A Two-Phase Surrogate Approach for High-Dimensional Constrained Discrete Multi-Objective Optimization

Rommel G. Regis
rregis@sju.edu
Saint Joseph's University
Philadelphia, Pennsylvania, USA

ABSTRACT

This paper presents a two-phase surrogate approach for high-dimensional constrained discrete multi-objective optimization. In Phase I, the algorithm searches for a feasible point using surrogates for the constraints and objectives. In Phase I iterations, the algorithm identifies the infeasible points that are nondominated according to three criteria: number of constraint violations, maximum constraint violation, and sum of squares of constraint violations. Moreover, the function evaluation point is chosen from a large number of trial points in the neighborhood of a current nondominated point according to the predicted values of the above criteria. In Phase II, the algorithm searches for Pareto optimal solutions using surrogates for the objectives and constraints. In Phase II iterations, the function evaluation point is chosen from trial points that are predicted to be feasible and nondominated in the neighborhood of a current nondominated point using distance criteria in the objective and decision spaces. The algorithm is implemented using RBF surrogates and tested on the Mazda benchmark problem that has 222 discrete variables, 54 constraints and 2 objectives. The proposed method found feasible points much more quickly and obtained much better sets of nondominated objective vectors than an NSGA-II implementation given a budget of only 3330 simulations.

CCS CONCEPTS

• **Mathematics of computing** → **Mathematical software**; • **Computing methodologies** → **Genetic algorithms**; • **Applied computing** → *Engineering*.

KEYWORDS

multi-objective optimization, constrained optimization, discrete variables, high-dimensional optimization, surrogate models, radial basis functions

ACM Reference Format:
Rommel G. Regis. 2021. A Two-Phase Surrogate Approach for High-Dimensional Constrained Discrete Multi-Objective Optimization. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3449726.3463204

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

ACM ISBN 978-1-4503-8351-6/21/07…$15.00
https://doi.org/10.1145/3449726.3463204

1 INTRODUCTION

Surrogate approaches have been proposed for computationally expensive multi-objective optimization (e.g., [1, 12, 16, 19, 25]) and some of these methods can handle black-box inequality constraints (e.g., [5, 8, 11, 15, 22, 24]). However, relatively few surrogate-based or surrogate-assisted optimization methods have been proposed for problems with discrete variables [3]. Even fewer methods can handle high-dimensional multi-objective problems with discrete variables and many black-box constraints. For example, Brownlee and Wright [4] proposed a variant of NSGA-II for constrained mixed-integer multi-objective optimization that uses radial basis function (RBF) networks for fitness approximation and applied it to a building design problem with 50 decision variables, of which 20 are discrete variables. Moreover, Ohtsuka et al. [17] proposed a decomposition-based constrained multi-objective evolutionary algorithm CMOEA/D that uses an Extreme Learning Machine (ELM) and applied it to the 3-car Mazda benchmark problem [6, 13] that involves 222 discrete variables, 54 black-box inequality constraints and 2 objective functions. However, the ELM-assisted CMOEA/D in [17] aggregated the constraints for each type of car, resulting in only 3 constraints that are each modeled by an ELM. This paper proposes a two-phase surrogate approach to handle challenging multi-objective problems such as the Mazda benchmark.

Formally, the focus of this paper is on solving the following multi-objective optimization problem:

$$\begin{aligned} &\min\ F(x) = (f_1(x), \ldots, f_k(x)) \\ \text{s.t.}\ & \\ &G(x) = (g_1(x), \ldots, g_m(x)) \leq 0 \\ &x^{(i)} \in D_i \subset \mathbb{R} \text{ with } |D_i| < \infty,\ i = 1, \ldots, d \end{aligned} \tag{1}$$

where $x^{(i)}$ is the ith decision variable and the ith entry of $x \in \mathbb{R}^d$, and D_i is the finite set of the possible discrete ordinal values of $x^{(i)}$. In practical applications, the values in D_i are *not* necessarily integers. They could be fractional settings that are allowed for the ith variable. For example, $x^{(i)}$ could take on values from the set $D_i = \{1.2, 1.5, 2.3, 2.5\}$. Moreover, the objective functions $f_i : \mathbb{R}^d \longrightarrow \mathbb{R}$, $i = 1, \ldots, k$ and constraint functions $g_j : \mathbb{R}^d \longrightarrow \mathbb{R}$, $j = 1, \ldots, m$ are black-box and computationally expensive. For example, the values of these functions might be obtained by running an expensive simulation. For now, assume that there is no noise in the calculation of the f_i's and g_j's. Future work will address the issue of noise in the objective and constraint functions.

In problem (1), the finite set $\prod_{i=1}^{d} D_i \subset \mathbb{R}^d$ is the *search space* for problem. The lower and upper bounds of the variables are given by $\ell_i := \min D_i$ and $u_i := \max D_i$ for $i = 1, \ldots, d$. Clearly, the box $[\ell, u] = \prod_{i=1}^{d} [\ell_i, u_i]$ encloses the search space of problem (1). Throughout this paper, assume that one *simulation* at a given point

x in the search space yields the values of all the objective and constraint functions at x, i.e., all the components of $F(x)$ and $G(x)$.

The POSEIDON algorithm [23] is a recently proposed surrogate approach that can handle computationally expensive and high-dimensional constrained multi-objective problems with ordinal discrete variables and many black-box constraints, i.e., problem (1) with large dimension d and number of constraints m. POSEIDON using RBF surrogates for each of the objectives and constraints significantly outperformed an implementation of NSGA-II on the 3-car Mazda Benchmark problem that has 222 ordinal discrete decision variables, 54 black-box inequality constraints and 2 objective functions [6] when given a relatively limited computational budget. Note that in contrast to the ELM-assisted CMOEA/D in [17], POSEIDON uses a surrogate for each for the 54 constraints in the Mazda benchmark. However, POSEIDON assumes that a feasible initial point is given, which is sometimes difficult to obtain in some applications. This paper proposes the *Two-Phase POSEIDON* algorithm, which extends the capabilities of POSEIDON to solve constrained discrete multi-objective problems when no feasible initial point is provided. Moreover, this paper also proposes a modification to the original POSEIDON algorithm that improves its ability to explore and expand the extremes of the set of nondominated objective vectors in the objective space. Finally, the main ideas in Two-Phase POSEIDON can be used for surrogate-assisted evolutionary algorithms for constrained discrete multi-objective optimization.

