Structural and Multidisciplinary Optimization
https://doi.org/10.1007/s00158-020-02592-6

RESEARCH PAPER

Adaptive in situ model refinement for surrogate-augmented population-based optimization

Payam Ghassemi[1] · **Ali Mehmani**[2] · **Souma Chowdhury**[1]

Received: 16 October 2019 / Revised: 9 March 2020 / Accepted: 29 March 2020

Abstract

In surrogate-based optimization (SBO), the deception issues associated with the low fidelity of the surrogate model can be dealt with in situ model refinement that uses infill points during optimization. However, there is a lack of model refinement methods that are both independent of the choice of surrogate model (neural networks, radial basis functions, Kriging, etc.) and provides a methodical approach to preserve the fidelity of the search dynamics, especially in the case of population-based heuristic optimization processes. This paper presents an adaptive model refinement (AMR) approach to fill this important gap. Therein, the question of when to refine the surrogate model is answered by a novel hypothesis testing concept that compares the distribution of model error and distribution of function improvement over iterations. These distributions are respectively computed via a probabilistic cross-validation approach and by leveraging the probabilistic improvement information uniquely afforded by population-based algorithms such as particle swarm optimization. Moreover, the AMR method identifies the size of the batch of infill points needed for refinement. Numerical experiments performed on multiple benchmark functions and an optimal (building energy) planning problem demonstrate AMR's ability to preserve computational efficiency of the SBO process while providing solutions of more attractive fidelity than those provisioned by a standard SBO approach.

Keywords Adaptive model refinement · Surrogate-based optimization · Predictive estimation of model fidelity · Sequential sampling · Particle swarm optimization

1 Introduction

Optimizing complex systems often involves computationally expensive simulations (e.g., FEA) to evaluate system behavior and estimate quantities of interest. While computationally efficient alternatives are often available for system or function evaluation, for example in the form of simplified analytical models, coarse grid models, or surrogate models (to name a few), they tend to compromise on the fidelity of their estimations. These low-fidelity models often mislead the search process during optimization, leading to sub-optimal or even infeasible solutions. *Variable-fidelity optimization* approaches seek to address these issues and offer attractive trade-offs between computational efficiency and fidelity of the optimal solutions obtained—i.e., the ability to quickly arrive at optimal solutions that can be relied upon. In these approaches, model management strategies, for instance, model selection, switching, and/or refinement, adaptively integrate models with different levels of fidelity and (computational) cost into the optimization process.

Surrogate-based optimization (SBO) constitutes one of the most important implementations of variable-fidelity concepts in design optimization and optimal planning. Surrogate models or metamodels are data-driven models that are trained using a carefully designed set of (high-fidelity simulation) experiments, and are inexpensive to implement once trained (Simpson et al. 2008; Jin 2011; Fernández-Godino et al. 2016). SBO typically uses one or more surrogate models in place of or in addition

Responsible Editor: Nathalie Bartoli

✉ Souma Chowdhury
soumacho@buffalo.edu

Payam Ghassemi
payamgha@buffalo.edu

[1] Department of Mechanical and Aerospace Engineering, University at Buffalo, Buffalo, NY, 14260, USA

[2] Data Science Institute, Columbia University, New York, NY, 10027, USA

Published online: 26 May 2020

to physics-based models to perform objective and/or constraint function evaluation in an optimization process. Both interpolating-type (e.g., radial basis functions) and regression-type (e.g., quadratic response surfaces) surrogate modeling methods are popular in design optimization.

The reliability of the optimization search process in SBO depends on the interplay of model uncertainty and function improvement over iterations. Our paper presents a new SBO approach that seeks to *address the challenges associated with characterizing and thereby regulating this interplay through situation-adaptive refinement. Specifically, this is accomplished in a manner in which the reliability of the search process is preserved without compromising computational efficiency*. In doing so, we take a model-type-independent approach that also exploits crucial characteristics of population-based optimization algorithms, namely distributed information processing and the ability to deal with non-convex functions. These contributions extend and build upon the concept and framework presented earlier in our conference papers (Mehmani et al. 2015a, 2016; Chowdhury et al. 2016).

The remaining portion of this introduction section provides a review of model management strategies in general, and those specific to SBO, and a summary of the objectives of this paper.

1.1 Variable-fidelity optimization

Different model management approaches have been reported in the literature for integrating low-fidelity models within optimization processes. One of the classical model management strategies is that of involving *Trust-Region* methods (Booker et al. 1999b; Alexandrov et al. 1999; Rodriguez et al. 2001; Marduel et al. 2006; Robinson et al. 2008). In these methods (Alexandrov et al. 1998), the trust-region radius parameter is adaptively increased (or decreased) depending on the ratio of the actual improvement to the predicted improvement (given by the low-fidelity model) in the objective function. Some *Trust-Region* methods also seek the agreement of the function and its gradient values in the low-fidelity model with those estimated in the high-fidelity model (March et al. 2011). However, these techniques may not be directly applicable in problems where gradients are expensive to evaluate, or where zero-order algorithms are being used for optimization.

In another class of model management strategies, focused on surrogate-based optimization, the accuracy and robustness of the surrogate model are improved during the optimization process by adding infill points where additional evaluations of the high-fidelity model or experiment are desired to be performed. Over the last two decades, different surrogate-based optimization strategies have been developed (Jones et al. 1998; Duan et al. 1992; Kleijnen et al. 2012; Bichon et al. 2013; Moore et al. 2011). In SBO approaches that seek to refine the surrogate model during the optimization process, infill points are generally added in (i) the region where the optimum is likely located (local exploitation); (ii) the region(s) where the uncertainty induced by the model is predicted to be high; and/or (iii) the entire design space (global exploration) (Keane and Nair 2005; Forrester et al. 2008; Sugiyama 2006). Infill points can be added in a fully sequential (one-at-a-time), or a batch sequential, manner. Various criteria exist for determining the locations of the infill points including (i) index-based criteria (e.g., (integrated- and maximum) mean squared error (MSE) and maximum entropy criteria) and (ii) distance-based criteria (e.g., Euclidean distance, Mahalonobis distance, and weighted distance criteria) (Jones et al. 1998; Moore et al. 2011; Keane 2006; Williams et al. 2011; Booker et al. 1999; Audet et al. 2000; Rai 2006).

Variations of Bayesian optimization (Jones et al. 1998; Pelikan 2005; Snoek et al. 2012; Tajbakhsh et al. 2013) feature prominently among SBO-type model management strategies, with *efficient global optimization* (EGO) (Jones et al. 1998) being one of the most popular BO variants in the design optimization domain. These BO variants typically use criteria such as "expected improvement" or "probability of improvement," which are evaluated from a Gaussian process model that is trained and refined in situ during the BO process. Translating the capabilities of these methods to SBO based on other surrogate models (instead of GP) is challenging; however, important strides have been made in this direction through heuristics such as importing uncertainty measures from one surrogate model to another. Substantial amount of work has been done in advancing the EGO paradigm in other ways (Tajbakhsh et al. 2013; Hennig and Schuler 2012; Viana et al. 2013), a detailed review of which is beyond the scope of this paper. Other, more generally applicable (w.r.t. model type) SBO methods, with many using radial basis functions as the surrogate model, are discussed next.

Major surrogate modeling methods that are used in SBO include polynomial response surfaces (Jin et al. 2001), Kriging (Simpson et al. 2001; Forrester and Keane 2009; Lulekar et al. 2018), moving least square (Choi et al. 2001; Toropov et al. 2005), radial basis functions (RBF) (Hardy 1971), support vector regression (SVR) (Clarke et al. 2005), artificial neural networks (Yegnanarayana 2004; Liu et al. 2018), and hybrid surrogate models (Zhang et al. 2012). RBFs have seen popular usage in various SBO implementations, which include unconstrained local optimization (Wild et al. 2008), multi-objective optimization (Jakobsson et al. 2010), high-dimensional optimization (Regis and Shoemaker 2013;

Regis 2014), and handling computationally expensive constraints (Peng et al. 2014). Notable examples of performing SBO with neural networks include the work by Kourakos and Mantoglou (2009) and Yao et al. (2014). The primary gap in these existing SBO techniques, especially ones that provide model refinement and are model independent in implementation, can be summarized as follows: lack of a coherent approach to inform *when* to refine (during optimization iterations) and *how many* infill points to add during refinement, such that the improvements observed during optimization remain reliable in light of the uncertainty of the function evaluations performed by a surrogate model. In this context, we hypothesize that population-based optimization algorithms uniquely provision statistical information on the degree of improvement incurred across iterations.

