

A Comparison-Relationship-Surrogate Evolutionary Algorithm for Multi-Objective Optimization

Christopher M. Pierce^{a,*}, Young-Kee Kim^a, Ivan Bazarov^b

^a*University of Chicago, 5640 S Ellis Ave., Chicago, 60637, IL, USA*

^b*Cornell University, 245 Feeney Way, Ithaca, 14853, NY, USA*

Abstract

Evolutionary algorithms often struggle to find well converged (e.g small inverted generational distance on test problems) solutions to multi-objective optimization problems on a limited budget of function evaluations (here, a few hundred). The family of surrogate-assisted evolutionary algorithms (SAEAs) offers a potential solution to this shortcoming through the use of data driven models which augment evaluations of the objective functions. A surrogate model which has shown promise in single-objective optimization is to predict the “comparison relationship” between pairs of solutions (i.e. who’s objective function is smaller). In this paper, we investigate the performance of this model on multi-objective optimization problems. First, we propose a new algorithm “CRSEA” which uses the comparison-relationship model. Numerical experiments are then performed with the DTLZ and WFG test suites plus a real-world problem from the field of accelerator physics. We find that CRSEA finds better converged solutions than the tested SAEAs on many of the *medium-scale, biobjective* problems chosen from the WFG suite suggesting the “comparison-relationship surrogate” as a promising tool

*cmpierce@uchicago.edu

for improving the efficiency of multi-objective optimization algorithms.

Keywords: Surrogate-Assisted Evolutionary Algorithm (SAEA), Expensive Optimization, Medium-Scale Multi-objective Optimization, Comparison-Relationship Surrogate, Pairwise Comparison.

1. Introduction

Across a diversity of fields, the users of optimization algorithms often face problems that involve multiple expensive-to-evaluate objective functions [1, 2, 3, 4, 5]. The expense is usually the time taken to numerically evaluate the objectives. However, more and more, it can also be a monetary cost associated with a real-world measurement [6, 7, 8]. For users operating on a budget, this requires the optimization task to be completed using only a small fixed number of function evaluations. In the context of this paper, we take that number to be a few hundred in total.

Solving multi-objective optimization problems with potentially black-box functions is commonly achieved using a genetic algorithm. Unfortunately, these methods are considered by many to be sample inefficient and can require a large number of function evaluations (extending to the tens of thousands) to produce a reasonable set of solutions [9, 10]. When faced with a limited budget, this inefficiency may force users to sacrifice the quality of their results. Additionally, some problems may be so expensive that they are not solvable, even approximately, within the user's limits. This points to a need for new efficient multi-objective genetic algorithms that produce well-converged solutions using only a small number of function evaluations.

A popular approach to increasing the sample efficiency of multi-objective

genetic algorithms is to introduce a surrogate model: a fast-to-evaluate model of the objectives based on already evaluated data. These surrogate models (also called meta-models) can act as a stand-in for the expensive objective function evaluations in existing optimization algorithms. For example, in K-RVEA [11], a model of the objective functions is generated from an initially evaluated set of solutions and the optimization is performed against the model using a reference-vector-based evolutionary algorithm. The model is then periodically updated with real calls to the objective functions for candidate solutions. For a review of surrogate-assisted evolutionary algorithms (SAEAs), readers are pointed to [9, 12, 13].

Instead of directly modeling the objective functions' values, a possible alternative is to build a surrogate of their comparison relationships. That is, a model of the boolean function,

$$c_i(\vec{x}_1, \vec{x}_2) := f_i(\vec{x}_1) < f_i(\vec{x}_2), \quad (1)$$

where \vec{x}_1 and \vec{x}_2 are the decision variables for the two candidate solutions being compared, and f_i are the objective functions. The problem of building an accurate surrogate model has now become one of classification instead of regression and the domination relationships can be inferred from the comparisons between objective functions. In this work, we explore integrating this type of model into a multi-objective genetic algorithm to improve its efficiency in solving expensive optimization problems.

The main contributions of this work can be summarized as the following.

1. A surrogate model of the comparison relationships between the objec-

tive functions of two solutions is proposed for solving expensive multi-objective optimization problems. This model is implemented using a neural network with an architecture designed to reflect the symmetry properties of the comparison operator.

2. A method of reconstructing the ranking of solutions from their approximate comparisons (which may not satisfy the transitive property) is explored.
3. The model and its associated improvements are integrated into a new SAEA (called “CRSEA”) which is then compared experimentally on a set of benchmark problems to contemporary algorithms for solving expensive multi-objective optimization problems on a budget.

The rest of this article is dedicated to exploring the use of a comparison-relationship-surrogate model to improve the sample efficiency of multi-objective genetic algorithms. In Sec. 2, we review the existing literature surrounding surrogate-assisted evolutionary algorithms (SAEAs) which are powered with classification-based models and the state of comparison-relationship models for optimization. We describe the new SAEA that uses a model of the comparison relationships in Sec. 3. The construction of the model using a neural network is also described and architectural / training details related to the symmetries of the comparison operator are discussed. In Sec. 4, we report on numerical experiments that compare the new algorithm to a diverse set of existing SAEAs. These experiments are performed on several well-known suites of test problems. Results from the algorithm on an example real-world optimization problem are also given in Sec. 5. We conclude the paper in Sec. 6 and give suggestions for future areas of study related to

the comparison-relationship-surrogate optimization algorithm.

2. Related Work

2.1. Regression-Based SAEAs

One approach to building a data-driven model of the already evaluated solutions is to treat the situation as a regression problem. We can attempt to predict the values of the objective functions or metrics related to them using the decision variables.

The algorithm ParEGO [14], for example, uses a Gaussian process model of a scalarization function of the objectives. New points are chosen to maximize the expected improvement of the solution in a manner similar to the kriging method [15]. SMS-EGO extended the idea to maximize improvement in the hypervolume (S-metric) instead [16]. In K-RVEA, the regression model is used directly in place of the expensive objective function evaluations in a reference-vector guided evolutionary algorithm [11]. Multi-objective optimization problems are first decomposed into single objective sub-problems in the algorithm MOEA/D-EGO [17]. A Gaussian process is used to model each of the scalar sub-problems which are then solved simultaneously to suggest new candidate solutions to evaluate.

2.2. Classification-Based SAEAs

Surrogate assisted evolutionary algorithms driven by classification models have shown some success in improving the efficiency with which expensive optimization problems can be solved on a budget. In early work, the algorithm Pareto-SVM [18] was introduced which uses a support vector machine

to classify regions of decision space into those which map to points in objective space that are part of the dominated set, the current Pareto front, or unexplored regions. A variant of the algorithm [19] followed shortly after and introduced a new classification model that accepts pairs of solutions and approximates their domination relationship directly. The themes of classifying either regions of decision space or the relationships between pairs of solutions can be seen throughout later work and we use them to help organize the remainder of this section.

2.2.1. Decision Space Classification

We begin by taking a look at some of the algorithms that use classification models that accept a single point in decision space as their input. In CPS-MOEA [20], the solutions in the training set are labeled into equally sized positive and negative classes using a combination of non-domination rank and crowding distance. A k-nearest neighbor classifier is then trained on the data and used to pre-filter candidate solutions before evaluation. The algorithm MOEA/D-SVM [21] explored using a scalarization function on the objectives to split the population into positive and negative classes. In one proposed algorithm [22], a “one-class classification model” avoided the problem of splitting the data by only considering positive training examples. CSA-MOEA [23] explored using a decision tree to sort regions of decision space into those which were and were not promising to find candidates in. New infill strategies were explored for choosing which candidates to evaluate. Multiple surrogate models were applied to solve “local” problems in a decomposition-based optimization algorithm to mitigate overfitting in MCEA/D [24]. The algorithm CR-SAEA [25] augments the multiple local

classification models approach with a global regression model to help guide its decomposition-based genetic algorithm.

2.2.2. Relationship-Based Surrogates

Modeling the relationships between solutions (e.g. domination) is also an important theme in multi-objective optimization algorithms aided by classification models. The concept of modeling the Pareto-dominance relationship was explored further in some work [26] which compared ten off-the-shelf classification models for use in this task. The algorithm CSEA [27] was introduced which uses a surrogate model to estimate domination between candidate solutions and selected already evaluated reference solutions. The algorithm also uses a measure of uncertainty in the predictions to choose which candidates to evaluate. Two surrogate models: one of the Pareto-domination relationships and a second of the θ -domination relationships between candidates are combined in the algorithm θ -DEA-DP [28]. In REMO [29], the relationships between candidate solutions are modeled with a neural network classifier trained on pairs of solutions which are grouped into equal sized sets based on domination. A related algorithm, named CREMO [30], also explored relationship learning using a convolutional neural network on the decision variables. In FAEA-SDR [31], the strengthened dominance relationship between solutions is modeled instead. Pairwise comparisons of the scalarized objective functions are learned and used in the optimization algorithm PC-SAEA [32]. The concept of domination modeling has also been applied to neural architecture search with promising results [33].

2.2.3. Comparison-Operator Models

Some work has been put into modeling the comparison relationships between the objective functions directly. That is, the value of the comparison operator evaluated on the objective functions between pairs of solutions. The use of these models has remained restricted to the case of single-objective optimization or constraint satisfaction problems. One such study [34] showed an algorithm that ranked solutions using a support vector machine could outperform evolutionary algorithms aided with regression models on some benchmark problems. Binary relationship learning for the pre-selection of solutions was investigated in Hao et al [35]. Multiple classification models were compared for the purpose of ranking solutions based on their comparison relationships in single-objective optimization. A wide variety of pairwise classification models and regression models were compared against each other on benchmark problems for single-objective optimization in [36] with some evidence for the success of the classification approach. A specially designed symmetric neural network architecture was created for the task of modeling the comparison operator between multiple objectives in [37] and applied to solve constraint satisfaction problems in integrated circuit design. While these results demonstrate the potential of a comparison-based model, they did not address the unique challenges of integrating one with a multi-objective optimization algorithm. This includes integrating the comparison predictions across all objectives, diversity preservation in higher dimensions, and dealing with predictions that may not satisfy important symmetries of the comparison operator which are relied on in (for example) domination sorting.

Algorithm 1: CRSEA Optimization Algorithm

Input: FE_{\max} , the maximum number of function evaluations; N , the population size; w_{\max} , the number of generations executed before updating the model; μ , the number of expensive function evaluations performed during a model update.

