

Efficient Parametric Optimization for Expensive Single Objective Problems

Jonathan M. Weaver-Rosen

Design Systems Laboratory,
Mechanical Engineering,
Texas A&M University,
College Station, TX 77843
e-mail: jonr23@tamu.edu

Richard J. Malak, Jr.

Design Systems Laboratory,
Mechanical Engineering,
Texas A&M University,
College Station, TX 77843
e-mail: rmalak@tamu.edu

Parametric optimization solves optimization problems as a function of uncontrollable or unknown parameters. Such an approach allows an engineer to gather more information than traditional optimization procedures during design. Existing methods for parametric optimization of computationally or monetarily expensive functions can be too time-consuming or impractical to solve. Therefore, new methods for the parametric optimization of expensive functions need to be explored. This work proposes a novel algorithm that leverages the advantages of two existing optimization algorithms. This new algorithm is called the efficient parametric optimization (EPO) algorithm. EPO enables adaptive sampling of a high-fidelity design space using an inexpensive low-fidelity response surface model. Such an approach largely reduces the required number of expensive high-fidelity computations. The proposed method is benchmarked using analytic test problems and used to evaluate a case study requiring finite element analysis. Results show that EPO performs as well as or better than the existing alternative, Predictive Parameterized Pareto Genetic Algorithm (P3GA), for these problems given an allowable number of function evaluations.

[DOI: 10.1115/1.4049519]

Keywords: design automation, design optimization, metamodeling

1 Introduction

Parametric optimization, also known as multi-parametric programming, is the process of solving an optimization problem as a function of one or more variables, termed parameters, over a given range [1–7]. In this context, parameters are interpreted as variables on which the optimal solution depends but that are not controlled by the optimization process. For example, an engineer may wish to perform an optimization study but believes several critical assumptions about the problem (e.g., those affecting boundary conditions, etc.) are likely to change during the course of the project. The engineer can proceed in several ways including: optimizing to current values, optimizing to worst-case values, and performing optimization under uncertainty (e.g., robust design optimization [8,9] or reliability-based design optimization [10]) based on beliefs about the likely values of the parameters. In contrast to these options, parametric optimization enables the engineer to find the optimal solution as a function of parameter values, and later, once true parameter values are determined, select the relevant solution from the parametric solution set.

Many engineering and systems design scenarios can benefit from parametric optimization. For example, an engineer may wish to perform an optimization study but believes that critical design requirements are likely to change. Such requirements could be represented using parameters in a parametric optimization framework [6]. A related example is when the engineer performing the optimization lacks security clearance to know the precise values of particular variables. In this case, parametric optimization can provide the optimal solution as a function of those variables and this solution can be used by individuals with appropriate security clearance. Another use of parametric optimization is to find optimal solutions for an adaptive system as a function of environmental or system conditions that change during system operation. Examples of this are flight conditions for a morphing wing [7,11] and ambient conditions for chemical process control [5,12].

Parametric optimization also has applications outside of engineering including to determine market equilibria as a function of exogenous parameters such as the cost of goods or labor [13].

Several specialized algorithms for parametric optimization exist [3,12,14–19]. These include techniques for restricted problem classes (e.g., linear parametric programming, quadratic parametric programming) as well as general nonlinear black-box models. For the special case of a single parameter, it may be practical to perform a parameter sweep with a unique optimization run at each sample point. However, this scales poorly with the number of parameters and has been shown to be inferior to specialized parametric algorithms [20].

One challenge for parametric optimization techniques is the application of advanced optimization strategies to deal with computationally intensive analyses (e.g., finite element analyses (FEA), computational fluid dynamics (CFD), or physical experiments). For non-parametric optimization, a common strategy is to combine response surface models with an adaptive sampling technique. These approaches operate with the response surface and execute expensive analyses only when the result is likely to be informative. The efficient global optimization (EGO) algorithm [21], Gutmann’s radial basis function (GRBF) method [22], and sequential approximation optimizations (SAO) [23] are examples of such techniques for single-objective optimization. The Pareto EGO (ParEGO) algorithm [24], Keane’s statistically based improvement (KSI) method [25], meta-model assisted evolution strategies (MAES) [26], and the predictive entropy search (PES) [27] extend these works to solve multi-objective problems. Furthermore, the parallelized multi-objective EGO (ParMOEGO) algorithm [28] seeks to improve efficiency of these methods even more by parallelizing high-fidelity computations to reduce “wall-clock” time. Another extension of these efficient algorithms, the Multi-Objective Bayesian Optimization (MOBO) algorithm [29] allows for the efficient optimization of problems with multiple information sources. However, the special role of parameter variables precludes the direct use of these techniques for parametric optimization.

In this paper, we extend the well-known EGO algorithm to parametric optimization problems. The new technique formulates a parametric version of the expected improvement function and

Contributed by the Design Automation Committee of ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received May 28, 2020; final manuscript received December 21, 2020; published online January 27, 2021. Assoc. Editor: Nam H. Kim.

Parametric	EPO (this work)	
	EGO ^[21] GRBF ^[22] SAO ^[23]	ParEGO ^[24] KSI ^[25] MAES ^[26] MOBO ^[29] ParMOEGO ^[28] PES ^[27]
Non-Parametric	Single Objective	Multiple Objectives

Fig. 1 A list of optimization methods that rely on adaptive sampling of a response surface sorted by problem type

uses an existing parametric optimization technique, the Predictive Parameterized Pareto Genetic Algorithm (P3GA) [18], to perform parametric expected improvement maximization. We also present a parametric version of the hypervolume indicator (HVI) metric as a means for comparing solutions and evaluating optimizer progress in parametric optimization. The new algorithm, Efficient Parametric Optimization (EPO), is benchmarked on two test problems with known solutions prescribing a fixed number of high-fidelity function evaluations and demonstrated on the parametric optimization of a structural stiffened plate requiring FEA. Figure 1 relates EPO to other efficient metamodel-based optimization algorithms by the type of problems they can solve.

2 Background

2.1 Parametric Optimization. Although most optimization methods only identify the optimal solution for a single combination of parameters, parametric optimization identifies the optimal solution as a function of the parameters. Once this is done, the optimal objective value and corresponding optimization variables can be easily solved using the parameter values. The single-objective parametric optimization problem can be formulated as follows [20]:

$$J^*(\theta) = \min_{\mathbf{x}} J(\mathbf{x}, \theta) \forall \theta \in [\theta_{lb}, \theta_{ub}]$$

subject to:

$$\begin{aligned} \mathbf{g}(\mathbf{x}, \theta) &\leq \mathbf{0} \\ \mathbf{h}(\mathbf{x}, \theta) &= \mathbf{0} \\ \mathbf{x}_{lb} &\leq \mathbf{x} \leq \mathbf{x}_{ub} \\ \mathbf{x} &\in \mathbb{R}^k \\ \theta &\in \mathbb{R}^m \end{aligned} \quad (1)$$

where J is the objective function to be minimized, \mathbf{x} is the vector of optimization variables bounded by \mathbf{x}_{lb} and \mathbf{x}_{ub} , θ is the vector of parameters bounded by θ_{lb} and θ_{ub} , and $\mathbf{g}(\mathbf{x}, \theta)$ and $\mathbf{h}(\mathbf{x}, \theta)$ represent inequality and equality constraints, respectively, as functions of both optimization variables and parameters. This problem formulation applies to cases where parameters are independent variables (e.g., operational conditions or currently unknown variables outside of the system). Existing methods solve the problem formulation in Eq. (1) for linear parametric problems [3,14], quadratic parametric problems [15,16], parametric problems with convex solutions [12,17], and nonlinear black-box parametric problems [18,19].

