

Niching Surrogate-Assisted Optimization for Simulation-Based Design Optimization of Cylinder Head Water Jacket

Ali Ahrari, Julian Blank, Kalyanmoy Deb, Xianren Li

COIN Report Number 2019009

Abstract

Many engineering design problems are associated with computationally expensive simulations for design evaluation, which makes the optimization process a time-consuming effort. In such problems, each candidate design should be selected carefully, even though it means extra algorithmic complexity. This study develops a niching-based surrogate-assisted evolutionary algorithm that aims at handling both single-objective and multi-objective computationally expensive problems. A trust-region concept in the optimization context is proposed to control the evaluation error. At the same time, maximizing the information about specific regions of the search space is pursued by proper selection of new candidate solutions. The proposed method is evaluated and compared to a recently developed surrogate-assisted evolutionary algorithm on multi-objective test problems. Thereafter, a case study involving a multi-objective design optimization of the cylinder head water jacket of a vehicle engine is presented and discussed.

Keywords: Multi-objective Optimization, Metamodeling, Real-World Optimization, Simulation-based Optimization

1 Introduction

Evolutionary algorithms are robust optimization tools that can handle different challenges in practical optimization problems such as uncertainty (Sahinidis 2004), multimodality (Goldberg and Richardson 1987), discontinuity (Mongeau 2009), mixed-variable (Cao, Jiang, and Wu 2000), conflicting objectives (Deb 2001) and black-box nature (Jones, Schonlau, and Welch 1998a). These algorithms do not require any simplification of the actual problem towards the solution process, a matter which is often required by the classical point-based methods (Bartz-Beielstein et al. 2010). This flexibility usually comes at the cost of relatively high number of solution evaluations. For example, the evaluation budget in BBOB2009 (Hansen et al. 2010) test suite was set to $10^6 D$, in which D is the number of variables for the defined problems to solve the problems close to their global optima. For computationally expensive problems, each solution evaluation may take a few hours to a few days, and thus, such a large evalu-

ation budget is impractical. Therefore, even if the evaluation process can be parallelized, the calculation time is the bottleneck of the optimization process.

In many applications, including optimal mechanical design problems, the evaluation budget is often limited to a few hundred or less. The goal in such problems is to make the most out of the available evaluation budget, which is fulfilling the design objective(s) as much as possible. A limited evaluation budget clarifies the importance of a careful selection of candidate solutions for evaluation, which motivates the use of surrogate models, also known as meta-models or response surfaces. A surrogate model is constructed from a few high-fidelity solutions. Because a surrogate model usually relies on assumptions on the actual model, a variety of techniques have been suggested in the context of optimization: Radial Basis Functions (Zapotecas Martínez and Coello Coello 2013), Support Vector Machine (Zapotecas Martínez and Coello Coello 2010), Kriging (Zhang et al. 2015) or Moving Least Squares (Filomeno Coelho, Lebon, and Bouillard 2011).

This study develops a surrogate-assisted evolutionary algorithm for single and multi-objective optimization problems. This method, called Niching-based Surrogate-Assisted Evolutionary Algorithm (NSA-EA), selects new candidate solutions following two goals. First, these solutions should optimize the predicted objective values, and second, they should maximize the information collected about specific regions of the problem landscape.

This paper is organized as follows: Section 2 provides an overview of the literature on surrogate-assisted optimization for single and multi-objective problems. Afterwards, the methodology and implementation details are elaborated in Section 3. Section 4 studies the effect of each component of NSA-EA numerically. A comparison with a recent surrogate-assisted multi-objective optimization method is performed in Section 5. A practical case on multi-objective design optimization of the cylinder head water jack of the vehicle engine is studied in Section 6. Finally, conclusions are made, and future research is discussed in Section 7.

2 Past Studies

Metamodel-assisted algorithms follow a general scheme which will be described in the following. First, the initial solutions are sampled uniformly in the search space, and their actual values are evaluated. The optimal number of initial high-fidelity solutions depends on the problem complexity (unknown) and the available budget (known). Second, a metamodel is built using the initial high-fidelity solutions. In principle, any surrogate model can be used, and a fine-tuning of metamodel hyper-parameters makes sense. Afterwards, an optimization algorithm searches for non-dominated solutions by using only predictions of the metamodel model, an internal optimization which is referred to as *virtual optimization*. Out of these solutions, some are selected for high-fidelity evaluation. If the termination criteria are not satisfied, the selected points are utilized to update the metamodel, and the optimization process continues.

A general review of metamodel-based optimization in engineering design can be found in (Wang and Shan 2007). A study that aims to solve problems with only 100 and 250 solution evaluations calls for multi-objective problems was made by Knowles (Knowles, Corne, and Reynolds 2009). The proposed

algorithm ParEGO (Jones, Schonlau, and Welch 1998b) is used to solve single-objective problems composed by Tchebycheff function (Steuer and Choo 1983). The focus of this study is the limited evaluation budget and the evaluation in noisy optimization problems. Results showed that this method outperforms NSGAI on DTLZ and some other test problems.

Another surrogate-assisted multi-objective algorithm, Gap Optimized Multi-objective Optimization using Response Surfaces (GOMORS) (Akhtar and Shoemaker 2016), a parallel algorithm for multi-objective optimization, is compared with the ParEGO results on the ZDT, LZJ test and a groundwater remediation real-world problem with 200 solution evaluations. GOMORS uses either NSGA-II (Deb et al. 2002) or MOEA/D (Zhang and Li 2007) to find promising points on the metamodel. GOMORS was found to outperform ParEGO, especially on higher-dimensional problems.

Recently, Surrogate Optimization of Computationally Expensive Multi-Objective Problems (SOCEMO) (Mueller 2017) has been proposed, which focuses on solving large-scale multi-objective problems. The initial population is created by using Latin Hypercube Sampling and RBF with cubic functions is employed for the metamodeling. Different strategies to select new solutions for high-fidelity evaluation were employed to balance exploitation and exploration. The algorithm was tested on a large set of benchmark problems and two engineering application problems and showed to be much more efficient than MOGA. Also, a single-objective surrogate-based algorithm, Mixed-Integer Surrogate Optimization (MISO) (Mueller 2016) was proposed to optimize mixed-integer problems with expensive solution evaluations. The algorithm combines different sampling strategies and local search to obtain high-accuracy solutions. Also, real-valued problems can be solved using this algorithm.

Deb et al. (Deb et al. 2017) systematically analyzed how metamodels can be used for optimization. The proposed taxonomy shows the possibilities for using metamodels for either objective or constraint functions. Different combinations of modeling strategies to predict all objective and constraint function values with one model, create one model for each function and constraint or combine these two strategies in specific ways. Also, a classification of existing algorithms into the different categories is performed. Moreover, results for multi-objective optimization using Kriging as a metamodel are provided in (Hussein and Deb 2016).

3 Proposed Niching-based Surrogate-Assisted Optimization

In the following, the concept of surrogate-assisted optimization is explained. Afterwards, we propose the niching-based surrogate-assisted evolutionary algorithm (NSA-EA) and provide a detailed description of the method.

3.1 Proximity and Trust Region

A well-known concept in optimization is the exploration-exploitation trade-off (Karimzadehgan and Zhai 2010). The likelihood of missing the global optimum increases if the search is not explorative enough. On the other hand, when focusing more on exploration, no time might be left to exploit the gained information

about the design space. Only a reasonable trade-off will produce satisfactory results. Evolutionary algorithms generally focus on exploration in the beginning and gradually start to exploit available information. For example, initialization of the solutions is an entirely explorative process since the objective values of the previously initialized solutions are not considered.