Output: P , the final population

```
/* Initialize population and model */  
1  $P_t \leftarrow$  initial set of  $N_i$  solutions, Latin hypercube  
2  $P \leftarrow N$  solutions selected from  $P_t$  as in NSGA-II  
3  $\theta \leftarrow$  model parameters from initial training on  $P_t$   
4  $FE \leftarrow N_i$  // Track function evaluations  
5 while  $FE \leq FE_{\max}$  do  
    /* Run NSGA-II using the model */  
    6  $w \leftarrow 1$  // "Model generation" count  
    7  $P_m \leftarrow P$  // Model population  
    8 fit  $\leftarrow$  fitness calculated with nondominated sorting  
    9 while  $w \leq w_{\max}$  do  
        10  $c_m \leftarrow$  get  $N$  children, use "fit" in tournament  
        11  $P_m \leftarrow P_m \cup c_m$   
        12  $P_m \leftarrow$  select  $N$  solutions in  $P_m$  using model  
        13 fit  $\leftarrow$  calculate solutions' fitness with model  
        14  $w \leftarrow w + 1$   
    15 end  
    /* Evaluate Solutions */  
    16  $c \leftarrow$  evaluate  $\mu$  random candidate solutions from  $P_m$   
    17  $FE \leftarrow FE + \mu$   
    /* Update model */  
    18  $P_t \leftarrow P_t \cup c$  // Add to training set  
    19  $\theta \leftarrow$  train model on  $P_t$   
    /* Select solutions based on fitness */  
    20  $P \leftarrow P \cup c$   
    21  $P \leftarrow$  select  $N$  solutions from  $P$  as in NSGA-II  
22 end
```

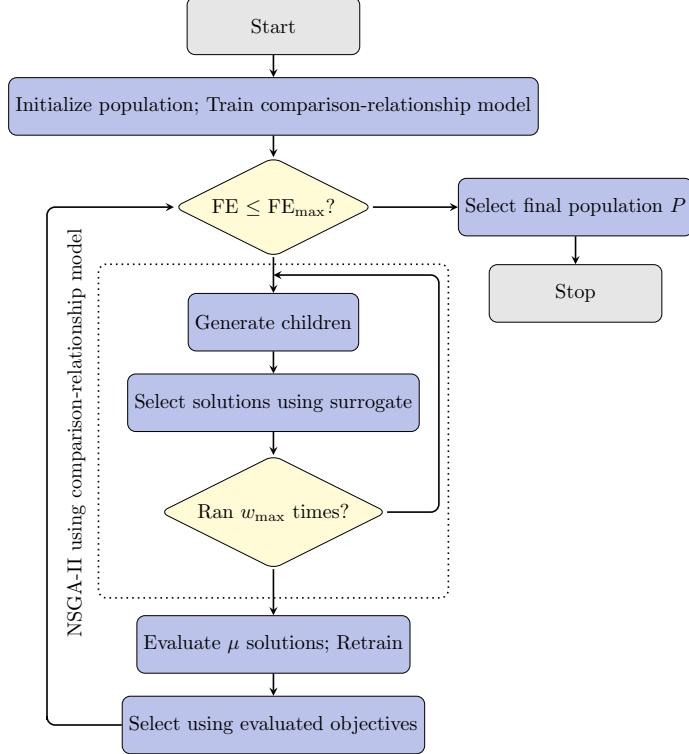


Figure 1: A diagram of the how the CRSEA algorithm (Alg. 1) works. An initial population is generated and the comparison-relationship model trained. Then, the comparison-relationship model is used in the place of the objective functions in an evolutionary algorithm for some number of generations (w_{\max}). Several (μ) randomly chosen solutions from the resulting candidate solutions are evaluated and used to retrain the model. The process is repeated until the budget of function evaluations (FE_{\max}) is exhausted.

2.3. Motivation

Unlike a model of the objective functions themselves, models of the comparison relationship are invariant under all monotonic transformations of the objective functions. Some point to the invariance properties of evolutionary algorithms as one reason for their observed robustness on a wide variety of problems [38, 39, 40]. When building a surrogate model for an evolutionary algorithm, it is therefore natural to consider one such as a comparison-based

surrogate that reflects the same symmetries.

In addition to the invariance properties of the model, a classification problem is always “simpler” than the equivalent regression problem in the sense that it is solved in the process of solving the regression problem, but does not solve the regression problem itself. That is, a model of the objective function values also gives the classifications whether they predict domination, comparisons, or something else. However, knowledge of what class a solution lies in is not enough information to determine the values of the objective functions. It may be that this “simpler” problem is easier to solve.

The task of learning the comparison relationships between the objectives also forms a naturally balanced classification problem. In a model of the domination relationships, for example, as the algorithm drifts closer to the Pareto front most new examples are dominated by the existing population. The comparisons between objective functions, on the other hand, can always be manipulated to have an equal number of positive and negative examples by swapping the order of the two solutions being compared.

With these arguments and the previously discussed success of comparison-relationship models in solving single-objective optimization problems, we are motivated to use a similar model to drive a genetic algorithm in solving the multi-objective case.

3. A Comparison-Relationship-Surrogate multi-objective Genetic Algorithm

With an accurate enough model of the comparison relationships (Eq. 1), the expense of evaluating the objective functions in an evolutionary algorithm

could be replaced completely with a cheaper call to the model. This approach for integrating information from the model into an optimization algorithm has seen success in other SAEAs (such as in K-RVEA [11]) which motivates us to handle the comparison-relationship model in a similar manner.

With this inspiration, we propose the algorithm described in Alg. 1. The model is first initialized using training points that are drawn from the decision space with Latin hypercube sampling [41] and evaluated using the real objective functions. Candidate solutions are found from the model using an evolutionary algorithm. To combat the model’s loss of accuracy when it is evaluated far from the points it was trained on, the model is periodically updated using a small number of candidate solutions for which the real objective function values are computed. This process (solving the optimization problem and updating the model) is repeated until the budget of function evaluations is exhausted. A diagram of the algorithm is shown in Fig. 1.

The main difference between the new algorithm and existing SAEAs is the use of the comparison-relationship model which directly models the comparison relationships between each objective of every pair of solutions. As discussed in the motivation section, a benefit of this model is that it has the same invariance under monotonic transformations of the objective functions as the genetic algorithms it is being integrated with. It also forms a naturally balanced classification problem solving some difficulties in other classification-based SAEAs. On the other hand, diversity preservation with a model that doesn’t have information about the distance between solutions may be challenging. In the following subsections, we explain in detail the integration of the comparison-relationship model into a genetic algorithm.

3.1. Integrating the Comparison-Relationship Model with an Evolutionary Algorithm

The major step in the proposed multi-objective optimization algorithm is generating candidate solutions using only the model. Because the surrogate model can be evaluated cheaply, this step may be performed with a conventional multi-objective genetic algorithm. The lack of information about the values of the objectives from the comparison-relationship model, however, imposes some limits on which types of algorithms can be run on it. As an example, algorithms that use reference vectors for diversity preservation require some knowledge of the distance or angle between solutions and are ruled out.

With this restriction in mind, we propose finding optimal solutions from the model using NSGA-II [42]. Briefly summarizing the algorithm, in each generation, a set of children is produced with mutation, crossover, and selection operators (in this paper, polynomial mutation, simulated binary crossover, and tournament selection [43]). After evaluating the objective functions, the population is sorted by increasing rank of non-domination. Within those ranks, the “crowding distance” metric is used to break ties in order to promote diversity. Solutions are selected from the newly sorted list with the smallest rank, largest crowding distance solutions being the most fit. We suggest running the algorithm using only information from the surrogate model.

Although NSGA-II is well suited for use with the comparison-relationship model, there are still a few challenges that come with it. Each is described in detail in the following subsections and we propose ways of mitigating them.

First, we find that approximate models of the comparison relationships which do not satisfy the transitive properties must have their results corrected before use in an optimization algorithm. We then discuss the prediction of domination from a model that only predicts strict inequality between solutions. We conclude by talking about the crowding distance metric and the difficulty encountered in using it during selection with a model that only predicts comparisons.

3.1.1. Violation of the Transitive Property

The comparison operator satisfies the transitive property. For the objective functions (single objective for simplicity) of three solutions, f_1, f_2, f_3 , if $f_1 < f_2$ and $f_2 < f_3$ then $f_1 < f_3$ must be true. Approximate models of the comparison operator may violate the transitive property. This can lead to paradoxical situations such as the model predicting a solution 1 dominates solution 2, solution 2 dominates solution 3, but solution 3 is not dominated by solution 1. Such nonsensical results can wreak havoc on algorithms such as the fast non-dominated sorting algorithm from Ref. [42] used in NSGA-II. To integrate a comparison-relationship model into the genetic algorithm, we must find a way of avoiding these situations.

We propose “cleaning” the predictions of the comparison model so that they respect the transitive property. For all solutions x_i in \mathbb{P} , the population being sorted, we can calculate the noisy predictions from the comparison model. The predictions take the form of a probability that the comparison is true, $\mathcal{P}(f(x_i) < f(x_j))$, where $f(x)$ is the objective function (a single

objective for simplicity). To clean the predictions, we calculate the value,

$$\tau_i = \frac{1}{|\mathbb{P}|} \sum_{j \in \mathbb{P}} \mathcal{P}(f(x_i) < f(x_j)), \quad (2)$$

from Ref. [44]. We take the value of $\mathcal{P}(f(x_i) < f(x_i))$ as zero. The quantity τ_i can be interpreted as the probability of the objective function of solution i being smaller than that of a solution chosen at random from \mathbb{P} . The comparison predictions defined by $f(x_i) < f(x_j) := \tau_i > \tau_j$ now satisfy the transitive property. This cleaning procedure is repeated for all objective functions and the domination relationships can be calculated from there, allowing the fast non-dominated sorting algorithm to be run using the comparison model.

3.2. Equality and the Comparison-Relationship Model

Assuming a minimization problem, one solution dominates another if all of its objectives are less than or equal to the other's and at least one of its objectives is strictly less than the corresponding objective in the other solution. Unfortunately, the comparison-relationship model seems to only predict the case of strict inequality. This prevents us from predicting domination in the cases where some of the objective for two solutions are equal. However, equality happens exactly at the point in between the region where $f(x_1) < f(x_2)$ and the region where $f(x_2) < f(x_1)$. In terms of the model, these are exactly the points which lie along the classifier's decision boundary! This allows us to define domination as predicted from the model as $\tau_i \leq \tau_j$ for all objectives and $\tau_i < \tau_j$ for at least one using our “cleaned” predictions as described in the previous section.

3.2.1. Crowding Distance

With the ability to clean the model’s predictions from violations of the transitive property, we can now use it to predict the non-domination rank of solutions. In NSGA-II, “crowding distance” is then used to break ties of non-domination rank for the purpose of selection. This acts as a form of diversity preservation.

