

DETC2017-67541

A MULTIOBJECTIVE ADAPTIVE SURROGATE MODELING-BASED OPTIMIZATION (MO-ASMO) FRAMEWORK USING EFFICIENT SAMPLING STRATEGIES

Yong Hoon Lee, R. E. Corman, Randy H. Ewoldt, James T. Allison
University of Illinois at Urbana-Champaign, Urbana, IL, 61801
Email: {ylee196, corman1, ewoldt, jtalliso}@illinois.edu

ABSTRACT

A novel multiobjective adaptive surrogate modeling-based optimization (MO-ASMO) framework is proposed to utilize a minimal number of training samples efficiently for sequential model updates. All the sample points are enforced to be feasible, and to provide coverage of sparsely explored sparse design regions using a new optimization subproblem. The MO-ASMO method only evaluates high-fidelity functions at feasible sample points. During an exploitation sample phase, samples are selected to enhance solution accuracy rather than the global exploration. Sampling tasks are especially challenging for multiobjective optimization; for an n -dimensional design space, a strategy is required for generating model update sample points near an $(n - 1)$ -dimensional hypersurface corresponding to the Pareto set in the design space. This is addressed here using a force-directed layout algorithm, adapted from graph visualization strategies, to distribute feasible sample points evenly near the estimated Pareto set. Model validation samples are chosen uniformly on the Pareto set hypersurface, and surrogate model estimates at these points are compared to high-fidelity model responses. All high-fidelity model evaluations are stored for later use to train an updated surrogate model. The MO-ASMO algorithm, along with the set of new sampling strategies, are tested using two mathematical and one realistic engineering problems. The second mathematical test problems is specifically designed to test the limits of this algorithm to cope with very narrow, non-convex feasible domains. It involves oscillatory objective functions, giving rise to a discontinuous set of Pareto-optimal solutions. Also, the third test problem demonstrates that the MO-ASMO algorithm can handle a practical engineering problem

with more than 10 design variables and black-box simulations. The efficiency of the MO-ASMO algorithm is demonstrated by comparing the result of two mathematical problems to the results of the NSGA-II algorithm in terms of the number of high fidelity function evaluations, and is shown to reduce total function evaluations by several orders of magnitude when converging to the same Pareto sets.

1 NOMENCLATURE

Variables and Parameters

x	Design variable	<i>dim</i>	Dimension
f	Objective function value	<i>idx</i>	Index
lb	Lower bound	<i>k</i>	Iteration number
ub	Upper bound	<i>n</i>	Number
c	Constraint function	ϵ	Error (L^2 -norm)
\mathcal{D}	Data storage	i	Array of indices
\mathcal{M}	Surrogate model	<i>i, j</i>	Loop index
s	Displacement	v	Velocity

Subscripts

x	In design space	smp	Sample
f	In objective function space	tmp	Temporary
l	Linear	rem	Remaining
nl	Nonlinear	ii	Initial
k	Iteration number	pop	GA Population
mi	Maximum iteration	hf	High fidelity
cr	Criteria	v	Validation
u	Update	er	Exploration
st	Stored	et	Exploitation
sc	Scaled	LH	Latin Hypercube Sampling
P	Pareto set	ctr	Centroid
p	Predicted		

Functions

$r \leftarrow \text{choose}(i, j, 'rand')$	Choose ['rand', 'big', 'small'] from i and j
$\mathbf{x}_{\text{ctr}} \leftarrow \text{cluster_ctr}(\mathbf{x}, n)$	Get n cluster centroids from \mathbf{x} using k-means
$\mathbf{x} \leftarrow \text{descale}(\mathbf{x}, \mathbf{lb}, \mathbf{ub})$	Convert \mathbf{x} from $[0, 1]$ to $[\mathbf{lb}, \mathbf{ub}]$ space
$d \leftarrow \text{dist}(\mathbf{x}_1, \mathbf{x}_2)$	Compute distances between \mathbf{x}_1 and \mathbf{x}_2
$\mathbf{i} \leftarrow \text{get_idx}(\mathbf{a}, \mathbf{b})$	Get indices of \mathbf{a} in set \mathbf{b}
$\mathbf{f} \leftarrow \text{hf_func_eval}(\mathbf{x})$	High fidelity function evaluation
$[\mathbf{x}, \mathbf{f}] \leftarrow \text{NSGA-II}(\mathfrak{M}, \mathbf{x})$	NSGA-II [1], multiobjective optimizer
$\mathbf{b} \leftarrow \text{rand}(dim)$	Get random numbers in $[0, 1] \in \mathbb{R}^{dim}$ space
$\mathbf{x} \leftarrow \text{rand_pop}()$	Get random populations for GA [2]
$\mathbf{x} \leftarrow \text{scale}(\mathbf{x}, \mathbf{lb}, \mathbf{ub})$	Convert \mathbf{x} from $[\mathbf{lb}, \mathbf{ub}]$ to $[0, 1]$ space
$\mathbf{b} \leftarrow \text{sign}(\mathbf{a})$	if $\mathbf{a}(i) > 0$, $\mathbf{b}(i) = 1$, otherwise $\mathbf{b}(i) = -1$
$\mathbf{x} \leftarrow \text{smp_exploit}(\mathbf{x}_{p,p})$	Sampling for exploitation
$\mathbf{x} \leftarrow \text{smp_explore}()$	Sampling for exploration
$\mathbf{x} \leftarrow \text{smp_lh}(n, dim)$	Latin Hypercube Sampling
$\mathbf{x} \leftarrow \text{smp_uniform}(\mathbf{x})$	Uniformly sampling points
$\mathbf{x} \leftarrow \text{smp_validate}(\mathbf{x}_{p,p}, \mathbf{f}_{p,p})$	Sampling for validation
$\mathbb{A} \leftarrow \text{sort}(\mathbb{A}, j, 'asc')$	Sort \mathbb{A} in ['asc', 'dec'] order for column j
$\mathfrak{M} \leftarrow \text{surr_model_constr}(\mathbf{x}, \mathbf{f})$	Surrogate model construction
$\mathbf{f} \leftarrow \text{surr_model_eval}(\mathbf{x})$	Surrogate model evaluation
$\mathbf{i} \leftarrow \text{unique}(\mathbf{i}, \mathbb{A})$	Remove duplicated data from $\mathbb{A}(\mathbf{i})$

2 INTRODUCTION

Many practical design problems in engineering and other fields have multiple design objectives, which form multiobjective decision-making problems [3]. Generally, some of these design objectives conflict with each other and the decision problem, therefore, involves a trade-off between objectives. For example, system performance and cost are common competing objectives [4, 5, 6], while different types of system performance metrics also can compete with each other, such as minimizing friction and maximizing load capacity for fluid power systems [7, 8]. Thus, the solution of such a problem is generally expressed as a set of non-dominated (Pareto-optimal) design alternatives. A design is called non-dominated if no objective function can be improved without degrading any other objective functions [3, 7, 9].

Several Multiobjective Evolutionary Algorithms (MOEAs) have been developed and demonstrated to be effective in finding Pareto-optimal solutions [1, 10, 11]. MOEAs obtain a set of multiple Pareto-optimal design solutions simultaneously in a single optimization algorithm execution. Another type of algorithm for solving multiobjective optimization problems is scalarization [12, 13], examples of which include the weighted-sum method [14], the ϵ -constraint method [15], and the Pascoletti-Serafini method [16]. This type of algorithm converts a single multiobjective optimization problem into a set of single-objective optimization problems, each of which produces a Pareto-optimal point. An important drawback of both MOEAs and scalarization is the potential for a large required number of function evaluations. This is particularly concerning for problems based on high-fidelity simulations.

To mitigate the computational burden for solving high-fidelity optimization problems, surrogate modeling (or metamodeling) is a well-established strategy [17, 18, 19]. A limited set of design samples is evaluated using the high-fidelity model, and an approximation model is constructed based on these sample

points. This approximation could be realized using one of several strategies, such as polynomial response surfaces [20], kriging (or Gaussian process regression) [21, 17], support vector machines [22], or radial basis function methods [23]. Optimization is then performed using the inexpensive surrogate model instead of with the high-fidelity model. Surrogate models provide insight into the design problems using a limited number of high-fidelity function evaluations. Constructing a surrogate model that is accurate across the entire design domain is computationally expensive, and for optimization purposes is inefficient. The model only needs to be accurate in the neighborhood of the solution. An important strategy for limiting high-fidelity evaluations is to begin with a coarse initial sample, and then adaptively select new sample points to update the surrogate model [24, 25, 26, 27]. New samples often are either chosen in regions near the predicted optimum, or in sparsely sampled regions to improve the probability of finding a global optimum. Adaptive surrogate modeling strategies can be classified based on updating sequence and other characteristics. Surveys of surrogate modeling methods and strategies for design optimization are presented in Refs. [28, 29, 30].

Managing constraints is challenging when utilizing surrogate model-based optimization. Eldred and Dunlavy [31] considered constraints only during the optimization phase; constraints are not considered by the surrogate model. Hussein and Deb [32] allowed infeasible training points, but used a convexifying term involving a sum of constraint violations. These strategies, however, are less effective for problems where constraints are coupled densely with design variables (e.g., a multidimensional geometric mesh with gradient limiter). In addition, many engineering simulation models may fail when attempting to evaluate infeasible design points. Methods should prevent high-fidelity evaluation of infeasible points. This could be accomplished by filtering, but in problems with narrow, non-convex feasible domains this may require discarding a large portion of sample points generated using conventional strategies (e.g., latin hypercube sampling). Processing and filtering a large number of infeasible sample points to obtain a few feasible points may dominate computational effort. Here we seek strategies that by nature generate only feasible sample points.