For metamodel-assisted optimization, each high-fidelity solution provides some new information on the landscape of the objective function(s). If this solution is far from all the existing high-fidelity solutions, the new information on the landscape will be significant. This information can be used to improve the accuracy of the existing metamodel on-the-fly, resulting in more accurate predictions in subsequent generations. Consequently, it is possible to improve exploration by emphasizing the diversity of the newly introduced solutions for high-fidelity evaluation.

To take the diversity of new solutions into account, we define the concept of proximity measure. It defines an infeasible spherical region of radius R_{prox} around each existing solution so that subsequent solutions are not selected close to existing ones. A greater R_{prox} enforces more exploration, which can be beneficial in the early stages of the optimization.

The reliability of the predicted values strongly depends on the accuracy of the metamodel. Predictions close to observations are more accurate than predictions far from them. Therefore, we propose the concept of the trust region to control the amount of acceptable uncertainty in the search process. The trust region defines spherical regions of radius R_{trust} around each solution. Subsequent solutions for high-fidelity evaluation must be selected in the trust region. If R_{trust} is large, points with high predicted fitness are probably those with high uncertainty. This can mislead the search to regions with high metamodel uncertainty. The value of R_{trust} can control the exploitation of the optimization process: a small R_{trust} forces the algorithm to select new solutions close to existing solutions, where the prediction error is small.

Altogether, combining both concepts, a search region is defined where all infill points should be. On one hand, they should be smaller than R_{trust} to satisfy the current amount of allowed uncertainty. On the other hand, they should be larger than R_{prox} to improve the diversity of infill solutions. $R_{trust} \geq R_{prox} > 0$ should gradually decrease to allow for a gradual transient from explorative to exploitative search. For simplicity, we set $R_{trust} \propto R_{prox}$.

Figure 1 shows different search regions during an optimization run and visualizes the gradual movements from exploration to exploitation. The proximate region is shown in red and the trust region green. In the initial epoch, exploration is emphasized by having a very large R_{trust} and also a large R_{prox} value. In intermediate epochs, both radii have decreased to improve exploitation. During the final epochs, both regions are relatively small to maximize exploitation.

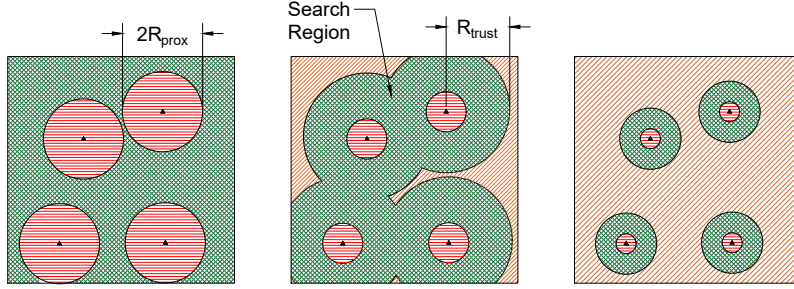


Figure 1: Proximal and trust region.

As shown in the figure above, the proximate and trust region shrink over time. The shrinking is based on the evaluation budget and the number of already evaluated solutions (excluding the initial sample solutions). Both radii are decreased using an exponential function. To update the value of R_{trust} and R_{prox} , first, the reduction rate is calculated as follow:

$$a = \left(1 - \frac{\maxEvals - n_{eval}}{\maxEvals - n_{init}} \right), \quad (1)$$

where \maxEvals is the overall solution evaluation budget, n_{eval} the number of solution evaluations so far and n_{init} the number of points used for the initial population. At each epoch, R_{prox} is calculated as follow:

$$R_{prox} = \max(r_{init} \cdot a^{\tau_R} \epsilon_{prox}), \quad (2)$$

where r_{init} is the initial minimum distance between two solutions after initial sampling and τ_R is the proximate reduction exponent. $\epsilon_{prox} > 0$ is the lower bound for a distance between two solutions. R_{trust} is then defined proportionally to R_{prox} :

$$R_{trust} = \gamma \cdot R_{prox}. \quad (3)$$

The reduction procedure introduces three new parameters: γ as a proportion between proximate and trust region, τ_R as reduction exponent and ϵ_{prox} as a tolerance value. ϵ_{prox} is the lower bound for R_{prox} which can not be fallen below even though the exponential decreasing function would say so. Therefore, it can also be interpreted as the overall precision of the algorithm. For our experiments $\gamma = 4$ and $\epsilon_{prox} = 10^{-6} \cdot \|X^{(U)} - X^{(L)}\|_2$ which is 10^{-6} times the euclidean distance of the upper minus the lower bound for each variable. A parameter study is performed in Section 4.1 to find a good value for τ_R .

These principles can be embedded into any optimization algorithm by adding constraints that declares solutions as infeasible when they are in the proximal or outside of the trust region. Therefore, it is independent of the metamodel itself and only part of the optimization process.

3.2 Selection Error Probability

The goodness of approximation models is crucial for the performance of surrogate-assisted algorithms. Inaccurate predictions lead the optimization to search re-

gions that are, in fact, not as good as expected. Often, several metamodel types are considered, and the best one out of these needs to be selected. Also, for metamodels that have control parameters, the best parameter setting should be chosen to keep the prediction error as small as possible. To do this, a metric to quantify the goodness of a metamodel should be defined. An intuitive metric for metamodel goodness is Mean Squared Error (MSE):

$$e_{mse}(x) = \frac{1}{n} \sum_{i=1}^n \left(\hat{f}(x_i) - f(x_i) \right)^2, \quad (4)$$

where n is the number of available solutions, $\hat{f}(x_i)$ is the prediction and $f(x_i)$ the real value. It sums up the square error for each prediction and averages it.

Note that building a metamodel can have different intentions. It can be used to get an idea of a function by interpolating between existing points. However, in the case of optimization, this is not even necessary. It is a sufficient condition if the metamodel preserves the order of solutions since many algorithms use a rank-based selection. For multi-objective problems, the metamodel is used by an optimization algorithm to find non-dominated solutions. Optimization algorithms make pairwise comparisons to determine if a solution is dominating another. If the outcome is equivalent to any comparison made on the high-fidelity function, the metamodel is meant to make no error despite having a deviation between prediction and exact values.

Figure 2 illustrates an exemplary scenario for solutions comparisons. The true function values are shown in blue and the predictions in red. The absolute metamodel error is the integral of the difference between both functions. The optimization algorithm aims to minimize $f(x)$ (True) by using $\hat{f}(x)$ (Prediction). The optimization algorithm will make pairwise comparisons. For instance, if point x_1 and x_2 are compared, then $f(x_1) > f(x_2)$ and $\hat{f}(x_1) > \hat{f}(x_2)$. The metamodel does predict the domination relation correctly. Contrarily, when x_1 and x_3 are compared, $f(x_1) > f(x_3)$, but $\hat{f}(x_1) < \hat{f}(x_3)$. The metamodel prediction leads to an incorrect comparison result. Considering minimizing using this metamodel the algorithm will favor solution x_1 over x_3 which is indeed wrong because $f(x_1) > f(x_3)$.