To calculate crowding distance, each solution is assigned a value of zero to start with. For each objective, the solutions are sorted by the value of their objective function. The extreme solutions (highest and lowest value) are assigned a crowding distance of infinity. Every other solution has its crowding distance incremented by the distance between its two nearest neighbors for this objective normalized by full range spanned by all solutions (distance between the biggest and smallest values).

The comparison-relationship model, of course, does not have information on the distance between neighbors. Instead we suggest using the values of τ from Eq. 2 to predict the solutions with extreme values for the objective functions (smallest and largest). These solutions are preferred when breaking ties of non-domination rank. This should achieve a similar effect to setting their crowding distance to infinity in NSGA-II, but we do expect worse diversity preservation since the surrogate model cannot predict the distance between solutions.

With these two adjustments (cleaning the predictions to satisfy the transitive property, and breaking ties with predictions of extreme solutions instead of crowding distance) the comparison-relationship model may be used in an NSGA-II-like algorithm to perform selection.

3.3. Modeling the Comparison Relationship

One way to approach the task of modeling the comparison relationships between objectives is by considering it as multiple binary classification problems. The training data here is the set of pairs of solutions in the evaluated population with the labels being the m comparisons of objective functions ($f_i(x_1) < f_i(x_2)$) for each example. We are interested in building a model from the data to predict the comparisons for the unseen pairs of solutions. Due to their expressive power and success in solving problems with high dimensional inputs, we used a deep neural network to approximate the comparison relationship.

The comparison operator has the symmetry,

$$f_i(x_1) < f_i(x_2) \implies f_i(x_1) \not> f_i(x_2), \quad (3)$$

which is not guaranteed in a naive implementation using a neural network. We fix this by using the architecture proposed in [37]. A sketch of the architecture is shown in Fig. 2 and we briefly describe it here.

Considering a network that outputs predictions with the one-hot encoding scheme, the requirement in Eq. 3 is equivalent to requiring that the output vector of the network is reversed on permutation of the two inputs. To achieve this, the decision variables for the two solutions being compared are each first fed into a feature extraction network that consists of conventional dense layers with ReLU activation and has shared weights and biases between the two inputs. The output features of the two solutions are concatenated into a single feature vector with the elements from the second network being reversed so that permuting the solutions is equivalent to reversing the full

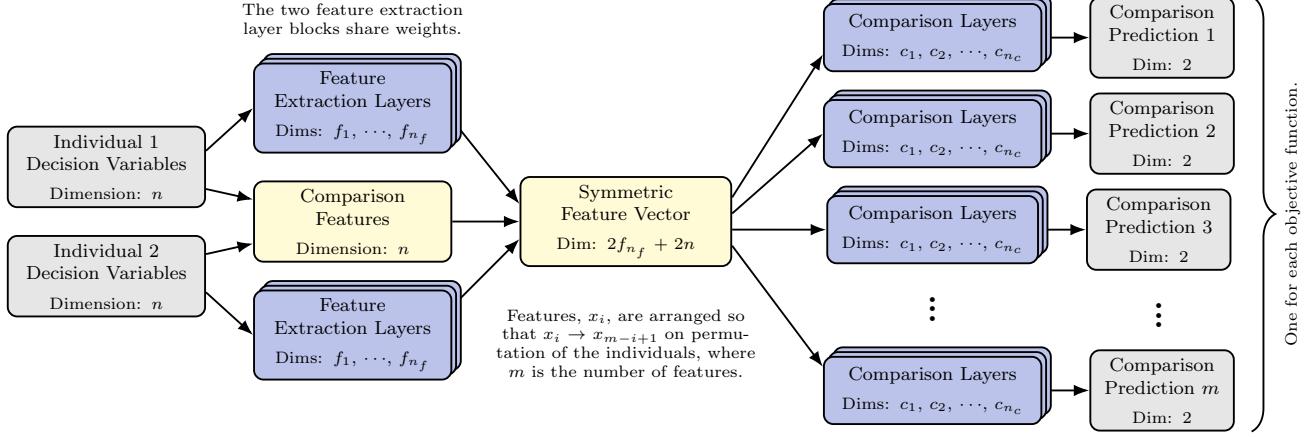


Figure 2: A schematic of the comparison surrogate model. Decision variables of the two solutions being compared are fed into feature extraction networks made of densely connected layers with shared weights. Additional features in the form of the comparisons $x_{1,i} < x_{2,i}$ are generated. All features are combined into a symmetric vector such that a permutation of the two input solutions will cause the feature vector to reverse itself ($f_i \rightarrow f_{n-i+1}$ for a vector f with n elements). These features are fed into comparison networks (one for each objective) which use special symmetric dense layers to ensure the correct behavior under permutation of the inputs.

feature vector. That is,

$$(\vec{f}_a, \vec{f}_b) \rightarrow [f_{a,1}, f_{a,2}, \dots, f_{a,n}, f_{b,n}, \dots, f_{b,2}, f_{b,1}], \quad (4)$$

where \vec{f}_a and \vec{f}_b are the outputs of the two feature extraction networks and n is the size of the final layer in the feature extraction network.

The complete feature vector is then fed into a separate ‘‘comparison network’’ for each objective. Because the requirement from Eq. 3 is already satisfied by the feature vector, to ensure it for the whole network we only need to preserve the symmetry in the comparison layers,

$$y_i = f(w_{i,j}x_j + b_j), \quad (5)$$

where $f(x)$ is the activation function, $w_{i,j}$ and b_i are the weights and biases, and we are using Einstein summation notation. The layers will preserve our required symmetry as long as they satisfy the conditions,

$$w_{i,j} = w_{m-i+1,n-j+1}, \quad b_i = b_{n-i+1}, \quad (6)$$

where m and n are the dimensions of the weights matrix. In our work, the ReLU non-linearity was used for these comparison layers except for the output layer which used the softmax activation function. All weights are initialized to be uniformly distributed in the way described in Ref. [45]. With these constraints, the full model will only produce predictions with the symmetry described in Eq. 3 on permutation of the inputs.

Besides features learned from the training data, additional features may be added manually as long as they are introduced to the final feature vector in a way that preserves its symmetry. For example, we suggest adding the features $x_{1,i} < x_{2,i}$ ($1 \leq i \leq n$) to allow the comparison networks to learn from the raw comparisons themselves. These are added to the middle of the feature vector along with a reversed and inverted copy to ensure the correct behavior under permutation of the inputs. These extra features could be useful in several practical applications where one of the objectives may be a monotonic function of a single decision variable.

Training data can be generated from a set of decision variables and the corresponding values of the objective functions. For each pair of solutions in the training set (discounting duplicates like $a < b$ and $b < a$), the comparisons between objective functions are evaluated. An error term is calculated from the model for each objective using the binary cross entropy loss function.

The total loss is then optimized using mini-batch gradient descent with the AdamW optimizer [46] and cosine annealing with warm restarts [47] where the learning rate scheduler is “restarted” at the beginning of every retraining step in the algorithm. When the model is used in a multi-objective optimiza-

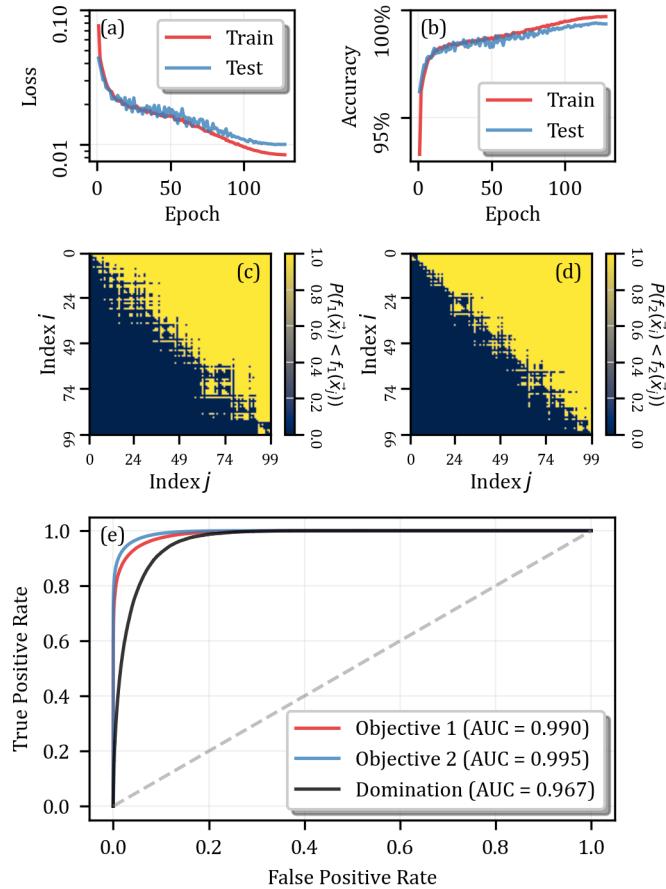


Figure 3: An example of the comparison model trained on data from the WFG8 ($n = 16$) problem. (a) loss function on the train and test set during training; (b) the accuracy during training; (c) and (d) predicted comparisons for both objectives on never-before-seen samples where solutions are indexed in sorted order for the objective being compared (i.e. predictions from a perfect classifier would show up as a solid yellow upper triangle); (e) the receiver operating characteristic curve for each model output as well as domination predictions.

tion algorithm, we retain all evaluated solutions for use in the training set and retrain the model at each generation of the optimizer (as in Alg. 1).

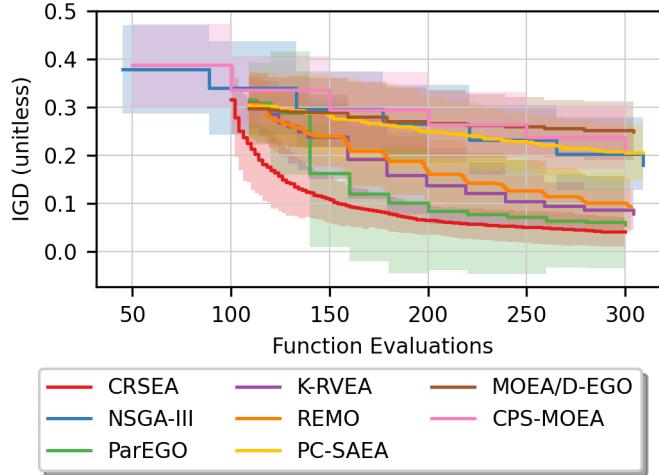


Figure 4: Mean value of the IGD metric and 95% prediction interval for non-dominated solutions to DTLZ5 ($m = 3$, $n = 10$).