Another sampling challenge touched on above is efficient sampling for updating surrogate models, balancing exploration (global search) and exploitation (precise solution) of the design space [33, 34, 35]. Previous studies aimed to achieve the following goals simultaneously: 1) finding a global solution, 2) finding an accurate local solution, and 3) limiting the number of high-fidelity simulations. Many efforts in adaptive surrogate modeling have been successful in achieving these objectives for single-objective problems. Extension to multiobjective optimization, however, gives rise to new challenges that have not been addressed thoroughly. For example, regarding the exploitation objective, an accurate local solution requires that the surrogate model is accurate in the neighborhood of the estimated Pareto set

(often an $(n - 1)$ hypersurface in the design space), as opposed to just being accurate in the neighborhood of the estimated optimal design (a single design point). Quantifying this local accuracy and validating the surrogate model is much more complicated for multiobjective problems compared to single objective problems. This motivates new efficient sampling strategies for model validation.

This article presents advancements of adaptive surrogate modeling for multiobjective optimization that aim to reduce the number of required simulations to obtain optimal solutions (Pareto sets). These advancements aim to cope with challenging problem characteristics, such as tightly-constrained design spaces (small, complex feasible domains) and the need to evaluate only feasible sample points. Contributions of this study include: (1) a novel global sampling strategy for tightly-constrained design spaces, and (2) a novel adaptive sampling strategy. The proposed global sampling algorithm can generate globally distributed feasible training points. This advantage is especially useful when the feasible design space is extremely small relative to the full design space, which often makes standard sampling strategies impractical. In addition, the new adaptive sampling algorithm utilizes a novel force directed point locating strategy for generating sample points near the estimated Pareto set. This approach is inspired by force directed layout algorithms used for graph visualization [36].

3 METHOD

3.1 MO-ASMO approach

The proposed multiobjective adaptive surrogate modeling optimization (MO-ASMO) method [37] solves multiobjective optimization problems on a constructed surrogate model, which is updated iteratively using new training sample points. The flowchart in Fig. 1 illustrates this method, which this can be classified as a *direct sampling approach* using Wang and Shan’s categorization strategy [28]. Initial training samples are generated, and function values are obtained for them using high fidelity function evaluation. Using all existing results of the high fidelity model, a surrogate model is constructed, which is then used with a multiobjective optimization algorithm to generate an estimate of the Pareto set for the original full-fidelity problem. The framework is independent of multiobjective optimization

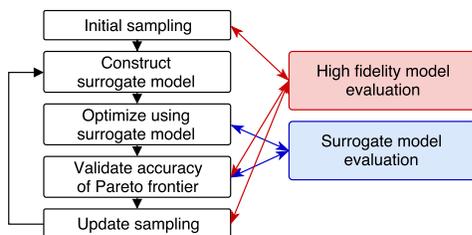


FIGURE 1: Flow chart of proposed MO-ASMO framework

algorithm choice. Any type of MOEAs (e.g., NSGA-II algorithm [1]) or gradient-based multiobjective optimization strategies (e.g., scalarization methods [12, 13]) can be used. Here we used NSGA-II for our test problems, assuming that computational costs for the optimization using surrogate model is very low compared to high-fidelity model evaluations. Even if many thousands of function evaluations are required to obtain a good Pareto set estimate based on the surrogate model, this computational expense is still a fraction of high-fidelity simulation expense. This framework is capable of handling a large number of design variables, as well as more than two objective functions. To avoid excessive memory consumption in these multidimensional problems, we refrained from using full factorial type loops in the algorithm.

3.2 MO-ASMO Algorithm

The MO-ASMO algorithm is presented in Alg. 1 as pseudocode, and illustrates how the data flow of training points and model evaluations is managed. The first stage is a problem initialization (Line 1). Variable dimensions, bounds, and constraints of the multiobjective optimization problem are defined here. Equality and inequality constraints can be given in both linear ($\mathbb{A}\mathbf{x} \leq \mathbf{b}$, $\mathbb{A}_{\text{eq}}\mathbf{x} = \mathbf{b}_{\text{eq}}$) and nonlinear ($\mathbf{c}(\mathbf{x}) \leq \mathbf{0}$, $\mathbf{c}_{\text{eq}}(\mathbf{x}) = \mathbf{0}$) forms. After initializing, the first set of training samples are generated in the global design space (Line 2). The details of this step are presented in Sec. 3.3, which involves a novel sampling strategy for tightly-constrained design spaces.

The main loop of the iterative enhancement phase using adaptive sampling is given in Lines 3-27. First, as stated in Line 5, the script performs high fidelity function evaluations for the given samples. For the first iteration, initial samples are used as update samples to be evaluated (Line 2). In Lines 6-11, these high fidelity results are combined with already stored high-fidelity results from previous iterations, and a surrogate model is constructed using all available high-fidelity results (Line 12).

Here NSGA-II is used as the multiobjective optimization solver. Lines 13-18 show how the initial populations are prepared for the optimization solver. The predicted Pareto set in the previous iteration is used as the initial population. Also, the set of stored high-fidelity results are sorted by objective function value, and are sampled uniformly (including anchor points). These sampled points are also useful to help preserve the anchor points of existing data. During the first iteration, however, the initial population is selected randomly as shown in Line 19. After obtaining the surrogate model and initial populations, the NSGA-II algorithm is called to solve the optimization problem using the surrogate model as shown in Line 21.

After a Pareto set is obtained by the multiobjective optimization solver, the solution must be checked to determine whether it satisfies a specified level of accuracy. If it does, the algorithm stops. If not, the process is repeated with an enhanced surrogate

model. From the obtained Pareto set, samples for validating accuracy of the Pareto frontier are chosen as shown in Line 22. The detailed algorithm for choosing these samples is given in Section 3.4. Using the validation samples, resulting objective function values from both the surrogate and high-fidelity model are obtained and compared to store the estimated error in Lines 23-24.

Before proceeding to the next iteration with updated samples and an updated surrogate model, new samples must be identified to aid both exploration (regions with low information), and exploitation (near the estimated Pareto set). This sampling is performed in Lines 25 and 26, and added to the set of update samples for the next iteration (Line 27).

3.3 Sampling for exploration

A novel sampling strategy was created for generating exploration training samples (defined in Alg. 2). This algorithm is specifically useful when the design space is tightly constrained, and general space-filling sampling algorithms cannot easily produce a reasonable number of feasible sample points. In Lines 4 and 5, the specified number of samples are generated using the latin hypercube sampling method [38] and rescaled appropriately for design variables ranges. This process generates samples of orthogonal designs throughout a hypercube of the design space

Algorithm 1 MO-ASMO parent script

```

1: Set  $dim_x, dim_f, \mathbf{lb}_x, \mathbf{lb}_f, \mathbf{ub}_x, \mathbf{ub}_f, \mathbf{c}_1, \mathbf{c}_n$ ;  $k \leftarrow 0$ 
2:  $\mathcal{D}_k \cdot \mathbf{x}_u \leftarrow \text{smp\_explore}(\text{initial sampling})$ 
3: while ( $k < n_{mi}$ ) & ( $\varepsilon > \varepsilon_{cr}$ ) do
4:    $k \leftarrow k + 1$ 
5:    $\mathcal{D}_k \cdot \mathbf{f}_{u,hf} \leftarrow \text{hf\_func\_eval}(\mathcal{D}_k \cdot \mathbf{x}_u)$ 
6:   if ( $k > 1$ ) then
7:      $\mathcal{D}_k \cdot \mathbf{x}_{st} \leftarrow [\mathcal{D}_{k-1} \cdot \mathbf{x}_{st}, \mathcal{D}_{k-1} \cdot \mathbf{x}_v, \mathcal{D}_k \cdot \mathbf{x}_u]$ 
8:      $\mathcal{D}_k \cdot \mathbf{f}_{st,hf} \leftarrow [\mathcal{D}_{k-1} \cdot \mathbf{f}_{st,hf}, \mathcal{D}_{k-1} \cdot \mathbf{f}_{v,hf}, \mathcal{D}_k \cdot \mathbf{f}_{u,hf}]$ 
9:   else
10:     $\mathcal{D}_k \cdot \mathbf{x}_{st} \leftarrow [\mathcal{D}_k \cdot \mathbf{x}_u]$ 
11:     $\mathcal{D}_k \cdot \mathbf{f}_{st,hf} \leftarrow [\mathcal{D}_k \cdot \mathbf{f}_{u,hf}]$ 
12:    $\mathcal{M}_k \leftarrow \text{surr\_model\_constr}(\mathcal{D}_k \cdot \mathbf{x}_{st}, \mathcal{D}_k \cdot \mathbf{f}_{st,hf})$ 
13:   if ( $k > 1$ ) then
14:      $\mathbf{i} \leftarrow []$ 
15:     for  $j \leftarrow$  (each dimension number of  $\mathbf{x}_{st}$ ) do
16:        $\mathbf{x}_{tmp} \leftarrow \text{sort}(\mathcal{D}_k \cdot \mathbf{x}_{st}, j, \text{'asc'})$ 
17:        $\mathbf{i} \leftarrow [\mathbf{i}, \text{get\_idx}(\text{smp\_uniform}(\mathbf{x}_{tmp}), \mathbf{x}_{tmp})]$ 
18:      $\mathbf{x}_{ii,pop} \leftarrow [\mathcal{D}_{k-1} \cdot \mathbf{x}_{p,p}, \mathcal{D}_k \cdot \mathbf{x}_{st}(\mathbf{i})]$ 
19:   else
20:      $\mathbf{x}_{ii,pop} \leftarrow \text{rand\_pop}()$ 
21:    $\mathcal{D}_k \cdot [\mathbf{x}_{p,p}, \mathbf{f}_{p,p}] \leftarrow \text{NSGA-II}(\mathcal{M}_k, \mathbf{x}_{ii,pop})$ 
22:    $\mathcal{D}_k \cdot [\mathbf{x}_v, \mathbf{f}_{v,p}] \leftarrow \text{smp\_validate}(\mathcal{D}_k \cdot [\mathbf{x}_{p,p}, \mathbf{f}_{p,p}])$ 
23:    $\mathcal{D}_k \cdot \mathbf{f}_{v,hf} \leftarrow \text{hf\_func\_eval}(\mathcal{D}_k \cdot \mathbf{x}_v)$ 
24:    $\varepsilon_k \leftarrow \mathcal{D}_k \cdot \|\mathbf{f}_{v,p} - \mathbf{f}_{v,hf}\|$ 
25:    $\mathbf{x}_{et} \leftarrow \text{smp\_exploit}(\mathcal{D}_k \cdot \mathbf{x}_{p,p})$ 
26:    $\mathbf{x}_{er} \leftarrow \text{smp\_explore}()$ 
27:    $\mathcal{D}_{k+1} \cdot \mathbf{x}_u \leftarrow [\mathbf{x}_{et}, \mathbf{x}_{er}]$ 