Existing model management strategies used with population-based (metaheuristics) optimization algorithms have instead mostly focused on deciding which members of the population should be evaluated with high-fidelity models (individual-based evolution control), or which iteration/generation of the optimization should use high-fidelity evaluation of the entire population (a.k.a. generation-based evolution control) (Jin et al. 2002). Graning et al. (2007) have explored different individual-based evolution frameworks such as (i) the Best Strategy (Jin 2005), where the best individuals at each generation are selected as controlled individuals; (ii) the Pre-Selection method (Ulmer et al. 2004), where the offspring of the best individuals are selected as controlled individuals and (iii) the Clustering Technique (Jin and Sendhoff 2004), where the *k-means* clustering technique is used to find the "controlled individual cluster" based on the distance from the best individual.

Note that, given the statistical information on function improvement implicit to population-based algorithms, they offer the opportunity to establish a reliability criterion to determine "when to refine" and "with how many infill points"; this opportunity remains largely unexploited by existing population-based SBO implementations. Moreover, successful formulation of such a reliability criterion is also contingent on our ability to capture the uncertainty induced by the surrogate model being used for optimization. Unfortunately, other than variants of Gaussian processes, very few surrogate model types directly provide measures of induced uncertainty. In this paper, we aim to address these gaps by developing and testing a new adaptive model refinement criterion, which leverages (1) the above-stated opportunity provided by population-based algorithms and (2) a relatively recent approach to quantifying the uncertainty induced by surrogate models. The specific objectives of this paper are summarized next.

1.2 Surrogate-based optimization with adaptive refinement

The primary objective of this paper is to develop a new surrogate-based optimization method with an in situ adaptive model refinement approach that seeks to reduce the computational cost of optimization while converging to the optimum/optima with an acceptable level of fidelity. The proposed SBO method is designed to have the following characteristics:

1. Independent of the type of surrogate models (radial basis functions, neural networks, Gaussian processes, etc.) being used in SBO.
2. Efficiently preserves the reliability of the optimization search process, by predicting when the surrogate models (used for function evaluations) need to be refined with infill points, to be performed in-between optimization iterations (hence "in situ").
3. Determines the optimal batch size for the infill points, once the refinement event is triggered; this adaptive approach is conceived to provide further computational efficiency benefits over one-at-a-time sequential approaches and approaches where the batch size is a user-defined parameter (Marmin et al. 2015).

The second objective of the paper is to analyze the performance of our new SBO method by applying it to benchmark functions and comparing with standard SBO and direct high-fidelity optimizations, and to demonstrate the effectiveness of our method by applying it to a practical optimal planning problem in the building energy domain. Note that, with respect to the work earlier presented by us as shorter conference papers (Mehmani et al. 2015a, 2016; Chowdhury et al. 2016), this journal paper provides an extended literature review (earlier in this section), and more importantly, provides unique methodological extensions and experimental studies. These extensions include (i) upgraded formulations to effectively compute the number of infill points to be added during each refinement event and the locations of those infill points; (ii) comprehensive parametric analysis of the new SBO method; and (iii) new benchmark comparisons and a completely new practical application example.

The remainder of the paper is organized as follows. The next section presents the formulation of the novel adaptive model refinement (AMR) method in SBO. In addition, Section 2 summarizes the model uncertainty quantification method, the model refinement approach, and the optimization algorithm used along with AMR. Section 3 presents numerical experiments with benchmark functions to demonstrate the benefits of the proposed AMR method, followed

by application of AMR on a complex operations planning problem (building thermal management) where time-efficient decision-making is crucial. Section 5 summarizes our concluding remarks.

2 Variable-fidelity optimization with AMR

Performing model refinement (i.e., increasing model fidelity) too early in a SBO process can be computationally expensive while wasting resources to explore undesirable regions of the design domain. On the other hand, refining the surrogate model too late might mislead the search process early on, that is, leading to scenarios where the global (or even more attractive, local) optimum is outside of the region spanned by the population of candidate solutions in later iterations. The AMR method thus seeks to accomplish a desirable balance between "effectiveness of the SBO search process" and its "computational efficiency." This approach automatically determines the in situ time of refinement (in-between optimization iterations), and the number and location of infill points to be added (adaptive).

Figure 1 provides a flowchart of the AMR approach, assuming implementation in a generalized population-based optimization process. As seen from Fig. 1, the AMR approach can be divided into the following five major steps:

Step 1: A set of initial sampling points are generated in the design space using a standard design of experiments method (Latin hypercube sampling (LHS) (McKay et al. 1979) with maximin criterion is used here). The pertinent objective function is evaluated at these sample points using the high-fidelity model. An initial surrogate model is then constructed using this initial set of sample points.

Step 2: A initial population is then generated for optimization (at iteration $t = 1$), using this surrogate model.

Step 3: At every iteration (t) of the population-based optimization process, the current surrogate model is used to evaluate the function values of the candidates in the population, and then the optimization algorithm-specific steps are conducted to update the population, before the iteration is incremented, i.e., $t = t + 1$. In this paper, particle swarm optimization (PSO) is chosen as the optimization algorithm.

Step 4: Standard stopping criteria, namely change in the global best compared with set tolerance values, or maximum allowed iterations or function evaluations, are used here. If any of the termination criteria is satisfied, the current optimum (the best global solution in the case of PSO) is identified as the final optimum and the optimization process is terminated. Otherwise, we go to **Step 5**.

Step 5: The AMR metric, which serves as the criterion to decide whether to refine the surrogate model or not, is evaluated. If the AMR metric is satisfied, a model refinement step is invoked, and we go to Step 5. Otherwise, we directly go back to **Step 3**.

Step 6: The surrogate model is refined by a series of sub-steps, as shown on the right side in Fig. 1. These sub-steps do the following: estimate the required number of infill points; generate, evaluate, and use these infill points to refine the model; and evaluate the fidelity of the refined model. Then, we go back to **Step 3**.

In practice, the AMR metric (step 5) does not need to be applied at every iteration; the user can specify that it be applied at a prescribed interval of iterations. In the flowchart in Fig. 1, the metric is shown to be applied at every iteration, for the sake of simplicity. In the following subsections, we describe the novel components of AMR:

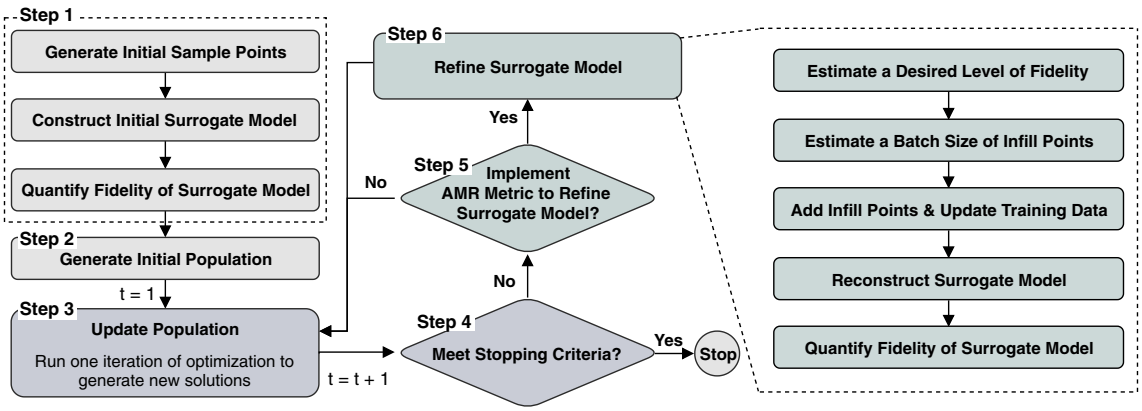


Fig. 1 Adaptive model refinement in surrogate-based optimization

the AMR metric and the subsequent model refinement strategy. Subsequently, we provide an overview of the model error measurement technique used in AMR and the mixed-discrete PSO algorithm, which are used to implement the AMR method and investigate its performance.