An example of using the comparison-relationship-surrogate model is shown in Fig. 3. A training set of 300 solutions was generated for the problem WFG8 ($n = 16$) [48] and the model was trained on it for 128 epochs. The resulting learning curves are shown in subfigures (a) and (b) using an 80%-20% train-test split of the labeled comparison data. Predicted comparisons were then evaluated for every pair in a second set of 128 randomly generated solutions. The results are plotted in subfigures (c) and (d) where the solutions are indexed according to their position in the array sorted by the objective being evaluated. This representation helps visualize when and where errors occur in the comparisons. For example, many of the out-of-place elements are close to the diagonal meaning that errors tend to be between solutions that are similar in value for that objective. In subfigure (e) we show the

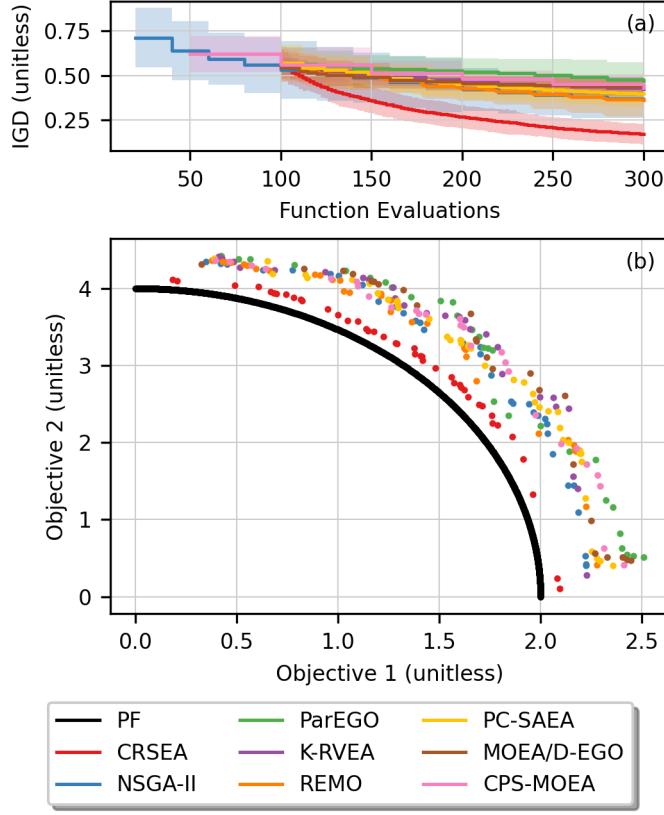


Figure 5: Results from the experiments performed in Tab. 2 for the WFG7 ($n=16$) test problem : (a) Mean value of the IGD metric and 95% prediction interval for non-dominated solutions; (b) Solutions after 300 function evaluations from the optimization with the median value of the IGD metric for each algorithm. The analytical Pareto front (labeled “PF”) is plotted in black.

receiver operating characteristic curve and list the area under curve metric for the comparison models as well as for the resulting domination classifier. As expected, the domination predictions are degraded as a correct prediction requires that both underlying comparisons are predicted correctly.

4. Empirical Studies of Performance

In this section, we evaluate the performance of the comparison-relationship-surrogate optimizer through a set of numerical experiments. In each experiment, the new optimizer, as well as a set of contemporary algorithms, are used to solve test problems on a fixed budget of 300 function evaluations. The performance of the algorithms is compared based on the inverted generational distance [49] of the returned solutions to the analytical Pareto fronts. Due to the stochastic nature of evolutionary optimization, these comparisons are carried out using the Wilcoxon rank sum statistical test at a significance level of 95% with a sample size of 32.

Three different classes of algorithms were chosen to compare against: conventional genetic algorithms, regression-based surrogate-assisted algorithms, and classification-based surrogate-assisted algorithms. From the first class, NSGA-II [42] was chosen for test problems with two objectives. This algorithm was selected for its popularity in applications as well as the fact that the new comparison-surrogate-assisted algorithm is strongly influenced by it. For the problems with more than two objectives, NSGA-III [50] was chosen as the conventional optimizer due to its improved performance on many-objective problems. From the category of regression-based surrogate-assisted optimizers, we chose ParEGO [14], K-RVEA [11], and MOEA/D-EGO [17] for a combination of their performance on test problems and popularity in the literature. Because the new algorithm being studied uses a classification-based surrogate model, we feel it is important to include similar algorithms in the numerical experiments. For these purposes, we chose an example of an algorithm that classifies the regions of decision space, CPS-MOEA [20],

as well as two that use a pairwise model of the domination relationships, REMO [29] and PC-SAEA [32].

All algorithms except for the new comparison-relationship-surrogate-based optimizer were evaluated using their implementation in PlatEMO [51] version 4.11 with default settings for all parameters. For algorithms that require a population size, it was set to 50 in all cases, except for NSGA-II where it was set to 20 following the recommendation of Ref. [14] and NSGA-III where it was set based on the number of objectives using the table in Ref. [27]. For surrogate-assisted algorithms that use an initial population size, it was set to $N_i = 11n - 1$ as recommended for the Latin hypercube sampling technique [41] and was capped at $N_i = 100$ for problems with more than ten decision variables to avoid immediately exhausting the budget of function evaluations.

Parameters for the new comparison-relationship-surrogate optimizer were selected as follows. The number of new function evaluations (μ in Alg. 1) was set to two and the number of “model generations” where the optimizer is run only on the surrogate model (w_{\max}) was set to 32. We discuss more about this choice in a sensitivity study on the parameter in the following subsection. Two hidden layers of size 64 were used in the feature extraction networks and one hidden layer of size 32 was used for the comparison networks. The number of epochs in the initial training of the model was 128 with 16 epochs used in every retraining in further generations. The learning rate was 0.001 at the beginning of each training cycle. An L2 regularization loss term was used (i.e. through the AdamW optimizer [46]) with a weight of $\lambda = 0.1$.

The first experiment was performed with the popular DTLZ test suite

Table 1: Statistical results for the IGD metric of 32 runs of each optimizer on the DTLZ problem set with ten decision variables. Cells that are among the best performing for that problem up to a 95% significance level are highlighted.