```

without considering any constraints. Next, an optimization problem is solved for each sample point to satisfy constraints, while minimizing the relocation distance of the point and maximizing the sum of distances from existing points (Lines 6-8). Since the subproblem given in Line 7 is a single-objective optimization problem, any appropriate choice of nonlinear programming algorithm can be used (sequential quadratic programming (SQP) used here). All relocated training samples then satisfy the constraints, and have reasonably even distribution.

Algorithm 2 Sampling in a densely constrained design space

```

1: function smp_explore ()
2:   if initial sampling then  $n \leftarrow n_{ii,er,smp}$ 
3:   else  $n \leftarrow n_{er,smp}$ 
4:    $\mathbf{x}_{tmp,sc} \leftarrow \text{smp\_lh}(n, dim_x)$ 
5:    $\mathbf{x}_{tmp} \leftarrow \text{descal}(\mathbf{x}_{tmp,sc}, \mathbf{lb}_x, \mathbf{ub}_x)$ 
6:   for  $j \leftarrow 1, 2, \dots, n$  do
7:     Solve:  $\min \{f(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_{tmp}(j)\|$ 
8:            $+ \sum_i [-\ln(\|\mathbf{x} - \mathbf{x}_{stored}(i)\| \neq 0)], \forall i\}$ ,
9:     subject to  $\{\Delta \mathbf{x} - \mathbf{b} \leq \mathbf{0}, \Delta_{eq} \mathbf{x} - \mathbf{b}_{eq} = \mathbf{0},$ 
10:               $\mathbf{c}(\mathbf{x}) \leq \mathbf{0}, \mathbf{c}_{eq}(\mathbf{x}) = \mathbf{0}\}$ 
11:      $\mathbf{x}_{smp}(j) \leftarrow \mathbf{x}$ 
12:   return  $\mathbf{x}_{smp}$ 

```

Algorithm 3 Sampling for validation of surrogate model

```

1: function smp_validate ( $\mathbf{x}_{p,p}, \mathbf{f}_{p,p}$ )
2:    $\mathbf{x}_{tmp,1} \leftarrow \text{scale}(\mathbf{x}_{p,p}, \mathbf{lb}_x, \mathbf{ub}_x)$ 
3:    $\mathbf{f}_{tmp,1} \leftarrow \text{scale}(\mathbf{f}_{p,p}, \mathbf{lb}_f, \mathbf{ub}_f)$ 
4:    $\mathbf{i} \leftarrow []$ 
5:   for  $j \leftarrow$  (each dimension number of  $\mathbf{f}$ ) do
6:      $\mathbf{i} \leftarrow [\mathbf{i}, \text{get\_idx}(\min(\mathbf{f}_{tmp,1}(j)))]$ 
7:      $\mathbf{i} \leftarrow [\mathbf{i}, \text{get\_idx}(\max(\mathbf{f}_{tmp,1}(j)))]$ 
8:    $\mathbf{i} \leftarrow \text{unique}(\mathbf{i}, [\mathbf{x}_{tmp,1}, \mathbf{f}_{tmp,1}])$ 
9:    $\mathbf{x}_{tmp,v} \leftarrow \mathbf{x}_{tmp,1}(\mathbf{i}); \mathbf{f}_{tmp,v} \leftarrow \mathbf{f}_{tmp,1}(\mathbf{i})$ 
10:   $n \leftarrow n_{v,smp} - \text{count}(\mathbf{x}_{tmp,v})$ 
11:   $\mathbf{x}_{rem} \leftarrow \mathbf{x}_{tmp,1}(\text{idx} \neq \mathbf{i}); \mathbf{f}_{rem} \leftarrow \mathbf{f}_{tmp,1}(\text{idx} \neq \mathbf{i})$ 
12:  while ( $\text{count}(\mathbf{x}_{rem}) > n$ ) do
13:     $\mathbf{d} \leftarrow []; \mathbf{i} \leftarrow []; \mathbf{j} \leftarrow []$ 
14:    for  $j \leftarrow$  (each index of  $\mathbf{x}_{rem}$ ) do
15:      for  $i \leftarrow$  (each index of  $\mathbf{x}_{rem}$ ) do
16:        if  $i \neq j$  then
17:           $\mathbf{d} \leftarrow [\mathbf{d}, \text{dist}(\mathbf{x}_{rem}(i), \mathbf{x}_{rem}(j))]$ 
18:           $[\mathbf{i}, \mathbf{j}] \leftarrow [[\mathbf{i}, i], [\mathbf{j}, j]]$ 
19:         $[\mathbf{i}, \mathbf{j}, \mathbf{d}] \leftarrow \text{sort}([\mathbf{i}, \mathbf{j}, \mathbf{d}], 3, \text{'asc'})$ 
20:         $i \leftarrow \mathbf{i}(1); j \leftarrow \mathbf{j}(1)$ 
21:         $r \leftarrow \text{choose}(i, j, \text{'rand'})$ 
22:         $\mathbf{x}_{rem} \leftarrow \mathbf{x}_{rem}(\text{idx} \neq r); \mathbf{f}_{rem} \leftarrow \mathbf{f}_{rem}(\text{idx} \neq r)$ 
23:   $\mathbf{x}_{v,smp} \leftarrow \text{descal}([\mathbf{x}_{tmp,v}, \mathbf{x}_{rem}], \mathbf{lb}_x, \mathbf{ub}_x)$ 
24:   $\mathbf{f}_{v,smp} \leftarrow \text{descal}([\mathbf{f}_{tmp,v}, \mathbf{f}_{rem}], \mathbf{lb}_f, \mathbf{ub}_f)$ 
25:  return  $[\mathbf{x}_{v,smp}, \mathbf{f}_{v,smp}]$ 

```

3.4 Sampling for validation

Another sampling strategy generates sample points for surrogate model validation (Alg. 3). An easy approach for choosing validation samples would be to use estimated Pareto set points—resulting from the MOEA—directly. However, it is desirable to select a set of uniformly distributed points throughout the Pareto set hypersurface. This requires calculation of distances between pairs of Pareto-optimal points. First, the predicted Pareto set is scaled in Lines 2-3 before measuring normalized distance. Since each design variable may have its own scale, variable scaling is important for measuring distances in the design space. The anchor points in the objective function space are important for characterizing the solution prediction. Thus all the anchor points in the objective function space are selected for the validation in Lines 5-7. The remaining validation sample points are chosen as uniformly distributed samples selected from the Pareto set. Distances between all pair combinations of points are computed in Line 17. Distances are sorted in ascending order, and one point out of a pair with the minimum distance is eliminated in Lines 12-22 until the number of remaining points matches the required number of samples for validation. Since points are scaled, rescaling is performed in Lines 23-24.