In Phase I of the proposed method, all sample points are infeasible and the algorithm searches for a feasible point using surrogates for the constraints and objectives. In each iteration of Phase I, the algorithm identifies the infeasible points that are nondominated according to three criteria: number of constraint violations, maximum constraint violation, and sum of squares of the constraint violations. Moreover, the next simulation point is chosen from a large number of trial points in the neighborhood of a current nondominated point according to the predicted values of the above criteria. In Phase II, the algorithm searches for Pareto optimal solutions using surrogates for the objectives and constraints. In each iteration of Phase II, the next simulation point is chosen from trial points that are predicted to be feasible and predicted to be nondominated in the neighborhood of a current nondominated point using distance criteria in the decision and objective spaces.

The proposed Two-Phase POSEIDON algorithm is implemented using RBF surrogates and tested on the 3-car Mazda benchmark problem using a computational budget of only $15d = 3330$ simulations. The proposed algorithm found feasible points much more quickly and yielded much better sets of nondominated objective vectors than an implementation of NSGA-II that uses discrete variable encoding. In addition, the method significantly improves on an initial feasible design on the Mazda benchmark given the limited computational budget.

2 THE TWO-PHASE POSEIDON ALGORITHM

2.1 Algorithm Description

This paper proposes the *Two-Phase POSEIDON (Pareto Optimization using Surrogates for Expensive Inequality-constrained Discrete Ordinal and Nonlinear problems)* algorithm for solving the computationally expensive constrained discrete multi-objective optimization problem in (1). The proposed method is an extension of the surrogate-based POSEIDON algorithm in [23] that employs a two-phase approach and can be used when no feasible point is available among the initial sample points. The original POSEIDON algorithm assumes that a feasible initial point is provided but this might not be the case in some practical applications. The first phase of Two-Phase POSEIDON finds a feasible point while the second phase searches for Pareto optimal solutions or at least a set of non-dominated solutions that improve on the feasible point found in the first phase. Because the objective and constraint functions are computationally expensive, the two phases are implemented using a relatively limited computational budget of simulations and both phases employ surrogates for the objective and constraints. The use of a two-phase approach in the context of surrogate-based constrained optimization with continuous variables has been employed in the COBRA algorithm [21]. However, Two-Phase POSEIDON not only deals with discrete variables, it uses different strategies for using surrogates to select sample points in both phases.

Two-Phase POSEIDON begins by selecting an initial set of points in the search space where the objective and constraint functions will be evaluated. These points could be generated uniformly at random from the discrete search space, or by using a space-filling design that is suitable for discrete search spaces. Then the algorithm runs simulations to obtain the objective and constraint function values at the initial points to obtain information needed to fit the initial surrogate models. For problems with black-box constraints, there is no guarantee that a feasible point is available among the initial points. Hence, Phase I of the algorithm uses surrogate models for the constraints and the objectives to find a feasible point for the problem. Then, Phase II searches for Pareto optimal solutions using again the surrogates for the constraints and objectives. If there is a feasible point among the initial sample points, then the algorithm simply proceeds to Phase II as was done in the original POSEIDON algorithm [23].

In each iteration of Phase I, the algorithm fits or updates the surrogates, one for each objective function and for each inequality constraint function, and then identifies or updates the set of infeasible sample points that are nondominated according to three criteria: number of constraint violations, maximum constraint violation, and sum of squares of the constraint violations. Then, the algorithm randomly chooses one of the current nondominated points and then selects the next point where the simulation will take place from a large number of trial points in the neighborhood of the chosen nondominated point. The trial points are generated by perturbing some or all of the components of the chosen nondominated point. The generation of trial points are described below after the pseudocode. To obtain the simulation point, the algorithm uses the surrogates for the constraints to identify the trial points that are predicted to be nondominated according to the predicted values of the above criteria, namely predicted number of constraint violations, maximum predicted constraint violation, and sum of squares of the predicted constraint violations. Note that if there are trial points that are predicted to be feasible, then the predicted values for the above criteria are all zero for these trial points. In this case, the algorithm determines which of these trial points are predicted to be nondominated based on the surrogates of the objectives, and among these trial points, selects the one that is farthest from all previously evaluated points to promote exploration of the search space. On the other hand, if none of the trial points are predicted to be feasible, then the algorithm collects all trial points with the minimum number of predicted constraint violations from among those that are predicted to be nondominated according to the predicted values of the three criteria above. The simulation point is then chosen from these trial points as the one with the best weighted combination of scaled values of the maximum predicted constraint violation (*MPCV criterion*) and the sum of squares of the predicted constraint violations (*SPCV criterion*). Next, a simulation is performed to obtain the objective and constraint values at the chosen trial point. The algorithm goes through the iterations in Phase I until a feasible point is found, after which the algorithm proceeds to Phase II.

In each iteration of Phase II, the algorithm again fits the surrogates of the objectives and constraints and identifies or updates the nondominated set of points from all previous feasible sample points. In contrast to Phase I, nondomination among feasible points in Phase II is now based on the objective function values. Next, the algorithm chooses one of the current nondominated points in the decision space and perturbs it many times to generate a large number of trial points in the neighborhood of the nondominated point. The nondominated point that will be perturbed may be chosen uniformly at random among all current nondominated points (*Random (RND)* strategy) or it may be chosen to be the one whose objective vector is the most isolated (in terms of Euclidean distance) from the other nondominated objective vectors in the objective space (*Objective Space Distance (OSD)* strategy). These two strategies were previously employed in the original version of POSEIDON [23]. This paper employs another strategy for POSEIDON called the *Extreme Objective Vector (EOV)* strategy where the nondominated point chosen minimizes one of the objective functions among all current nondominated points. This strategy is meant to promote exploration of the extremes of the current approximate Pareto front.