Problem	m	NSGA-III	ParEGO	K-RVEA	MOEA/D-EGO	CPS-MOEA	REMO	PC-SAEA	CRSEA
DTLZ1	3	9.09e+11 (1.57e+1) -	6.73e+11 (5.57e+0) +	9.01e+11 (2.27e+1) -	8.43e+11 (1.62e+1) \approx 8.00e+11 (1.57e+1) \approx	5.40e+11 (1.87e+1) +	8.61e+11 (1.66e+1) \approx 8.11e+11 (1.65e+1)		
	4	6.04e+11 (1.62e+1) \approx	5.63e+11 (7.24e+0) +	6.23e+11 (1.41e+1) \approx 6.21e+11 (1.23e+1) \approx 5.94e+11 (1.35e+1) \approx	3.78e+11 (1.20e+1) +	7.03e+11 (1.60e+1) -	6.26e+11 (1.23e+1)		
	6	3.11e+11 (1.07e+1) +	3.71e+11 (5.59e+0) +	2.94e+11 (7.54e+0) +	3.32e+11 (7.47e+0) +	3.09e+11 (8.56e+0) +	1.53e+11 (4.56e+0) +	2.76e+11 (1.12e+1) +	4.21e+11 (7.12e+0)
	8	1.08e+11 (5.16e+0) +	1.55e+11 (4.15e+0) \approx	9.37e+10 (3.43e+0) +	1.28e+11 (3.27e+0) +	1.01e+11 (3.56e+0) +	4.55e+10 (2.28e+0) +	9.16e+10 (3.17e+0) +	1.53e+11 (5.47e+0)
	10	5.22e+11 (2.29e+1) +	4.58e+11 (8.84e-2) +	3.68e+11 (1.07e+1) -	4.61e+11 (1.62e+1) +	3.99e+11 (8.58e-2) +	2.95e+11 (6.39e-2) +	3.91e+11 (8.21e-2) +	6.91e+11 (2.99e-1)
DTLZ2	3	2.93e-1 (2.69e-2) -	1.60e-1 (3.75e-2) -	1.73e-1 (4.10e-2) -	3.54e-1 (2.27e-2) -	2.86e-1 (2.10e-2) -	2.09e-1 (2.56e-2) -	2.80e-1 (2.39e-2) -	8.36e-2 (1.03e-2)
	4	3.52e-1 (2.95e-2) -	4.28e-1 (3.04e-2) -	2.41e-1 (3.13e-2) -	3.95e-1 (2.07e-2) -	3.95e-1 (3.16e-2) -	2.71e-1 (2.33e-2) -	3.51e-1 (2.01e-2) -	2.17e-1 (2.44e-2)
	6	4.83e-1 (3.55e-2) \approx	5.33e-1 (2.92e-2) -	3.37e-1 (2.38e-2) +	4.81e-1 (1.72e-2) \approx	5.36e-1 (2.93e-2) -	4.04e-1 (4.00e-2) +	4.69e-1 (2.70e-2) +	4.85e-1 (2.62e-2)
	8	5.83e-1 (2.78e-2) -	6.59e-1 (2.77e-2) -	4.67e-1 (3.70e-2) +	5.18e-1 (1.50e-2) +	5.98e-1 (2.90e-2) -	5.13e-1 (4.67e-2) +	5.71e-1 (2.07e-2) -	5.31e-1 (1.87e-2)
	10	6.71e-1 (2.99e-2) -	7.00e-1 (2.87e-2) -	5.41e-1 (2.32e-2) \approx	5.69e-1 (3.21e-2) -	6.55e-1 (3.95e-2) -	6.12e-1 (4.50e-2) -	6.72e-1 (2.91e-2) -	5.40e-1 (1.71e-2)
DTLZ3	3	2.65e+2 (6.68e+1) -	1.75e+2 (8.48e+0) +	2.24e+2 (5.49e+1) \approx	2.03e+2 (3.32e+1) +	2.13e+2 (3.41e+1) \approx	1.63e+2 (4.38e+1) +	2.38e+2 (4.10e+1) -	2.17e+2 (2.71e+1)
	4	1.82e+2 (5.01e+1) \approx	1.50e+2 (9.92e+0) +	1.73e+2 (2.97e+1) \approx	1.66e+2 (2.30e+1) \approx	1.58e+2 (3.09e+1) +	1.24e+2 (2.88e+1) +	2.01e+2 (3.98e+1) -	1.72e+2 (2.00e+1)
	6	9.47e+11 (3.70e+1) +	9.64e+11 (1.23e+1) +	8.25e+11 (2.29e+1) +	9.53e+11 (1.59e+1) +	9.32e+11 (2.06e+1) +	5.60e+11 (1.81e+1) +	1.01e+2 (2.66e+1) +	1.13e+2 (1.04e+1)
	8	3.27e+11 (1.27e+1) +	4.61e+11 (1.00e+1) \approx	2.80e+11 (1.07e+1) +	3.62e+11 (1.23e+1) +	3.41e+11 (1.33e+1) +	1.87e+11 (8.97e+0) +	2.55e+11 (8.64e+0) +	4.52e+11 (1.13e+1)
	10	2.02e+0 (1.75e+0) +	1.58e+0 (4.97e-1) +	1.22e+0 (3.10e-1) +	1.43e+0 (4.37e-1) +	2.20e+0 (1.36e+0) \approx	6.93e+11 (2.68e-1) +	1.29e+0 (2.72e-1) +	1.90e+0 (6.49e-1)
DTLZ4	3	7.34e-1 (1.92e-1) \approx	5.31e-1 (1.31e-1) +	4.15e-1 (1.19e-1) +	6.01e-1 (6.79e-2) +	5.62e-1 (6.06e-2) +	2.62e-1 (1.27e-1) +	5.26e-1 (8.64e-2) +	7.26e-1 (1.71e-1)
	4	7.36e-1 (1.60e-1) \approx	6.93e-1 (1.02e-1) \approx	5.12e-1 (9.17e-2) +	6.55e-1 (5.16e-2) \approx	6.16e-1 (3.91e-2) +	3.22e-1 (7.95e-2) +	6.01e-1 (5.30e-2) +	7.06e-1 (1.96e-1)
	6	7.74e-1 (9.65e-2) -	5.00e-1 (7.57e-2) +	5.42e-1 (4.58e-2) +	6.54e-1 (3.23e-2) +	6.34e-1 (2.36e-2) +	5.10e-1 (4.82e-2) +	7.06e-1 (3.64e-2) -	6.05e-1 (5.73e-2)
	8	7.54e-1 (6.06e-2) -	5.50e-1 (3.76e-2) +	5.62e-1 (3.80e-2) +	6.29e-1 (1.83e-2) \approx	6.31e-1 (1.36e-2) \approx	5.92e-1 (4.34e-2) +	6.62e-1 (1.98e-2) -	6.31e-1 (4.40e-2)
	10	7.15e-1 (6.09e-2) -	5.97e-1 (2.37e-2) -	6.09e-1 (2.62e-2) -	6.13e-1 (1.16e-2) -	6.19e-1 (8.78e-3) -	6.43e-1 (2.86e-2) -	6.35e-1 (2.13e-2) -	5.84e-1 (2.03e-2)
DTLZ5	3	2.03e-1 (3.90e-2) -	5.63e-2 (4.49e-2) -	8.68e-2 (2.19e-2) -	2.52e-1 (3.16e-2) -	2.14e-1 (3.31e-2) -	1.00e-1 (2.86e-2) -	2.06e-1 (2.98e-2) -	4.09e-2 (1.58e-2)
	4	1.89e-1 (2.61e-2) -	1.06e-1 (6.76e-2) +	6.11e-2 (2.28e-2) +	2.25e-1 (2.87e-2) -	1.95e-1 (3.11e-2) -	1.19e-1 (2.52e-2) \approx	1.89e-1 (2.26e-2) -	1.10e-1 (2.49e-2)
	6	1.35e-1 (3.17e-2) -	1.03e-1 (3.61e-2) +	5.23e-2 (1.20e-2) +	1.41e-1 (1.59e-2) +	1.46e-1 (2.54e-2) +	6.45e-2 (1.99e-2) +	1.18e-1 (1.95e-2) \approx	1.20e-1 (2.37e-2)
	8	8.73e-2 (1.44e-2) -	7.40e-2 (9.77e-3) -	2.37e-2 (5.86e-3) +	6.85e-2 (7.06e-3) \approx	7.43e-2 (1.12e-2) -	3.45e-2 (9.26e-3) +	5.50e-2 (5.78e-3) +	6.89e-2 (1.02e-2)
	10	2.95e-2 (5.14e-3) -	1.94e-2 (1.73e-3) \approx	1.25e-2 (2.08e-3) +	1.94e-2 (2.52e-3) \approx	1.90e-2 (2.53e-3) \approx	1.22e-2 (2.51e-3) +	1.39e-2 (1.56e-3) +	2.01e-2 (3.68e-3)
DTLZ6	3	6.08e+0 (3.33e-1) -	9.22e-1 (1.87e-1) +	2.88e+0 (4.35e-1) \approx	2.21e+0 (8.20e-1) +	3.48e+0 (5.78e-1) -	4.30e+0 (6.05e-1) -	6.06e+0 (3.90e-1) -	2.77e+0 (9.65e-1)
	4	5.31e+0 (3.56e-1) +	9.95e-1 (1.77e-1) +	1.97e+0 (3.26e-1) +	1.71e+0 (5.97e-1) +	3.21e+0 (6.65e-1) +	4.10e+0 (7.11e-1) +	5.24e+0 (2.20e-1) +	5.44e+0 (3.04e-1)
	6	3.76e+0 (2.91e-1) +	6.31e-1 (1.26e-1) +	1.50e+0 (3.49e-1) +	1.20e+0 (4.96e-1) +	2.20e+0 (5.10e-1) +	2.27e+0 (6.98e-1) +	3.64e+0 (2.37e-1) +	4.07e+0 (1.24e-1)
	8	2.14e+0 (2.80e-1) +	4.35e-1 (1.12e-1) +	5.13e-1 (2.30e-1) +	4.92e-1 (2.18e-1) +	1.04e+0 (3.77e-1) +	8.31e-1 (4.07e-1) +	1.89e+0 (1.76e-1) +	2.37e+0 (9.13e-2)
	10	5.31e-1 (2.14e-1) +	1.44e-1 (5.45e-2) +	6.25e-2 (2.98e-2) +	2.21e-1 (8.24e-2) +	6.70e-2 (3.34e-2) +	8.50e-2 (9.64e-2) +	4.82e-1 (1.98e-1) +	7.01e-1 (3.69e-1)
DTLZ7	3	4.87e+0 (8.89e-1) -	1.51e-1 (1.48e-2) +	1.29e-1 (1.90e-2) +	3.04e-1 (1.04e-1) +	3.73e+0 (1.30e+0) -	8.11e-1 (5.20e-1) -	4.78e+0 (1.13e+0) -	5.11e-1 (2.17e-1)
	4	5.66e+0 (1.41e+0) -	3.04e-1 (3.35e-2) +	3.15e-1 (3.18e-2) +	5.32e-1 (5.53e-2) +	4.49e+0 (2.12e+0) -	1.27e+0 (4.78e-1) -	6.25e+0 (1.05e+0) -	7.12e-1 (2.63e-1)
	6	8.57e+0 (2.66e+0) -	7.48e-1 (3.70e-2) +	6.17e-1 (2.84e-2) +	9.28e-1 (7.24e-2) +	4.23e+0 (2.51e+0) -	2.86e+0 (4.69e-1) -	8.03e+0 (2.52e+0) -	1.10e+0 (1.13e-1)
	8	8.80e+0 (3.16e+0) -	1.18e+0 (4.96e-2) +	9.63e-1 (6.06e-2) +	1.12e+0 (4.02e-2) +	2.79e+0 (1.94e+0) -	3.53e+0 (5.02e-1) -	6.35e+0 (2.83e+0) -	1.77e+0 (2.09e-1)
	10	3.03e+0 (1.51e+0) -	1.41e+0 (5.75e-2) +	1.05e+0 (2.92e-2) +	1.26e+0 (3.21e-2) +	1.56e+0 (1.09e-1) +	2.21e+0 (3.91e-1) -	1.79e+0 (2.79e-1) +	1.89e+0 (2.11e-1)

Table 2: Statistical results for the IGD metric of 32 runs of each optimizer on the WFG problem set with two objectives. Cells which are among the best performing for that problem up to a 95% significance level are highlighted.

