for n decision variables, the initial population should be $11n - 1$. So we used the same logic to choose the third sample size, which was 109.

In Tables 5, 6 and 7, one can see the results for different sample sizes for all the surrogate models that were tested (the best results are highlighted in boldface in each table). It is worth mentioning that each model was trained ten times, where each time the training sample was selected randomly (in the feasible space), and the R^2 value [31, 32] was used to evaluate how accurate the surrogates were. In the tables we show the average of these ten runs. Besides, a random assign algorithm was used to create a sample pool (for all sample sizes) for training the surrogate models. Here, 70 percent of the sample size was used to train the surrogates, and the remaining 30 percent was used to test them.

	Annual operation cost	Annual CO_2 emissions	Resilience
Bayesian	0.711	0.856	0.618
Random forest	0.626	0.687	**0.802**
SVR-linear	0.454	0.368	0.516
SVR-RBF	0.498	0.435	0.482
SVR-polynomial	0.457	0.625	0.664
Kriging-Default	0.734	0.904	0.523
Kriging-RBF	0.756	0.908	0.540
Kriging-Matern	**0.768**	0.911	0.625
Kriging-ESS	**0.768**	**0.914**	0.600
Kriging-RQ	0.765	0.911	0.265

Table 5: Average of the R^2 values for different surrogate models with sample size 35.

	Annual operation cost	Annual CO_2 emissions	Resilience
Bayesian	**0.912**	0.921	0.788
Random forest	0.869	**0.922**	0.768
SVR-linear	0.498	0.400	0.498
SVR-RBF	0.645	0.578	0.651
SVR-polynomial	0.747	0.704	0.784
Kriging-Default	0.764	0.914	0.632
Kriging-RBF	0.788	0.917	0.592
Kriging-Matern	0.883	0.901	**0.816**
Kriging-ESS	0.760	0.810	0.762
Kriging-RQ	0.775	0.921	0.545

Table 6: Average of the R^2 values for different surrogate models with sample size 70

	Annual operation cost	Annual CO_2 emissions	Resilience
Bayesian	0.827	0.887	0.788
Random forest	**0.836**	0.864	0.768
SVR-linear	0.476	0.471	0.516
SVR-RBF	0.745	0.564	0.683
SVR-polynomial	0.765	0726	0.767
Kriging-Default	0.689	0.723	0.727
Kriging-RBF	0.689	0.834	0.727
Kriging-Matern	0.825	**0.893**	**0.800**
Kriging-ESS	0.754	0.854	0.762
Kriging-RQ	0.795	0.891	0.698

Table 7: Average of the R^2 values for different surrogate models with sample size 109

As one can see, SVR surrogates did not perform as well as the others. This could be because of their hyper-parameter tuning. On the other hand, Kriging had the best performance for at least two objectives with different training sample sizes. Besides, the uncertainty information that Kriging provides can be utilized in interactive K-RVEA. Moreover, these results are only based on the initial populations, and the performance of Kriging models will improve as we update them during the solution process. Based on the results provided, we could conclude that Kriging models have competitive performance, and we selected them to be used in the interactive K-RVEA method.

References

[1] K. Miettinen, *Nonlinear Multiobjective Optimization.* Kluwer Academic Publishers, 1999.

[2] C.-L. Hwang and A. S. M. Masud, *Multiple Objective Decision Making Methods and Applications: a State-of-the-Art Survey.* Springer, 1979.

[3] K. Miettinen, J. Hakanen, and D. Podkopaev, "Interactive nonlinear multiobjective optimization methods," in *Multiple Criteria Decision Analysis* (S. Greco, M. Ehrgott, and J. R. Figueira, eds.), pp. 927–976, Springer, 2nd ed., 2016.

[4] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms.* John Wiley & Sons, 2001.

[5] J. Branke, K. Deb, K. Miettinen, and R. Słowiński, eds., *Multiobjective Optimization: Interactive and Evolutionary Approaches.* Springer, 2008.

[6] H. Wang, M. Olhofer, and Y. Jin, "A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges," *Complex & Intelligent Systems*, vol. 3, no. 4, pp. 233–245, 2017.

[7] 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. 41256–41279, 2018.

[8] R. C. Purshouse, K. Deb, M. M. Mansor, S. Mostaghim, and R. Wang, "A review of hybrid evolutionary multiple criteria decision making methods," in *Proceedings of the 2014 IEEE Congress on Evolutionary Computation*, pp. 1147–1154, 2014.

[9] T. Rodemann, "A comparison of different many-objective optimization algorithms for energy system optimization," in *Applications of Evolutionary Computation* (P. Kaufmann and P. Castillo, eds.), pp. 1–16, Springer, 2019.

[10] R. Cheng, T. Rodemann, M. Fischer, M. Olhofer, and Y. Jin, "Evolutionary many-objective optimization of hybrid electric vehicle control: From general optimization to preference articulation," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 2, pp. 97–111, 2017.

[11] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.

[12] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, "A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms," *Soft Computing*, vol. 23, no. 9, pp. 3137–3166, 2019.

[13] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

[14] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya, "A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 129–142, 2018.

[15] J. Sacks, S. B. Schiller, and W. J. Welch, "Designs for computer experiments," *Technometrics*, vol. 31, no. 1, pp. 41–47, 1989.

[16] J. Hakanen, T. Chugh, K. Sindhya, Y. Jin, and K. Miettinen, "Connections of reference vectors and different types of preference information in interactive multiobjective evolutionary algorithms," in *Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8, 2016.

[17] A. P. Wierzbicki, "The use of reference objectives in multiobjective optimization," in *Multiple Criteria Decision Making Theory and Application* (G. Fandel and T. Gal, eds.), pp. 468–486, Springer, 1980.

[18] P. Fritzson and P. Bunus, "Modelica-a general object-oriented language for continuous and discrete-event system modeling and simulation," in *Proceedings of the 35th Annual Simulation Symposium*, pp. 365–380, IEEE, 2002.

[19] R. Yang and L. Wang, "Multi-objective optimization for decision-making of energy and comfort management in building automation and control," *Sustainable Cities and Society*, vol. 2, no. 1, pp. 1–7, 2012.

[20] J. A. Cornell, *Experiments with Mixtures: Designs, Models, and the Analysis of Mixture Data.* John Wiley & Sons, 2011.

[21] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 42, no. 1, pp. 55–61, 2000.

[22] "Web-based parallel coordinate plot." Avalable: https://dgoldri25.github.io/Categorical-ParallelCoordinatePlot/. (accessed: 13.2.2019) [online].

[23] K. Li, K. Deb, and X. Yao, "R-metric: Evaluating the performance of preference-based evolutionary multiobjective optimization using reference points," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 6, pp. 821–835, 2017.

[24] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[25] T. Bartz-Beielstein and M. Zaefferer, "Model-based methods for continuous and discrete global optimization," *Applied Soft Computing*, vol. 55, pp. 154–167, 2017.

[26] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in Neural Information Processing Systems*, pp. 155–161, 1997.

[27] A. Liaw, M. Wiener, *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[28] K. W. Fornalski, "Applications of the robust bayesian regression analysis," *International Journal of Society Systems Science*, vol. 7, no. 4, pp. 314–333, 2015.

[29] J. Knowles, "ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50–66, 2006.

[30] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by MOEA/D with Gaussian process model," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 456–474, 2010.

[31] J. H. Torrie, *Principles and Procedures of Statistics: with Special Reference to the Biological sciences.* McGraw-Hill, 1960.

[32] S. A. Glantz and B. K. Slinker, *Primer of Applied Regression and Analysis of Variance.* McGraw-Hill, 1990.