From the collection of all trial points in a given iteration of Phase II, the surrogates for the constraints are used to identify the trial points that are predicted to be feasible or those that have the minimum number of predicted constraint violations. Among these eligible trial points, the surrogates for the objectives are used to identify the trial points that are predicted to be nondominated by the other eligible trial points and by the current nondominated points. From this subset of eligible trial points that are predicted to be nondominated, we then select the best trial point according to a weighted combination of scaled values of two criteria: (i) minimum distance between the predicted objective vector of the trial point and the current nondominated objective vectors (in the objective space) (*MDOS criterion*); and (ii) minimum distance between the trial point and the current nondominated points (in the decision space) (*MDDS criterion*). Hence, the algorithm selects a trial point whose predicted objective vector is far from the nondominated objective vectors to promote good spacing in the approximate Pareto front in the objective space. At the same time, it also selects a trial point that is far from the current nondominated points to promote exploration of the decision space. The weights for these two criteria are set with more weight on the former than the latter. Once the best trial point is selected, a simulation is performed to obtain the objective and constraint values at that point. The algorithm goes through the iterations in Phase II until the computational budget is exhausted.

Below is a pseudo-code that outlines the main steps of the Two-Phase POSEIDON algorithm for solving the constrained discrete multi-objective optimization problem of the form (1) .

Two-Phase POSEIDON Algorithm

(1) *(Initial Simulations)* Perform simulations to obtain objective and constraint function values at initial set of points in the search space. If one of these initial points is feasible, go to Step 3 (Phase II).

(2) *(Phase I Iterations)* While a feasible sample point has not been found or while the computational budget has *not* been exhausted do:

- **(a)** *(Determine Nondominated Set)* Identify the previously evaluated points (all infeasible) that are nondominated according to three criteria: number of constraint violations, maximum constraint violation, and sum of squares of the constraint violations.
- **(b)** *(Fit Surrogates)* Fit or update surrogate for each objective and each constraint.
- **(c)** *(Generate Trial Points)* Select a point uniformly at random from the current set of nondominated points (all infeasible) and generate a large number of trial points in its neighborhood.
- **(d)** *(Identify Trial Points Predicted to be Nondominated)* Evaluate the surrogates for the constraints at the trial points in Step 2(c) and identify the trial points that are predicted to be nondominated according to the predicted values of the three criteria above.
- **(e)** *(Select Simulation Point)* If one of the trial points in Step 2(d) is predicted to be feasible, then do (i) below, else do (ii).
  - (i) Evaluate the surrogates for the objectives at the trial points that are predicted to be feasible and determine which of these trial points are predicted to be nondominated based on the surrogates of the objectives. Among these trial points, select the one that is farthest from all previously evaluated points.
  - (ii) Collect all trial points with the minimum number of predicted constraint violations from among those that are predicted to be nondominated based on the predicted values of the three criteria above. From these trial points, choose the one with the best weighted combination of scaled values of the MPCV and SPCV criteria.
- **(f)** *(Simulate)* Perform one simulation to obtain the objective and constraint function values at the trial point chosen in Step 2(e).

end.

(3) *(Phase II Iterations)* While the computational budget has *not* been exhausted, do:

- **(a)** *(Determine Nondominated Set)* Identify the set of nondominated points among feasible points obtained so far.
- **(b)** *(Fit Surrogates)* Fit or update surrogate for each objective and each constraint.
- **(c)** *(Generate Trial Points)* Select one of the nondominated points using the RND, OSD or EOV strategies and generate

a large number of trial points in the neighborhood of the chosen nondominated point.

(d) *(Determine Eligible Trial Points)* Evaluate the surrogates for the constraints at the trial points in Step 3(c) and identify trial points that are predicted to be feasible or those with the minimum number of predicted constraint violations.

(e) *(Identify Eligible Points Predicted to be Nondominated)* Evaluate the surrogates for the objectives at the eligible trial points obtained in Step 3(d) and identify those that are predicted to be nondominated by the other eligible trial points and by the current nondominated points.

(f) *(Select Simulation Point)* From the eligible trial points predicted to be nondominated in Step 3(e), choose the best one according to a weighted combination of scaled values of the MDOS and MDDS criteria.

(g) *(Simulate)* Perform one simulation to obtain the objective and constraint function values at the trial point chosen in Step 3(f).

end.

(4) *(Return Nondominated Set)* Return set of nondominated points and corresponding objective vectors and constraint function values.

Note that the criteria used for nondomination differ in the two phases. In Phase I, the nondominated set of points is identified from the set of all previous sample points, which are all infeasible, based on the number of constraint violations, maximum constraint violation, and sum of squares of the constraint violations. In Phase II, the nondominated set of points is obtained from all previous *feasible* sample points using their objective function values.

In each iteration of Phase I or Phase II, a large number of trial points is generated in the neighborhood of the chosen nondominated point. A trial point is obtained by perturbing some of the components (values of the variables) of the chosen nondominated point. As with the original POSEIDON [23], each component is perturbed with a certain probability denoted by p_{pert}. A perturbation consists of changing the current setting of a variable by either increasing or decreasing its value by a few discrete steps. The neighborhood depth parameter, denoted by $depth_{\text{nbhd}}$, indicates the fraction of the number of settings that the variable is allowed to increase or decrease. For example, suppose the current setting of variable $x^{(i)}$ in the current best point is 1.2 and there are 10 possible settings given by $\{0.5, 0.7, 0.9, 1.0, 1.2, 1.5, 1.8, 1.9, 2.0, 2.2\}$. If $depth_{\text{nbhd}} = 20\%$, then $x^{(i)}$ is allowed to take 20% of the possible discrete settings for that variable above or below the current setting. Hence, $x^{(i)}$ may be increased or decreased from 1.2 up to $0.2(10) = 2$ discrete steps, and so, $x^{(i)}$ may take on the possible values $\{0.9, 1.0, 1.5, 1.8\}$. Note that the current setting of 1.2 is excluded from the possible values, to force the value of $x^{(i)}$ to change.