Problem	n	NSGA-II	ParEGO	K-RVEA	MOEA/D-EGO	CPS-MOEA	REMO	PC-SAEA	CRSEA
WFG1	16	1.44e+0 (2.41e-1) \approx 1.32e+0 (4.70e-3) + 1.35e+0 (4.90e-2) + 1.38e+0 (2.90e-2) \approx 1.36e+0 (1.33e-2) \approx 1.30e+0 (5.23e-2) +	1.42e+0 (1.27e-1) \approx 1.50e+0 (1.69e-1)						
	32	1.57e+0 (3.33e-1) \approx 1.32e+0 (3.71e-3) + 1.33e+0 (3.50e-2) + 1.40e+0 (4.26e-2) + 1.37e+0 (1.22e-2) +	1.32e+0 (4.91e-2) + 1.58e+0 (2.94e-1) \approx 1.53e+0 (1.79e-1)						
	64	1.49e+0 (2.52e-1) + -	1.36e+0 (4.86e-2) + 1.42e+0 (4.65e-2) + 1.37e+0 (8.70e-3) +	1.34e+0 (4.97e-2) + 1.40e+0 (1.21e-1) + 1.58e+0 (1.73e-1)					
WFG2	16	5.14e-1 (1.31e-1) + 6.32e-1 (3.14e-2) -	4.85e-1 (6.56e-2) + 5.37e-1 (4.14e-2) + 5.72e-1 (5.51e-2) \approx 4.87e-1 (6.19e-2) +	5.47e-1 (6.66e-2) \approx 5.66e-1 (6.39e-2)					
	32	6.34e-1 (1.53e-1) + 6.69e-1 (2.12e-2) \approx 6.56e-1 (2.73e-2) \approx 5.97e-1 (4.17e-2) + 6.05e-1 (5.70e-2) +	7.16e-1 (4.09e-2) \approx 6.68e-1 (3.26e-2) + 6.54e-1 (4.50e-2) +	5.43e-1 (6.10e-2) + 6.64e-1 (4.92e-2) \approx 6.58e-1 (4.87e-2)					
	64	6.51e-1 (1.22e-1) + -		6.15e-1 (5.01e-2) + 6.97e-1 (4.31e-2) \approx 7.09e-1 (2.78e-2)					
WFG3	16	4.00e-1 (5.11e-2) - 6.04e-1 (2.17e-2) -	3.04e-1 (6.78e-2) - 4.90e-1 (2.61e-2) - 5.21e-1 (3.37e-2) -	4.27e-1 (4.36e-2) - 4.31e-1 (4.07e-2) - 2.24e-1 (4.96e-2)					
	32	4.93e-1 (4.00e-2) - 6.54e-1 (1.53e-2) -	5.35e-1 (4.02e-2) - 5.35e-1 (2.31e-2) - 5.76e-1 (2.55e-2) -	5.24e-1 (3.69e-2) - 5.53e-1 (2.22e-2) - 4.61e-1 (4.58e-2)					
	64	5.86e-1 (2.48e-2) + -	6.33e-1 (1.61e-2) - 6.38e-1 (1.49e-2) - 6.16e-1 (2.79e-2) \approx 5.96e-1 (2.15e-2) + 6.31e-1 (9.93e-3) - 6.15e-1 (2.38e-2)						
WFG4	16	2.68e-1 (7.70e-2) - 3.86e-1 (2.41e-2) -	3.06e-1 (2.78e-2) - 3.36e-1 (6.10e-2) - 3.09e-1 (2.73e-2) -	2.66e-1 (2.94e-2) - 3.00e-1 (5.66e-2) - 1.49e-1 (1.45e-2)					
	32	3.36e-1 (6.65e-2) - 4.75e-1 (1.97e-2) -	3.78e-1 (2.01e-2) - 3.85e-1 (5.12e-2) - 3.63e-1 (1.94e-2) -	3.24e-1 (4.93e-2) - 3.45e-1 (3.48e-2) - 2.40e-1 (2.49e-2)					
	64	4.01e-1 (9.31e-2) \approx -	4.21e-1 (2.72e-2) - 4.00e-1 (2.09e-2) - 3.88e-1 (2.22e-2) \approx 3.76e-1 (3.46e-2) \approx 3.98e-1 (2.38e-2) - 3.83e-1 (2.56e-2)						
WFG5	16	3.66e-1 (4.02e-2) - 1.09e-1 (1.38e-2) -	1.78e-1 (2.93e-2) - 2.20e-1 (4.24e-2) - 3.30e-1 (2.44e-2) -	2.84e-1 (4.19e-2) - 4.37e-1 (3.56e-2) - 9.37e-1 (1.50e-2)					
	32	4.83e-1 (5.60e-2) - 1.66e-1 (5.15e-2) -	3.39e-1 (3.24e-2) - 3.09e-1 (5.78e-2) - 3.78e-1 (2.42e-2) -	4.03e-1 (3.84e-2) - 4.98e-1 (1.95e-2) - 1.17e-1 (1.48e-2)					
	64	5.69e-1 (4.75e-2) - -	4.88e-1 (3.11e-2) - 4.13e-1 (4.95e-2) - 4.20e-1 (1.95e-2) -	5.13e-1 (3.44e-2) - 5.78e-1 (1.89e-2) - 1.79e-1 (2.58e-2)					
WFG6	16	4.11e-1 (5.76e-2) - 2.96e-1 (6.06e-2) \approx -	5.04e-1 (5.19e-2) - 4.45e-1 (8.56e-2) - 6.24e-1 (3.75e-2) -	4.75e-1 (5.08e-2) - 5.37e-1 (3.64e-2) - 3.26e-1 (8.07e-2)					
	32	5.58e-1 (4.90e-2) - 2.84e-1 (6.22e-2) \approx -	5.81e-1 (4.22e-2) - 6.20e-1 (8.78e-2) - 7.23e-1 (3.89e-2) -	5.72e-1 (4.80e-2) - 6.42e-1 (3.11e-2) - 3.46e-1 (1.62e-1)					
	64	6.75e-1 (3.76e-2) - -	6.93e-1 (2.55e-2) - 6.89e-1 (5.92e-2) - 7.51e-1 (2.78e-2) -	6.72e-1 (3.18e-2) - 7.15e-1 (3.99e-2) - 4.66e-1 (1.03e-1)					
WFG7	16	3.62e-1 (5.56e-2) - 4.69e-1 (5.63e-2) -	4.23e-1 (3.19e-2) - 4.37e-1 (2.62e-2) - 4.32e-1 (2.24e-2) -	3.64e-1 (5.33e-2) - 3.99e-1 (3.77e-2) - 1.72e-1 (2.94e-2)					
	32	4.51e-1 (5.29e-2) - 5.83e-1 (2.44e-2) -	5.28e-1 (1.77e-2) - 4.93e-1 (3.09e-2) - 4.85e-1 (1.74e-2) -	4.48e-1 (3.01e-2) - 4.71e-1 (2.74e-2) - 3.53e-1 (3.03e-2)					
	64	5.31e-1 (5.64e-2) - -	5.76e-1 (9.12e-3) - 5.43e-1 (1.65e-2) - 5.18e-1 (2.08e-2) -	5.13e-1 (2.97e-2) \approx 5.32e-1 (2.52e-2) - 5.05e-1 (2.14e-2)					
WFG8	16	4.75e-1 (4.88e-2) - 6.46e-1 (4.49e-2) -	4.32e-1 (3.53e-2) - 5.79e-1 (2.84e-2) - 5.74e-1 (4.10e-2) -	4.65e-1 (4.16e-2) - 5.63e-1 (4.06e-2) - 3.50e-1 (3.54e-2)					
	32	5.10e-1 (5.30e-2) - 6.73e-1 (2.88e-2) -	4.71e-1 (3.25e-2) - 5.75e-1 (2.38e-2) - 5.83e-1 (2.93e-2) -	4.87e-1 (4.13e-2) - 5.71e-1 (4.74e-2) - 4.31e-1 (3.41e-2)					
	64	5.68e-1 (5.70e-2) - -	5.85e-1 (2.68e-2) - 6.04e-1 (2.51e-2) - 5.93e-1 (2.85e-2) -	5.17e-1 (2.93e-2) - 5.76e-1 (2.54e-2) - 4.86e-1 (3.17e-2)					
WFG9	16	3.90e-1 (7.57e-2) - 3.61e-1 (6.84e-2) -	3.84e-1 (8.81e-2) - 5.27e-1 (6.49e-2) - 3.96e-1 (5.62e-2) -	3.66e-1 (1.10e-1) - 4.18e-1 (7.64e-2) - 2.13e-1 (4.14e-2)					
	32	5.25e-1 (1.01e-1) - 4.17e-1 (7.48e-2) \approx -	6.23e-1 (1.02e-1) - 6.48e-1 (4.87e-2) - 4.46e-1 (4.53e-2) -	5.49e-1 (1.29e-1) - 6.15e-1 (6.07e-2) - 3.87e-1 (1.11e-1)					
	64	6.59e-1 (6.99e-2) \approx -	7.77e-1 (3.87e-2) - 7.30e-1 (4.86e-2) - 5.48e-1 (8.14e-2) + 6.83e-1 (8.90e-2) \approx 7.14e-1 (2.67e-2) - 6.74e-1 (8.00e-2)						
+/-/≈		5/18/4	2/12/4	4/21/2	5/21/1	5/18/4	7/17/3	1/21/5	

of problems [52] with ten decision variables and three, four, six, eight, and ten objectives. These results are shown in Tab. 1. Each cell lists the mean value of the inverted generational distance of the 32 final approximate Pareto fronts to points from the analytical Pareto front with the standard deviation of the values shown in parentheses. These values can be sensitive to the number of points used in the reference set and in our work we chose 1000 values evenly distributed along the Pareto front. Also in each cell is either a plus sign, minus sign, or equals sign indicating whether, for a single problem, the algorithm beats (plus), loses to (minus), or is indistinct from (equals) the new comparison-relationship-surrogate algorithm up to a significance level of 95%. Cells are highlighted if they are among the best performing on that problem up to the same significance level. The performance of the algorithms

on DTLZ5 ($m = 3$) is also shown in Fig. 4.

Although the new algorithm is among the best performing on a handful of problems in the DTLZ test suite, it did not achieve that distinction on the majority of them. Because the algorithm lacks a diversity preservation mechanism (known to be important for many-objective problems [53]) this was not a surprising result. We also point out here that the recall of the domination classifier is $r = p^m$ for a problem with m objectives and an underlying comparison model with recall p for each objective. That is, a true positive prediction of domination requires the correct prediction of every comparison of the objectives. This points to degraded predictions as the number of objectives grows. These two properties of the algorithm may explain its performance on the DTLZ test suite.

With a better understanding of the types of problems the comparison-relationship-surrogate optimizer is suited for, we sought out a test with fewer objectives and more decision variables to take advantage of the unique benefits of a neural-network-based surrogate model over those which use a Gaussian process. To this end, we performed tests on the biobjective WFG test suite [48] with 16, 32, and 64 decision variables. The number of “position parameters” (k in Ref. [48]) was fixed at two. These results are shown in Tab. 2 and show a clear improvement in the quality of solutions generated by the comparison-relationship-surrogate optimizer over the other algorithms studied. The new algorithm produced among the best solutions seen for 70% of the problems studied. For the case of $n = 64$, ParEGO was allowed to run on the problems for several days before terminating after not enough progress was made. An example of the solutions generated for WFG7 ($n = 16$) is

Table 3: Sensitivity study of the w_{\max} parameter in CRSEA. Statistical results (mean with standard deviation in parenthesis) are shown for the value of the IGD metric of 32 runs using each setting on the WFG ($n=16$) problems. Cells which are among the best performing for that problem up to a 95% significance level are highlighted.

Problem	n	CRSEA ($w_{\max} = 4$)	CRSEA ($w_{\max} = 8$)	CRSEA ($w_{\max} = 16$)	CRSEA ($w_{\max} = 32$)	CRSEA ($w_{\max} = 64$)
WFG1	16	1.84e+0 (4.43e-2) -	1.75e+0 (7.98e-2) -	1.65e+0 (1.54e-1) -	1.47e+0 (1.86e-1)	1.42e+0 (1.43e-1) \approx
WFG2	16	5.46e-1 (6.70e-2) \approx	5.48e-1 (6.01e-2) \approx	5.63e-1 (6.22e-2) \approx	5.57e-1 (5.33e-2)	5.90e-1 (5.35e-2) -
WFG3	16	3.08e-1 (6.37e-2) -	2.57e-1 (4.37e-2) -	2.44e-1 (4.04e-2) \approx	2.32e-1 (4.90e-2)	2.15e-1 (4.22e-2) \approx
WFG4	16	1.99e-1 (3.57e-2) -	1.78e-1 (2.89e-2) -	1.62e-1 (2.17e-2) -	1.50e-1 (1.37e-2)	1.52e-1 (1.30e-2) \approx
WFG5	16	1.81e-1 (2.99e-2) -	1.13e-1 (2.15e-2) -	9.52e-2 (7.45e-3) -	9.15e-2 (9.40e-3)	9.04e-2 (6.84e-3) \approx
WFG6	16	4.14e-1 (6.30e-2) -	3.34e-1 (6.17e-2) \approx	3.24e-1 (6.09e-2) \approx	3.16e-1 (7.45e-2)	2.80e-1 (6.75e-2) +
WFG7	16	2.53e-1 (4.37e-2) -	2.03e-1 (4.59e-2) -	1.88e-1 (3.82e-2) \approx	1.71e-1 (3.28e-2)	1.76e-1 (3.86e-2) \approx
WFG8	16	3.91e-1 (3.02e-2) -	3.65e-1 (3.46e-2) -	3.61e-1 (3.65e-2) -	3.47e-1 (3.06e-2)	3.51e-1 (3.57e-2) \approx
WFG9	16	2.92e-1 (9.69e-2) -	2.46e-1 (6.59e-2) -	2.24e-1 (5.34e-2) \approx	2.07e-1 (6.07e-2)	2.25e-1 (4.68e-2) \approx
+/-/ \approx		0/8/1	0/7/2	0/4/5		1/1/7

shown in Fig. 5b. The evolution of the quality of the solutions (measured by the IGD metric) during an optimization is visualized for the same example in Fig. 5a.