However, parameters can be dependent and/or independent variables. The examples presented in this work focus on dependent parameters only. In this case, the parametric optimization problem

formulation can be rewritten as:

$$J^*(\theta) = \min_{\mathbf{x}} J(\mathbf{x}, \theta(\mathbf{x})) \forall \theta \in [\theta_{lb}, \theta_{ub}]$$

subject to:

$$\begin{aligned} \mathbf{g}(\mathbf{x}, \theta(\mathbf{x})) &\leq \mathbf{0} \\ \mathbf{h}(\mathbf{x}, \theta(\mathbf{x})) &= \mathbf{0} \\ \mathbf{x}_{lb} &\leq \mathbf{x} \leq \mathbf{x}_{ub} \\ \mathbf{x} &\in \mathbb{R}^k \\ \theta &\in \mathbb{R}^m. \end{aligned} \quad (2)$$

The generalized formulation given in Eq. (2) is used throughout the rest of the paper. The solution to a parametric problem is typically given as $J^*(\theta)$ or $\mathbf{x}^*(\theta)$ where $J^*(\theta) = J(\mathbf{x}^*(\theta), \theta)$. It has been shown that $\mathbf{x}^*(\theta)$ may not be unique or even defined when mathematical singularities are present [30–32]. However, many engineering case studies do not suffer from this limitation. Therefore, the authors assume that $\mathbf{x}^*(\theta)$ both exists and is unique.

One algorithm for parametric optimization is the Predictive Parameterized Pareto Genetic Algorithm, or P3GA. Inspired by NSGA-II [33], P3GA is an algorithm for parametric optimization of multi-objective black-box problems [18]. The novel aspect of P3GA is the use of a machine learning technique (specifically a support vector domain description, or SVDD) to predict parametric Pareto dominance. Parametric Pareto dominance is an extension of classical Pareto dominance that considers both objectives and parameters. Simply, parametric Pareto dominance is the application of classic Pareto dominance to configurations with identical parameter values [34]. With the ability to predict parametric Pareto dominance, P3GA can evaluate a problem as a genetic algorithm ranking and sorting data with multiple objectives and multiple parameters. To the authors' knowledge, P3GA is the only parametric optimization algorithm that handles the case when parameters are dependent variables (e.g., system attributes). In this paper, P3GA is used to benchmark the novel EPO algorithm and maximize an expected improvement metric within EPO (see Sec. 3).

2.2 Efficient Global Optimization. For optimization problems with expensive objective functions, engineers are limited in how many times expensive functions can be evaluated. Response surface modeling of these functions can allow an engineer to search more of the design space while also predicting the uncertainty of the response surface.

EGO is a methodology that iteratively updates and uses response surface models to efficiently search the design space in order to reduce the number of expensive function evaluations required to reach the global optimum [21]. To accomplish this, the expensive (or high-fidelity) function is evaluated a number of times via a design of experiments (DOE) to train an initial response surface. Then, the less expensive response surface model is searched via branch and bound optimization to find the design with the highest expected improvement based on the response surface prediction of objective value and its uncertainty. This expected improvement formulation is [21]:

$$\begin{aligned} E[I(\mathbf{x})] &= (J_{\min} - \hat{J}(\mathbf{x}))\Phi\left(\frac{J_{\min} - \hat{J}(\mathbf{x})}{s(\mathbf{x})}\right) \\ &+ s(\mathbf{x})\phi\left(\frac{J_{\min} - \hat{J}(\mathbf{x})}{s(\mathbf{x})}\right) \end{aligned} \quad (3)$$

where J_{\min} is the minimum objective value found so far (from the high-fidelity samples) and \hat{J} and s are the response surface model predictions of the objective value and standard error, respectively, for given optimization variables \mathbf{x} . Additionally, Φ represents the standard normal cumulative distribution function and ϕ is the standard normal probability distribution function.

Finally, the design with the greatest expected improvement is evaluated by the expensive model, and the response surface model is subsequently improved. In this way, the entire high-fidelity design space is sampled in an adaptive way using the lower-fidelity response surface model to predict beneficial and/or uncertain designs. While this method requires more steps and multiple optimizations of a response surface model, the number of times that the expensive model is sampled can be drastically reduced when compared with a direct optimization of the expensive model. See Algorithm 1 for a more compact representation of this algorithm.

Algorithm 1 EGO [21]

```

1:  $J(\mathbf{x}) \leftarrow$  Expensive model
2:  $\hat{J}(\mathbf{x}) \leftarrow$  Initial response surface
3:  $N \leftarrow$  Number of expensive evaluations performed
4: input:  $N_{\max} \leftarrow$  Maximum allowable expensive evaluations
5:  $J_{\min} \leftarrow$  Minimum objective value from all expensive samples
6: while  $N < N_{\max}$  do
7:    $\mathbf{x}_{new} \leftarrow \operatorname{argmax}_{\mathbf{x}} E[I(\mathbf{x})]$ 
8:   if  $E[I(\mathbf{x}_{new})] < tol$  then
9:     break while
10:  else
11:     $J_{new} \leftarrow J(\mathbf{x}_{new})$ 
12:     $\hat{J}(\mathbf{x}) \leftarrow$  Update response surface with  $\mathbf{x}_{new}$  and  $J_{new}$ 
13:     $N \leftarrow N + 1$ 
14:    if  $J_{new} < J_{\min}$  then
15:       $J_{\min} \leftarrow J_{new}$ 
16:    end if
17:  end if
18: end while
19: return  $J_{\min}$ 

```

3 Efficient Parametric Optimization

The novel method explored herein is an extension of EGO to be applicable to parametric optimization problems. Although there are newer and more specialized techniques for adaptive sampling in an optimization context, such techniques rely on the same underlying principles of exploitation and exploration accomplished by the expected improvement function of EGO [35]. Therefore, this work showcases the key novelty of including parameters in efficient optimization by extending the seminal work, EGO, which is widely known in the community. In the same manner as EGO, EPO iteratively updates and uses response surface models to efficiently search the design space to find the parametrically nondominated frontier. The basic procedure is to initialize the response surface model via DOE, parametrically maximize the expected improvement function to sample the expensive model, update the response surface, and repeat until stopping criteria are met. While this is very similar to the procedure of EGO, these steps are performed differently to take parameters into account.

First, the presence of parameters introduces the potential for increasing numbers of necessary response surface models. In EGO, the objective and/or constraint functions could be expensive. In EPO, any combination of objective, constraint, and parameter functions could be expensive. One response surface is trained for each expensive function predicting the objective, constraint, or parameter value given a set of optimization variables.