2.2 Radial Basis Function Interpolation

The Two-Phase POSEIDON algorithm can be implemented using any type of surrogate, including popular choices such as Kriging or Gaussian process modeling [9], radial basis functions (RBF) (e.g., [1, 21]) and neural networks (e.g., [7]). The numerical experiments below use the RBF interpolation model in Powell [20], which differs from the more popular RBF network in the machine learning literature. In this RBF model, each data point is a center, the basis functions are not necessarily Gaussian, and training involves solving a linear system that has desirable mathematical properties. This RBF model is suitable for applying Two-Phase POSEIDON on the Mazda benchmark because it has been successfully applied to high-dimensional problems with hundreds of decision variables and many black-box constraints (e.g., see [2, 21]). In contrast, Kriging models tend to have numerical issues and can be time-consuming to build on high dimensional problems with hundreds of variables.

To describe how to train this RBF model, suppose we are given n distinct points $x_1, \ldots, x_n \in \mathbb{R}^d$ and the function values $u(x_1), \ldots, u(x_n)$, where $u(x)$ is either an objective or constraint function. The RBF interpolation model from Powell [20] has the form:

$$s(x) = \sum_{i=1}^{n} \lambda_i \phi(\|x - x_i\|) + p(x),\ x \in \mathbb{R}^d,$$

where $\|\cdot\|$ is the Euclidean norm, $\lambda_i \in \mathbb{R}$ for $i = 1, \ldots, n$, $p(x)$ is a linear polynomial in d variables, and ϕ has the *cubic* form: $\phi(r) = r^3$. The function ϕ can take other forms such as the thin plate spline ($\phi(r) = r^2 \log r$) and the Gaussian form ($\phi(r) = \exp(-\gamma r^2)$, where γ is a hyperparameter). A cubic RBF model is used because it does not require a hyperparameter that needs to be tuned and also because of its success in prior RBF methods (e.g., [21]). More details about how to fit this model are found in Powell [20].

3 NUMERICAL EXPERIMENTS

3.1 Description of the Mazda Benchmark

The proposed Two-Phase POSEIDON algorithm is tested on the 3-car Mazda Benchmark problem [10, 13, 18], which was jointly developed by the Mazda Motor Corporation, Japan Aerospace Exploration Agency, and Tokyo University of Science. This benchmark problem is a constrained discrete multi-objective optimization problem involving 222 ordinal discrete variables representing the thicknesses of structural parts, 54 inequality constraints such as collision safety performance requirements, and 2 objective functions one of which is the total weight of three types of Mazda cars and the other is the number of common gauge parts. It is among the largest black-box optimization benchmark problems that are based on a real application. Because of practical considerations, the thickness of a structural part can only take values from a finite set of discrete settings. However, for each variable, the number of settings range from 4 to 18, and the total number of possible combinations of settings of the discrete variables is 4.4427×10^{198}, which is beyond astronomical. The multi-objective problem is to determine trade-off feasible solutions that minimize the total weight of the three types of cars (sport utility vehicle Mazda CX-5 (SUV), large vehicle Mazda 6 (LV), and small vehicle Mazda 3 (SV)) and maximize the number of common gauge parts. Design optimization of car structures involve simulations that are computationally very expensive. However, the code for the Mazda benchmark evaluates relatively quickly because the collision safety constraints are modeled by response surface approximations [13]. For more details of the problem, check out the webpage https://ladse.eng.isas.jaxa.jp/benchmark/index.html.

3.2 Experimental Setup

Two-Phase POSEIDON is implemented using RBF surrogates and two variants are considered: *POSEIDON (RBF-local)*, which is more focused on local search, and *POSEIDON (RBF-global)*, which performs more global search. For Phase II of both variants, the OSD strategy to select the nondominated point that is perturbed to generate the trial points is applied to an entire cycle of iterations followed by the RND strategy for the next cycle of iterations, then the OSD strategy again for the third cycle, and then the EOV strategy for the fourth cycle, and then this entire sequence of cycles repeats until the computational budget is exhausted. Moreover, in each of the two phases, both variants generate the trial points from the chosen nondominated point by performing cycles of iterations where the perturbation probability p_{pert} and the neighborhood depth parameter $depth_{\text{nbhd}}$ vary within the cycle. The POSEIDON (RBF-local) variant uses a cycle of four iterations where the control parameters (p_{pert}, $depth_{\text{nbhd}}$) take on the values $\langle(0.5, 30\%), (0.1, 10\%), (0.05, 10\%), (0.01, 10\%)\rangle$ as in [23]. The POSEIDON (RBF-global) variant uses a cycle of five iterations where the control parameters (p_{pert}, $depth_{\text{nbhd}}$) take on the values $\langle(0.5, 50\%), (0.3, 50\%), (0.2, 10\%), (0.1, 10\%), (0.05, 10\%)\rangle$ again as in [23].

For a given perturbation probability p_{pert}, the number of variables perturbed follows a binomial distribution and the mean number of variables perturbed is $d \cdot p_{\text{pert}}$. For the Mazda benchmark, when POSEIDON uses $p_{\text{pert}} = 0.01$, the mean number of variables perturbed is $222(0.01) = 2.22$, and when combined with a depth parameter of 10%, the generation of trial points is highly local and close to the chosen nondominated point. On the other hand, when POSEIDON uses $p_{\text{pert}} = 0.5$, the mean number of variables perturbed is $222(0.5) = 111$, and when combined with a depth parameter of 50%, the trial points generated facilitate global search.

The two variants of Two-Phase POSEIDON are compared with NSGA-II with discrete variable encoding and using various crossover fractions as implemented in the NPGM software [14]. Discrete variable encoding is used for NSGA-II because previous numerical results from [6] indicate that this encoding yields better results than continuous variable encoding. In particular, the discrete settings for each variable are converted to the integers 0, 1, 2, up to the maximum number of possible discrete settings minus one. Moreover, intermediate crossover is used since this is the only option supported by NPGM [14]. In addition, NSGA-II is run with a population size of 300 as in [13] and with the various crossover fractions set at 1.0, 0.75, 0.5 and 0.25. Ideally, POSEIDON should be compared with a surrogate-assisted constrained NSGA-II, but a code is not publicly available.