4.1. Sensitivity to Number of Iterations Using Model

In this section, we evaluate the sensitivity of the CRSEA algorithm to the number of iterations of the evolutionary algorithm run using the model only (w_{\max} in Alg. 1). This parameter should be tuned to squeeze the most information out of the model without drifting too far into the regions where it does not have training data. For a test of how this parameter affects the performance of the algorithm, we have rerun CRSEA on the WFG problem suite with $n = 16$ as we vary w_{\max} . We report this data in Tab. 3. All of the settings besides w_{\max} were kept the same as in previous experiments. Every problem was evaluated 32 times and the Wilcoxon rank sum test used to compare them to the suggested set-point. These results suggest $w_{\max} = 32$ as a reasonable choice for this set of problems.

5. Performance on a Real-World Problem

As another benchmark of the comparison-relationship-surrogate optimizer, we chose to evaluate it on a practical medium-scale problem from the field of accelerator physics. This problem is a photoinjector optimization for ultrafast electron diffraction applications [54] with 13 decision variables. Due to the high computational cost of the test (20 to 30 minutes per function evaluation) and limits on the programming language the simulation tool and its interfaces are written in, it is restricted to a comparison between the new comparison-based evolutionary algorithm and a single conventional optimizer (NSGA-II here).

The problem setup is identical to the DC gun ultrafast electron diffraction optimization problem described in Ref. [54], with the following changes. First, constraints are not used in this test to make it compatible with the new optimization algorithm. Because these constraints protect the optimization against parts of decision space that may include beam loss, we also reduce the range of the following decision variables to avoid that condition. The two solenoid current settings are constrained to the ranges 2 A to 3 A and 1.5 A to 2.5 A respectively. The buncher's peak voltage is also restricted to a range of 10 kV to 30 kV. With these changes, we found good transmission of the beam across the space of decision variables. We also manually confirmed that the non-dominated solutions returned from the optimization algorithm had no beam loss. The cathode's mean transverse energy was set to 100 meV and the number of macroparticles used was 10000. The two objectives used in the optimization are: transverse emittance of the beam at the sample location (minimize) and the delivered bunch charge (maximize).

Each optimization was run 8 times with a budget of 500 function evaluations. The same algorithm settings were used as in the biobjective experiments in Sec. 4 except for starting with an initial population size of 50. The hypervolume of the approximate fronts was calculated using a reference point of (0 fC, 9 nm). The mean hypervolume and a band indicating the 95% prediction interval are shown versus the number of function evaluations in Fig. 6a. Examples of the non-dominated solutions returned from the optimization algorithms are also shown in Fig. 6b.

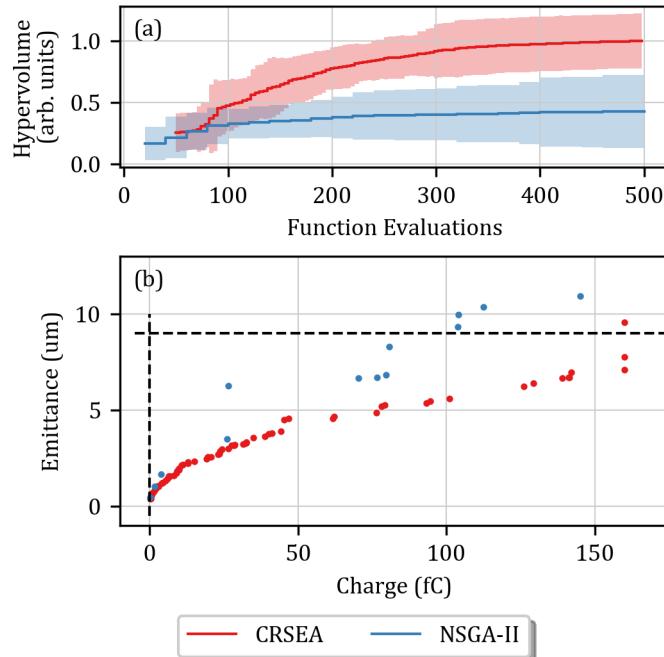


Figure 6: Results from the ultrafast electron diffraction photoinjector optimization problem: (a) Mean hypervolume metric for non-dominated solutions with the 95% prediction interval shown as the shaded regions; (b) The solutions after 500 function evaluations from the optimization with median value of the hypervolume for each algorithm. The reference point is shown on this plot as the dashed lines.

6. Conclusion

In this work, we investigated the use of a comparison-relationship-surrogate model to improve the quality of solutions achievable with multi-objective genetic algorithms on a fixed budget of function evaluations. This type of model, given two individuals, will predict which of them has the smaller value of each objective function. Domination relationships can then be inferred after a “cleaning” procedure allowing the model to be integrated into a non-dominated sorting genetic algorithm. Its performance was then compared to that of a selection of other algorithms on the DTLZ and WFG test suites as well as a real-world problem from the field of accelerator physics.

Our numerical experiments found that the comparison-relationship-surrogate optimizer produced among the best solutions of all the algorithms studied for many *medium-scale, biobjective* problems (specifically WFG3-WFG9 in Tab. 2). Tests on the real-world optimization problem also showed the ability of the algorithm to find a larger solution hypervolume than the tested conventional genetic optimizer (NSGA-II). Results were not as promising on the DTLZ test suite where solution diversity may have limited performance.

Finding diversity preservation methods which are compatible with predictions from the comparison-relationship-surrogate model will be an important line of research coming out of this work. Inclusion of constraints using a model of feasibility is also a natural extension to the work as it can be naturally added to the prediction of domination in the same way as in Ref. [42]. With continued work in this area, these algorithms could help to improve the results available to the scientific and industrial users of optimization algorithms that face expensive multi-objective problems on a limited budget.

7. CRediT Authorship Contribution Statement

Christopher M. Pierce: Conceptualization, Investigation, Software, Visualization, Writing – original draft, Writing – review & editing. **Young-Kee Kim:** Funding acquisition, Supervision, Writing – review & editing. **Ivan Bazarov:** Funding acquisition, Supervision, Writing – review & editing.

8. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

9. Acknowledgments

This work was supported by the U.S. National Science Foundation under Award PHY-1549132, the Center for Bright Beams.

References

- [1] M. John, M. J. Ammann, Antenna Optimization With a Computationally Efficient Multiobjective Evolutionary Algorithm, *IEEE Transactions on Antennas and Propagation* 57 (1) (Jan. 2009). doi:10.1109/TAP.2008.2009775.
- [2] A. Grosdidier, V. Zoete, O. Michelin, EADock: Docking of small molecules into protein active sites with a multiobjective evolutionary

- optimization, Proteins: Structure, Function, and Bioinformatics 67 (4) (2007). doi:10.1002/prot.21367.
- [3] M. G. Castillo Tapia, C. A. Coello Coello, Applications of multi-objective evolutionary algorithms in economics and finance: A survey, in: 2007 IEEE Congress on Evolutionary Computation, 2007. doi:10.1109/CEC.2007.4424516.
 - [4] I. V. Bazarov, C. K. Sinclair, Multivariate optimization of a high brightness dc gun photoinjector, Physical Review Special Topics - Accelerators and Beams 8 (3) (Mar. 2005). doi:10.1103/PhysRevSTAB.8.034202.
 - [5] Y. Lian, M.-S. Liou, Multi-Objective Optimization of Transonic Compressor Blade Using Evolutionary Algorithm, Journal of Propulsion and Power 21 (6) (Nov. 2005). doi:10.2514/1.14667.
 - [6] R. Roussel, A. Hanuka, A. Edelen, Multiobjective Bayesian optimization for online accelerator tuning, Physical Review Accelerators and Beams 24 (6) (Jun. 2021). doi:10.1103/PhysRevAccelBeams.24.062801.
 - [7] J. Knowles, Closed-loop evolutionary multiobjective optimization, IEEE Computational Intelligence Magazine 4 (3) (Aug. 2009). doi:10.1109/MCI.2009.933095.
 - [8] A. D. Clayton, A. M. Schweidtmann, G. Clemens, J. A. Manson, C. J. Taylor, C. G. Niño, T. W. Chamberlain, N. Kapur, A. J. Blacker, A. A. Lapkin, R. A. Bourne, Automated self-optimisation of multi-step reaction and separation processes using machine learning, Chemical Engineering Journal 384 (Mar. 2020). doi:10.1016/j.cej.2019.123340.

- [9] Y. Jin, Surrogate-assisted evolutionary computation: Recent advances and future challenges, *Swarm and Evolutionary Computation* 1 (2) (Jun. 2011). [doi:10.1016/j.swevo.2011.05.001](https://doi.org/10.1016/j.swevo.2011.05.001).
- [10] D. A. V. Veldhuizen, G. B. Lamont, Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art, *Evolutionary Computation* 8 (2) (Jun. 2000). [doi:10.1162/106365600568158](https://doi.org/10.1162/106365600568158).
- [11] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, K. Sindhya, A Surrogate-Assisted Reference Vector Guided Evolutionary Algorithm for Computationally Expensive Many-Objective Optimization, *IEEE Transactions on Evolutionary Computation* 22 (1) (Feb. 2018). [doi:10.1109/TEVC.2016.2622301](https://doi.org/10.1109/TEVC.2016.2622301).
- [12] Y. Jin, A comprehensive survey of fitness approximation in evolutionary computation, *Soft Computing* 9 (1) (Jan. 2005). [doi:10.1007/s00500-003-0328-5](https://doi.org/10.1007/s00500-003-0328-5).
- [13] K. Deb, P. C. Roy, R. Hussein, Surrogate Modeling Approaches for Multiobjective Optimization: Methods, Taxonomy, and Results, *Mathematical and Computational Applications* 26 (1) (Mar. 2021). [doi:10.3390/mca26010005](https://doi.org/10.3390/mca26010005).
- [14] J. Knowles, ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems, *IEEE Transactions on Evolutionary Computation* 10 (1) (Feb. 2006). [doi:10.1109/TEVC.2005.851274](https://doi.org/10.1109/TEVC.2005.851274).