Next, the expected improvement function (Eq. (3)) must be redefined such that it is dependent on the parameters. In general, the response surface model predictions \hat{J} and s become functions of both the optimization variables and parameters. However, since this paper deals with only dependent parameters, these notations do not change. J_{\min} , however, *must* become a function of the parameters since each location in parameter space ought to have a different (but not entirely independent) baseline. Therefore, the

parametric expected improvement function is given by

$$E[I(\mathbf{x}, \boldsymbol{\theta})] = (J_{\min}(\boldsymbol{\theta}) - \hat{J}(\mathbf{x}))\Phi\left(\frac{J_{\min}(\boldsymbol{\theta}) - \hat{J}(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x})\phi\left(\frac{J_{\min}(\boldsymbol{\theta}) - \hat{J}(\mathbf{x})}{s(\mathbf{x})}\right) \quad (4)$$

if and only if parameters are fully dependent on the optimization variables. In the case when parameters functions are also expensive, $\boldsymbol{\theta}$ is replaced by its response surface model prediction $\hat{\boldsymbol{\theta}}$ as a function of the optimization variables.

Unlike the scalar J_{\min} in EGO, the function $J_{\min}(\boldsymbol{\theta})$ is not trivial to compute. For the expected improvement formulation to be valid, a J_{\min} value is necessary for all parameter vectors. Therefore, EPO utilizes P3GA to find $J_{\min}(\boldsymbol{\theta})$ by parametrically optimizing the low-fidelity response surface model prediction only. Then, EPO uses this $J_{\min}(\boldsymbol{\theta})$ to parametrically optimize for maximum expected improvement, Eq. (4), via P3GA to determine new designs with which to sample the high-fidelity model. In this way, EPO requires *two* runs of parametric optimization (specifically P3GA in this work) on the response surface models in every loop without any expensive function calls.

While doubling the number of analytic optimizations performed per loop (compared with EGO) is not ideal, this disadvantage is balanced and possibly outweighed by the advantage of finding multiple designs with high expected improvement across the parameter space per loop. Since P3GA results in a parametrically nondominated set of designs in terms of maximum expected improvement, multiple different designs can be selected to sample the high-fidelity model and retrain the response surface model before repeating. This is beneficial for parametric problems because the engineer is interested in the entire parametric solution, while the non-parametric EGO only aims to find the single best design. The benefit of choosing multiple designs per adaptive sampling step is better explained by a comparison to the procedure of ParEGO [24] where the solution is a Pareto frontier. In ParEGO, a specific trade-off vector (between multiple objectives) is chosen each time the response surface is optimized for expected improvement. In the ParEGO framework, choosing a new trade-off vector would require an additional optimization. In contrast, multiple parameter locations can be chosen via the parametric optimization of EPO *without* requiring additional optimizations. The number of designs chosen to sample the high-fidelity model is essentially a meta-parameter of the algorithm. It is not necessary to choose only one (as in EGO), but it should also not be excessive such that the algorithm rarely updates the response surface. Tuning this part of the algorithm is ongoing work, but the cases studied in the next section consider multiple values of this meta-parameter.

The procedure of EPO is thus: (i) perform DOE to train the initial response surface, (ii) parametrically optimize the response surface prediction to find $J_{\min}(\boldsymbol{\theta})$, (iii) parametrically maximize the expected improvement (via Eq. (4)), (iv) select n points to evaluate with the high-fidelity model, (v) update the response surface model, and (vi) repeat steps ii-v until the maximum number of high-fidelity samples have been taken or the maximum expected improvement is below some threshold. This procedure is displayed in Algorithm 2 for the case of dependent parameters.

4 Benchmarking

Before analyzing an engineering case study, analytic test functions with known solutions are evaluated to determine the viability of the new algorithm. The test functions used to benchmark EPO are parametric variants of the DTLZ test suite of problems [36]. The two scalable test functions chosen from Galvan et al. [20] have been shown to be non-trivial to solve and are used to assess the performance of other parametric optimization algorithms. Furthermore, these problems are chosen because the mathematic formulation scales to any combination of objectives and parameters. For both

test problems, the hypervolume indicator (HVI) is used to compare the results to the true solution by the following:

$$\delta = HVI_{true} - HVI_{calc} \quad (5)$$

where HVI_{true} is the hypervolume indicator of the true solution, HVI_{calc} is the hypervolume indicator calculated from the data, and δ is simply the difference between the true and calculated HVI values. The method of calculating the HVI for parametric problems is discussed in the next subsection.

Algorithm 2 EPO (dependent parameters)

```

1:  $F(\mathbf{x}) = [J(\mathbf{x}), \boldsymbol{\theta}(\mathbf{x})]$  ← Expensive model
2:  $\hat{F}(\mathbf{x}) = [\hat{J}(\mathbf{x}), \hat{\boldsymbol{\theta}}(\mathbf{x})]$  ← Initial response surface
3:  $N \leftarrow$  Number of expensive evaluations performed
4: input:  $N_{max} \leftarrow$  Maximum allowable expensive evaluations
5: while  $N < N_{max}$  do
6:    $J_{min}(\hat{\boldsymbol{\theta}}) \leftarrow \min_{\mathbf{x}} \hat{J}(\mathbf{x}) \forall \hat{\boldsymbol{\theta}}$ 
7:    $\mathbf{x}^*(\hat{\boldsymbol{\theta}}) \leftarrow \operatorname{argmax}_{\mathbf{x}} E[J(\mathbf{x}, \hat{\boldsymbol{\theta}})]$ 
8:   if  $\text{all}(E[J(\mathbf{x}^*(\hat{\boldsymbol{\theta}}), \hat{\boldsymbol{\theta}})] < tol)$  then
9:     break while
10:  else
11:    Define  $n$   $\mathbf{x}_{new}$  tuples such that:  $\mathbf{x}_{new} \subseteq \mathbf{x}^*(\hat{\boldsymbol{\theta}})$ 
12:     $J_{new}, \boldsymbol{\theta}_{new} \leftarrow F(\mathbf{x}_{new})$ 
13:     $\hat{F}(\mathbf{x}) \leftarrow$  Update response surface via  $\mathbf{x}_{new}, \boldsymbol{\theta}_{new}$ , and  $J_{new}$ 
14:     $N \leftarrow N + n$ 
15:  end if
16: end while
17: return all high-fidelity samples

```

4.1 Parametric Hypervolume Indicator. The hypervolume indicator is a popular measure of evolutionary algorithm performance in multi-objective optimization. The hypervolume indicator (HVI) essentially gives an indication of how much space is “covered” (or dominated) by the data [37]. In the context of multi-objective optimization, this indicator is used to compare solution sets by determining which set dominates more of the reference space or to check for convergence to a single HVI value. In general, solutions with greater HVI values are closer to the true solution and can dominate solutions with lesser HVI values. The HVI can be directly calculated given a data set by summing the mutually exclusive sizes of the hypercube dominated by each point with respect to some reference point. See Refs. [37–39] for more information on calculating multi-objective hypervolume indicators. The benefit of calculating the HVI metric in evolutionary optimization is two-fold. First, an algorithm or engineer can monitor this value to detect convergence (or the lack thereof). This allows an optimization routine to be shortened or extended as needed. The second benefit is realized when comparing various solutions. Knowing that the true solution maximizes the HVI metric, an engineer should prefer solutions yielding a greater HVI value.