Two sets of experiments are performed. In the first set of experiments, the initial simulation points used by the POSEIDON algorithms do not include any feasible points. In the second set of experiments, the initial simulation points include the feasible design that is provided with the benchmark. In each set of experiments, each POSEIDON algorithm is run for 5 trials each with a computational budget of $15d = 3330$ simulations while each NSGA-II algorithm is run for 10 trials each with a computational budget of 3600 simulations (12 generations). To ensure a fair comparison, the initial set of points used for POSEIDON is the same as the initial population of 300 points used by NSGA-II. Moreover, for the first set of experiments, the first trial used the same initial population as the one used in [6], while for the other trials, an initial population was obtained by choosing points uniformly at random from the unimaginably vast search space. For the second set of experiments, the different trials used the same initial populations as in the first set of experiments except that the initial feasible design provided with the benchmark was included as the first individual and the last individual was removed to keep the number of initial points to the population size of 300.

All numerical experiments are performed in Matlab 9.4 on an Intel(R) Core(TM) i7-7700T CPU @ 2.90GHz, 2904 Mhz, 4 Core(s), 8 Logical Processor(s) machine. The Mazda benchmark was released as a C++ source code, so a Matlab interface was created to run the executable in the Matlab environment.

3.3 Comparison of Performance When None of the Initial Points is Feasible

In the first set of experiments, the local and global variants of Two-Phase POSEIDON that uses RBF surrogates are compared with NSGA-II algorithms when the initial population does not include any feasible points. First, the algorithms are compared in how quickly they are able to obtain a feasible sample point. Table 1 shows the number of simulations (including those for the initial population) it took the different algorithms to obtain a feasible sample point and the corresponding feasible objective vector for 5 trials. The last column of the table shows the mean of number of simulations needed to achieve feasibility for the algorithm.

The results in Table 1 clearly show that the POSEIDON-RBF algorithms are able to achieve feasibility much more quickly than any of the NSGA-II algorithms. In fact, removing the initial population size of 300 from the mean values in the table, it only takes an average of 29.2 and 16.4 simulations for POSEIDON (RBF-local) and POSEIDON (RBF-global), respectively, to find a feasible point. In contrast, the best performing NSGA-II algorithm (the one with crossover = 100%) requires an average of 2273 simulations to obtain a feasible point, which is about 78 times more than the average number of simulations needed by POSEIDON (RBF-local) and about 139 times more than the average number of simulations needed by POSEIDON (RBF-global). In terms of the quality of the first feasible objective vector obtained, the NSGA-II algorithms obtained better results but this was because of the large number of simulations that have been expended. As will be seen below, the POSEIDON-RBF algorithms are able to obtain much better feasible objective vectors when the computational budget is fixed.

Figure 1 shows the scatter plots of the nondominated objective vectors obtained by one trial of the POSEIDON-RBF and NSGA-II algorithms for the given computational budgets and starting with the same initial population of 300 sample points. Recall that the goal is to simultaneously minimize f_1 (the total weight of three types of Mazda cars) and maximize f_2 (the number of common gauge parts). The plots show that the POSEIDON-RBF algorithms obtained much better sets of nondominated objective vectors than the ones obtained by NSGA-II. In fact, the nondominated objective vectors obtained by each POSEIDON-RBF algorithm significantly dominate all the nondominated objective vectors obtained by any of

Table 1: Number of simulations to feasibility and first feasible sample point obtained for 5 trials of the POSEIDON algorithms and NSGA-II with various crossover fractions on the Mazda 3-Car Benchmark Problem. The first entry of each feasible point is the total weight of the 3 types of Mazda cars while the second entry is the number of common gauge parts.

Algorithm	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Mean
POSEIDON (RBF-local)	342	330	315	329	330	329.2
	[3.0348,2]	[3.0658,3]	[3.1258,2]	[3.0478,4]	[3.0528,5]	
POSEIDON (RBF-global)	319	319	313	318	313	316.4
	[3.1005,1]	[3.0967,2]	[3.1230,3]	[3.0991,3]	[3.1074,3]	
NSGA-II (crossover = 100%)	2732	2805	2748	2414	2166	2573.0
	[2.9891,19]	[2.9949,7]	[3.0627,3]	[2.9837,6]	[2.9872,14]	
NSGA-II (crossover = 75%)	2807	2858	2533	2961	2423	2716.4
	[3.0064,6]	[2.9664,6]	[3.0534,6]	[3.0332,7]	[3.0203,6]	
NSGA-II (crossover = 50%)	2965	3007	2724	3055	3436	3037.4
	[3.0209,7]	[3.0642,7]	[3.0392,4]	[3.0463,5]	[3.0265,7]	
NSGA-II (crossover = 25%)	3587	> 3600	> 3600	3466	> 3600	> 3570.6
	[2.9904,10]	NA	NA	[3.0397,4]	NA	

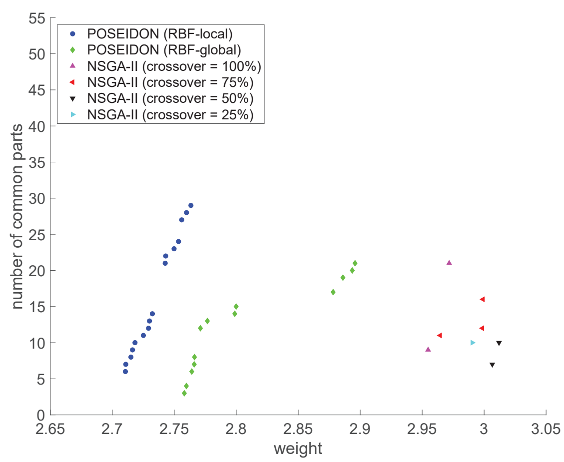

Figure 1: Nondominated objective vectors obtained by two POSEIDON algorithms that use RBF surrogates and when none of the initial points is feasible. The NSGA-II results shown used a population size of 300, discrete variable encoding and crossover fractions from 25% to 100%. The computational budget is $15d = 3330$ simulations for POSEIDON and 3600 simulations (12 generations) for NSGA-II.

the NSGA-II algorithms. Moreover, the local variant of POSEIDON-RBF obtained a much better set of nondominated points than the one obtained by the global variant.