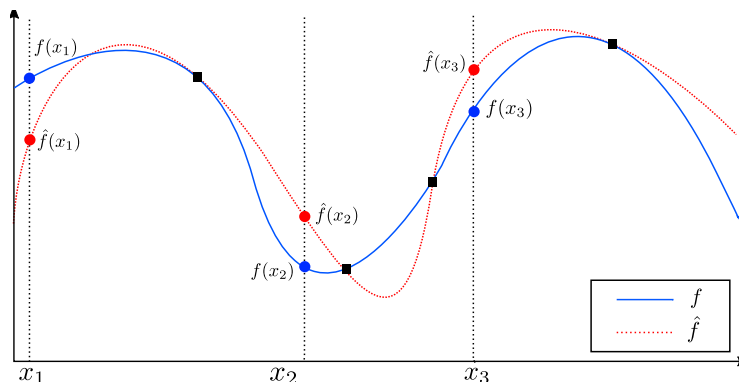


Figure 2: Selection Error Probability: Pairwise comparison between high-fidelity and prediction values

To define a metric for metamodel goodness that takes the discussed criteria into account, we propose the Selection Error Probability (SEP). It is based on the concept of ranking under uncertainty investigated in (Hansen et al. 2009). SEP considers all possible pairwise combinations and sums up the number of times the prediction is misleading:

$$e_{sep}(x) = \frac{1}{0.5 \cdot n \cdot (n - 1)} \sum_{i=1}^n \sum_{j=i+1}^n q(x_i, x_j), \quad (5)$$

where

$$q(x_i, x_j) = \begin{cases} 1, & \text{if } (f(x_i) - f(x_j)) \cdot (\hat{f}(x_i) - \hat{f}(x_j)) < 0, \\ 0, & \text{otherwise.} \end{cases}$$

The two sums iterate over pairwise comparisons of the existing solutions. The number of possible pairs with n solutions is $0.5 \cdot n \cdot (n - 1)$. For each pair $q(x_i, x_j)$ returns 1 if the comparison of two solutions using their predictions is not the same as using their true values. It will return 0 if the comparison is correct.

3.3 Niching Surrogate-Assisted Evolutionary Algorithm (NSA-EA)

This section presents Niching Surrogate-Assisted Evolutionary Algorithm (NSA-EA), which aims at solving single and multi-objective problems when the evaluation budget is highly limited. It utilizes the trust and proximity concept during the optimization and the SEP metric to select the best metamodel. The pseudo-code of NSA-EA is presented in algorithm 1.

Selection of initial high-fidelity solutions is a crucial phase because the sampling strategy directly affects the goodness of the initial metamodel. Since the metamodel must be fitted using existing solutions, it is important to cover the search space as good as possible. A space-filling approach is used in this study in order to generate initial solutions. A distance variable r is set to the maximum distance that can exist in the search space $r = \|X^{(U)} - X^{(L)}\|_2$ between two points. The initial point is generated randomly. Then, iteratively new points are sampled and accepted if they maintain a distance of at least r from all existing points. If n successive attempts are rejected, r is slightly reduced. This process continues until all initial points are generated. Finally, all sampled points have at least euclidean distance r to each other. We use this radius for the reduction later on, and save the resulting minimum distance as r_{init} after sampling all initial points.

Afterwards, the proximate and trust region are formed around the sampled high-fidelity solutions, and the solution evaluation counter n_{eval} is updated. Then as long as $n_{eval} \leq maxEvals$, a new virtual optimization is performed. Each metamodel update, virtual optimization, and select of new infill solutions constitutes an *epoch*.

At the beginning of each epoch, the best metamodel parameters should be found for each objective. NSA-EA uses DACE-Kriging (DK) as the metamodel. We use the DACEFIT module (Nielsen, Lophaven, and Søndergaard 2002) provided by the Matlab Surrogate Model Optimization Toolbox. The module allows setting the regression type, correlation function and an initial θ . The regression

Algorithm 1: Niching Surrogate-Assisted Evolutionary Algorithm (NSA-EA)

Result: Non-dominated Solutions for a given problem defined by evaluation function f

Input : $MaxEvals$ (Solution Evaluation Budget), $iniFr$ (Initial Fraction of Solutions), N_{epoch} (Points per Epoch), τ_R (Reduction Exponent), γ (Trust Region Ratio), ϵ_{prox} (Reduction Tolerance)

Output: Optimized X , F

```

/* Initialize Parameters and sample initial points          */
1  $n_{eval} \leftarrow 0$ 
2  $n_{init} \leftarrow MaxEvals \cdot iniFr$ 
3  $X, r_{init} \leftarrow \text{sample}(n_{init})$ 
4  $R_{prox} \leftarrow r_{init}$ 
5  $R_{trust} \leftarrow \gamma \cdot R_{prox}$ 
6  $F \leftarrow f(X)$ 
7  $n_{eval} \leftarrow n_{eval} + n_{init}$ 
8 while  $n_{eval} \leq maxEvals$  do
    /* Begin Epoch                                          */
    /* Find best model parameter setting for each objective */
    /*
    9  $P \leftarrow \text{kriging\_parameter\_settings}()$ 
    10  $M \leftarrow [M_0, \dots, M_{n_{obj}}]$ 
    11 for  $k = 0; k < n_{obj}; k = k + 1$  do
    12      $e_{min} \leftarrow 1.0$ 
    13     foreach  $p \in P$  do
    14          $m_k \leftarrow \text{build\_model}(X, F_k)$ 
    15          $e_{sep} \leftarrow \text{calc\_error\_by\_crossvalidation}(m_k, X, F_k)$ 
    16         if  $e_{sep} < e_{min}$  then
    17              $e_{min} \leftarrow e_{sep}$ 
    18              $M_k \leftarrow m_k$ 
    19         end
    20     end
    21 end

    /* Optimize on the surrogate model and select  $N_{epoch}$  */
    /* solutions                                          */
    22 if single-objective then
    23      $\hat{X}, \hat{F} \leftarrow \text{optimize\_and\_select}(M, R_{prox}, R_{trust}, \text{IPOP-CMA-ES},$ 
    24          $N_{epoch})$ 
    25 else
    26      $\hat{X}, \hat{F} \leftarrow \text{optimize\_and\_select}(M, R_{prox}, R_{trust}, \text{NSGAIII}, N_{epoch})$ 
    27  $X \leftarrow X \cup \hat{X}$ 
    28  $F \leftarrow F \cup f(\hat{X})$ 
    29  $n_{eval} \leftarrow n_{eval} + N_{epoch}$ 

    /* Update the proximity and trust region              */
    30  $R_{prox} = \max(r_{init} \cdot (1 - \frac{\maxEvals - n_{eval}}{\maxEvals - n_{init}})^{\tau_R}, \epsilon_{prox})$ 
    31  $R_{trust} = \gamma \cdot R_{prox}$ 

    /* End Epoch                                          */
32 end
return  $\text{non\_dominated}(X, F)$ 

```

can either be constant, linear or quadratic. For the correlation, an exponential, generalized exponential, Gaussian, linear, spherical or cubic spline function can be used. The theta bounds were set to $[10^{-6}, 10^2]$ for which the candidate values are set to $[10^{-5}, 10^{-4}, \dots, 10^0]$.

Whenever a metamodel should be updated, we execute an exhaustive search over a candidate set of these hyper-parameters. For the first epoch, this candidate set includes all possible parameter combinations. At each epoch, a fraction of worst candidate metamodel settings is discarded such that at the last epoch, only two parameter settings are tested.

The goodness of a metamodel is evaluated using stratified k-fold cross-validation, according to which all existing solutions are used as training and validation sets. Each time one solution is excluded from the training set, and its value is predicted using the trained metamodel. This process continues until the predicted values of all solutions are calculated. Having the actual and predicted values of each solution, SEP can easily be calculated using Equation 5. The best-found parameter setting is selected using SEP as a metric. This setting is then used to train the metamodel using all the available solutions.