While this indicator is widely used in multi-objective optimization, it has not yet been used in parametric optimization. It is important to note that the hypervolume indicator is conceptually the same for both parametric and non-parametric data. In both cases, the hypervolume indicator measures the size of the dominated space. Despite this, the HVI cannot be calculated for parametric data in the same way that it is calculated for multi-objective data. This is because the space dominated by parametric data is not as simple as overlapping hypercubes. Instead, one must rely on the space that is believed to be dominated based on the data. P3GA uses a machine learning technique (specifically a support vector domain description, or SVDD) to predict parameterized Pareto dominance (PPD) [18]. By sufficiently sampling the reference space (e.g., a Monte Carlo technique as in Bader et al. [40]) and predicting whether the points are dominated or not via PPD, a parametric HVI value can be

estimated as the fraction of the reference space dominated by the data. This HVI estimation is bounded from 0 to 1 where a value of 0 means that none of the space is dominated and a value of 1 means that the entire space is dominated. In this paper, a Latin hypercube sampling of 4000 points (chosen through a convergence study) is used to sample the attribute space (objective and parameter space) to predict the HVI of parametric data. Note that these 4000 points are *not* evaluated by the expensive or cheap functions. These points are generated directly in the attribute space for the sole purpose of estimating the HVI value.

4.2 Test Problem #1. The first example analyzed is test problem C from Galvan et al. [20] with one objective and five optimization variables. This scalable problem is evaluated for one and two parameters. This test function is known to have a discontinuous solution, a non-convex solution, and a biased solution density. The specific problem formulation where p denotes the number of parameters is follows:

$$J^*(\boldsymbol{\theta}) = \min_{\mathbf{x}} \left(1 + \frac{9}{4} \sum_{i=p+1}^5 x_i \right) (3(1+p) - \sum_{i=1}^p (2\theta_i(\mathbf{x})) + \sin(3\pi\theta_i(\mathbf{x}))) \quad (6)$$

subject to:

$$0 \leq x_i \leq 1 \quad \forall i \in [1, \dots, 5]$$

where:

$$\theta_i(\mathbf{x}) = \frac{1}{2} (1 + \sin(20x_i)) \quad \forall i \in [1, \dots, p].$$

4.2.1 One Parameter. To demonstrate the new framework, EPO is performed 20 times (randomly restarting). For the single parameter variant, an initial DOE of 50 designs is used and the response surface model is updated with two or five designs per sampling step (i.e., $n=2$ or $n=5$; cf. Algorithm 2). The maximum number of high-fidelity function evaluations is set to 150 (i.e., $N_{max}=150$; cf. Algorithm 2). Then, P3GA is performed 20 times (also randomly restarting) using the high-fidelity function directly prescribing 10 generations of 15 population members (to match 150 high-fidelity samples per run). To visualize the results, all parametrically nondominated points from all runs are displayed graphically in Fig. 2 along with the known true solution. To quantitatively compare the results, the hypervolume indicator is calculated for each run from all methods and its difference from the true solution is displayed in Table 1 where the HVI of the true solution is calculated to be 0.8803.

Theoretically, a value of δ equal to zero would mean that the data perfectly captures the true solution, and negative values of δ are not possible. However, the HVI calculated here relies on a prediction of the dominated space and is thus subject to some inherent error. Note that this error is relatively small and solution sets with higher HVI values (or smaller δ) are closer to the true solution. Performing a statistical t -test on the final HVI values reveals that the results of EPO and P3GA are statistically different (99% significance level), but the results of the two EPO runs are not statistically different.

The HVI calculated throughout each run is plotted against the number of high-fidelity function evaluations in Fig. 3. The point denotes the average value, the whiskers denote the maximum and minimum values, and the boxes denote the interquartile range (middle 50%) for all 20 runs. EPO approaches the true solution in fewer function evaluations compared to P3GA alone. EPO appears to perform better when fewer high-fidelity samples are taken each iteration (i.e., small n value), but this increased performance is coupled with increased wall-clock time. However, one should note that when the high-fidelity function becomes increasingly time-consuming, the increased time of the EPO algorithm becomes negligible.

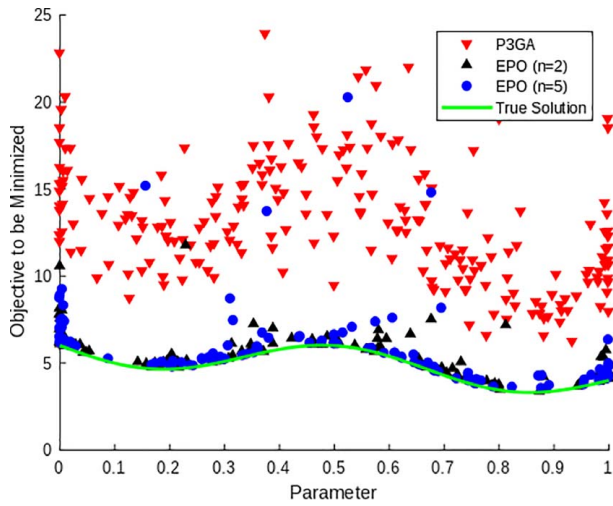


Fig. 2 Parametrically nondominated solutions found from all 20 runs for each method against the true solution for test problem #1 with one parameter

Table 1 Final mean, minimum, and maximum difference (Eq. (5)) from true HVI for 20 random restarts of each method on test problem #1 with one parameter (Eq. (6))

Method	Mean δ	Min δ	Max δ
P3GA	0.182	0.117	0.282
EPO ($n=2$)	0.010	0.000	0.042
EPO ($n=5$)	0.022	-0.001	0.145

Note: Theoretically, negative δ values are not possible, but there is a small error inherent in the calculated HVI. Each method evaluates the high-fidelity model only 150 times.

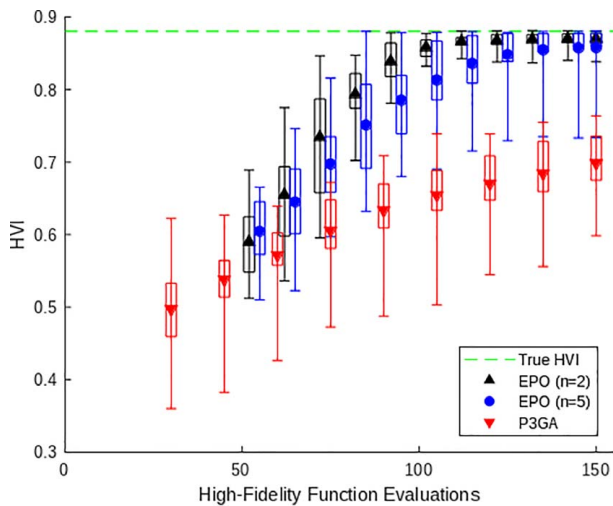


Fig. 3 Average HVI versus the number of high-fidelity function evaluations for test problem #1 with one parameter

4.2.2 Two Parameters. The same procedure is repeated scaling test problem #1 up to two parameters. The only differences are that: the initial response surface is trained by a DOE of 100 designs, the number of high-fidelity samples evaluated per sampling step is increased to 10 or 15, and the maximum number of high-fidelity samples is increased to 600 (i.e., $n=10$ or $n=15$ and $N_{\max}=600$). To account for the increase in allowable function evaluations,

Table 2 Final mean, minimum, and maximum difference (Eq. (5)) from true HVI for 20 random restarts of each method on test problem #1 with two parameters (Eq. (6))

Method	Mean δ	Min δ	Max δ
P3GA	0.116	0.037	0.254
EPO ($n=10$)	0.056	0.015	0.105
EPO ($n=15$)	0.067	0.018	0.168

Note: Each method evaluates the high-fidelity model 600 times.