3.5 Sampling for exploitation

A new exploitation sampling strategy was developed, inspired by force-directed layout algorithms, and is presented in Alg. 4. In this sampling stage, sampling points must be placed not on, but near the Pareto set hypersurface to improve surrogate model accuracy in the neighborhood of the estimated solution (Pareto set). Sample points are subjected to two types of predefined forces, and move dynamically toward equilibrium. One type of force is an attractive force between new sample points and base points, as quantified in Eqn. (1):

$$\vec{\mathbf{F}} = C_1 \ln\left(\frac{d}{C_2}\right) \hat{\mathbf{s}} \quad (1)$$

where $d = \|\mathbf{x}_{\text{base}} - \mathbf{x}_{\text{new}}\|$, $\hat{\mathbf{s}} = (\mathbf{x}_{\text{base}} - \mathbf{x}_{\text{new}})/d$. The base points are defined as n_2 number of centroids of the predicted Pareto set. The other type of force is a repulsive force between the new points and the points from the Pareto set, as defined in Eqn. (2):

$$\vec{\mathbf{F}} = \sum_{i=1}^{n_p} \left(\frac{C_3}{d_i^2}\right) \hat{\mathbf{s}}_i \quad (2)$$

where $d_i = \|\mathbf{x}_{\text{scaled},i} - \mathbf{x}_{\text{new}}\|$, $\hat{\mathbf{s}}_i = (\mathbf{x}_{\text{scaled},i} - \mathbf{x}_{\text{new}})/d_i$, and n_p is the number of points in the Pareto set. This force ensures that the new samples maintain an appropriate distance from the Pareto set hypersurface.

The Pareto set is scaled in Line 3, and the centroid of the Pareto set is computed in Line 4. From the centroid, n_1 design points with the largest distance from the centroid are selected as “base points” in Lines 5-8. Also, centroids of clustered Pareto

Algorithm 4 Sampling for exploitation of predicted Pareto set

```

1: function smp_exploit( $\mathbf{x}_{p,p}$ )
2:    $n_{\text{et},\text{smp},1} \leftarrow 20\%$  of  $n_{\text{et},\text{smp}}$ ;  $n_{\text{et},\text{smp},2} \leftarrow n_{\text{et},\text{smp}} - n_{\text{et},\text{smp},1}$ 
3:    $\mathbf{x}_{\text{sc}} \leftarrow \text{scale}(\mathbf{x}_{p,p}, \mathbf{lb}_x, \mathbf{ub}_x)$ 
4:    $\mathbf{x}_{\text{ctr},1} \leftarrow \text{cluster\_ctr}(\mathbf{x}_{\text{sc}}, 1)$ 
5:    $\mathbf{d} \leftarrow \text{dist}(\mathbf{x}_{\text{sc}}, \mathbf{x}_{\text{ctr},1})$ 
6:    $\mathbf{i} \leftarrow \text{get\_idx}(\text{sort}([\mathbf{x}_{\text{sc}}, \mathbf{d}], 2, 'des'))$ 
7:    $\mathbf{i} \leftarrow \text{unique}(\mathbf{i}(idx = 1, 2, \dots, n_{\text{et},\text{smp},1}))$ 
8:    $\mathbf{x}_{\text{ctr},2} \leftarrow \mathbf{x}_{\text{sc}}(\mathbf{i})$ 
9:    $\mathbf{x}_{\text{ctr},2} \leftarrow [\mathbf{x}_{\text{ctr},2}, \text{cluster\_ctr}(\mathbf{x}_{\text{sc}}, n_{\text{et},\text{smp},2})]$ 
10:   $\mathbf{x}_{\text{FDL}} \leftarrow \mathbf{x}_{\text{ctr},2} + 0.1 \times \text{sign}(\text{rand}(\text{dim}_x) - 0.5) \mathbf{x}_{\text{ctr},2}$ 
11:   $\mathbf{v}_{\text{FDL}} \leftarrow \mathbf{0}$ 
12:  while ( $\neg(\text{enough iterations})$ )  $\parallel$  ( $\neg(\mathbf{v}_{\text{FDL}} \ll \epsilon)$ ) do
13:    for  $j \leftarrow$  (each index of  $\mathbf{x}_{\text{FDL}}$ ) do
14:       $d_a \leftarrow \text{dist}(\mathbf{x}_{\text{ctr},2}(j), \mathbf{x}_{\text{FDL}}(j))$ 
15:       $\mathbf{s}_a \leftarrow (\mathbf{x}_{\text{ctr},2}(j) - \mathbf{x}_{\text{FDL}}(j))/d_a$ 
16:       $\mathbf{F}(j) \leftarrow C_{a1} \ln(d_a/C_{a2}) \mathbf{s}_a$ 
17:    for  $j \leftarrow$  (each index of  $\mathbf{x}_{\text{FDL}}$ ) do
18:      for  $i \leftarrow$  (each index of  $\mathbf{x}_{\text{sc}}$ ) do
19:         $d_b \leftarrow \text{dist}(\mathbf{x}_{\text{sc}}(i), \mathbf{x}_{\text{FDL}}(j))$ 
20:         $\mathbf{s}_b \leftarrow (\mathbf{x}_{\text{sc}}(i) - \mathbf{x}_{\text{FDL}}(j))/d_b$ 
21:         $\mathbf{F}(j) \leftarrow \mathbf{F}(j) + C_{b1}/(d_b)^2 \mathbf{s}_b$ 
22:       $\mathbf{v}_{\text{FDL}} \leftarrow (\mathbf{v}_{\text{FDL}} + (\mathbf{F}/M)\Delta t)/C_d$ 
23:       $\mathbf{x}_{\text{FDL}} \leftarrow \mathbf{x}_{\text{FDL}} + \mathbf{v}_{\text{FDL}}\Delta t$ 
24:   $\mathbf{x}_{\text{smp}} \leftarrow \text{descale}(\mathbf{x}_{\text{FDL}}, \mathbf{lb}_x, \mathbf{ub}_x)$ 
25:  return  $\mathbf{x}_{\text{smp}}$ 

```

set points are obtained to add as “base points” using k-means clustering in Line 9. From these base points, the same number of new samples are generated and placed near the corresponding base points in Line 10. Dynamic simulation of generated points in multidimensional space, subject to the forces acting on them, is performed iteratively in Lines 12-23 until the point positions have converged. The first force type is calculated in Lines 13-16 for each generated point. Also, the second force type is calculated in Lines 17-21 and added together in multidimensional vector form for each point j . The velocity values for the generated points are updated as given in Line 22 and Eqn. (3a) with a damping factor of $1/C_d$, which results in position convergence for each point. As given in Line 23, the resulting location at the next time step can be computed iteratively using the current locations of the generated sample points and the velocity values of corresponding points. Updates for velocity and position are given in Eqns. (3a) and (3b):

$$\vec{\mathbf{v}}_{\text{new}}^{(r+1)} = \left\{ \vec{\mathbf{v}}_{\text{new}}^{(r)} + \vec{\mathbf{a}}\Delta t \right\} / C_d \quad (3a)$$

$$\vec{\mathbf{x}}_{\text{new}}^{(r+1)} = \vec{\mathbf{x}}_{\text{new}}^{(r)} + \vec{\mathbf{v}}_{\text{new}}^{(r+1)} \Delta t, \quad (3b)$$

where $\vec{\mathbf{a}} = \vec{\mathbf{F}}/M$, $(r+1)$: next step, (r) : current step, M : mass, C_d : damping constant, and Δt : time step size. If the damping constant C_d is greater than 1, the velocity approaches zero, leading to an equilibrium state.

4 TEST PROBLEMS

4.1 Osyczka and Kundu problem

The Osyczka and Kundu problem [39] was first introduced to demonstrate the performance of a multiobjective evolutionary algorithm, and has been used widely since for testing multiobjective optimization algorithms. It involves six design variables, six inequality constraints, and two objective functions as given in Eqns. (4) and (5):

$$\min \mathbf{f}(\mathbf{x}) = [f_1, f_2] \quad \text{subject to} \quad \begin{aligned} g_1 &\leq 0, g_2 \leq 0, g_3 \leq 0, \\ g_4 &\leq 0, g_5 \leq 0, g_6 \leq 0, \end{aligned} \quad (4)$$

where

$$f_1(\mathbf{x}) = -[25(x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 1)^2] \quad (5a)$$

$$f_2(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 \quad (5b)$$

$$g_1(\mathbf{x}) = -x_1 - x_2 + 2 \quad (5c)$$

$$g_2(\mathbf{x}) = x_1 + x_2 - 6 \quad (5d)$$

$$g_3(\mathbf{x}) = -x_1 + x_2 - 2 \quad (5e)$$

$$g_4(\mathbf{x}) = x_1 - 3x_2 - 2 \quad (5f)$$

$$g_5(\mathbf{x}) = (x_3 - 3)^2 + x_4 - 4 \quad (5g)$$

$$g_6(\mathbf{x}) = -(x_5 - 3)^2 - x_6 + 4. \quad (5h)$$

This test problem was selected due to its low computational expense for initial studies, and due to its non-convex and sharply-kinked Pareto frontier. This leads to difficulty in approximating the function response accurately using surrogate models. While the computational expense of individual function evaluations is low, we can assess the computational advantage of the new MO-ASMO algorithm based on the total number of function evaluations.

Figure 2 shows the resulting Pareto frontiers for this problem obtained through application of the NSGA-II algorithm [1], as well as the proposed MO-ASMO algorithm. The Pareto frontiers are compared in the objective function space in Fig. 2(a), and intermediate solutions computed by NSGA-II for three different

cases are given in Fig. 2(b). We used population sizes (n_{pop}) of 500, 1000, and 2000 for solving this problem, and the corresponding line colors in this figure are orange, blue, and black. Dashed lines, dotted lines, and solid lines represent intermediate solutions after about 20000, 100000, and 400000 function evaluations (n_{hf}), respectively. After 40,3307 function evaluation ($n_{\text{pop}} = 2000$), the solution converged to the known Pareto frontier of the Osyczka and Kundu problem. Thus, we see that this problem is solved efficiently using a standard MOEA in terms of the number of function evaluations.