2.1 The AMR metric

In this paper, it is assumed that the uncertainty associated with the surrogate model can be evaluated in the form of an error distribution, $\mathbb{P}_{\text{SM}}$. Under this assumption, the function values evaluated using the current surrogate model (in the SBO process) can be related to the corresponding high-fidelity function evaluation as:

$$Y = \widehat{y}_{\text{SM}} + \varepsilon \tag{1}$$

In (1), $\widehat{y}_{\text{SM}}$ and ε, respectively, represent the function approximation given by the current surrogate model and the stochastic error associated with it, and Y is the corresponding high-fidelity function evaluation given by high-fidelity computational simulations or physical experiments. The relative improvement in the fitness function value (Δf) can be considered to follow a distribution, Θ, over the population of solutions. Here, Δf at any t^{th} iteration ($t \geq 2$) is given by:

$$\Delta {f_k}^t = \left| \frac{{f_k}^t - {f_k}^{t-\tau}}{f_{REF}^t} \right| \tag{2}$$

where ${f_k}^t$ is the function value of the local best of particle k at the t-th iteration; and f_{REF}^t is a reference value used for normalization. Here, τ ($\in \mathbb{Z}_{<t^*}$) is a user-defined interval that regulates the frequency of "surrogate model refinement" checks in the proposed SBO approach. Numerical experiments exploring different values for the parameter τ indicate that $3 \leq \tau \leq 5$ works well.

The model switching criterion in the AMR technique is defined based on the following notion: "*whether the uncertainty associated with a surrogate model response dominates the observed improvement in the relative fitness function of the population*." Since most heuristic population-based algorithms use the objective function value of candidate designs to implement intra-population comparisons that drive the dynamics of evolution of the population, the above definition applies to most heuristic algorithms. For example, in PSO, the individual local best of each particle and the global best are both updated at each iteration based on comparisons of the objective function value of the solutions visited by the particles. Inexact function evaluations are likely to make a fraction of these comparisons incorrect. The greater the model error relative to the comparative difference (in objective function value) between solutions, the more unreliable are the comparisons, and the ensuing search dynamics. Hence, our AMR metric seeks to capture if, depending on user's needs, a statistically significant fraction of the population has really improved upon their previous generation(s) in terms of objective function value. Specifically, in the case of PSO, the AMR metric is designed to test if at least a s fraction of the particles in the population have registered an amount of improvement in their individual local best (in terms of objective function value) that is greater than the expected maximum error measure of the best $(1 - s)$ fraction of the evaluations made by the surrogate model.

Since AMR is designed to work with diverse surrogate models (RBF, ANN, SVR, etc.), as opposed to being limited by model dependencies (such as Bayesian optimization with GP), we need to use a generalized measure of model error (ε) or model uncertainty (σ_ε). Generalized local error measures are typically unavailable. Thus, the model switching criterion is designed using the *stochastic global measures of model error* and the *distribution of solution improvement*. Let's say, based on designer's/user's preferences of design needs, η is the acceptable threshold value of error in the optimum yielded by the SBO process (that is using a model with an error distribution $\mathbb{P}_{\text{SM}}$). A *critical probability*, p_{cr}, can then be defined as the probability of the model error to be less than η, expressed as:

$$p_{\text{cr}} = \mathbb{P}(\varepsilon \leq \eta) = \int_0^{\eta} \mathbb{P}_{\text{SM}}(\varepsilon')\, d\varepsilon' \tag{3}$$

The critical probability (p_{cr}) indicates a critical bound in the error distribution $\mathbb{P}_{\text{SM}}$ ($0 \leq \varepsilon \leq \eta$). If the predefined

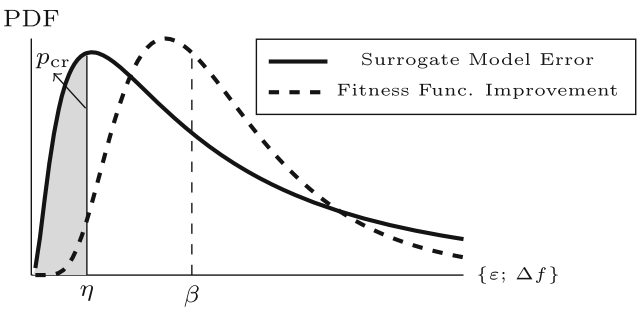


Fig. 2 Illustration of the AMR metric

cutoff value (β) of the distribution of fitness function improvement (across generations) over the population, namely the Θ distribution, lies within this critical bound, the current surrogate model is considered to be no more reliable for use in the optimization process (i.e., when $\eta \geq \beta$). Conversely, the surrogate model with the $\mathbb{P}_{\text{SM}}$ error distribution can be continued to be used in the optimization process, if $\eta \leq \beta$. An illustration of this second situation is given in Fig. 2.

Now, we can formulate the AMR metric as a hypothesis testing that is defined by a comparison between:

- The distribution of the relative fitness function improvement (Θ) over the entire population, and
- The distribution of the error associated with the current surrogate model ($\mathbb{P}_{\text{SM}}$) over the design space.

Hence, this AMR statistical test for the current surrogate model can be stated as:

$$\begin{aligned} &H_0 : \mathbb{Q}_{\mathbb{P}_{\text{SM}}}(p_{cr}) \geq \mathbb{Q}_{\Theta}(1 - p_{cr}) \\ &H_1 : \mathbb{Q}_{\mathbb{P}_{\text{SM}}}(p_{cr}) < \mathbb{Q}_{\Theta}(1 - p_{cr}) \\ &0 < p_{cr} < 1 \end{aligned} \tag{4}$$

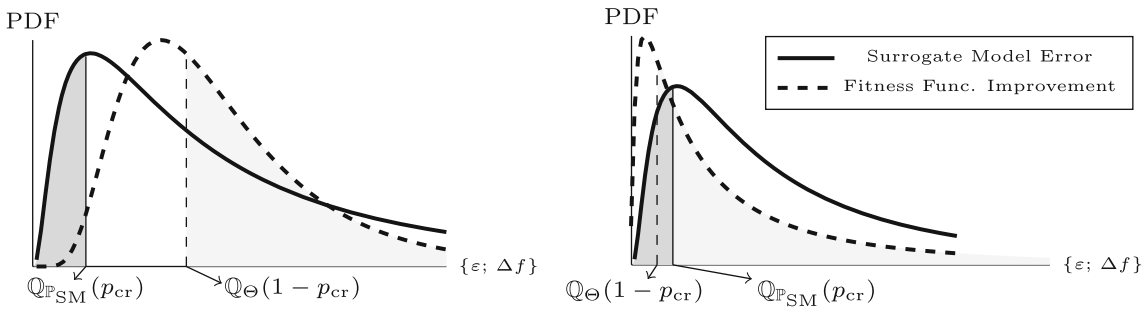
Here, $\mathbb{Q}$ represents a quantile function of a distribution, with the p-quantile, for a given distribution function, Ψ, being defined as (Meeker et al. (2017)):

$$\mathbb{Q}_{\Psi}(p) = \inf\{x \in \mathbb{R} \mid p \leq \Psi_{(c.d.f.)}(x)\} \tag{5}$$

In (4), the critical probability, p_{cr}, can be seen as an *Indicator of Conservativeness* (IoC); in other words, the IoC is a user-defined parameter, which regulates the trade-off between optimal solution reliability and computational cost in the AMR-based optimization process. Generally, the higher the IoC (closer to 1), the higher the desired solution reliability and the greater the computational cost; in such cases, model switching events will occur earlier on in the optimization process.

Figure 3a and b respectively illustrate the two scenarios, i.e., one where the AMR metric is satisfied, and another where it is not satisfied and in situ model refinement is triggered. In the first scenario (Fig. 3a), $\mathbb{Q}_{\Theta} > \mathbb{Q}_{\mathbb{P}_{\text{SM}}}$, and thus the null hypothesis will be rejected, and the optimization process will continue to use the current surrogate model. This outcome can be interpreted as the least erroneous $(p_{cr} \times 100)\%$ of the surrogate model predictions are expected to be bounded by a maximum error measure value that is smaller than the lower bound of the greatest $((1 - p_{cr}) \times 100)\%$ of the fitness function improvements across the population. Conversely, **if** $\mathbb{Q}_{\Theta} < \mathbb{Q}_{\mathbb{P}_{\text{SM}}}$, the null hypothesis will be accepted, as shown in Fig. 3b. In this scenario, the surrogate model will need to be refined with infill points before the optimization can progress further.