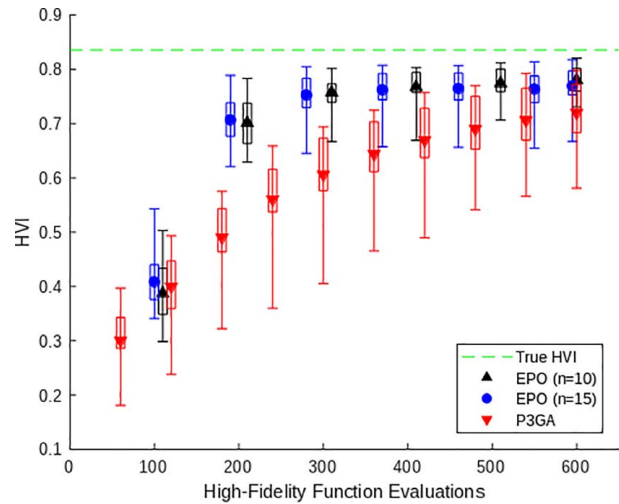


Fig. 4 Average HVI versus the number of high-fidelity function evaluations for test problem #1 with two parameters

the benchmarking P3GA directly evaluates 20 generations of 30 population members (for a total of 600 high-fidelity evaluations). The hypervolume indicator is calculated for each run from all methods and its difference from the true solution is displayed in Table 2 where the HVI of the true solution is calculated to be 0.8356. Furthermore, the hypervolume indicator is plotted against the number of high-fidelity function evaluations in Fig. 4. For both the one and two parameter variants of this benchmarking problem, the new EPO approaches the true solution in fewer function evaluations than the existing P3GA.

4.3 Test Problem #2. The second example analyzed is test problem B from Galvan et al. [20] with one objective and five optimization variables. This scalable problem is also evaluated for one and two parameters. This multimodal test function is known to have a biased solution density and a variable solution density. The specific problem formulation where p denotes the number of parameters is follows:

$$J^*(\theta) = \min_{\mathbf{x}} \left(2 + p + \sum_{i=p+1}^5 \cos(i\pi x_i^{0.02}) \right) (1 + p + \sum_{i=1}^p (\theta_i(\mathbf{x})^2 + \cos(\pi(1 - \theta_i(\mathbf{x})))^2)) \quad (7)$$

subject to:

$$0 \leq x_i \leq 1 \forall i \in [1, \dots, 5]$$

where:

$$\theta_i(\mathbf{x}) = \frac{1}{2} (1 + \sin(2\pi \cos(5\pi x_i))) \forall i \in [1, \dots, p].$$

In the same way as the previous test problem, 20 random restarts are performed for each method. The results of the single parameter

Table 3 Final mean, minimum, and maximum difference (Eq. (5)) from true HVI for 20 random restarts of each method on test problem #2 with one parameter (Eq. (7))

Method	Mean δ	Min δ	Max δ
P3GA	0.203	0.126	0.275
EPO ($n=2$)	0.194	0.143	0.237
EPO ($n=5$)	0.199	0.144	0.230

Note: Each method evaluates the high-fidelity model only 150 times.

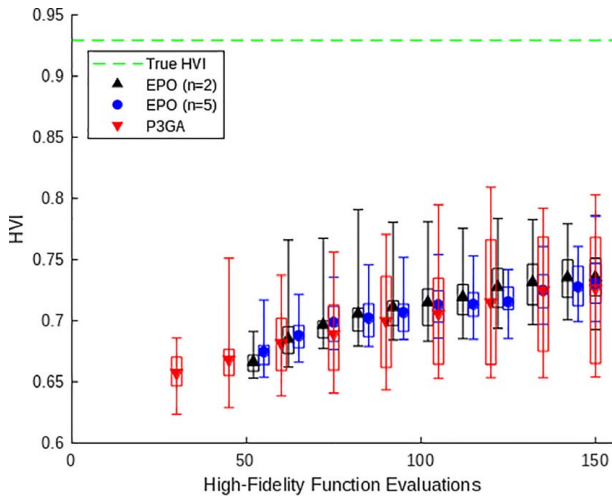


Fig. 5 Average HVI versus the number of high-fidelity function evaluations for test problem #2 with one parameter

formulation of test problem #2 can be found in Table 3 and Fig. 5 (where $HVI_{true} = 0.9292$), and the results for the two-parameter formulation can be found in Table 4 and Fig. 6 (where $HVI_{true} = 0.8833$).

In both cases of test problem #2, the best result of the existing method (P3GA) outperforms the best result of the new method (EPO). However, the worst result of P3GA is also worse than the worst result of EPO. For this problem, EPO results in less solution variance when compared to the existing P3GA. This benchmarking problem does not show EPO outperforming the existing P3GA, but rather the performances of EPO and P3GA are very similar. Through an inspection of how EPO chooses new designs to sample for this problem, we hypothesize that dependent parameters defined by highly multi-modal functions are not sampled efficiently with EPO. The algorithm was not able to quickly find the nondominated designs as seen in the previous benchmarking problem because it took many samples to develop a trustworthy response surface. This is not surprising for noisy problems, but it is promising that the novel EPO does not perform worse than P3GA despite this disadvantage. Future work will include a more detailed study to determine what properties this test problem has that cause EPO to not be more efficient than P3GA.

Table 4 Final mean, minimum, and maximum difference (Eq. (5)) from true HVI for 20 random restarts of each method on test problem #2 with two parameters (Eq. (7))

Method	Mean δ	Min δ	Max δ
P3GA	0.085	0.021	0.221
EPO ($n=10$)	0.145	0.107	0.164
EPO ($n=15$)	0.128	0.074	0.171

Note: Each method evaluates the high-fidelity model only 600 times.