Figure 3 illustrates the predicted Pareto set obtained from the surrogate model (black symbols), and the predicted and true values of chosen validation points (red symbols). The “×” points represent the predicted value on the surrogate model, “o” points correspond to accurate results from high-fidelity function evaluations (which in most cases will be numerical simulations). During the early iterations the surrogate model cannot represent the original response accurately. As we see in Fig. 3(a), the predicted Pareto frontiers have significant error (for each validation design point, we show the difference between the result predicted by the surrogate model and by the original model). These erroneous predictions are corrected by including original function evaluation results for validation points in the set of training points for

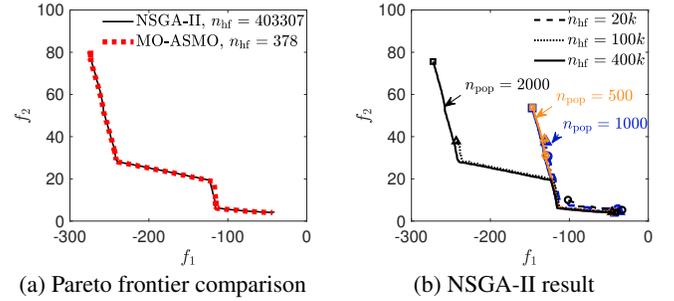


FIGURE 2: Comparison of Pareto frontiers of Osyczka and Kundu problem obtained by NSGA-II and MO-ASMO algorithms. Intermediate solutions of NSGA-II for 3 cases of different population number are also given in (b).

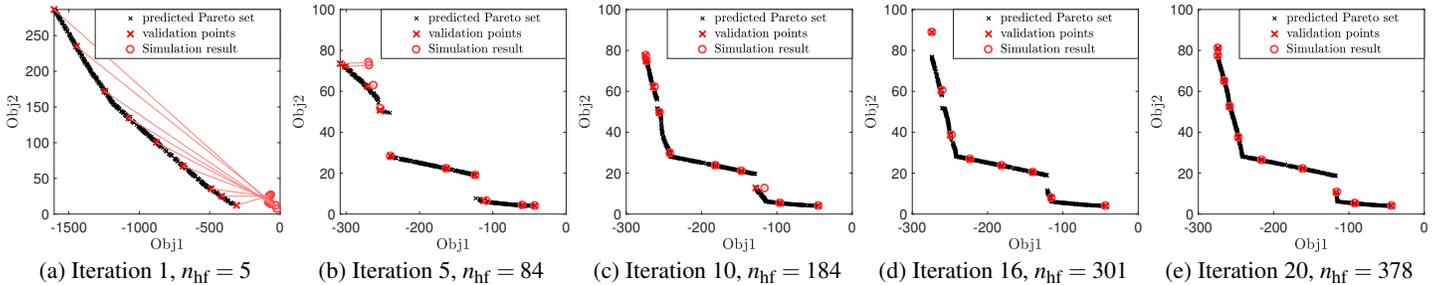


FIGURE 3: Predicted Pareto set and comparison between true and predicted results at the validation points at each iteration of MO-ASMO for the Osyczka and Kundu problem.

the next surrogate model construction.

The total number of function evaluations required to obtain an accurate Pareto set using the NSGA-II algorithm, $n_{hf} = 403,307$, is much larger than the number of original function evaluations required by the MO-ASMO algorithm to obtain approximately the same Pareto set ($n_{hf} = 378$, 20 iterations as shown in Fig. 3(e)). If we can sacrifice some solution accuracy, a nearly optimal solution can be obtained after ten iterations, with $n_{hf} = 184$ original function evaluations, as shown in Fig. 3(c).

4.2 Valley-shaped constraint problem

We developed a new test problem defined in Eqns. (6) and (7), illustrated in Fig. 4, to test more thoroughly the ability of the proposed framework to manage tightly-constrained feasible domains. This problem combines sinusoidal and exponential objec-

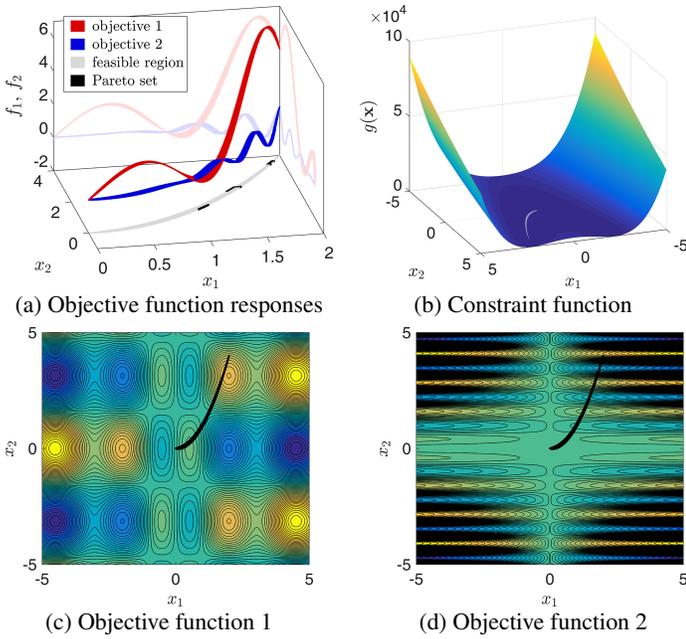


FIGURE 4: Illustration of valley shaped constraint problem

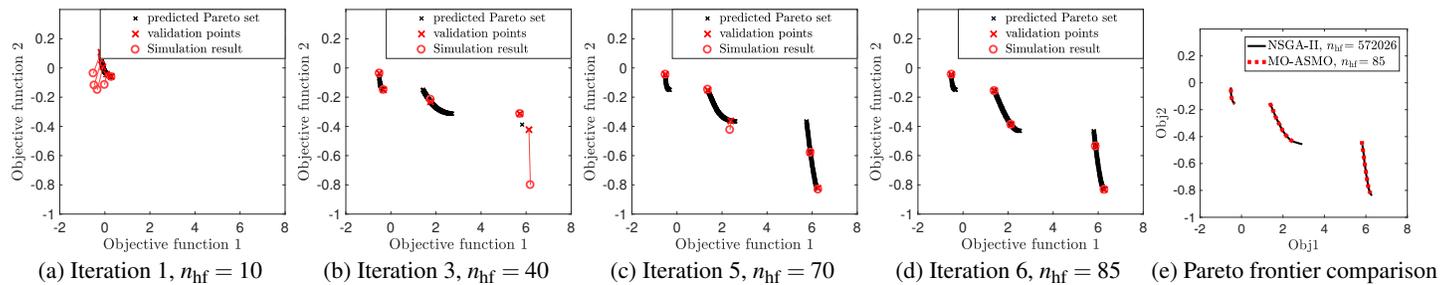


FIGURE 5: Predicted Pareto set and comparison between true and predicted results at the validation points at each iteration of MO-ASMO for the valley shaped constraint problem.

tive functions with a shifted Rosenbrock's valley function [40] as a constraint:

$$\min \mathbf{f}(\mathbf{x}) = [f_1, f_2] \quad \text{subject to } g \leq 0, \quad (6)$$

where

$$f_1(\mathbf{x}) = (3 \sin(2.5x_1) - 2x_1) (\cos(x_2) \exp(-0.001x_2^2)) \quad (7a)$$

$$f_2(\mathbf{x}) = 3(0.3|x_1|)^{19/25} ((1/8)x_2 \sin(5x_2)) \quad (7b)$$

$$g(\mathbf{x}) = (100(x_2 - x_1^2)^2 + (x_1 - 1)^2) - 1. \quad (7c)$$

The feasible region in the design space is shown in Figs. 4(a) and (b), and is plotted with light gray. For this constraint function, the feasible design space is long, narrow, non-convex, and nearly flat. Thus, it is not efficient to generate samples for training the surrogate model using standard sampling strategies. Also, the Pareto frontier of this problem is neither continuous in the objective function space, nor in the design space (see Figs. 5(e) and 4(a)), which increases design exploration difficulty. The proposed MO-ASMO algorithm demonstrates an advantage that stems from utilizing only feasible sample points.

Figure 5 illustrates the predicted Pareto set obtained from the surrogate model (small black symbols) and predicted and true values of chosen validation points (large symbols). As with the preceding figures, the symbol “ \times ” represents the values predicted by the surrogate model, while the symbol “ \circ ” corresponds to true values obtained using the original function. Figure 5(a)–(c) shows selected intermediate steps, and Fig. 5(d) shows the converged solution. Figure 5(e) presents a comparison of Pareto frontiers obtained by NSGA-II and MO-ASMO algorithms. NSGA-II requires $n_{hf} = 572,026$ original function evaluations to obtain an accurate Pareto set (population size of 2,000). In contrast, MO-ASMO requires only $n_{hf} = 85$ function evaluations (six iterations) to obtain approximately the same solution (Fig. 5(e)).