For evaluating the AMR metric, we need to estimate the Θ and $\mathbb{P}_{\text{SM}}$ distributions. In this paper, kernel density estimation (KDE) is adopted to model the distribution of the relative improvement (Θ) in the fitness function over consecutive τ iterations, i.e., across the population of particles in the PSO algorithm. Since the nature of distribution of fitness function improvement over the population is problem dependent, and is sometimes observed to be multimodal (over our numerical experiments with benchmark problems), the non-parametric KDE approach is a suitable choice in this context. For modeling the distribution of the error associated with the current surrogate model ($\mathbb{P}_{\text{SM}}$) over the design space, we use the log-normal distribution, based on early numerical experiments presented in Mehmani et al. (2015). Note that both distributions (Θ and $\mathbb{P}_{\text{SM}}$) are made to appear similar to the shape of a log-normal distribution in the representative Figs. 2 and 3, just for the sake of illustration aesthetics.



(a) Rejection of the test; **Don't Refine a model** (b) Acceptance of the test; **Refine a model**

Fig. 3 Illustration of the AMR hypothesis test (comparing the surrogate model error distribution ($\mathbb{P}_{\text{SM}}$) and the distribution of fitness function improvement (Θ)), that decides whether to trigger a refinement event

The uncertainty associated with surrogate models, and the batch size for the infill samples to be added in the model refinement process of AMR are determined using a surrogate model error estimation method called *predictive estimation of model fidelity* (PEMF) (Mehmani et al. 2015). The PEMF method can be perceived as a sequential stochastic extension of the K-fold cross-validation with the eventual measures represented in terms of an error distribution (instead of outlier-sensitive root mean squared measures) (Mehmani et al. 2015). Brief descriptions of KDE and the PEMF method are provided in Appendices A and C, respectively. Next, we describe how the PEMF method is used to design the refinement process.

2.2 Model refinement based on PEMF

2.2.1 Determining the batch size of infill points

The PEMF method tracks the distribution of the (median) K-fold error estimate with increasing K, and uses it to predict the error of the surrogate model that uses all samples for training. This approach in the PEMF method (Mehmani et al. 2015) can be extrapolated to determine the batch size of infill points to be added to the current sample set (X^C), such that it is forecasted to provide a certain level of needed fidelity in future iterations.

The inputs and outputs of PEMF, as used in the AMR method, can be expressed as:

$$\left[\mathbb{P}_{\mathrm{SM}^C}, \varepsilon^{\mathrm{C}}_{mod}\right] = f_{\mathrm{PEMF}}\left(\widehat{y}_{\mathrm{SM}}, X^{\mathrm{C}}, y^{\mathrm{C}}\right) \tag{6}$$

where X^{C} and y^{C} represent the sample data (input and output) used for training the surrogate model. Here, $\mathbb{P}_{\mathrm{SM}^C}$ represents the distribution of the error in the surrogate model $\widehat{y}_{\mathrm{SM}}$, with PEMF predicting the distribution over a median value of error on different cross-validation test sets. This probability distribution is used to perform the hypothesis tests that determine the AMR criteria (4). The mode, $\varepsilon^{\mathrm{C}}_{mod}$, of the error distribution is typically considered the final error measure provided by PEMF.

Model refinement is performed to refine the current surrogate model such that it retains the desired fidelity for a certain number of upcoming iterations of SBO, at the expense of infill points added in the current iteration. The desired fidelity, ε^*_{mod}, is determined using the history of the fitness function improvement in the optimization process, which is given by:

$$\varepsilon^*_{mod} = \left|\frac{\mathbb{Q}^{t=t^*}_{\Theta}}{\mathbb{Q}^{t=t^*-\tau}_{\Theta}}\right| \times \varepsilon^{C}_{mod} \tag{7}$$

The desired batch size (N^{Infill}) is then estimated by using the inverse of the regression functions used to represent the variation of error with sample density in the PEMF method (Mehmani et al. 2015). PEMF uses either a power or exponential regression for computing the extrapolated error measure. Based on the previously estimated "desired fidelity" (ε^*_{mod}), the batch size of infill points can be computed for these two PEMF regression models by using the following expressions:

$$N^{\mathrm{Infill}} = \begin{cases} \text{Power: } \left\lceil \frac{\ln(\varepsilon^*_{mod}) - \ln(a^C)}{b^C} \right\rceil - N^C_s \\ \text{Exponential: } \left\lceil e^{\left(\frac{\ln(\varepsilon^*_{mod}) - \ln(a^C)}{b^C}\right)} \right\rceil - N^C_s \end{cases} \tag{8}$$

where N^C_s represents the size of the current set of sample points, and a^C and b^C are the two coefficients in the PEMF regression functions.

2.2.2 Determining the infill point locations

Here, the location of the new infill points ($\mathrm{X}^{\mathrm{Infill}}|\mathrm{N}(\mathrm{X}^{\mathrm{Infill}}) = \mathrm{N}^{\mathrm{Infill}}$) in the input space are determined using two different approaches: (1) *Global Hypercube*: Infill points are added within a global hypercube (F_{g}), whose lower bound ($L^F_j = X^{\min}_j$) and upper bound ($U^F_j = X^{\max}_j$) w.r.t. the j^{th} dimension are respectively given by the lower ($X^{\min}_j$) and upper ($X^{\max}_j$) bounds of the design space. (2) *Local Hypercube*: Infill points are added within a local hypercube (F_{l}) that encloses promising current candidate designs in the optimization process. The bounds of the local hypercube are defined as $L^F_j = x^{\min}_j$, and $U^F_j = x^{\max}_j$, where, $x^{\min}_j$ and $x^{\max}_j$ are respectively the lower and the upper bounds of the j^{th} design variable spanned by the current population of particles and their individual local best. While the global infill point addition is expected to be relatively more robust to search discrepancies attributed to model errors (as it does not restrict model refinement to specific regions), the local infill point addition is expected to be greedier and thus provides faster convergence.

The optimum location of the new infill points within the previously computed hypercube, i.e., $X^{\mathrm{Infill}} \subset F$, is determined based on their distances from the current sample points (X^C). The objective of this criterion is to minimize the design space correlation between the current and the new points, while seeking to preserve the random uniform distribution of samples. Thus, the infill point locations can be computed by solving the following optimization problem:

$$\max_{X^{\mathrm{Infill}} \subset F} \min_{\mathbf{x}^{\mathbf{s}}, \mathbf{x}^{\mathbf{t}} \in X^{\mathrm{Active}}} \mathbb{D}(\mathbf{x}^{\mathbf{s}}\mathbf{x}^{\mathbf{t}}) \tag{9}$$

where $\mathbb{D}$ denotes the Euclidean distance. X^{Active} is the current data set enclosed by the refinement input space

F, and X^{Updated} is the updated training set that is used to refine (reconstruct) the current surrogate model; i.e., $X^{\text{Active}} = (X^C \subset F) \cup X^{\text{Infill}}$, $X^{\text{Updated}} = X^C \cup X^{\text{Infill}}$, and $N(X^{\text{Updated}}) = N_s^C + N^{\text{Infill}}$. An appealing feature of this distance-based criterion is its ease of implementation in a batch sequential manner. However, the above approach tends to be pre-dominantly explorative; thus, in future, approaches that adaptively balance exploitation and exploration in adding infill samples can be investigated. Note that, due to its model-independent nature, the AMR method can readily use other approaches of locating infill points, without requiring significant modifications of the other steps in AMR.

With the batch of infill points fully determined at this step, the expensive high-fidelity simulation/experiment is evaluated at these infill points, and the generated samples thereof are used to reconstruct the corresponding surrogate model (model structure and kernel choices are kept the same during this reconstruction process, at least for the case studies in this paper). The PEMF method is then called again to evaluate the fidelity of this refined surrogate model, $\epsilon_{mod}\left(X^{\text{Updated}}\right)$, which will be used here onwards in the optimization process to determine the next instance of refinement.