After selecting and training the best metamodel, an evolutionary algorithm is employed to perform virtual optimization. In principle, any algorithm made for black-box optimization can be used, since no assumption about the function to optimize can be made. Depending on the number of objectives, we use two different methods for optimization:

- For single objective problems, we use the Covariance Matrix Adaptation Evolution Strategy (IPOP-CMA-ES) with restarts (Auger and Hansen 2005), a successful method for continuous single-objective optimization. Since new solutions undergo high-fidelity evaluation at the end of each epoch, IPOP-CMA-ES is executed multiple times consecutively, generating one new point each time. This new solution does not affect the metamodel, but it changes the optimization problem landscape since each new point modifies the feasible region defined by R_{prox} and R_{trust} . At the same time, R_{prox} and R_{trust} are updated whenever a new solution is generated.
- For multi-objective problems, we use the recent unified version of Non-Dominated Sorting Genetic Algorithm III (NSGA-III) (Seada and Deb 2015). The solutions obtained by optimization are candidate solutions, from which solutions are selected for high-fidelity evaluation. As a constraint, each solution must be in R_{trust} , but not be in R_{prox} . The selection procedure is similar to NSGA-III but uses a clearing strategy. Existing points are assigned to reference lines. The solution assigned to the least crowded reference line is selected for high-fidelity evaluation, and all points within proximity region of this solution are cleared. If another point must be selected but all points have been cleared, we rerun NSGA-III after applying the effect of recent solution on the feasible region. For practical problems, the user might manually select new points from candidate solutions.

4 Descriptive Experiments

We design and perform a few descriptive experiments to demonstrate the effect of different components of NSA-EA. For this purpose, we select three test problems:

- Rastrigin function, a highly multimodal function with a symmetric bowl-shaped global structure. The local minimum gradually gets better when approaching the global minima.

$$f(\mathbf{x}) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i)), \quad -5 \leq x_i \leq 5.$$

- Schwefel function, a multimodal function in which local minima lie close to the corners of search space. Approaching any corner means going away from the rest of good local minima.

$$f(\mathbf{x}) = 418.9829D - \sum_{i=1}^D (x_i \sin \sqrt{|x_i|}), \quad -500 \leq x_i \leq 500.$$

- ZDT3 function, a two-objective test problem with disjoint Pareto front for the mathematical definition). The first objective function is unimodal, while the second objective function is multimodal.

$$\begin{aligned} f_1(\mathbf{x}) &= x_1, \\ f_2(\mathbf{x}) &= g(\mathbf{x}) h(f_1(\mathbf{x}), g(\mathbf{x})), \\ g(\mathbf{x}) &= 1 + \frac{9}{D-1} \sum_{i=2}^D x_i, \\ h(f_1(\mathbf{x}), g(\mathbf{x})) &= 1 - \sqrt{\frac{f_1(\mathbf{x})}{g(\mathbf{x})}} - \left(\frac{f_1(\mathbf{x})}{g(\mathbf{x})} \right) \sin(10\pi f_1(\mathbf{x})), \\ 0 &\leq x_i \leq 1. \end{aligned}$$

For the single-objective test functions, we use the median of the best solutions found in 100 independent runs. For the bi-objective test problem, we use the median of the median of relative hypervolume (RHV) in 100 independent runs, which is the ratio of the measured hypervolume defined by non-dominated solutions divided by the hypervolume defined by the true Pareto-optimal front. We use the Nadir point (1 0.852) as the reference point for calculation of the hypervolume. These problems are tested in 5-D space.

4.1 Effect of R_{prox}

One of the main advantage of NSA-EA over available surrogate-assisted optimization algorithms is the notation of proximity regions when selecting new infill solutions for high-fidelity evaluation. This allows a gradual shift from exploration to exploitation, one the principle concepts of population-based optimization methods. To show the effect of this factor, we test NSA-EA for

different values of τ_R (see Equation 2). A higher τ_R makes a faster transition from exploration to exploitation. $\tau_R = \infty$, for example, suddenly reduces R_{prox} to ϵ_{prox} , which actually suppress the effect of R_{prox} concept. The budget of high-fidelity solutions is set to 20 times the number of variables (20D), and different values for fraction of initial high-fidelity solutions ($iniFr$) are tried. For this experiment, R_{prox} is set to ∞ to exclude the effect of R_{trust} . Figure 3 illustrates the performance metric for each setting. As it can be observed:

- In general, proper reduction rate τ_R can significantly improve final results. This is more detectable for Rastrigin and Schwefel functions. $\tau_R = 2$ results in significantly better final solutions, when compared to $\tau_R = \infty$ or $\tau_R = 0.5$
- A gradual reduction of R_{prox} (for example, $\tau_R = 2$) improves the robustness of the method to the fraction of initial solutions. When $\tau_R = \infty$, exploration is limited to the initialization phase. For a small value of $iniFr$, this results in a more considerable performance drop compared to $\tau_R = 2$, in which exploration diminishes gradually. For the same reason, a higher $iniFr$ can help by improving the exploration when $\tau_R = \infty$.
- Suppressing the idea of R_{prox} is advantageous for ZDT3 problem, possibly because of the simplicity of the objective functions in this problem, or the fact that in multi-objective problems, diversity of infill solutions in the objective space can help diversity of solutions in the variable space.

Consequently, a gradual reduction of R_{prox} , motivated by gradual shift from exploration to exploitation, can improve both the quality of final solutions and robustness with respect to fraction of initial solutions.

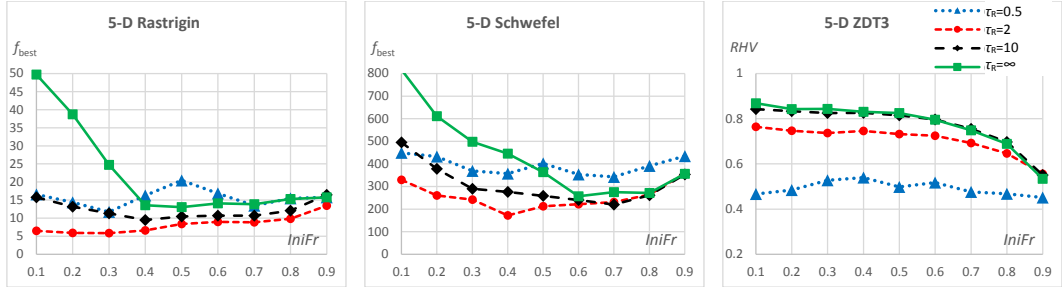


Figure 3: Median of the performance measure found over 100 independent runs for each test problem (maxEvals=100, MaxEpoch=20)

4.2 Effect of N_{epoch}

A smaller N_{epoch} is desirable from application point of view since it allows parallel evaluation of the new infill solutions; however, it may degrade optimization performance by postponing exploitation of information on true values of the new infill solutions. To investigate these factors, we test NSA-EA with different values for N_{epoch} and $maxEvals$. The median of performances is calculated over the

median of 100 independent run, which is plotted in Figure 4. It demonstrates that:

- A higher value of N_{epoch} improves the results for the Rastrigin and the Schwefel function, but its effect on ZDT function is not spectacular. This can again be associated with simplicity of the ZDT3 landscape, for which a gradual refinement of the metamodel is not critical.
- The extra gain for a higher $N_{epoch} > 16$ is barely detectable, independent of the value of $maxEvals$. Assuming the required time of an evaluation dominates the metamodel optimization and training time, this trade-off makes decision-making easier on the value of $maxEvals$ and N_{epoch} considering the available computation resources.
- A higher $maxEvals$ results in better solutions, especially for the Rastrigin and the Schwefel function. For the ZDT3 problem, increasing $maxEvals$ up to 100 has a dramatic effect, while further improvement is marginal after that. This can be lack of diversity of the non dominating solutions.