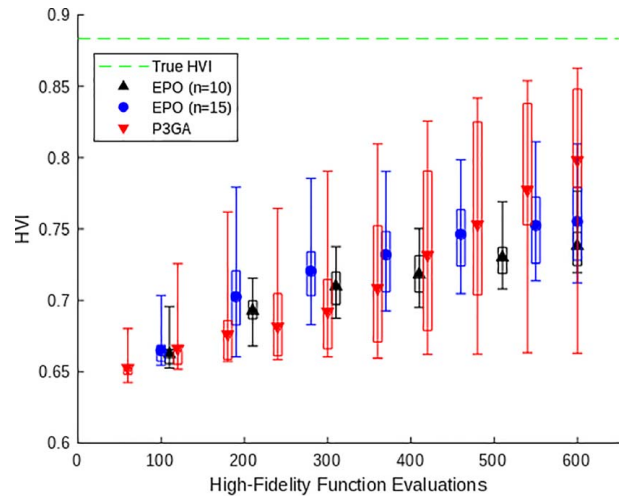


Fig. 6 Average HVI versus the number of high-fidelity function evaluations for test problem #2 with two parameters

5 Finite Element Case Study

The engineering case study presented in this section is the design of a panel consisting of a thin plate stiffened by L-beams under compression where the primary failure mode is buckling. As such, an eigenvalue buckling analysis is required to calculate the load magnitude that causes the structure to buckle. The design problem is to find the stiffened panel of minimum mass for various load magnitudes prescribing a safety factor of 1.5. In this way, the objective is to minimize mass, and the unknown parameter is the allowable loading magnitude. In a practical sense, an engineer may believe that the allowable loading requirement may change over the course of the project. Another possibility is that the engineer is interested in developing a product platform for which multiple scaled designs are required, each for different loading magnitudes which have not yet been decided. In either case, parametric optimization is well suited to solve this problem.

The stiffened panel is shown in Fig. 7. The outer dimensions of the plate are fixed at 60 in. (1.52 m) wide and 90 in. (2.29 m) tall. Additionally, the entire assembly is assumed to be aluminum with

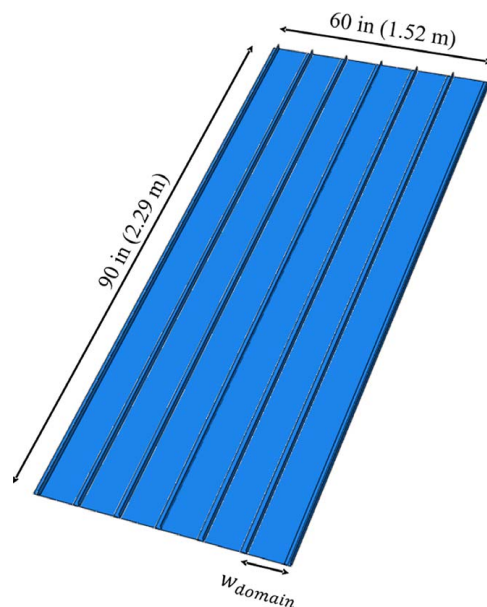


Fig. 7 Stiffened panel assembly

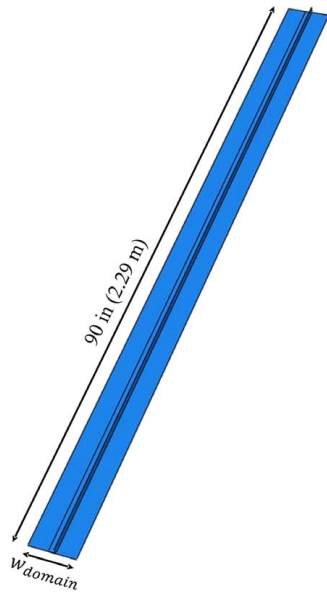


Fig. 8 Single section of the full panel analyzed by ABAQUS

the following material properties: $\sigma_y = 40$ ksi (275.79 MPa), $\nu = 0.3$, $\rho = 0.0975$ lbs/in³ (2698.79 kg/m³), and $E = 10.4E3$ ksi (71.71 GPa). The relevant optimization variables are as follows: the plate thickness, the stiffener thickness, the number of stiffeners per 60 in. (1.52 m) panel (n_{stiff}), and the stiffener cross-sectional dimensions. Note that one of these variables (n_{stiff}) is discrete while the other four are continuous.

For simplicity in the FEA model, only one segment of the panel is analyzed as shown in Fig. 8 where the width of a single panel segment, w_{domain} , is defined by the full panel width and number of stiffeners (assuming that all stiffeners are equally spaced). This is simply

$$w_{domain} = \frac{60}{n_{stiff}} \quad (8)$$

where 60 represents the total panel width in inches. To improve the accuracy of this reduced order model, two boundary conditions on the long edges are considered separately, and the worst case scenario is used to determine the allowable load. The first boundary condition is symmetry along the two long edges, and the second

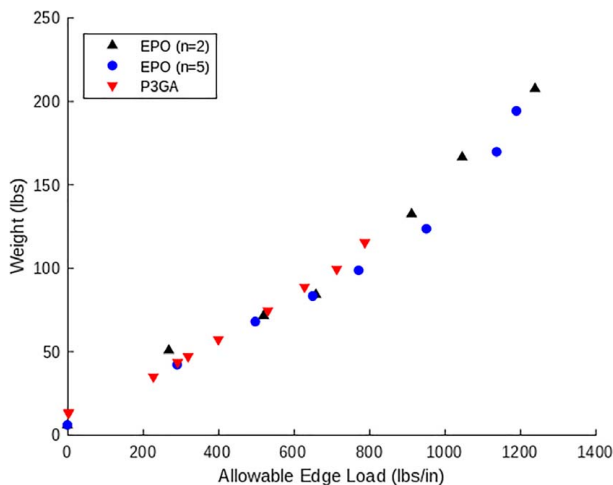


Fig. 9 Performance of parametrically nondominated solutions for each method for the finite element analysis case study

boundary condition is anti-symmetry along the two long edges. Therefore, two FEA evaluations are required to evaluate each design. For both cases, the loading is evenly distributed along one of the short edges while the other short edge is fully encastered. This case study demonstrates the ability of EPO to learn and predict the underlying physics of a black-box model to identify and exploit trends that could be unknown to an engineer. Since the demonstration herein is focused on solving general black-box models, the FEA model is not simplified further. In practice, however, an engineer should simplify their model as much as possible without inhibiting their trust in the model (e.g., using theoretical relationships [41], CAD feature suppression [42]) as doing so can reduce the expense incurred for every expensive function call.

The parametric problem is then:

$$W^*(\theta) = \min_{\mathbf{x}} \rho V(\mathbf{x})$$

where

$$\begin{aligned} \theta(\mathbf{x}) &= \min[\theta_1(\mathbf{x}), \theta_2(\mathbf{x})] \\ \theta_1(\mathbf{x}) &= \frac{\lambda_1(\mathbf{x})}{1.5(w_{domain})} \\ \theta_2(\mathbf{x}) &= \frac{\lambda_2(\mathbf{x})}{1.5(w_{domain})} \end{aligned} \quad (9)$$

where $W^*(\theta)$ denotes the minimum weight as a function of the parameter and $V(\mathbf{x})$ denotes the volume of the entire assembly (cf. Fig. 7). Furthermore, the parameter $\theta(\mathbf{x})$ denotes the allowable compressive edge load for a safety factor of 1.5, and λ_1 and λ_2 denote the critical buckling eigenvalues for the symmetric and anti-symmetric boundary conditions, respectively, calculated via ABAQUS simulation of a single section (as shown in Fig. 8).