4.3 Suspension with viscoelastic damper problem

Another problem we developed for testing the MO-ASMO is an optimization of a quarter car suspension with viscoelastic

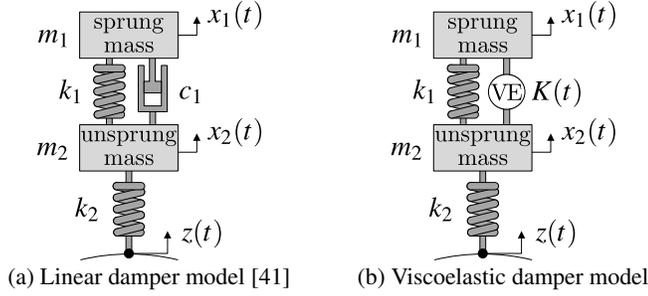


FIGURE 6: Quarter car suspension models with linear damper and viscoelastic damper.

damper, which is given in Fig. 6(b). The quarter car suspension model shown in Fig. 6(a) is a common simplification for analyzing and designing vehicle dynamics, while maintaining practical suspension design elements [42, 41, 43]. Sprung mass m_1 represents the vehicle body mass, while unsprung mass m_2 represents the mass of the suspension components and wheel set. A previous study demonstrated that viscoelasticity can improve vibration isolation performance throughout a wide range of frequencies [44, 45]. We replaced the linear damper c_1 between sprung and unsprung mass in Fig. 6(a) with a linear viscoelastic damper VE in Fig. 6(b). This VE damper was parameterized using single- and multi-mode Maxwell models [46]. In this problem, design variables are parameters defining the VE element and the spring constant k_1 . Other parameters, m_1 , m_2 , k_2 , and road input profile $z(t)$ of 100 meters are predefined after Ref. [47].

The design objectives of this problem are (1) enhancing the comfort metric by minimizing f_1 in Eqn. (8a) and (2) enhancing the handling metric by minimizing f_2 in Eqn. (8b). The first objective function f_1 uses peak acceleration amplitude as a proxy

metric for vehicle comfort, since International Standard Organization (ISO 2631) recommends acceptable vibration levels based on a correlation between acceleration values and their surveyed human comfort levels [48]. The second objective function f_2 uses a difference between displacements of the unsprung mass and the road profile, which represents a deflection of the tire, as a proxy for vehicle handling performance. Reducing this metric corresponds to less variance in tire contact force, helping to enhance cornering and traction performance [49].

$$f_1 = \max |\ddot{x}_1(t)| \quad (8a)$$

$$f_2 = \max |x_2(t) - z(t)| \quad (8b)$$

The linear viscoelastic behavior can be described by a time-dependent function, the relaxation kernel, $K(t)$. Using this relaxation kernel, the time-dependent force through a one-dimensional viscoelastic element, F_{ve} , can be represented as a convolution integral:

$$F_{ve}(t) = \int_{-\infty}^t K(t-t') \dot{x}(t') dt' = \int_0^\infty K(s) \dot{x}(t-s) ds \quad (9)$$

where \dot{x} is the deformation velocity of the viscoelastic element and $s = t - t'$. In general, the relaxation kernel $K(t)$ does not require a structure of linear springs and dashpots. Here, however, we utilize the multi-mode Maxwell model, which utilizes this specific structure, for our parameterization of the relaxation kernel function to simplify design representation. This model includes a series of parallel connections of multiple Maxwell elements, and a resultant expression of the relaxation kernel function for an M -mode Maxwell model can be defined by:

$$K(t) = \sum_{m=1}^M K_m \exp(-t/\lambda_m) \quad (10)$$

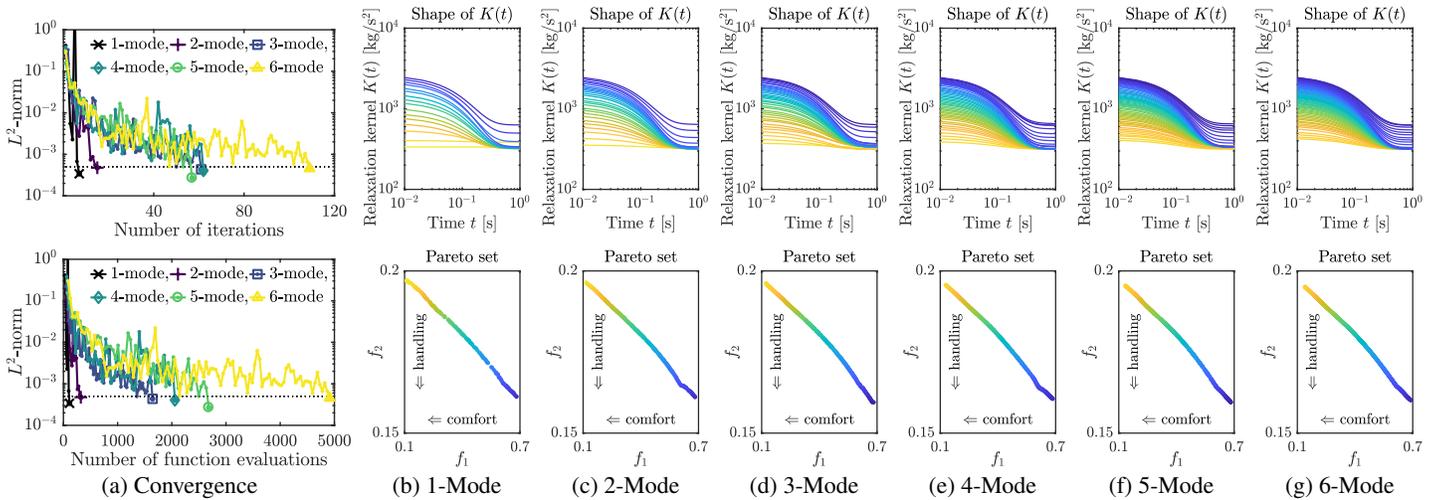


FIGURE 7: MO-ASMO convergence (a), viscoelastic kernel design (upper row of (b)–(g)), and corresponding Pareto frontier in objective function space (lower row) of the quarter car suspension optimization problem with viscoelastic damper using multi-mode Maxwell model.

where K_m are the Maxwell spring constants and λ_m are the relaxation times. A Maxwell element consists of a linear spring and a linear dashpot connected in series. Since this parameterization can have the same (or similar) relaxation kernel function with many different sets of parameters, we limited λ_m to have monotonically increasing values using the constraints $\lambda_{i+1} > \lambda_i$. Also, to provide a fair comparison between cases with a different number of modes M , the sum of K_m is capped by an upper bound; these values should not exceed the resultant Maxwell spring constant for the single mode case. This requirement is implemented using the constraint $\sum_m K_m < (\mathbf{ub} \text{ of } K_1)$. Although these constraints are in the form of a linear inequality, it is not possible to utilize linear constraints for this optimization problem because the quarter car suspension model is scaled in log space. These constraints, however, can be evaluated without running expensive simulation, but may shrink the feasible design region significantly, which is a challenge that this MO-ASMO algorithm was developed specifically for.

A simulation of this quarter suspension design problem is solved using the Euler predictor-corrector method with a numerical evaluation of the nested convolution integral given in Eqn. (9) for each time derivative function evaluation. This solution approach is a type of single-shooting method for dynamic system optimization, and with the numerous required convolution integral evaluations, the computational cost is very high compared to the other test problems. As a result, the MO-ASMO solution method is beneficial not only in terms of total function evaluations, but also computational time.

Convergence behaviors in terms of the L^2 -norm for six cases, ranging from single- to six-mode Maxwell models, are shown in Fig. 7(a) and the shape of the relaxation kernels and corresponding points in the Pareto frontier for each case are shown in (b)–(g). The number of design variables increases from 3 to 13 as the mode increases from 1 to 6. When using a lower-order Maxwell model, the required number of function evaluations is very low (i.e., $k = 7$, $n_{\text{hf}} = 112$ for 1-mode, $k = 15$, $n_{\text{hf}} = 321$ for 2-mode). However, a meaningful performance enhancement was not observed by increasing the order of

the Maxwell model, and this change in design representation increased the number of function evaluations ($k = 109$, $n_{\text{hf}} = 4911$ for six-mode) here. While the road profile has some spectral variance, the suspension is tested using a constant vehicle speed (60 mph). Lack of complexity in the test conditions may be important factors in the observed limited benefit from increasing the number of Maxwell modes. This problem is effective for testing and demonstrating the MO-ASMO algorithm, but further study is required to determine under what conditions a viscoelastic damper is beneficial. We expect that its frequency-dependent properties may prove to be valuable under more realistic test conditions. For example, varying the vehicle speed (and thus input frequency) according to a drive cycle such as the EPA standard cycle [50]. These additional studies are beyond the scope of this article, which is focused on understanding the new MO-ASMO framework.

5 ANALYSIS

5.1 Balance between exploration and exploitation

The purpose of adaptively refining surrogate models is to obtain an accurate set of Pareto-optimal designs. Two aspects are important in this adaptive procedure: 1) a thorough design space exploration to find regions that contains global optima, and 2) a precise local search to obtain an accurate solution. Having a good balance between these two objectives is a key for MO-ASMO algorithm effectiveness, but there is no universal principle to balance between exploration and exploitation for all cases.