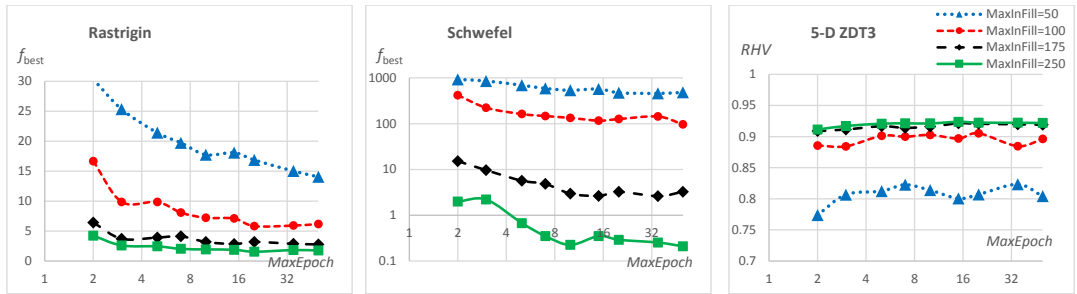


Figure 4: Median of the performance measure found over 100 independent runs for each test problem ($\tau_R = 2, iniFr = 0.5$)

5 Numerical Comparisons

This section compares the performance of NSA-EA with Surrogate Optimization of Computationally Expensive Multi-Objective (SOCEMO) (Müller 2017), a recently developed surrogate-assisted evolutionary algorithm for multi-objective optimization. The code of this method is available online ¹, and the default parameter setting, as suggested by the developer (Müller 2017), was used.

Table 1 presents six multi-objective test problems which are employed in this study for numerical evaluation and comparison. The family of ZDT and DTLZ test problems are widely used in the multi-objective optimization literature (Zitzler, Deb, and Thiele 2000; Deb et al. 2002). We eliminate a few of them either because of excessive simplicity (ZDT1 and ZDT2) or similarity to other selected problems. The problem dimension is 10 for all the problems, and the maximum number of evaluations is set to 100. This setting is based on the types

¹<https://ccse.lbl.gov/people/julianem/>

of practical problems that motivated this study, in which the problem dimension is about 10 and the maximum evaluation budget is about 100.

For each problem, 100 independent runs are performed, and Mean Relative Hypervolume (MRHV) is employed to compare the performance of the methods. The reference point for calculating the HV is selected as follows:

$$X_{REF} = \alpha(X_{NADIR} - X_{IDEAL}) + X_{IDEAL}, \quad (6)$$

in which X_{REF} , X_{NADIR} and X_{IDEAL} are the reference, Nadir and the Ideal point and $\alpha > 1$ is a scalar parameter specifying the location of the reference point on the line connecting the Ideal to the Nadir Point. By default, we select $\alpha = 1.01$; however, for some hard problems, this setting may result in $MRHV \approx 0$. If so, our measure may not determine which method has performed better. Alternatively, we try a greater α such that at least one method can reach $MRHV \approx 0.5$, if the default value results in a very small MRHV for both methods.

Table 2 presents the calculated MRHV for both methods. It reveals that:

Table 1: Test problems for empirical evaluation and comparison. For our simulation, $D = 10$ and $maxEvals = 100$.

PID	Function	No. of Objectives	X_{IDEAL}	X_{NADIR}
1	ZDT3	2	$[0, -0.7748]$	$[0.8518, 1]$
2	ZDT4	2	$[0, 0]$	$[1, 1]$
3	ZDT6	2	$[0, 0.28080]$	$[1, 0.9112]$
4	M-DTLZ1	3	$[0, 0, 0]$	$[0.5, 0.5, 0.5]$
5	DTLZ2	3	$[0, 0, 0]$	$[1, 1, 1]$
6	DTLZ6	3	$[0, 0, 0]$	$[1, 1, 1]$

Table 2: Relative hypervolume (median of 100 independent runs) for the SOCEMO and NSA-EA, and the selected value of α (see Equation 6).

PID	max-HV	α	SOCEMO	NSA-EA
1	0.8031	1.01	0.1097	0.900
2	3.6615	2	0.0409	0.2364
3	22.8240	1.01	0.5801	0.7477
4	15625	40	0.0485	0.4498
5	0.4472	1.01	0.4204	0.8586
6	5.4060	1.01	0.6270	0.9360

- NSA-EA outperforms SOCEMO for all investigated test problems. This superiority is more visible for ZDT3, ZDT4, M-DTLZ1 and DTLZ2.
- Except for M-DTLZ1, and to some extent for ZDT4, NSA-EA could satisfactorily approach the true Pareto front. This demonstrates its merit in practical problems, where we are interested in maximum gain for a very limited evaluation budget.

- Although the original DTLZ was modified to reduce its multimodality, still none of the methods could approach the true Pareto front, and thus a great value of α was selected to get meaningful MRHV.
- The internal computation time of SOCEMO is much less than NSA-EA, especially that SOCEMO employs RBF, which is computationally cheap to train; nevertheless, for the target problems of this project, in which each design evaluation may take from one hour to a few days, the computation time of NSA-EA is still negligible.
- NSA-EA has an important practical advantage: It can submit the new infill designs for external evaluation in a group of desired size. Based on our evaluation, SOCEMO does not have this flexibility and requires the evaluation of new infill solutions to proceed. This is a critically important feature when it comes to practical problems.

6 Case Study: Application to Cylinder Head Water Jacket Design

In this section, NSA-EA is employed to optimize the design of an engine cylinder head. In the following, the problem is described, the optimization procedure is explained, and finally, the obtained results are presented and discussed.

6.1 Problem Description

The problem has eight design parameters, which are the area of four inlets and four outlets of the cooling water jacket. These parameters are normalized with respect to the largest possible section. This way each area variable takes a value within $(x_i \in [0, 100], i = 1, 2, \dots, 8)$. In the base design, all inlets and outlets are set to their maximal size, which means $x_i = 100$ for all $i = 1, 2, \dots, 8$. Two conflicting objectives were defined – one ($f_1(x)$) is maximized, and the other ($f_2(x)$) is minimized. Each design evaluation requires a detailed CFD simulation which takes about one hour using 32 CPUs. The evaluation budget is limited to 61 designs to complete the optimization task in a reasonable computational time. Furthermore, two different boundary conditions are considered for CFD simulation, resulting in two separate problems. They are labelled B34 and B38 here. We solve both problems independently using our proposed approach. Figure 5 shows a measure related to the temperature distribution of the engine cylinder head water jacket at a given point of time during a CFD simulation. The objectives are derived from the flow of water through the jacket. Besides the variable bounds, there is no other constraint in the problem.

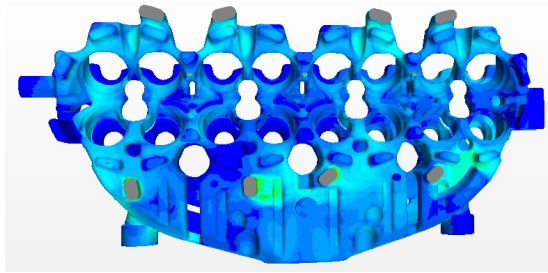


Figure 5: CFD simulation: Engine cylinder head water jacket.