Next, Table 2 shows the mean hypervolumes obtained by the POSEIDON-RBF and NSGA-II algorithms after normalizing the objective functions and using the reference point [1.1, 0] as suggested in [6]. The normalized objectives are obtained by: $\widetilde{f_1} = f_1 - 2$ and $\widetilde{f_2} = f_2/74$. The POSEIDON-RBF algorithms are run with a computational budget of $15d = 3330$ simulations while the NSGA-II

Table 2: Mean hypervolumes for 5 trials of the POSEIDON algorithms and 10 trials of the NSGA-II algorithms on the Mazda 3-Car Benchmark. The number inside the parenthesis is the standard error of the mean. The objective functions (total weight and number of common parts) are normalized as in [6] and the reference point is $[1.1, 0]$.

Algorithm	Mean Hypervolume
POSEIDON (RBF-local)	0.1367 (0.0072)
POSEIDON (RBF-global)	0.1041 (0.0076)
NSGA-II (crossover fraction = 100%)	0.0310 (0.0018)
NSGA-II (crossover fraction = 75%)	0.0235 (0.0010)
NSGA-II (crossover fraction = 50%)	0.0099 (0.0007)
NSGA-II (crossover fraction = 25%)	0.0018 (0.0015)

algorithms are run with a computational budget of 3600 simulations (12 generations). The mean hypervolumes are calculated over 5 trials for the POSEIDON-RBF algorithms and over 10 trials for the NSGA-II algorithms. Table 2 also reports the standard error of the mean hypervolumes for each of the algorithms. The table shows that the hypervolumes obtained by the POSEIDON-RBF algorithms are much better than the hypervolumes obtained by the NSGA-II algorithms even as the computational budget for POSEIDON-RBF (3330 simulations) is less than that for the NSGA-II algorithms (3600 simulations). Moreover, the mean hypervolume obtained by POSEIDON (RBF-local) is statistically significantly better than the one obtained by POSEIDON (RBF-global).

The results on the mean hypervolumes in Table 2 and the plot in Figure 1 suggest that the local variant of POSEIDON-RBF is more effective than the global variant. This is consistent with the numerical results obtained for the original POSEIDON-RBF in [23] where a feasible initial point is provided. A possible explanation for this is that the local variant makes more conservative changes to a current nondominated point, which facilitates the generation of feasible sample points. In contrast, for the global variant, making many changes to a current feasible nondominated point makes it more likely to violate one of the many black-box constraints, thereby making it more difficult to generate feasible sample points.

3.4 Comparison of Performance When a Feasible Initial Point is Given

In the second set of experiments, the local and global variants of Two-Phase POSEIDON that uses RBF surrogates are compared with NSGA-II algorithms when the initial population includes the feasible design provided with the benchmark, which has a feasible objective vector of $[f_1, f_2] = [3.0028, 35]$. In this case, the POSEIDON-RBF algorithms proceed directly to Phase II. Figure 2 shows the scatter plots of the nondominated objective vectors obtained for one trial of the POSEIDON-RBF and NSGA-II algorithms for the given computational budgets and starting with the same set of initial sample points of size 300 that includes the given initial feasible point. Note that this plot differs from the one for the original POSEIDON-RBF algorithms in [23] in that the new POSEIDON-RBF implementations include the EOV criterion for selecting the nondominated point to perturb in generating the trial points. Recall that the EOV criterion is meant to improve exploration in the extremes of the approximate Pareto front.

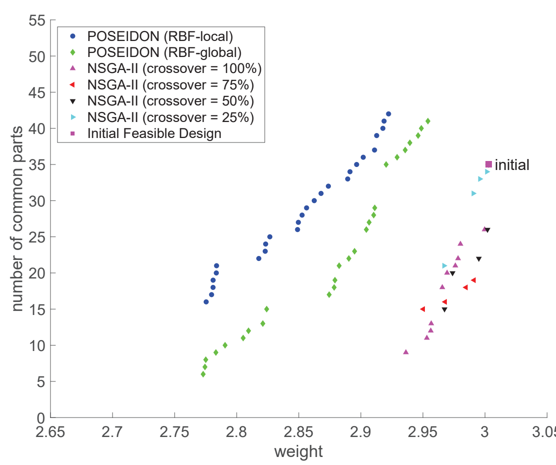

Figure 2: Nondominated objective vectors obtained by two POSEIDON algorithms that use RBF surrogates and by NSGA-II when a feasible initial point is provided. NSGA-II used a population size of 300, discrete variable encoding and crossover fractions from 25% to 100%. The computational budget is $15d = 3330$ simulations for POSEIDON and 3600 simulations (12 generations) for NSGA-II.

The plots in Figure 2 again show that the POSEIDON-RBF algorithms obtained much better sets of nondominated objective vectors than the ones obtained by the NSGA-II algorithms. As in the other set of experiments, the nondominated objective vectors obtained by each POSEIDON-RBF algorithm dominate all the nondominated objective vectors obtained by any of the NSGA-II algorithms with various crossover fractions. As before, the local variant of POSEIDON-RBF obtained a better set of nondominated objective vectors than the one obtained by the more global variant.

Note also that the POSEIDON-RBF algorithms obtained nondominated objective vectors that substantially improve on the initial feasible design for the Mazda benchmark. For example, one of the nondominated objective vectors obtained by POSEIDON (RBF-local) is $[f_1, f_2] = [2.9223, 42]$, which is a significant improvement over the initial feasible objective vector of $[f_1, f_2] = [3.0028, 35]$ given the limited computational budget of only $15d$ simulations. On the other hand, none of the NSGA-II algorithms yielded such an improvement over the initial feasible design even with a somewhat larger computational budget.

Table 3: Mean hypervolumes for 5 trials of the POSEIDON algorithms and 10 trials of the NSGA-II algorithms on the Mazda 3-Car Benchmark when a feasible initial point is provided. The number inside the parenthesis is the standard error of the mean. The objective functions (total weight and number of common parts) are normalized as in [6] and the reference point is $[1.1, 0]$.