2.3 Optimization algorithm: Particle swarm optimization

In this paper, AMR is implemented within an SBO process that uses the particle swarm optimization (PSO) algorithm (Kennedy and Eberhart 1995). While AMR can be in principle used in conjunction with other population-based optimization algorithms, PSO provides the advantage of easy tracking of population members (namely particles, and the fitness improvement in their local best) over iterations; this readily translates into the distribution of improvement (across iterations) used in AMR. In future, newer formulations can be pursued to quantify the distribution of improvement when extending the AMR technique to work with other evolutionary optimization algorithms, such as genetic algorithms, differential evolution (Tanabe and Fukunaga 2014), and covariance matrix adaptation evolution strategy (CMA-ES) (Hansen 2006).

Specifically, we use an advanced implementation of the PSO algorithm called mixed-discrete PSO (MDPSO), which was developed by Chowdhury et al. (2013). The advantages that the MDPSO algorithm provides over a conventional PSO algorithm include the following: (i) an ability to deal with both discrete and continuous design variables, and (ii) an explicit diversity preservation capability that mitigates premature stagnation of particles. Further description of the MDPSO algorithm can be found in the papers by Chowdhury et al. (2013) and Tong et al. (2016).

In numerical experiments with earlier versions of AMR, we had observed that the optimization can stagnate in regions with a local optimum, where the surrogate model keeps getting repeatedly refined (instead refinements are desirable closer to the region of the global optimum). To counter this phenomena in the current implementation, after each refinement event, we move a randomly chosen particle (other than the global best) to the location of the infill point ($P_i \in X^{\text{Updated}}$) with the best high-fidelity observation of the objective function. This modification was found to provide improved search dynamics in the benchmark problems.

3 Benchmark testing of AMR

In this section, we first use an analytical benchmark problem, the two-dimensional *six-hump camel back function* (Molga and Smutnicki 2005), to illustrate how the AMR technique operates. Then, we provide a comparative analysis using three additional popular benchmark prob-

Table 1 Case studies and problem settings

Case study	Problem	n_d	N_0	N_f	$Iter_{\max}$	p_{cr}	τ
Comparative analysis	Six-hump camel back	2	20	30	100	0.3	2
Further benchmark analysis	Branin-Hoo	2	20	30	100	0.3	2
	5D Dixon-Price	5	50	150	400	0.3	2
	20D Griewank	20	200	300	400	0.7	2
Parametric analysis (p_{cr})	Six-hump camel back	2	20	30	200	0.3	2
	Six-hump camel back	2	20	200	200	0.3	2
Parametric analysis (N_0)	Six-hump camel back	2	[15,20,25]	30	100	0.3	2

N_0, initial investment (sample size); N_f, final (total) sample size; n_d, dimension of the problem; p_{cr}, critical probability (indicator of conservativeness); τ, frequency check of surrogate model refinement

lems (Branin-Hoo, 5D Dixon-Price, and 20D Griewank functions) to demonstrate the effectiveness of AMR w.r.t. standard SBO with no in situ model refinement. Additionally, we perform two parametric studies to show the impact of prescribed parameters on the performance of AMR.

3.1 Case studies and settings

Table 1 summarizes the different benchmark case studies and their solution settings. In order to study how the AMR technique performs in comparison with single-stage SBO-based optimization methods, the corresponding pure high-fidelity optimization, and how the surrogates change with refinement (vs. the fixed surrogates), we use the six-hump camel back function.

We solve the optimization problem using five approaches: (1) *PSO-HF*: find the optimum by applying the MDPSO algorithm on the actual high-fidelity function, and run this 10 times; (2) *PSO-AMR*: generate 10 sets of N_0 samples using LHS and identify the best surrogate model (model type, kernel choice, and hyperparameter values) using the concurrent surrogate model selection (COSMOS) framework (Mehmani et al. 2018); solve the optimization problem 10 times with the surrogate model corresponding to each sample set using the new AMR technique integrated with MDPSO, during which a total of $N_f - N_0$ infill points are added based on the *Global Hypercube* method; (3) *PSO-AMR-Local*: similar to PSO-AMR, except it uses the *Local Hypercube* method for determining the allowed region of the infill points; (4) *PSO-SBO-k1*: fit surrogate models to 10 sets of N_f samples, and use them in the MDPSO algorithm without any in situ refinements (Chowdhury et al. 2013); again MDPSO is run 10 times w.r.t. the surrogate model for each data set; and (5) *PSO-SBO-k2*: similar to PSO-SBO-k1, except we fit to the N_f-sized data sets the same surrogate model and kernel type that is used in PSO-AMR. Using multiple data sets and multiple PSO runs respectively accounts for the stochasticity of the sampling and the PSO optimization processes. The *PSO-SBO-k2* approach allows us to demonstrate that any of the potential improvements provided by PSO-AMR over PSO-SBO is not necessarily attributed to the differences in fitted surrogate models. The fitted surrogate models might be different since they are chosen by the automated model selection method, COSMOS (Mehmani et al. 2018), which uses error measures to identify the model type, kernel type, and hyperparameter values, that best represents the data set (and the data set is different for different sampling sizes). COSMOS makes this choice from a list of candidates that include Kriging, radial basis functions (RBFs), and support vector regression (SVR). For the initial comparative studies, we use the following setting of the IoC: $p_{cr} = 0.3$.

For further comparative analysis among PSO-HF, PSO-AMR, and PSO-SBO-k1, three benchmark problems (Branin-Hoo, 5D Dixon-Price (Dixon and Price 1989), 20D Griewank functions (Griewank 1981)) are used. The sampling approach and multiple optimization run settings are similar to those used for the first case study with the six-hump problem.

Next, we perform two parametric analyses: (1) to study the impact of the critical probability (p_{cr}), which controls the trade-off between optimal solution reliability and computational cost of the optimization process, and (2) to study the impact of the initial investment (N_0), with the total investment remaining the same (where higher initial/total ratio is likely to allow a better surrogate model early on, but lower scope for targeted improvement during the AMR-SBO process). To these ends, we analyze the performance of PSO-AMR on the six-hump camel problem. For the first study, PSO-AMR is run under three different values of p_{cr}, 0.1, 0.5, and 0.9. For this purpose, we randomly generate 10 different sets of 20 samples each using Latin hypercube sampling (McKay et al. 1979) with maximin criterion; and identify the best surrogate model using COSMOS (Mehmani et al. 2018).

Then, we develop two different case studies, one where the total allowed investment is set at $N_f = 30$ and another where it is set at $N_f = 200$. This is to understand if and how the choice of the critical probability is linked with the total available sample investment. For both cases, the PSO-AMR process is run 10 times for each surrogate model (corresponding to each LHS) to find the optimum. For the second study (analyzing the impact of N_0, by varying it among 15, 20, and 25), we similarly generate 10 different samples using LHS and obtain their corresponding surrogate models using COSMOS. We run the AMR approach for each case 10 times, where the critical probability is set at 0.2 (i.e., $p_{cr} = 0.2$), and the maximum investment is set at 30 samples. The population size (N_{pop}) and the maximum iteration (Iter_{max}) are set at 20 and 100, respectively. The iteration frequency of checking the AMR metric is set at $\tau = 2$ for both studies.