6.2 Methodology

Two optimization approaches are tested on problems B34 and B38 in parallel and independently. In the first approach, a commercial software is utilized by engine design engineers for surrogate model training and optimization. The selection of new infill solutions, however, is performed manually by the engineering team. This approach is denoted by CS (Commercial Software).

In the second approach, the proposed NSA-EA is employed by the research team at Michigan State University (MSU) for design optimization. For NSA-EA, selection of new infill solutions is performed automatically by the algorithm, except for the final two epochs, in which the engineering team acted as decision-makers to choose preferred solutions from NSA-EA results. At the end of each epoch, three to five new solutions obtained by NSA-EA are sent to the engineering team for evaluation.

The CS approach was run independently by the engine design engineers, while NSA-EA was run at MSU with assistance on solution evaluations and preference information.

6.3 Results

The research team at MSU interacted with the optimization progress in two ways:

- At the end of the ninth epoch, they manually reduced the search range based on lower and upper values of the variables among all non-dominated solutions, as depicted in Figure 6.
- For the last two epochs, the MSU team was informed by the engine design engineering team that they are interested in solutions with $f_2 \leq 10.0$, with f_1 as large as possible. Therefore, for the last two epochs, the MSU team selected the new infill solutions from the search results manually. The chosen solutions are marked in Figure 7 for epoch 9 for both B34 and B38 cases. In both cases, a well-distributed set of points within $f_2 \in [9.4, 10]$ are chosen. For B34 problem, engineering team was interested in applying two extra epochs with continued preference information.

Figure 8 shows the predicted values of new infill solutions, as well as their true values after CFD simulation for both methods (CS and NSA-EA) and both problems (B34 and B38). All generated solutions by both methods are illustrated in Figure 9. The region of interest of the engine design engineering

team is focused in Figure 10, which also demonstrates the final solutions selected by the engineering team for fabrication and experimental testing.

6.4 Discussion

Based on the obtained results, the following conclusions can be made:

- NSA-EA and CS generate solutions that dominate most initial infill solutions (Figure 9). This can be considered as a checkpoint for validity of optimization process even when the evaluation budget is highly limited.
- The selected solutions for fabrication are from NSA-EA results for problem B34, and from CS results for problem B38 (Figure 10). For both solutions, f_2 is slightly greater than 10, as chosen by the engineering team.
- The prediction error (the difference between a solid line and the high-fidelity evaluated points) is initially high for both methods and both objectives. The error remains high for the CS method until the end, but gradually reduces for the NSA-EA method with more solution evaluations (Figure 8). This is more detectable for f_2 , for which the prediction error becomes almost zero in final epochs when using NSA-EA. This advantage of NSA-EA is presumably the result of a better exploration of promising regions in early epochs and a better exploitation in the final epochs, as well as manual reduction of the search range.
- Compared to the base design, the selected solutions show a maximum of 88% and 114% improvement of f_1 for problems B34 and B38, respectively. This demonstrates impact of algorithmic optimization in comparison with intuition-based design.
- One interesting and unexpected feature of the selected design for problem B34 is that one of the outlets is almost removed (x_6 being close to zero in Figure 6). This unexpected observation demonstrates possibility of using optimization to come up with innovative knowledge about key features of optimally solving a problem. Such information can be used in earlier stages of design to determine the number of inlets/outlets or even their locations. Although simultaneous optimization of different features is more challenging than optimization of only sizes of inlets/outlets, it is predictably much more rewarding as well.
- For this problem, considering the soft constraint $f_2 \leq 10$ from the beginning on could have been advantageous. It would automatically concentrate the search to the region of interest in the f -space, however such a knowledge about the upper bound on f_2 may not have been known a priori and may have resulted from the non-dominated solutions over epochs.

Although both methods were tested independently, the CS approach was run by the design engineering team, who had the knowledge of designers' region of interest. This significant privilege helps concentration of the search to a small region, which not only provides more infill solutions for that region but also improves the accuracy of the metamodel. For the NSA-EA, such information

was provided and used only for the last two epochs, when the exploration phase was almost concluded.

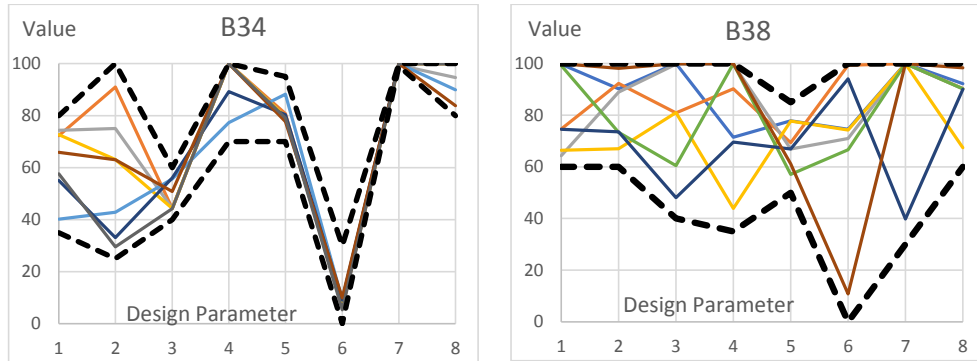


Figure 6: Manual reductions of the search range for problems *B34* (left) and *B38* (right) from the 9th Epoch.

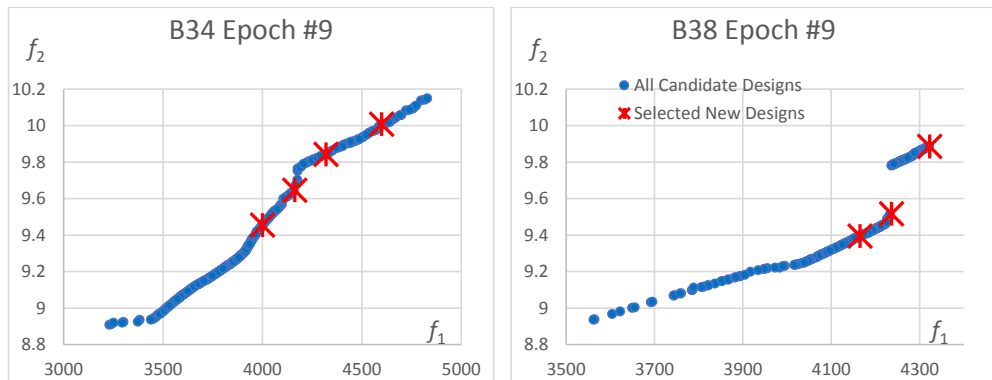


Figure 7: Manual selection of the new solutions from the predicted PO solutions in the ninth epoch for problems *B34* and *B38*.