Algorithm	Mean Hypervolume
POSEIDON (RBF-local)	0.1597 (0.0040)
POSEIDON (RBF-global)	0.1244 (0.0033)
NSGA-II (crossover fraction = 100%)	0.0572 (0.0009)
NSGA-II (crossover fraction = 75%)	0.0520 (0.0009)
NSGA-II (crossover fraction = 50%)	0.0494 (0.0010)
NSGA-II (crossover fraction = 25%)	0.0540 (0.0010)

Table 3 shows the average hypervolumes obtained by the various algorithms when a feasible initial point is provided. As before, the hypervolumes are calculated after normalizing the objective functions and using the reference point $[1.1, 0]$ as suggested in [6]. The objectives are normalized in the same way as before. Note that the average hypervolumes in this table are better than the ones in Table 2, and this is expected since we are using a feasible initial point in each trial for this set of experiments. The results are similar to the ones obtained when there are no feasible points in the initial populations. In particular, Table 3 shows that the average hypervolumes obtained by the POSEIDON-RBF algorithms are much better than the hypervolumes obtained by the NSGA-II algorithms even as the computational budget for POSEIDON (3330 simulations) is less than that for NSGA-II (3600 simulations).

3.5 Comparison with Earlier Results on the Mazda Benchmark

We now compare the results obtained in this paper with the earlier results reported for the 3-car Mazda benchmark problem from [17] and [13]. Assuming that those papers used the same initial population of 300 points that was provided with the Mazda benchmark problem, we can visually compare the quality of the nondominated objective vectors and the hypervolumes obtained by the Two-Phase POSEIDON-RBF algorithms with those from [17] and [13]. Comparing Figure 1 in this paper with Figure 2 in [17], note that many of the nondominated objective vectors obtained by POSEIDON-RBF dominate those obtained by ELM-assisted CMOEA/D in [17]. Moreover, the hypervolume obtained after 300 generations for ELM-assisted CMOEA/D in Figure 1 of [17] was less than 0.08 while the average hypervolumes obtained by POSEIDON (RBF-local) and POSEIDON (RBF-global) were 0.1367 and 0.1041 while using a computational budget of only $15d = 3330$ simulations (about 11 generations for a population size of 300).

Next, comparing Figure 1 in this paper with Figure 2 in [13], we also see that the nondominated objective vectors obtained by the POSEIDON-RBF algorithms using only $15d = 3330$ simulations are *not* dominated by those obtained in [13] with 30,000 simulations. Moreover, if we include the feasible design provided in [6] among the initial population, some of the nondominated objective vectors obtained by the POSEIDON-RBF algorithms in Figure 2 of this paper are also *not* dominated by those obtained in [13], again while expending much less simulations.

Finally, we compare Two-Phase POSEIDON with the original POSEIDON from [23] given the feasible design provided in [6] among the initial population. The main difference between Phase II of Two-Phase POSEIDON and the original POSEIDON is that the former uses the additional EOV strategy that is meant to promote exploration of the extremes of the approximate Pareto front. Since only one trial was performed in the original POSEIDON algorithms, we compare it with the first trial of the Two-Phase POSEIDON algorithms. The initial population is the same for both algorithms, so the results are directly comparable. The hypervolumes obtained for the first trial of Two-Phase POSEIDON (RBF-local) and the original POSEIDON (RBF-local) on the 3-car Mazda benchmark using $15d = 3330$ simulations are 0.1555 and 0.1524, respectively. Moreover, the hypervolumes obtained for the first trial of Two-Phase POSEIDON (RBF-global) and the original POSEIDON (RBF-global) are 0.1295 and 0.1182, respectively. In both cases, the EOV strategy resulted in an improvement in hypervolumes obtained (2.03% improvement for the local variant and 9.56% improvement for the global variant).

4 SUMMARY AND FUTURE WORK

This paper proposed the surrogate-based Two-Phase POSEIDON algorithm, which is an extension of the POSEIDON algorithm from [23] for high-dimensional and computationally expensive constrained discrete multi-objective optimization problems with ordinal decision variables and many black-box constraints. The original POSEIDON algorithm assumes that a feasible initial point is available while Two-Phase POSEIDON can be used even when a feasible point is not provided. In Phase I, the algorithm finds a feasible point by using the surrogates of the constraints and objectives. In Phase II, the algorithm finds a set of nondominated objective vectors that improve on the feasible point found in Phase I. Moreover, this paper incorporates the EOV strategy as an additional strategy for selecting a nondominated point to perturb when generating trial solutions from which the simulation point is chosen. This strategy is meant to improve the ability of the algorithm to explore and expand the extremes of the approximate Pareto front.

Local and global variants of Two-Phase POSEIDON that utilize RBF surrogates are tested on the 3-car Mazda benchmark problem involving 222 discrete decision variables, 54 black-box inequality constraints, and 2 objective functions. Two sets of experiments are performed comparing POSEIDON with an NSGA-II implementation. For the first set of experiments, none of the initial points are feasible. For the second set of experiments, the feasible point provided with the Mazda benchmark is included among the initial points. The results for both sets of experiments show that when the computational budget is only about $15d = 3330$ simulations, the POSEIDON algorithms yielded much better nondominated objective vectors than NSGA-II that uses discrete variable encoding. Moreover, both sets of experiments indicate that the local variant of POSEIDON performs better than the global variant. In addition, the local variant of POSEIDON yielded a significant improvement over the given initial feasible design for both objectives while none of the NSGA-II algorithms yielded such an improvement with the given computational budget. Finally, the Two-Phase POSEIDON algorithms found many nondominated objective vectors that either dominate or are at least *not* dominated by the nondominated objective vectors from earlier results using other algorithms on the 3-car Mazda benchmark reported in [17] and [13] while consuming much less simulations. These results suggest that Two-Phase POSEIDON is a promising approach for high-dimensional and computationally expensive constrained discrete multi-objective optimization.

The current implementation of Two-Phase POSEIDON assumes that all of the objective and constraint functions are computationally expensive. However, there are applications where only some of the objectives and constraints are black-box and expensive. It is possible to develop extensions to Two-Phase POSEIDON that take advantage of explicitly defined objectives and constraints. Moreover, other strategies for generating simulation points may be explored including by solving a multi-objective subproblem where the original objective and constraint functions are replaced by their surrogates. It would also be of interest to explore how well Two-Phase POSEIDON performs when the feasible region is highly disconnected. By setting the perturbation probability and neighborhood depth parameter to large values, Two-Phase POSEIDON is able to perform a more global search that has the ability to traverse infeasible regions of the search space, but it may be possible to improve its performance on problems with disconnected feasible regions by considering multiple neighborhoods in different regions that can be explored simultaneously through parallel processing. Finally, as mentioned earlier, the collision safety constraints in the Mazda benchmark are modeled by response surface approximations [13], so the resulting constraint functions are smooth. It would be interesting to see how Two-Phase POSEIDON will perform when using the constraint function values obtained directly from the simulations. It would also be worth exploring how the RBF model interacts with these response surface approximations and whether the performance of Two-Phase POSEIDON is sensitive to the choice of the surrogate model.