For this problem, only one run of each method is performed with the same initializing DOE of 50 designs. Then, P3GA performs 20 generations with a population size of 5 (yielding 100 additional function evaluations). Furthermore, EPO is used to find 100 additional designs by two procedures. In one run, two high-fidelity samples are taken per iteration, and the other run takes five (i.e., $n = 2$ and $n = 5$; cf. Algorithm 2). The attributes of all parametrically nondominated designs are displayed in Fig. 9, and the HVI is plotted against the number of high-fidelity function evaluations in Fig. 10. Given the same design initialization, both runs of EPO are able to find designs that can withstand larger loads while P3GA is unable to find any design that can withstand an edge load larger than 800 lbs/in (140.1 N/mm) resulting in inferior HVI values. This highlights the advantage of EPO. Because it is a

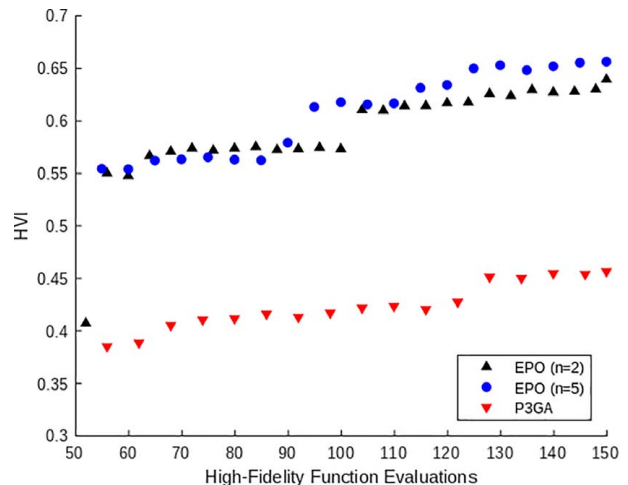


Fig. 10 HVI versus the number of high-fidelity function evaluations for the finite element analysis case study

global optimization technique, one would expect P3GA to find solutions for larger edge loads if given sufficient runtime. In contrast, EPO is able to identify these solutions quickly due to its use of a response surface model and adaptive sampling strategy.

6 Conclusions

This work details a new methodology that extends the efficient global optimization (EGO) framework to parametric optimization. The novel EPO shows promise for the problems considered. For the analytical test problem #1 and the engineering case study, EPO consistently outperformed the existing P3GA. The expected improvement formulation allows EPO to locate promising designs with less exploration of the high-fidelity design space compared to the alternative, P3GA. For the analytical test problem #2, the performances of EPO and P3GA were largely the same. For each problem considered, two variations of the EPO algorithm are performed updating the response surface with a different number of high-fidelity samples per iteration, but no statistical difference is found in these results. The problems considered in this paper are only a small subset of problems applicable for EPO, but it is promising that EPO performed as well as or better than the competition for all problems considered.

7 Future Work

There are several avenues of future work motivated by this paper. First, the new EPO algorithm will be evaluated on a wider variety of benchmarking problems with varying numbers of parameters and optimization variables to fully assess the scalability and limitations of the algorithm. Furthermore, the algorithm will be evaluated on problems with independent parameters in addition to dependent parameters to assess whether this has an impact on algorithmic performance. To further showcase the impact of this novel algorithm, EPO will be used to solve more engineering case studies comparing its performance to that of existing parametric algorithms. The results presented herein do not give a strong indication of how the number of high-fidelity samples evaluated per sampling step, n , affects the algorithmic performance. Therefore, this will be studied in more depth using a wider range of values in order to draw a conclusion. Finally, an extension of this algorithm to solve parametric and multi-objective problems is the topic of ongoing research.

Acknowledgment

This work is supported by the NASA *University Leadership Initiative* (ULI) program under federal award number NNX17AJ96A, titled Adaptive Aerostructures for Revolutionary Civil Supersonic Transportation. The authors would also like to thank Dr. Darren Hartl and his Aerospace Structural Design class (AERO 604 at Texas A&M University) for the engineering case study problem formulation.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The authors attest that all data for this study are included in the paper.

Nomenclature

s = response surface model prediction of standard error
 \mathbf{x} = optimization variable vector

J = high-fidelity objective function
 \hat{J} = response surface model prediction of objective
 θ = parameter vector
 $\hat{\theta}$ = response surface model prediction of parameters