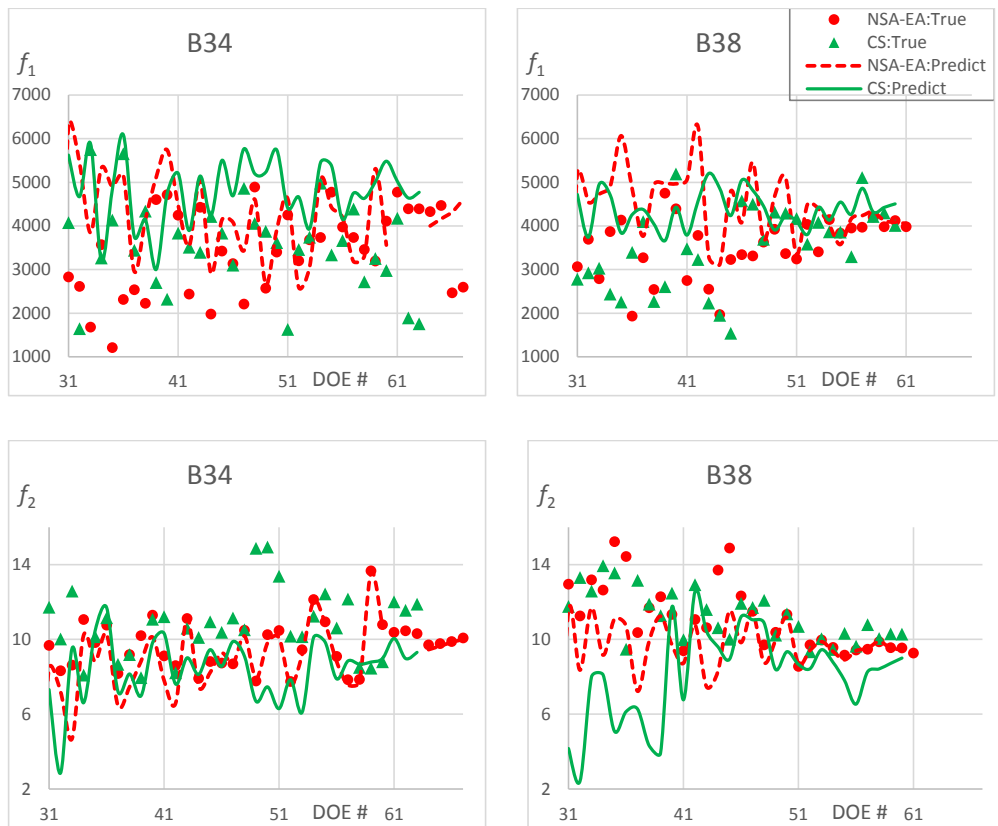


Figure 8: Predicted values of new solutions and their true values after CFD simulation for both methods, both problems, and both objectives.

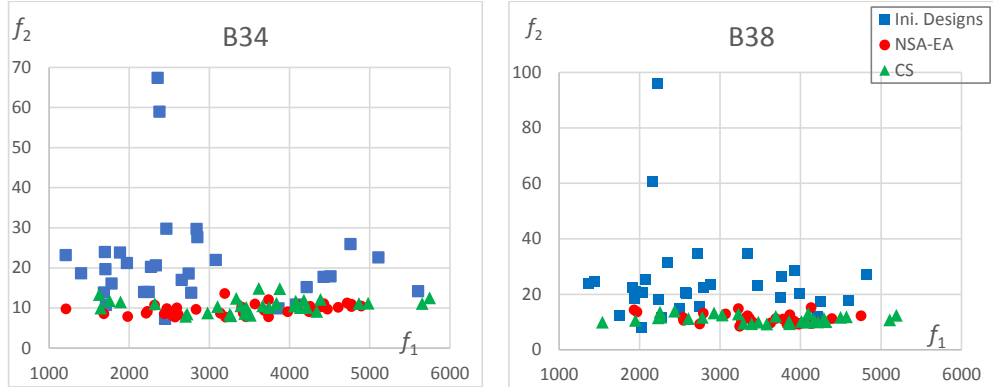


Figure 9: All generated solutions by the research team at MSU and engineering team for problems $B34$ and $B38$.

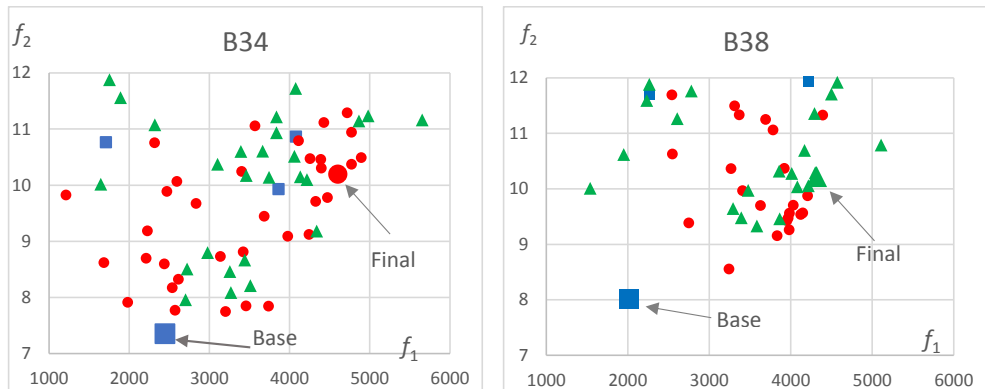


Figure 10: Solutions in the vicinity of the interest region of the decision-making team for problems $B34$ and $B38$. The base and the selected designs for fabrication is demonstrated by arrows.

7 Conclusion

In many applications, including optimal engineering design, a solution evaluation requires a costly computer simulation or even experiments, which limits the number of designs that can be evaluated in an optimization task. This study has developed a niching-based surrogate-assisted evolutionary algorithm (NSA-EA) for single and multi-objective optimization of computationally expensive problems. NSA-EA has introduced the concepts of proximity measure and a trust region concept for balancing the trade-off between exploration and exploita-

tion. By gradually shrinking two radius parameters defining proximity and trust region, the exploration aspect has been relatively reduced to provide more exploitation of already-found good solutions. Our descriptive experiments have demonstrated the importance of such a gradual transition. Moreover, NSA-EA has been evaluated on six test problems and compared with SOCEMO – a recent surrogate-assisted multi-objective optimization algorithm. Results have demonstrated the superiority of NSA-EA over SOCEMO. Furthermore, NSA-EA has been employed to optimize the design of an engine head water jacket. The case study illuminates the importance of interactive and collaborative optimization for solving practical problems. For example, user preference can clarify which part of the Pareto front is more practical. This is crucially important when the evaluation budget is limited since the scarcity of infill solutions cannot provide a good estimation of the whole Pareto-optimal front.

This initial study on a practical problem and executed in collaboration with an industry partner opens up several practical research issues, some of which we plan to study next. An essential practical feature of collaborative optimization is the number of infill points which industry can afford to evaluate at a time. In this study, we provided 3-5 infill solutions after every epoch. In certain situations, an industry may be interested in evaluating an order of magnitude more solutions at a time in order to reduce their efforts and time. In such a situation, we shall be left with fewer epochs to complete the whole optimization task but will have the privilege of using many infill points to make a better surrogate model at every epoch. A study exploring the trade-off between the two aspects should be executed on a case by case basis. The possibility of evaluating designs in parallel should be exploited in this context. In this study, the concept of trust region has shown promising results especially on multimodal single-objective problems; however, an adaptive scheme can be more effective than a predefined dynamic reduction of the proximity and trust regions. This can be the subject of future research in the domain of this study. Although Kriging surrogate modeling approach is used in this study, other approaches, such as radial basis function, neural networks, etc., can be considered after each epoch for each objective function modeling.

Multi-objective optimization resorts to a number of trade-off solutions, but decision-making of solutions in terms of preference information about a problem is also an equally important task. An optimization using a surrogate modeling approach must consider both aspects of optimization and decision-making. Moreover, real-world optimization is collaborative between optimization researchers and practitioners. In this paper, we have demonstrated one viable approach for achieving the collaborative optimization task on a real-world design problem. More such developmental studies must now be done to make the whole approach smooth and systematic.

Acknowledgements

This study was supported by Ford Motor Company under the Ford-MSU Alliance framework.