Figure 8 compares error and Pareto set convergence for different balance ratios between exploitation and exploration, based on the Osyczka and Kundu problem. Here we define this balance as the ratio of new sample points placed according to exploitation objectives to the number of new sample points placed according to exploration objectives. Since the MO-ASMO framework is based on training samples generated using randomized algorithms such as Latin hypercube sampling, it is very difficult to characterize convergence using only a few tests. However, we see in Fig. 8(a) that error for all different ratios have a decrease-

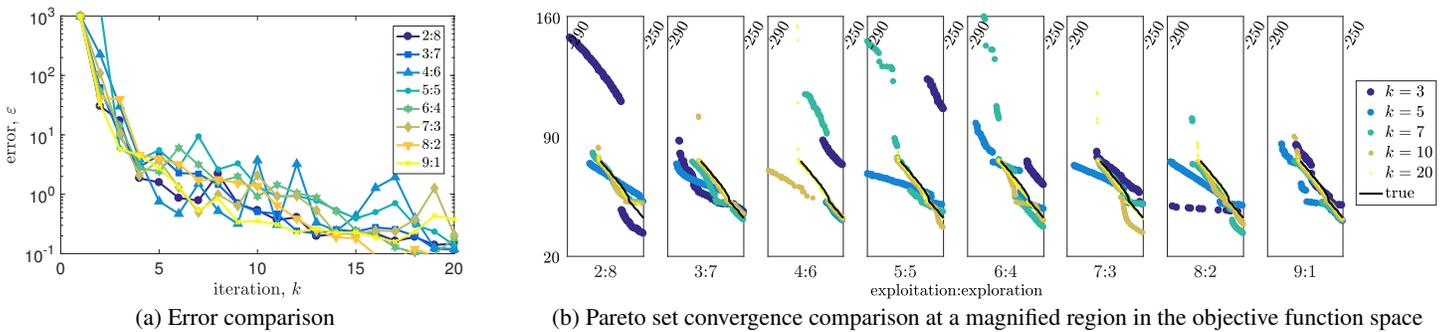


FIGURE 8: MO-ASMO convergence comparisons with different ratios of exploitation and exploration samples for the Osyczka and Kundu problem.

ing trend as iteration number increases. A few cases, such as 4:6 (4 samples for exploitation and 6 samples for exploration), exhibit oscillatory convergence characteristics. 5:5 and 6:3 cases exhibit relatively slow convergence, while 9:1, 8:2, 2:8, and 3:7 cases exhibit rapid and stable diminishing error values. This indicates that a proper balance between exploration and exploitation cannot be achieved simply through use of an equal number of samples for the two categories. More study on systematically balancing samples for exploration and exploitation is required (for multi-objective surrogate modeling methods in particular).

Figure 8(b) shows a magnified view of the Pareto set for selected iterations ($k = 3, 5, 7, 10,$ and 20) for each distinct balance ratio. At iteration 20 (yellow), most cases are converged to the true solution (black, obtained by NSGA-II). However, intermediate solutions are different across the distinct cases. As we see from Fig. 8(a), when sample points are allocated equally to exploitation and exploration, the MO-ASMO algorithm exhibits some convergence difficulties. One possible approach for resolving this problem is the observation that the number of sample points for these two categories may vary with respect to how well the surrogate model represents the trend of global responses, such as inflections, convexity, and monotonicity. Thus, ongoing work aims to formulate how the ratio and distribution of samples can be improved during surrogate model refinement iterations.

5.2 Hybrid DO/ASMO approach

Overall, the surrogate modeling optimization strategy is efficient compared to direct optimization. However, during the initial surrogate model refinement stages, predicted solution values can be highly inaccurate. Also, when exploring high-dimension design spaces, it can be difficult to explore extreme design points, which are often important as these are where anchor points usually are found. Finding Pareto frontier anchor points in the objective function space is not an easy task for MOEAs as they are based on heuristic strategies for expanding the region of solutions considered. To explore these extreme design points, direct optimization may be useful if the computational cost is not excessive. Thus, ongoing investigation is addressing how a hybrid direct and adaptive surrogate modeling-based optimization (hybrid DO/ASMO) may further accelerate Pareto set identification. When finding anchor points, we can use a single-objective optimization formulation for each objective function as a supplement. After obtaining these anchor points, we can use this information as a part of the initial population for the MOEA used within MO-ASMO.

6 CONCLUSION

A multiobjective adaptive surrogate modeling-based optimization (MO-ASMO) framework is introduced here with novel sampling strategies to address shortcomings of conventional multi-objective optimization methods, specifically for problems with

computationally expensive simulations and tightly constrained feasible domains. First, it is possible to sample designs in a tightly-constrained design space without violating constraints by running an optimization subproblem for each sample point. In the exploration stage, the objective function of this optimization subproblem minimizes relocation distance, while maximizing the distance between existing points and the new point. Second, generating samples to validate the predicted Pareto frontier requires even sampling across the hypersurface of the estimated Pareto set. Finally, a sampling strategy for exploitation needs to place new sample points near, but not on, the Pareto set hypersurface. A force-directed layout algorithm was used to place these sample points.

The MO-ASMO algorithm was demonstrated using the Osyczka and Kundu problem, and a new valley-shaped constraint test problem. These problems normally require many thousands of function evaluations to solve. However, the proposed MO-ASMO algorithms obtained the Pareto optimal solutions with orders of magnitude fewer evaluations. For example, the NSGA-II algorithm required 403,307 function evaluations to solve the Osyczka and Kundu problem, whereas the MO-ASMO algorithm required only 378 function evaluations. An even larger difference was observed with the valley-shaped constraint problem. NSGA-II required 572,026 function evaluations, whereas the MO-ASMO algorithm required just 85 original function evaluations. These ratios, however, cannot be used directly for assessing *computational cost*. Additional factors include calculation of new samples, including solution of optimization subproblems for sample placement, and MOEA solution with the surrogate model. If the original system model exhibits significant computational expense, it is anticipated that MO-ASMO will result in significant practical efficiency improvements. The MO-ASMO algorithm was also applied to a suspension design problem with viscoelastic damping as an example of a practical engineering design problem. Unlike other two test problems, this problem has a relatively expensive black-box simulation and exhibits complicated output responses. Having 3 to 13 design optimization variables for the one-mode to six-mode cases, respectively, convergence characteristics in terms of number of iterations and number of function evaluations were compared. Although we could not observe a meaningful performance improvement through increasing the number of Maxwell modes, MO-ASMO was demonstrated to be effective for a relatively expensive and complicated practical engineering design problem.

Ongoing work is addressing the application of this framework to more expensive and realistic engineering problems, including fluid flow problems, such as surface texture design problems for lubricated sliding contact [7, 27]. The new MO-ASMO algorithm has the potential to support solution of complex problems such as those with increased fidelity beyond what is possible currently. This may help explore new types of designs with previously unexploited mechanisms to achieve new perfor-

mance levels or functionality. In the sliding contact problem, this would support inclusion of additional physical phenomena in the model, while still enabling solution in a practical time period. Another possible implication of this algorithm is to simultaneous design problems of rheologically complex material and the corresponding system [51, 44, 45]. The algorithm presented here would be especially beneficial in this case as material functions design, e.g., viscoelasticity, need to be constrained to align them more closely with physically realizable materials. These complex physical constraints may significantly shrink the feasible region and the developed sampling methods may reduce computation effort.

The MO-ASMO algorithm may be enhanced further via a hybrid approach as described in Sec. 5.2. This approach may support more efficient design space exploration without relying solely on randomized sampling strategies. Finally, a deeper investigation of balancing exploration and exploitation should be conducted to enhance our understanding of how best to use the MO-ASMO algorithm for challenging problems, aiming to reduce the required number of function evaluations, while still improving the likelihood of identifying globally optimal Pareto sets.

7 ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation Grant no. CMMI-1463203.