ACKNOWLEDGEMENTS

Thanks to Dr. Takehisa Kohira of the Mazda Corporation for clarifying the details of the Mazda benchmark. Thanks also to Dr. Song Lin for the Matlab NPGM code that implements NSGA-II with constraint handling and to Dr. Yi Cao for his Matlab codes that extract the set of nondominated solutions. I am grateful to Dr. Yohanes Bimo Dwianto for bringing the Mazda benchmark to my attention.

REFERENCES

[1] Taimoor Akhtar and Christine A. Shoemaker. 2016. Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection. *Journal of Global Optimization* 64 (2016), 17–32.

[2] Samineh Bagheri, Wolfgang Konen, Michael Emmerich, and Thomas Bäck. 2017. Self-adjusting parameter control for surrogate-assisted constrained optimization under limited budgets. *Applied Soft Computing* 61 (2017), 377–393.

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

[4] A. E. I. Brownlee and J. A. Wright. 2015. Constrained, mixed-integer and multi-objective optimisation of building designs by NSGA-II with fitness approximation. *Applied Soft Computing* 33 (2015), 114–126.

[5] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya. 2018. A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation* 22, 1 (2018), 129–142.

[6] The Mazda Motor Corporation. 2018. Mazda Benchmark Problem. https://ladse.eng.isas.jaxa.jp/benchmark/index.html. Accessed: 2020-04-30.

[7] A. Dushatskiy, A. M. Mendrik, T. Alderliesten, and P. A. N. Bosman. 2019. Convolutional neural network surrogate-assisted GOMEA. In *GECCO '19: Proceedings of the Genetic and Evolutionary Computation Conference*. 753–761.

[8] Paul Féliot, Julien Bect, and Emmanuel Vazquez. 2017. A Bayesian approach to constrained single- and multi-objective optimization. *Journal of Global Optimization* 67 (2017), 97–133.

[9] A. I. J. Forrester, A. Sobester, and A. J. Keane. 2008. *Engineering Design via Surrogate Modelling: A practical guide*. John Wiley & Sons.

[10] H. Fukumoto and A. Oyama. 2018. Benchmarking multiobjective evolutionary algorithms andconstraint handling techniques on a real-world car structure design optimization benchmark problem. In *GECCO '18: Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 177âĂŞ–178.

[11] A. Habib, H. K. Singh, T. Chugh, T. Ray, and K. Miettinen. 2019. A multiple surrogate assisted decomposition based evolutionary algorithm for expensive multi/many-objective optimization. *IEEE Transactions on Evolutionary Computation* 23, 6 (2019), 1000–1014.

[12] J. Knowles. 2006. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation* 10, 1 (2006), 50–66.

[13] T. Kohira, H. Kemmotsu, A. Oyama, and T. Tatsukawa. 2018. Proposal of benchmark problem based on real-world car structure design optimization. In *GECCO '18: Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 183–184.

[14] Song Lin. 2011. NPGM – A NSGA Program in Matlab, Version 1.4. http://www.mathworks.com/matlabcentral/fileexchange/31166.

[15] M. Mlakar, D. Petelin, T. Tušar, and B. Filipič. 2015. GP-DEMO: differential evolution for multiobjective optimization based on gaussian process models. *European Journal of Operational Research* 243, 2 (2015), 347–361.

[16] Juliane Müller. 2017. SOCEMO: Surrogate optimization of computationally expensive multi-objective problems. *INFORMS Journal on Computing* 29, 4 (2017), 581–596.

[17] H. Ohtsuka, M. Kaidan, T. Harada, and R. Thawonmas. 2018. Evolutionary algorithm using surrogate assisted model for simultaneous design optimization benchmark problem of multiple car structures. In *GECCO '18: Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 55–56.

[18] A. Oyama, T. Kohira, H. Kemmotsu, T. Tatsukawa, and T. Watanabe. 2017. Simultaneous structure design optimization of multiple car models using the K computer. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)* (Honolulu, HI). 1–4.

[19] P. S. Palar, Y. B. Dwianto, L. R. Zuhal, and T. Tsuchiya. 2016. Framework for Robust Optimization Combining Surrogate Model, Memetic Algorithm, and Uncertainty Quantification. In *Advances in Swarm Intelligence*. Springer International Publishing, Cham, Switzerland, 48–55.

[20] M. J. D. Powell. 1992. The theory of radial basis function approximation in 1990. In *Advances in Numerical Analysis, Volume 2: Wavelets, Subdivision Algorithms and Radial Basis Functions*, W. Light (Ed.). Oxford University Press, Oxford, U.K., 105–210.

[21] Rommel G. Regis. 2014. Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. *Engineering Optimization* 46, 2 (2014), 218–243.

[22] Rommel G. Regis. 2016. Multi-objective constrained black-box optimization using radial basis function surrogates. *Journal of Computational Science* 16 (2016), 140–âĂŞ155.

[23] Rommel G. Regis. 2020. High-Dimensional Constrained Discrete Multi-objective Optimization Using Surrogates. In *Machine Learning, Optimization, and Data Science. LOD 2020. Lecture Notes in Computer Science*, G. Nicosia et al. (Ed.). Vol. 12566. Springer Nature Switzerland AG, Cham, Switzerland, 203–214.

[24] Prashant Singh, Ivo Couckuyt, Francesco Ferranti, and Tom Dhaene. 2014. A constrained multi-objective surrogate-based optimization algorithm. In *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 3080–3087.

[25] K. Yang, L. Li, A. Deutz, T. Back, and M. Emmerich. 2016. Preference-based multiobjective optimization using truncated expected hypervolume improvement. In *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)* (Changsha, China). 276–281.