References

- Akhtar, Taimoor, and Christine A. Shoemaker. 2016. “Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection.” *Journal of Global Optimization* 64 (1): 17–32. <https://doi.org/10.1007/s10898-015-0270-y>.
- Auger, A., and N. Hansen. 2005. “A restart CMA evolution strategy with increasing population size.” In *2005 IEEE Congress on Evolutionary Computation*, Vol. 2, 1769–1776 Vol. 2.
- Bartz-Beielstein, Thomas, Mike Preuss, Karlheinz Schmitt, and Hans-Paul Schwefel. 2010. *Challenges for Contemporary Evolutionary Algorithms*. Cite-seer.
- Cao, Y.J., L. Jiang, and Q.H. Wu. 2000. “An evolutionary programming approach to mixed-variable optimization problems.” *Applied Mathematical Modelling* 24 (12): 931 – 942.
- Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. 2002. “A fast and elitist multiobjective genetic algorithm: NSGA-II.” *IEEE Transactions on Evolutionary Computation* 6 (2): 182–197.
- Deb, K., L. Thiele, M. Laumanns, and E. Zitzler. 2002. “Scalable multi-objective optimization test problems.” In *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, Vol. 1, May, 825–830.
- Deb, Kalyanmoy. 2001. *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc.
- Deb, Kalyanmoy, Rayan Hussein, Proteek Roy, and Gregorio Toscano. 2017. “Classifying Metamodeling Methods for Evolutionary Multi-objective Optimization: First Results.” In *9th International Conference on Evolutionary Multi-Criterion Optimization - Volume 10173*, EMO 2017, New York, NY, USA, 160–175. Springer-Verlag New York, Inc.
- Filomeno Coelho, Rajan, Jérémy Lebon, and Philippe Bouillard. 2011. “Hierarchical stochastic metamodels based on moving least squares and polynomial chaos expansion.” *Structural and Multidisciplinary Optimization* 43 (5): 707–729.
- Goldberg, David E., and Jon Richardson. 1987. “Genetic Algorithms with Sharing for Multimodal Function Optimization.” In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*, Hillsdale, NJ, USA, 41–49. L. Erlbaum Associates Inc. <http://dl.acm.org/citation.cfm?id=42512.42519>.
- Hansen, Nikolaus, Anne Auger, Raymond Ros, Steffen Finck, and Petr Pošík. 2010. “Comparing Results of 31 Algorithms from the Black-box Optimization Benchmarking BBOB-2009.” In *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO '10*, New York, NY, USA, 1689–1696. ACM. <http://doi.acm.org/10.1145/1830761.1830790>.

- Hansen, Nikolaus, André S. P. Niederberger, Lino Guzzella, and Petros Koumoutsakos. 2009. “A Method for Handling Uncertainty in Evolutionary Optimization With an Application to Feedback Control of Combustion.” *IEEE Trans. Evolutionary Computation* 13 (1): 180–197.
- Hussein, Rayan, and Kalyanmoy Deb. 2016. “A Generative Kriging Surrogate Model for Constrained and Unconstrained Multi-objective Optimization.” In *Proceedings of the Genetic and Evolutionary Computation Conference 2016, GECCO '16*, New York, NY, USA, 573–580. ACM. <http://doi.acm.org/10.1145/2908812.2908866>.
- Jones, Donald R., Matthias Schonlau, and William J. Welch. 1998a. “Efficient Global Optimization of Expensive Black-Box Functions.” *Journal of Global Optimization* 13 (4): 455–492. <https://doi.org/10.1023/A:1008306431147>.
- Jones, Donald R., Matthias Schonlau, and William J. Welch. 1998b. “Efficient Global Optimization of Expensive Black-Box Functions.” *J. of Global Optimization* 13 (4): 455–492. <https://doi.org/10.1023/A:1008306431147>.
- Karimzadehgan, Maryam, and ChengXiang Zhai. 2010. “Exploration-exploitation Tradeoff in Interactive Relevance Feedback.” In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM '10*, New York, NY, USA, 1397–1400. ACM. <http://doi.acm.org/10.1145/1871437.1871631>.
- Knowles, Joshua, David Corne, and Alan Reynolds. 2009. *Noisy Multiobjective Optimization on a Budget of 250 Evaluations*, 36–50. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Mongeau, Marcel. 2009. “Discontinuous Optimization.” In *Encyclopedia of Optimization, Second Edition*, edited by Christodoulos A. Floudas and Panos M. Pardalos, 739–744. Springer. https://doi.org/10.1007/978-0-387-74759-0_130.
- Mueller, Juliane. 2016. “MISO: mixed-integer surrogate optimization framework.” *Optimization and Engineering* 17 (1): 177–203. <https://doi.org/10.1007/s11081-015-9281-2>.
- Mueller, Juliane. 2017. “SOCEMO: Surrogate Optimization of Computationally Expensive Multiobjective Problems.” *INFORMS Journal on Computing* 29 (4): 581–596. <https://doi.org/10.1287/ijoc.2017.0749>.
- Müller, Juliane. 2017. “SOCEMO: Surrogate Optimization of Computationally Expensive Multiobjective Problems.” *INFORMS Journal on Computing* 29 (4): 581–596.
- Müller, Juliane. 2017. “SOCEMO: Surrogate Optimization of Computationally Expensive Multiobjective Problems.” *INFORMS Journal on Computing* 29 (4): 581–596. <https://doi.org/10.1287/ijoc.2017.0749>.
- Nielsen, H. B., S. N. Lophaven, and J. Søndergaard. 2002. “DACE - A Matlab Kriging Toolbox.” Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby. <http://www2.imm.dtu.dk/pubdb/p.php?1460>.

- Sahinidis, Nikolaos V. 2004. "Optimization under uncertainty: state-of-the-art and opportunities." *Computers & Chemical Engineering* 28: 971–983.
- Seada, Haitham, and Kalyanmoy Deb. 2015. *U-NSGA-III: A Unified Evolutionary Optimization Procedure for Single, Multiple, and Many Objectives: Proof-of-Principle Results*, 34–49. Cham: Springer International Publishing.
- Steuer, Ralph E., and Eng-Ung Choo. 1983. "An interactive weighted Tchebycheff procedure for multiple objective programming." *Mathematical Programming* 26 (3): 326–344. <https://doi.org/10.1007/BF02591870>.
- Wang, Gary G., and S. Shan. 2007. "Review of Metamodeling Techniques in Support of Engineering Design Optimization." *Journal of Mechanical Design* 129 (4): 370–380. <http://dx.doi.org/10.1115/1.2429697>.
- Zapotecas Martínez, Saúl, and Carlos A. Coello Coello. 2010. "A Multi-objective Meta-model Assisted Memetic Algorithm with Non Gradient-based Local Search." In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO '10*, New York, NY, USA, 537–538. ACM. <http://doi.acm.org/10.1145/1830483.1830581>.
- Zapotecas Martínez, Saúl, and Carlos A. Coello Coello. 2013. "MOEA/D Assisted by Rbf Networks for Expensive Multi-objective Optimization Problems." In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, GECCO '13*, New York, NY, USA, 1405–1412. ACM. <http://doi.acm.org/10.1145/2463372.2465805>.
- Zhang, Q., and H. Li. 2007. "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition." *IEEE Transactions on Evolutionary Computation* 11 (6): 712–731.
- Zhang, Yu, Sanbao Hu, Jinglai Wu, Yunqing Zhang, and Liping Chen. 2015. *Modeling and Multi-Objective Optimization of Double Suction Centrifugal Pump Based on Kriging Meta-models*, 251–261. Cham: Springer International Publishing.
- Zitzler, E., K. Deb, and L. Thiele. 2000. "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results." *Evolutionary Computation* 8 (2): 173–195.