3.2 Benchmark testing: Results and discussion

3.2.1 Analyzing AMR performance in comparison with other methods: Six-hump problem

Figure 4a shows the violin plots for each of the five methods, in terms of the high-fidelity evaluation of the objective function at the optimum solution obtained by the methods. These plots illustrate the statistical performance of PSO-HF, PSO-AMR, PSO-AMR-Local, and the two baseline SBO methods for the six-hump problem. The comparison of the

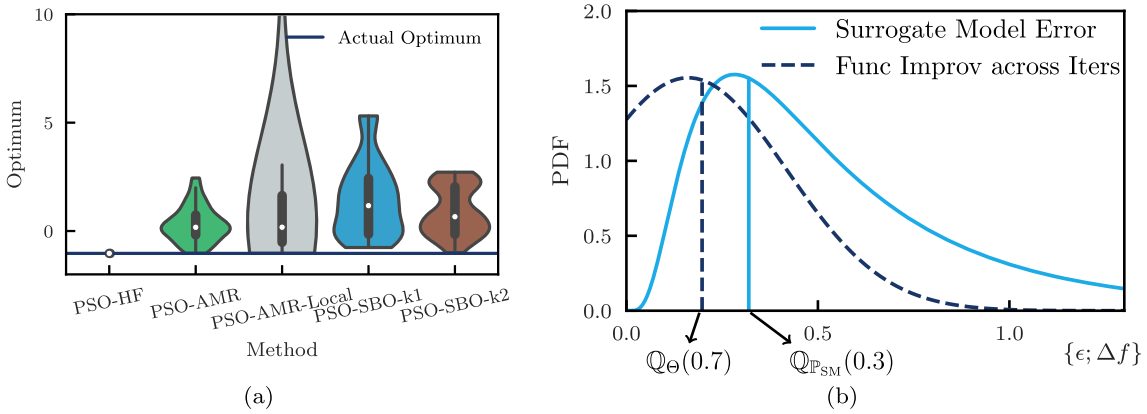


Fig. 4 Comparative analysis study results (six-hump camel back function): **a** optimization results for PSO-AMR, PSO-AMR-Local, PSO-SBO-k1, and PSO-SBO-k2 approaches, over 100 runs each (10 different LHS and 10 runs for each LHS and SM). The numerical values in terms of median and standard deviation for each method are reported in Table 3; **b** PSO-AMR: hypothesis test that triggered the refinement for the selected case of PSO-AMR, shown in Fig. 13b

primary statistical metrics (across 10 × 10 runs), namely median and standard deviation, of the four methods is provided in Table 3. As shown in Fig. 4a and Table 3, PSO-AMR outperformed the two SBO approaches both in terms of the median value and the variation of the results across multiple runs. While providing a median performance similar to PSO-AMR, PSO-AMR-Local expectedly suffers from greater variance across multiple runs. It is also observed that PSO-AMR and PSO-SBO-k2 have at least one outcome each that gets very close to the actual optimum. The worst case outcome of the PSO-AMR is in the interquartile range of PSO-SBO-k1 results. From Fig. 4a, note that the results of PSO-HF, and thus of PSO itself, is near perfect (reaches the optimum in all runs), for the six-hump problem. This outcome allows us to more readily attribute the observed performance superiority of PSO-AMR to the AMR approach itself, compared with the baseline SBO variants.

Table 2 shows the individual best case results, out of 10 × 10 runs each of PSO-AMR, PSO-AMR-Local, PSO-SBO-k1, and PSO-SBO-k2. Here again, the performances of PSO-AMR and PSO-AMR-Local can be seen to be noticeably better than that of the SBO variants w.r.t. closeness to the actual optimum. The best PSO-AMR optimum is only 0.47% away from the optimum obtained by the best case PSO-HF, while the total high-fidelity investment in PSO-AMR is only 10% of the number of high-fidelity evaluations required in PSO-HF. Additional illustrative plots for the six-hump problem are provided in Appendix F (Fig. 13), showing the results of the best individual runs of PSO-AMR, PSO-AMR-Local, PSO-SBO-k1, and PSO-SBO-k2—they mainly display the difference in the true and surrogate model estimated response surfaces (pre/post refinement), and the corresponding convergence histories of the optimization process.

Impact of global/local hypercube on AMR In order to provide an insightful understanding of how the local hypercube– and global hypercube–based infill sampling impact the AMR performance, we select a best case run of *PSO-AMR-Local* based on the smallest normal or standard score. The z-score approach (Glantz et al. 1990) is used

Table 2 Six-hump camel back function: the optimization results for PSO-AMR, PSO-AMR-Local, PSO-SBO-k1, and PSO-SBO-k2 approaches for the selected best case for each approach

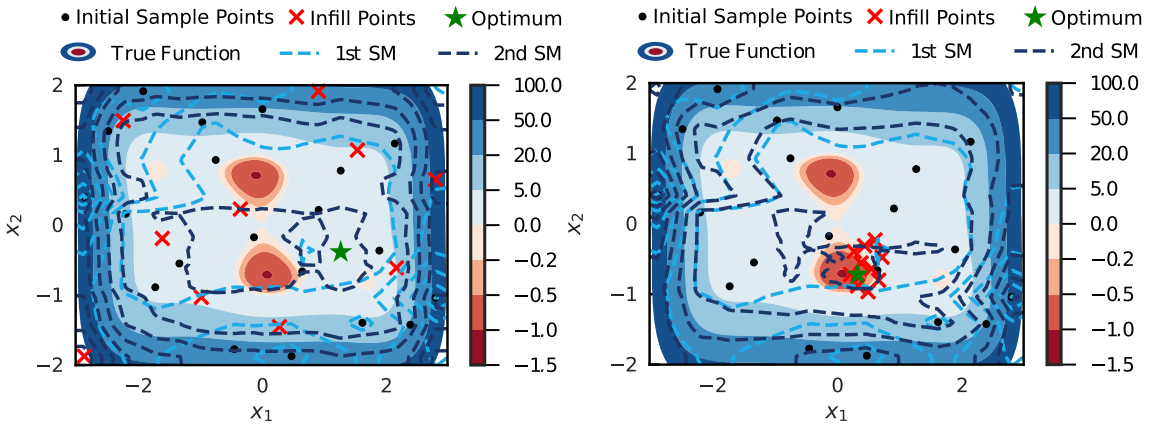
Approach	PSO-HF	PSO-AMR	PSO-AMR-Local	PSO-SBO-k1	PSO-SBO-k2
HF objective Func. value at optimum	−1.0316	−1.0268	−1.0265	−0.7612	−0.9963
RAE w.r.t. actual optimum (−1.0316)	0%	0.47%	0.49%	26.21%	3.42%
No. of HF Func. evaluations (DoE+Infill)	300	20+10	20+10	30+0	30+0
No. of SM Func. evaluations	–	240	400	180	400
Selected surrogate model/kernel	–	Kriging Cubic	Kriging Gaussian	Kriging Linear	Kriging Cubic

to select this run, which gives a measure of how far better is each result compared with the average performance of all results with the same initial LHS sample—this helps in clearly attributing the performance gains to the infill point additions. Illustration of the selected case of PSO-AMR-Local is shown in Fig. 5b, with Fig. 5a showing the illustration of PSO-AMR with global infill sampling for the same case (same initial sampling and surrogate model). In both of these figures, we show the response surface of the true function (filled contours) and those of the surrogate models (dashed contours); the infill points added during the refinement event are depicted by red $\times$ symbols, and the computed optimum solution is depicted by a green star symbol. It can be seen from Fig. 5b that due to better search performance of PSO, PSO-AMR-Local is able to identify a region in the neighborhood containing the true optimum for adding infill points; this in turn significantly improved the accuracy of the surrogate model, and helped achieve an optimum solution very close to the true optimum. In contrast, PSO-AMR experienced a global improvement in the accuracy of the model post-refinement, which did not particularly help in getting close to the true optimum (as seen from Fig. 5a), since the model accuracy did not significantly improve in the neighborhood of the optimum. Overall, this demonstrates that effective regional sampling can significantly benefit the AMR process, while being dependent on the performance of optimization algorithm's search process, which is stochastic in the case of PSO and other heuristic algorithms. Thus, the localized or regional sampling benefits are not guaranteed across different runs of the same problem under the current implementation (as seen from the violin plots in Fig. 4a). Therefore, regional sampling approaches that are more adaptive to the status of the PSO population w.r.t. exploitation/exploration balance is needed in the future.

3.2.2 Case study 2: Further benchmark analyses

Figure 6 shows the results obtained using different approaches for the three additional test problems, namely Branin-Hoo, 5D Dixon-Price, and 20D Griewank functions. Similar to the six-hump problem results, here the results are again shown in terms of violin plots over the respective multiple runs of each method. The main statistical metrics, i.e., median and standard deviation of the computed optimum, for these additional benchmark problems are also summarized in Table 3. We see from Fig. 6 that the *PSO-HF* approach finds the true optimum in all runs for these problems, thus again serving in an effective role of a benchmark to compare against.