- [15] M. A. OLIVER, R. WEBSTER, Kriging: A method of interpolation for geographical information systems, *International journal of geographical information systems* 4 (3) (1990) 313–332. doi:10.1080/02693799008941549.
- [16] W. Ponweiser, T. Wagner, D. Biermann, M. Vincze, Multiobjective Optimization on a Limited Budget of Evaluations Using Model-Assisted S-Metric Selection, *Parallel Problem Solving from Nature* (2008) 784–794doi:10.1007/978-3-540-87700-4_78.
- [17] Q. Zhang, W. Liu, E. Tsang, B. Virginas, Expensive Multiobjective Optimization by MOEA/D With Gaussian Process Model, *IEEE Transactions on Evolutionary Computation* 14 (3) (Jun. 2010). doi:10.1109/TEVC.2009.2033671.
- [18] I. Loshchilov, M. Schoenauer, M. Sebag, A mono surrogate for multi-objective optimization, in: *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, Association for Computing Machinery, New York, NY, USA, 2010, pp. 471–478. doi:10.1145/1830483.1830571.
- [19] I. Loshchilov, M. Schoenauer, M. Sebag, Dominance-Based Pareto-Surrogate for Multi-Objective Optimization, in: K. Deb, A. Bhattacharya, N. Chakraborti, P. Chakroborty, S. Das, J. Dutta, S. K. Gupta, A. Jain, V. Aggarwal, J. Branke, S. J. Louis, K. C. Tan (Eds.), *Proceedings of SEAL 2010*, Vol. 6457, Springer Berlin Heidelberg, 2010. doi:10.1007/978-3-642-17298-4_24.

- [20] J. Zhang, A. Zhou, G. Zhang, A classification and Pareto domination based multiobjective evolutionary algorithm, in: Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC), 2015. doi:10.1109/CEC.2015.7257247.
- [21] X. Lin, Q. Zhang, S. Kwong, A decomposition based multiobjective evolutionary algorithm with classification, in: 2016 IEEE Congress on Evolutionary Computation (CEC), 2016, pp. 3292–3299. doi:10.1109/CEC.2016.7744206.
- [22] J. Zhang, A. Zhou, G. Zhang, Preselection via one-class classification for evolutionary optimization, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO ’18, Association for Computing Machinery, New York, NY, USA, 2018, p. 259. doi:10.1145/3205651.3205717.
- [23] J. Li, P. Wang, H. Dong, J. Shen, C. Chen, A classification surrogate-assisted multi-objective evolutionary algorithm for expensive optimization, Knowledge-Based Systems 242 (Apr. 2022). doi:10.1016/j.knosys.2022.108416.
- [24] T. Sonoda, M. Nakata, Multiple Classifiers-Assisted Evolutionary Algorithm Based on Decomposition for High-Dimensional Multiobjective Problems, IEEE Transactions on Evolutionary Computation 26 (6) (2022) 1581–1595. doi:10.1109/TEVC.2022.3159000.
- [25] H. Geng, F. Song, J. Shen, J. Li, A classification and regression assisted optimization algorithm for high-dimensional expensive many-

objective problems, Neurocomputing 586 (2024) 127629. doi:10.1016/j.neucom.2024.127629.

- [26] S. Bandaru, A. H. Ng, K. Deb, On the performance of classification algorithms for learning Pareto-dominance relations, in: 2014 IEEE Congress on Evolutionary Computation (CEC), IEEE, Beijing, China, 2014, pp. 1139–1146. doi:10.1109/CEC.2014.6900641.
- [27] L. Pan, C. He, Y. Tian, H. Wang, X. Zhang, Y. Jin, A Classification-Based Surrogate-Assisted Evolutionary Algorithm for Expensive Many-Objective Optimization, IEEE Transactions on Evolutionary Computation 23 (1) (Feb. 2019). doi:10.1109/TEVC.2018.2802784.
- [28] Y. Yuan, W. Banzhaf, Expensive Multiobjective Evolutionary Optimization Assisted by Dominance Prediction, IEEE Transactions on Evolutionary Computation 26 (1) (Feb. 2022). doi:10.1109/TEVC.2021.3098257.
- [29] H. Hao, A. Zhou, H. Qian, H. Zhang, Expensive Multiobjective Optimization by Relation Learning and Prediction, IEEE Transactions on Evolutionary Computation 26 (5) (Oct. 2022). doi:10.1109/TEVC.2022.3152582.
- [30] H. Hao, A. Zhou, A Relation Surrogate Model for Expensive Multiobjective Continuous and Combinatorial Optimization, in: M. Emmerich, A. Deutz, H. Wang, A. V. Kononova, B. Naujoks, K. Li, K. Miettinen, I. Yevseyeva (Eds.), Evolutionary Multi-Criterion Op-

timization, Springer Nature Switzerland, Cham, 2023, pp. 205–217.
doi:10.1007/978-3-031-27250-9_15.

- [31] J. Shen, P. Wang, H. Dong, W. Wang, Expensive Many-Objective Optimization by Learning of the Strengthened Dominance Relation, in: 2023 IEEE Congress on Evolutionary Computation (CEC), 2023, pp. 1–8. doi:10.1109/CEC53210.2023.10254133.
- [32] Y. Tian, J. Hu, C. He, H. Ma, L. Zhang, X. Zhang, A pairwise comparison based surrogate-assisted evolutionary algorithm for expensive multi-objective optimization, Swarm and Evolutionary Computation 80 (2023) 101323. doi:10.1016/j.swevo.2023.101323.
- [33] L. Ma, N. Li, G. Yu, X. Geng, S. Cheng, X. Wang, M. Huang, Y. Jin, Pareto-Wise Ranking Classifier for Multiobjective Evolutionary Neural Architecture Search, IEEE Transactions on Evolutionary Computation 28 (3) (2024) 570–581. doi:10.1109/TEVC.2023.3314766.
- [34] T. P. Runarsson, Ordinal Regression in Evolutionary Computation, in: T. P. Runarsson, H.-G. Beyer, E. Burke, J. J. Merelo-Guervós, L. D. Whitley, X. Yao (Eds.), Proceedings of Parallel Problem Solving from Nature - PPSN IX, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2006. doi:10.1007/11844297_106.
- [35] H. Hao, J. Zhang, X. Lu, A. Zhou, Binary Relation Learning and Classifying for Preselection in Evolutionary Algorithms, IEEE Transactions on Evolutionary Computation 24 (6) (2020) 1125–1139. doi:10.1109/TEVC.2020.2986348.

- [36] P. S. Naharro, P. Toharia, A. LaTorre, J.-M. Peña, Comparative study of regression vs pairwise models for surrogate-based heuristic optimisation, *Swarm and Evolutionary Computation* 75 (2022) 101176. doi:[10.1016/j.swevo.2022.101176](https://doi.org/10.1016/j.swevo.2022.101176).
- [37] K. Hakhamaneshi, N. Werblun, P. Abbeel, V. Stojanovic, BagNet: Berkeley Analog Generator with Layout Optimizer Boosted with Deep Neural Networks, in: Proceedings of the 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2019.
- [38] I. Loshchilov, M. Schoenauer, M. Sebag, Comparison-Based Optimizers Need Comparison-Based Surrogates, in: R. Schaefer, C. Cotta, J. Kołodziej, G. Rudolph (Eds.), *Proceedings of Parallel Problem Solving from Nature, PPSN XI*, Krakow, Poland, 2010. doi:[10.1007/978-3-642-15844-5_37](https://doi.org/10.1007/978-3-642-15844-5_37).
- [39] S. Gelly, S. Ruette, O. Teytaud, Comparison-Based Algorithms Are Robust and Randomized Algorithms Are Anytime, *Evolutionary Computation* 15 (4) (Dec. 2007). doi:[10.1162/evco.2007.15.4.411](https://doi.org/10.1162/evco.2007.15.4.411).
- [40] A. Auger, N. Hansen, J. M. Perez Zerpa, R. Ros, M. Schoenauer, Experimental Comparisons of Derivative Free Optimization Algorithms, in: *Proceedings of the International Symposium on Experimental Algorithms*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. doi:[10.1007/978-3-642-02011-7_3](https://doi.org/10.1007/978-3-642-02011-7_3).
- [41] M. D. McKay, R. J. Beckman, W. J. Conover, A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of

Output From a Computer Code, *Technometrics* 42 (1) (Feb. 2000).
doi:10.1080/00401706.2000.10485979.

- [42] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation* 6 (2) (Apr. 2002). doi:10.1109/4235.996017.
- [43] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley-Interscience Series in Systems and Optimization, John Wiley & Sons, Chichester; New York, 2001.
- [44] N. B. Shah, M. J. Wainwright, Simple, Robust and Optimal Ranking from Pairwise Comparisons, *Journal of Machine Learning Research* 18 (Apr. 2018).
- [45] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feed-forward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010.
- [46] I. Loshchilov, F. Hutter, Decoupled Weight Decay Regularization (Jan. 2019). arXiv:1711.05101.
- [47] I. Loshchilov, F. Hutter, SGDR: Stochastic Gradient Descent with Warm Restarts (May 2017). arXiv:1608.03983.
- [48] S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit, *IEEE Transactions on Evolutionary Computation* 10 (5) (Oct. 2006). doi:10.1109/TEVC.2005.861417.

- [49] C. A. C. Coello, N. C. Cortes, Solving Multiobjective Optimization Problems Using an Artificial Immune System, *Genetic Programming and Evolvable Machines* 6 (2) (Jun. 2005). doi:10.1007/s10710-005-6164-x.
- [50] K. Deb, H. Jain, An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints, *IEEE Transactions on Evolutionary Computation* 18 (4) (Aug. 2014). doi:10.1109/TEVC.2013.2281535.
- [51] Y. Tian, R. Cheng, X. Zhang, Y. Jin, PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization [Educational Forum], *IEEE Computational Intelligence Magazine* 12 (4) (Nov. 2017). doi:10.1109/MCI.2017.2742868.
- [52] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable multi-objective optimization test problems, in: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02, 2002. doi:10.1109/CEC.2002.1007032.
- [53] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization, *IEEE Transactions on Evolutionary Computation* 20 (5) (Oct. 2016). doi:10.1109/TEVC.2016.2519378.
- [54] C. M. Pierce, M. B. Andorf, E. Lu, C. Gulliford, I. V. Bazarov, J. M. Maxson, M. Gordon, Y.-K. Kim, N. P. Norvell, B. M. Dunham,

T. O. Raubenheimer, Low intrinsic emittance in modern photoinjector brightness, Physical Review Accelerators and Beams 23 (7) (Jul. 2020).
doi:10.1103/PhysRevAccelBeams.23.070101.