References

- [1] Fiacco, A. V., 1976, "Sensitivity Analysis for Nonlinear Programming Using Penalty Methods," *Math. Program.*, **10**(1), pp. 287–311.
- [2] Pistikopoulos, E. N., and Georgiadis, C., 2007, *Multi-Parametric Programming: Theory, Algorithms, and Applications*. No. Centre for Process Systems Engineering. Efstratios, N. Pistikopoulos, ed., Vol. 1 in Process systems engineering, Wiley-VCH, Weinheim. OCLC: 180943251.
- [3] Filippi, C., 2004, "An Algorithm for Approximate Multiparametric Linear Programming," *J. Optim. Theory Appl.*, **120**(1), pp. 73–95.
- [4] Johansen, T., 2002, "On Multi-parametric Nonlinear Programming and Explicit Nonlinear Model Predictive Control," Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, NV.
- [5] Pistikopoulos, E. N., Dua, V., Bemporad, A., and Morari, M., 2004, "On-line Optimization Via Off-Line Parametric Optimization Tools," *Comput. Chem. Eng.*, **24**(2–7), pp. 183–188.
- [6] Hartl, D. J., Galvan, E., Malak, R. J., and Baur, J. W., 2016, "Parameterized Design Optimization of a Magnetohydrodynamic Liquid Metal Active Cooling Concept," *ASME J. Mech. Des.*, **138**(3), p. 031402.
- [7] Weaver-Rosen, J. M., Leal, P. B. C., Hartl, D. J., and Malak, R. J., 2020, "Parametric Optimization for Morphing Structures Design: Application to Morphing Wings Adapting to Changing Flight Conditions," *Struct. Multidiscipl. Optim.*, **62**(6), pp. 2995–3007.
- [8] Keane, A. J., and Voutchkov, I. I., 2020, "Robust Design Optimization Using Surrogate Models," *J. Comput. Design Eng.*, **7**(1), pp. 44–55.
- [9] Beyer, H.-G., and Sendhoff, B., 2007, "Robust Optimization—A Comprehensive Survey," *Comput. Methods Appl. Mech. Eng.*, **196**(33–34), pp. 3190–3218.
- [10] Li, M., and Wang, Z., 2018, "Confidence-Driven Design Optimization Using Gaussian Process Metamodeling With Insufficient Data," *ASME J. Mech. Des.*, **140**(12), p. 121405.
- [11] Gamboa, P., Vale, J., and Suleman, A., 2009, "Optimization of a Morphing Wing Based on Coupled Aerodynamic and Structural Constraints," *AIAA J.*, **47**(9), pp. 2087–2104.
- [12] Acevedo, J., and Salgueiro, M., 2003, "An Efficient Algorithm for Convex Multiparametric Nonlinear Programming Problems," *Ind. Eng. Chem. Res.*, **42**(23), pp. 5883–5890.
- [13] Spiegel, M., and Subrahmanyam, A., 1992, "Informed Speculation and Hedging in a Noncompetitive Securities Market," *Rev. Financial Studies*, **5**(2), pp. 307–329.
- [14] Gal, T., 1984, "Linear Parametric Programming—A Brief Survey," *Sensitivity, Stability and Parametric Analysis*, Fiacco, A. V., ed. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 43–68.
- [15] Spjøtvold, J., Tøndel, P., and Johansen, T. A., 2007, "Continuous Selection and Unique Polyhedral Representation of Solutions to Convex Parametric Quadratic Programs," *J. Optim. Theory Appl.*, **134**(2), pp. 177–189.
- [16] Bemporad, A., and Filippi, C., 2003, "Suboptimal Explicit Receding Horizon Control Via Approximate Multiparametric Quadratic Programming," *J. Optim. Theory Appl.*, **117**(1), pp. 9–38.
- [17] Dua, V., and Pistikopoulos, E. N., 1999, "Algorithms for the Solution of Multiparametric Mixed-Integer Nonlinear Optimization Problems," *Ind. Eng. Chem. Res.*, **38**(10), pp. 3976–3987.
- [18] Galvan, E., and Malak, R. J., 2015, "P3GA: An Algorithm for Technology Characterization," *ASME J. Mech. Des.*, **137**(1), p. 011401.
- [19] Hale, E. T., 2005, "Numerical Methods for D-parametric Nonlinear Programming with Chemical Process Control and Optimization Applications," Ph.D. thesis, The University of Texas at Austin, Austin, TX.
- [20] Galvan, E., Malak, R. J., Hartl, D. J., and Baur, J. W., 2018, "Performance Assessment of a Multi-objective Parametric Optimization Algorithm with Application to a Multi-physical Engineering System," *Struct. Multidiscipl. Optim.*, **58**(2), pp. 489–509.
- [21] Jones, D. R., Schonlau, M., and Welch, W. J., 1998, "Efficient Global Optimization of Expensive Black-Box Functions," *J. Global Optim.*, **13**(4), pp. 455–492.
- [22] Gutmann, H. M., 2001, "A Radial Basis Function Method for Global Optimization," *J. Global Optim.*, **19**(3), pp. 201–227.
- [23] Wang, D., Wu, Z., Fei, Y., and Zhang, W., 2014, "Structural Design Employing a Sequential Approximation Optimization Approach," *Comput. Struct.*, **134**(1), pp. 75–87.
- [24] Knowles, J., 2006, "ParEGO: A Hybrid Algorithm with on-line Landscape Approximation for Expensive Multiobjective Optimization Problems," *IEEE Trans. Evolut. Comput.*, **10**(1), pp. 50–66.
- [25] Keane, A. J., 2006, "Statistical Improvement Criteria for Use in Multiobjective Design Optimization," *AIAA J.*, **44**(4), pp. 879–891.
- [26] Emmerich, M., Giannakoglou, K., and Naujoks, B., 2006, "Single- and Multiobjective Evolutionary Optimization Assisted by Gaussian Random Field Metamodels," *IEEE Trans. Evolut. Comput.*, **10**(4), pp. 421–439.
- [27] Hernández-Lobato, J. M., Hoffman, M. W., and Ghahramani, Z., 2014, "Predictive Entropy Search for Efficient Global Optimization of Black-Box Functions," *Advances in Neural Information Processing Systems 27*,

- Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds., Curran Associates Inc., New York, pp. 918–926.
- [28] Sun, G., Tian, Y., Wang, R., Fang, J., and Li, Q., 2020, “Parallelized Multiobjective Efficient Global Optimization Algorithm and Its Applications,” *Struct. Multidiscipl. Optim.*, **61**(2), pp. 763–786.
- [29] Khatamsaz, D., Peddareddygar, L., Friedman, S., and Allaire, D. L., 2020, “Efficient Multi-Information Source Multiobjective Bayesian Optimization,” AIAA Scitech 2020 Forum, Orlando, FL.
- [30] Jongen, H. T., Jonker, P., and Twilt, F., 1986, “Critical Sets in Parametric Optimization,” *Math. Program.*, **34**(3), pp. 333–353.
- [31] Kojima, M., and Hirabayashi, R., 1984, “Continuous Deformation of Nonlinear Programs,” *Sensitivity, Stability and Parametric Analysis*, Vol. 21, A. V. Fiacco, ed., Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 150–198.
- [32] Lundberg, B. N., and Poore, A. B., 1993, “Numerical Continuation and Singularity Detection Methods for Parametric Nonlinear Programming,” *SIAM J. Optim.*, **3**(1), pp. 134–154.
- [33] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., 2002, “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II,” *IEEE Trans. Evolut. Comput.*, **6**(2), pp. 182–197.
- [34] Malak, R. J., and Paredis, C. J. J., 2010, “Using Parameterized Pareto Sets to Model Design Concepts,” *ASME J. Mech. Des.*, **132**(4), p. 041007.
- [35] Liu, H., Ong, Y.-S., and Cai, J., 2018, “A Survey of Adaptive Sampling for Global Metamodeling in Support of Simulation-Based Complex Engineering Design,” *Struct. Multidiscipl. Optim.*, **57**(1), pp. 393–416.
- [36] Deb, K., Thiele, L., Laumanns, M., and Zitzler, E., 2005, “Scalable Test Problems for Evolutionary Multiobjective Optimization,” *Evolutionary Multiobjective Optimization*, Abraham, A., Jain, L., and Goldberg, R., eds., Springer-Verlag, London, pp. 105–145.
- [37] Zitzler, E., Brockhoff, D., and Thiele, L., 2007, “The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration,” *Evolutionary Multi-Criterion Optimization*, Vol. 4403, Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., and Murata, T., eds., Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 862–876.
- [38] While, L., Bradstreet, L., and Barone, L., 2012, “A Fast Way of Calculating Exact Hypervolumes,” *IEEE Trans. Evolut. Comput.*, **16**(1), pp. 86–95.
- [39] Jaszekiewicz, A., 2018, “Improved Quick Hypervolume Algorithm,” *Comput. Operat. Res.*, **90**(1), pp. 72–83.
- [40] Bader, J., Deb, K., and Zitzler, E., 2010, “Faster Hypervolume-Based Search Using Monte Carlo Sampling,” *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems*, Vol. 634, Ehr Gott, M., Naujoks, B., Stewart, T., and Wallenius, J., eds. Lecture Notes in Economics and Mathematical Systems, Springer, Berlin, Heidelberg, pp. 313–326.
- [41] Timoshenko, S., and Gere, J. M., 2009, “Elastic Buckling of Bars and Frames,” *Theory of Elastic Stability*, 2nd ed., W. Prager, ed., Dover Publications, Mineola, NY, pp. 46–162.
- [42] Gao, S., Zhao, W., Lin, H., Yang, F., and Chen, X., 2010, “Feature Suppression Based CAD Mesh Model Simplification,” *Comput.-Aided Design*, **42**(12), pp. 1178–1188.