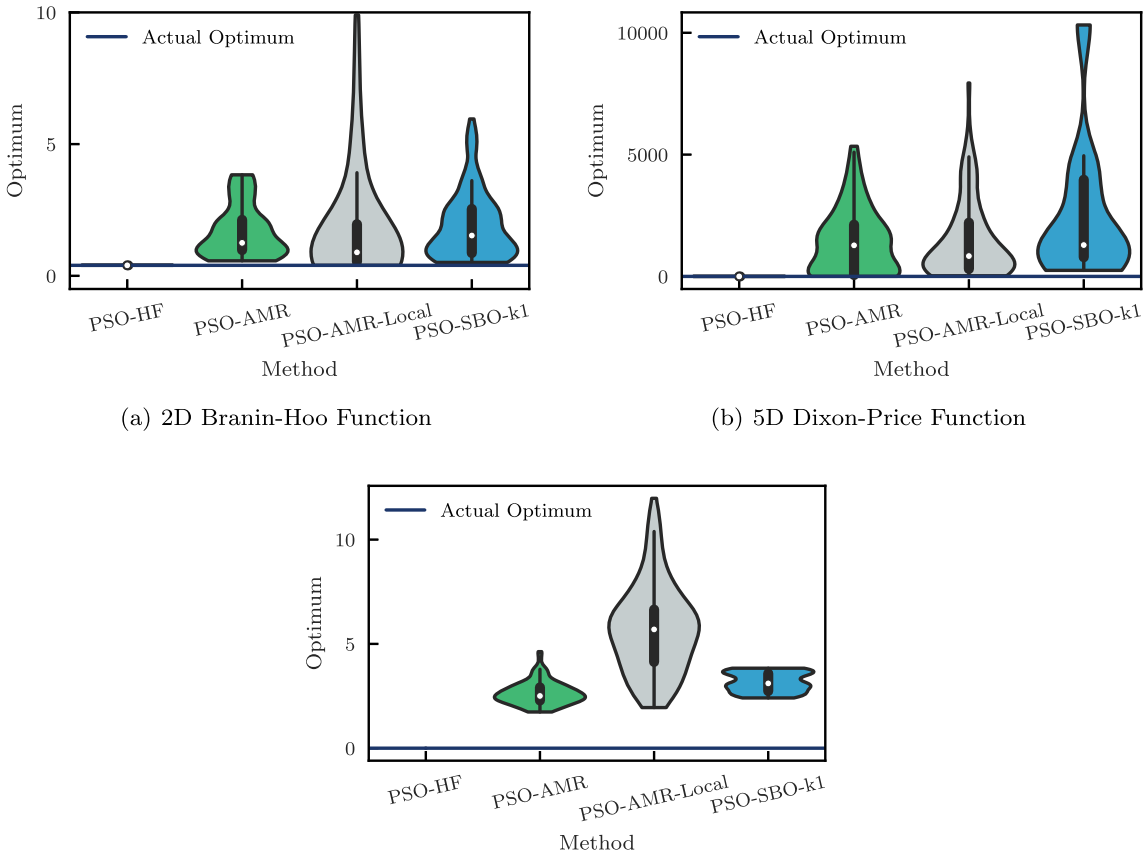
For the Branin-Hoo problem, Fig. 6a and Table 3 show that PSO-AMR-Local outperforms the PSO-AMR and PSO-ABO-k1 in terms of the median value and the best optimum (that gets closest to the true optimum), while experiencing larger variation in performance (across multiple runs) compared with the latter two methods. In the case of the 5D Dixon-Price problem (Fig. 6b), PSO-AMR-Local again provides the better median value compared



(a) PSO-AMR: contour plot

(b) PSO-AMR-Local: contour plot

Fig. 5 The results of PSO-AMR and PSO-AMR-Local for the six-hump problem, w.r.t. the same initial sample case that gives the best z-score for PSO-AMR-Local. The filled contour shows the true response surface. The light blue and dark dark dashed lines represent the initial surrogate model (1st SM) and the updated surrogate model after refinement (2nd SM), respectively. The black dot, red cross, and green star markers respectively depict the original sample points (same for both), the infill points, and the computed optimum given by each method



(a) 2D Branin-Hoo Function

(b) 5D Dixon-Price Function

(c) 20D Griewank Function

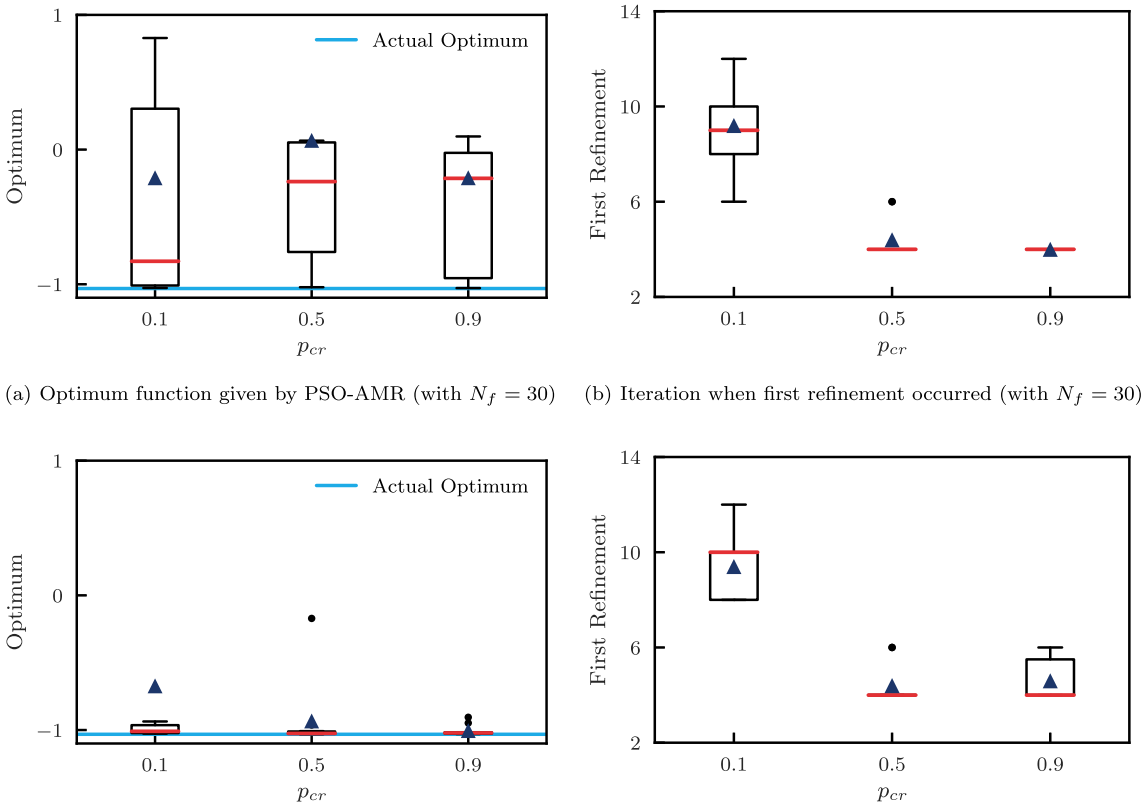
Fig. 6 Optimization results for PSO-HF, PSO-AMR, PSO-AMR-Local, and PSO-SBO-k1 approaches; **a** each case executed 100 times (10 different LHS and 10 runs for each LHS and SM); **b** each case executed 100 times (10 different LHS and 10 runs for this LHS and SM); **c** each case executed 100 times (10 different LHS and 10 runs for each LHS and SM). For all three cases, the numerical values in terms of median and standard deviation for each method are reported in Table 3

with PSO-AMR and PSO-SBO-k1, with its variance also being comparable to PSO-AMR. These observations point to the potential of the localized sampling approach in AMR, particularly from a median performance perspective. Lastly, for the 5D Dixon-Price problem, it can be seen that while the SBO results are comparable to the AMR results in terms of the median value, PSO-SBO-k1 experiences significantly large variance, and unlike AMR, it does not get close to the true optimum in any run.