REFERENCES

- [1] Deb, K., Pratap, A., Agarwal, S., and T., M., 2002. "A fast and elitist multiobjective genetic algorithm: NSGA-II". *IEEE Transactions on Evolutionary Computation*, **6**(2), pp. 182–197.
- [2] Holland, J. H., 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI.
- [3] Chankong, V., and Haimes, Y. Y., 1983. *Multiobjective Decision Making: Theory and Methodology*. North-Holland, Amsterdam.
- [4] Allison, J. T., 2013. "Plant-limited co-design of an energy-efficient counterbalanced robotic manipulator". *Journal of Mechanical Design*, **135**(10), p. 101003 (10pp).
- [5] Shirazi, A., Najafi, B., Aminyavari, M., Rinaldi, F., and Taylor, R. A., 2014. "Thermal-economic-environmental analysis and multi-objective optimization of an ice thermal energy storage system for gas turbine cycle inlet air cooling". *Energy*, **69**, pp. 212–226.
- [6] Kashfi, F., and Draper, J., 2013. "Multiobjective optimization of cost, performance and thermal reliability in 3DICs". In 2013 Euro-micro Conference on Digital System Design, IEEE, pp. 404–411.
- [7] Lee, Y. H., Schuh, J. K., Ewoldt, R. H., and Allison, J. T., 2017. "Enhancing full-film lubrication performance via arbitrary surface texture design". *Journal of Mechanical Design*, **139**(5), p. 053401.
- [8] Schuh, J. K., and Ewoldt, R. H., 2016. "Asymmetric surface textures decrease friction with Newtonian fluids in full film lubricated sliding contact". *Tribology International*, **97**, pp. 490–498.
- [9] Sawaragi, Y., Nakayama, H., and Tanino, T., 1985. *Theory of Multiobjective Optimization*. Academic Press, Orlando, FL.
- [10] Zitzler, E., and Thiele, L., 1999. "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach". *IEEE Transactions on Evolutionary Computation*, **3**(4), pp. 257–271.
- [11] Deb, K., and Jain, H., 2014. "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: solving problems with box constraints". *IEEE Transactions on Evolutionary Computation*, **18**(4), pp. 577–601.
- [12] Eichfelder, G., 2008. *Adaptive Scalarization Methods in Multiobjective Optimization*. Springer-Verlag, Berlin/Heidelberg.
- [13] Ehrgott, M., 2005. *Multicriteria Optimization*. Springer-Verlag, Berlin/Heidelberg.
- [14] Saaty, T., and Gass, S., 1954. "Parametric objective function (Part 1)". *Journal of the Operations Research Society of America*, **2**(3), pp. 316–319.
- [15] Haimes, Y. Y., Lasdon, L. S., and Wismer, D. A., 1971. "On a bi-criterion formulation of the problems of integrated system identification and system optimization". *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-1**(3), pp. 296–297.
- [16] Pascoletti, A., and Serafini, P., 1984. "Scalarizing vector optimization problems". *Journal of Optimization Theory and Applications*, **42**(4), pp. 499–524.
- [17] Jones, D. R., Schonlau, M., and Welch, W. J., 1998. "Efficient Global Optimization of Expensive Black-Box Functions". *Journal of Global Optimization*, **13**(4), pp. 455–492.
- [18] Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P. K., 2005. "Surrogate-based analysis and optimization". *Progress in Aerospace Sciences*, **41**(1), pp. 1–28.
- [19] Razavi, S., Tolson, B. A., and Burn, D. H., 2012. "Review of surrogate modeling in water resources". *Water Resources Research*, **47**, p. W07401 (pp32).
- [20] Box, G. E. P., and Wilson, K. B., 1951. "On the Experimental Attainment of Optimum Conditions". *Journal of the Royal Statistical Society, Series B*, **13**(1), pp. 1–45.
- [21] Matheron, G., 1989. *Estimating and Choosing*. Springer, Berlin.
- [22] Cortes, C., and Vapnik, V., 1995. "Support-Vector Networks". *Machine Learning*, **20**(3), pp. 273–297.
- [23] Powell, M. J. D., 1987. "Radial Basis Function for Multivariable Interpolation: A Review". In *Algorithms for Approximation*. Clarendon Press, Oxford.
- [24] Dennis, J. E., and Torczon, V., 1997. "Managing approximation models in optimization". In *Multidisciplinary Design Optimization: State of the Art*, N. M. Alexandrov and M. Y. Hussaini, eds., Vol. 80. Society for Industrial and Applied Mathematics, Philadelphia, PA, pp. 330–347.
- [25] Osio, I. G., and Amon, C. H., 1996. "An engineering design methodology with multistage Bayesian surrogates and optimal sampling". *Research in Engineering Design*, **8**(4), pp. 189–206.
- [26] Schonlau, M., Welch, W. J., and Jones, D. R., 1998. "Global versus local search in constrained optimization of computer models". In *New Developments and Applications in Experimental Design*, N. Flournoy, W. F. Rosenberger, and W. K. Wong, eds. Institute of Mathematical Statistics, Hayward, CA, pp. 11–25.

- [27] Rao, L. G., Schuh, J. K., Ewoldt, R. H., and Allison, J. T., 2015. “On Using Adaptive Surrogate Modeling in Design for Efficient Fluid Power”. In ASME 2015 IDETC/CIE, no. DETC2015-46832 in Volume 2B: 41st Design Automation Conference, ASME, p. V02BT03A024 (pp11).
- [28] Wang, G. G., and Shan, S., 2007. “Review of metamodeling techniques in support of engineering design optimization”. *Journal of Mechanical Design*, **129**(4), pp. 370–380.
- [29] Simpson, T. W., Toropov, V., Balabanov, V., and Viana, F. A. C., 2008. “Design and Analysis of Computer Experiments in Multidisciplinary Design Optimization: A Review of How Far We Have Come - or Not”. In 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA.
- [30] Forrester, A. I. J., and Keane, A. J., 2009. “Recent Advances in Surrogate-Based Optimization”. *Progress in Aerospace Sciences*, **45**(1), pp. 50–79.
- [31] Eldred, M. S., and Dunlavy, D. M., 2006. “Formulations for surrogate-based optimization with data fit, multifidelity, and reduced-order models”. In Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, no. AIAA 2006-2117, American Institute of Aeronautics and Astronautics, p. (20pp).
- [32] Hussein, R., and Deb, K., 2016. “A generative Kriging surrogate model for constrained and unconstrained multi-objective optimization”. In Proceedings of the Genetic and Evolutionary Computation Conference 2016, Association for Computing Machinery, pp. 573–580.
- [33] Crombecq, K., Gorissen, D., Deschrijver, D., and Dhaene, T., 2011. “A novel hybrid sequential design strategy for global surrogate modeling of computer experiments”. *SIAM Journal on Scientific Computing*, **33**(4), pp. 1948–1974.
- [34] Jiang, P., Shu, L., Zhou, Q., Zhou, H., Shao, X., and Xu, J., 2015. “A novel sequential exploration-exploitation sampling strategy for global metamodeling”. In 17th IFAC Symposium on System Identification, Vol. 48, International Federation of Automatic Control.
- [35] Ajdari, A., and Mahlooji, H., 2014. “An adaptive exploration-exploitation algorithm for constructing metamodels in random simulation using a novel sequential experimental design”. *Communications in Statistics - Simulation and Computation*, **43**(5), pp. 947–968.
- [36] Eades, P., 1984. “A heuristic for graph drawing”. *Congressus Numerantium*, **42**, pp. 149–160.
- [37] Lee, Y. H., 2017. Multiobjective Adaptive Surrogate Modeling-based Optimization Toolbox I, to appear. <https://github.com/yonghoonlee/MO-ASMO-I>.
- [38] Iman, R. L., Campbell, J. E., and Helton, J. C., 1981. “An approach to sensitivity analysis of computer models: Part I - Introduction, input, variable selection and preliminary variable assessment”. *Journal of Quality Technology*, **13**(3), pp. 174–183.
- [39] Osyczka, A., and Kundu, S., 1995. “A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm”. *Structural Optimization*, **10**(2), pp. 94–99.
- [40] Rosenbrock, H. H., 1960. “An automatic method for finding the greatest or least value of a function”. *The Computer Journal*, **3**(3), pp. 175–184.
- [41] Allison, J. T., 2008. “Optimal partitioning and coordination decisions in decomposition-based design optimization”. Ph.D. Dissertation, University of Michigan, Ann Arbor, MI, May.
- [42] Jazar, R. N., 2017. *Vehicle Dynamics: Theory and Application, 3rd Ed.* Springer, New York, NY.
- [43] Allison, J. T., Guo, T., and Han, Z., 2014. “Co-Design of an Active Suspension Using Simultaneous Dynamic Optimization”. *Journal of Mechanical Design*, **136**(8).
- [44] Rao, L. G., and Allison, J. T., 2015. “Generalized viscoelastic material design with integro-differential equations and direct optimal control”. In ASME 2015 IDETC/CIE, no. DETC2015-46768 in Volume 2B: 41st Design Automation Conference, ASME, p. V02BT03A008 (11pp).
- [45] Corman, R. E., Rao, L. G., Bharadwaj, N. A., Allison, J. T., and Ewoldt, R. H., 2016. “Setting material function design targets for linear viscoelastic materials and structures”. *Journal of Mechanical Design*, **138**(5), p. 051402 (12pp).
- [46] Bird, R. B., Armstrong, R. C., and Hassager, O., 1987. *Dynamics of Polymeric Liquids: Volume 1 Fluid Mechanics, 2nd Ed.* Wiley, New York, NY.
- [47] Allison, J. T., 2012. Simulation and Animation of a Quarter-Car Automotive Suspension Model. <https://www.mathworks.com/matlabcentral/fileexchange/35478>.
- [48] Yuen, T. J., Rahizar, R., Azman, Z. A. M., Anuar, A., and Afandi, D., 2013. “Design Optimization of Full Vehicle Suspension Based on Ride and Handling Performance”. In Proceedings of the FISITA 2012 World Automotive Congress, Volume 7: Vehicle Design and Testing (I), Springer, pp. 75–86.
- [49] Lu, J., and DePoyster, M., 2002. “Multiobjective Optimal Suspension Control to Achieve Integrated Ride and Handling Performance”. *IEEE Transaction on Control Systems Technology*, **10**(6), pp. 807–821.
- [50] EPA Urban Dynamometer Driving Schedule (UDDS). <https://www.epa.gov/emission-standards-reference-guide/epa-urban-dynamometer-driving-schedule-udds>.
- [51] Bharadwaj, N. A., Allison, J. T., and Ewoldt, R. H., 2013. “Early-Stage Design of Rheologically Complex Materials via Material Function Design Targets”. In ASME 2013 IDETC/CIE, no. DETC2013-13462 in Volume 3A: 39th Design Automation Conference, ASME, p. V03AT03A058 (pp10).