For the 20D Griewank problem, it can be observed from Fig. 6c and Table 3 that PSO-AMR provides better median performance than PSO-SBO-k1, and gets closer to

Table 3 Optimization results for PSO-HF, PSO-AMR, PSO-AMR-Local, PSO-SBO-k1, and PSO-SBO-k2 approaches: the median performance and the standard deviation in parenthesis are given for 100 times (10 different LHS and 10 runs for each LHS)

Method	Six-hump camel	Branin-Hoo	5D Dixon-Price	20D Griewank
PSO-HF	−1.032 (0.001)	0.398 (0.001)	0.049 (0.172)	0 (0)
PSO-AMR	0.172 (0.811)	1.251 (0.991)	1.276e3 (1.357e3)	2.505 (0.552)
PSO-AMR-Local	0.176 (9.417)	0.891 (3.361)	0.836e3 (1.578e3)	5.784 (2.595)
PSO-SBO-k1	1.170 (1.733)	1.529 (1.265)	1.287e3 (2.828e3)	3.106 (0.469)
PSO-SBO-k2	0.660 (1.095)	–	–	–
True optimum	−1.032	0.398	0	0



(a) Optimum function given by PSO-AMR (with $N_f = 30$) (b) Iteration when first refinement occurred (with $N_f = 30$)

(c) Optimum function given by PSO-AMR (with $N_f = 200$) (d) Iteration when first refinement occurred (with $N_f = 200$)

Fig. 7 Six-hump camel back function: analyzing the impact of critical probability (p_{cr}) on PSO-AMR. The red line and triangle depict the median and mean values, respectively

the true optimum with its best run (compared with that of PSO-SBO-k1). All surrogate-based methods noticeably trail the performance of PSO-HF for this problem, unlike in the lower dimensional problems; this is a persisting issue with surrogate modeling itself, when dealing with high-dimensional problems. In this 20D problem, PSO-AMR-Local in particular performs poorly, since the expected low fidelity of the surrogate tends to misdirect particles into undesirable regions, and local refinement approach exacerbates this issue.

For further insights on PSO-AMR's performance on this high-dimensional problem, we compare it with a standard implementation of EGO.[1] Here, EGO is similarly run 100 times. PSO-AMR outperforms EGO in terms of both the median value (2.505 vs. 109.280) and the variance (0.552 vs. 53.750) of the computed optimum. The EGO results obtained here are well aligned with those reported by Cheng et al. (2015), for this same 20D problem. Given that the focus of this paper is on AMR, more comprehensive comparison of PSO-AMR (which couples PSO and AMR performance) with other multi-fidelity optimization methods is considered to be a direction of future work.

[1] A Matlab implementation based on Jones et al. (1998)

3.2.3 Case study 3: Parametric analysis of AMR (using the six-hump camel function)

Impact of the critical probability (p_{cr}) Figure 7 shows how the prescribed critical probability (p_{cr}) influences the performance of AMR, under two different total (i.e., initial + infill) sample investments (N_f), with the initial investment (N_0) remaining the same. When comparing the outcomes under $N_f = 30$ (Fig. 7a) and $N_f = 200$ (Fig. 7c), it is readily evident that p_{cr} has a more noticeable impact on performance at sparser sample investments. The observed higher variance in general with $N_f = 30$ can be attributed to the larger variation in model accuracy due to the significantly smaller sample size. Expectedly, in both sample size cases, increasing the value of p_{cr}, which facilitates lesser compromise in fidelity, led the refinement events to occur much earlier (as seen from

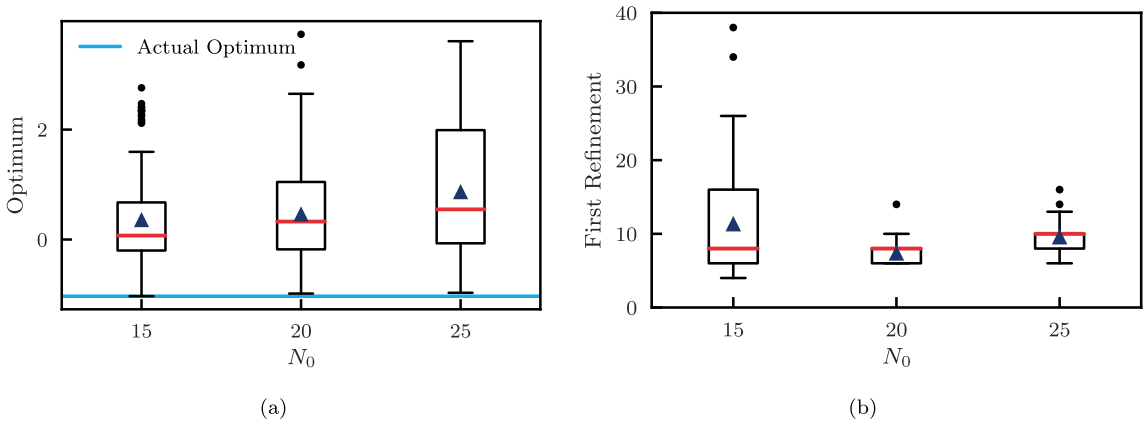


Fig. 8 Six-hump camel back function: analyzing the impact of the initial investment (N_0) on PSO-AMR. The red line and triangle depict the median and mean values, respectively

Fig. 7b and d). As a result, even though this helps in more aggressively preserving the fidelity of the search process early on, the swarm soon runs out of infill points when the total investment is capped at a smaller size (30 vs. 200). The search process in PSO is typically more explorative toward the start and becomes more localized later on, and thus the opportunity to achieve higher fidelity in smaller promising neighborhoods (i.e., close to the the optima) is lost if the process runs out of infill points, and thus unable to refine the objective function response in these smaller neighborhoods—which leads to the observed drop in performance with higher p_{cr} when total investment is frugal or sparse.

Impact of the initial sample investment (N_0) For this purpose, we use three different initial sample sizes (N_0): 15, 20, and 25, with the total investment ($N_f = 30$) remaining the same. The results obtained are shown in Fig. 8.

It can be observed that at least for the given problem, both the median performance and the variance (both smaller the better) improved with decreasing initial investment; the improvement in variance being more apparent. This observation can be attributed to the usage of greater opportunity for refinement where required in the design space, when PSO-AMR has more infill points available to it (i.e., when N_0/N_f is smaller). At the same time, we observe a large variance in the timing of the first refinement event in the case of the smallest initial investment of $N_0 = 15$, which can be attributed to the likely large variance in the accuracy of the initial SBO (across the ten LHS samples) when the initial sample size is small.

4 Application of AMR: Building energy management

For studying the effectiveness of AMR in more complex optimization applications, we use a *building thermal management* problem, which involves optimal planning of temperature settings for a large office building, with the

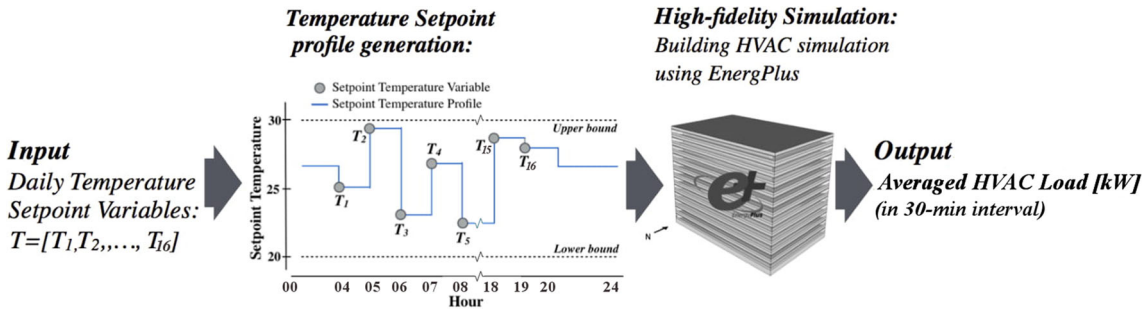


Fig. 9 Large office building problem: input and output in the high-fidelity simulation using EnergyPlus

objective to minimize energy consumption attributed to cooling loads. Further description of the problem, results of AMR application on it, and comparisons with high-fidelity optimization and standard SBO are presented in this section.

4.1 Electricity demand for cooling a large office building

This problem is based on a representative 12-floor building that utilizes an HVAC system for cooling during the summer season. An illustration of the problem is given in Fig. 9. The problem is defined as finding the temperature set point profile from 4 am to 8 pm on a typical hot summer day, such that it minimizes (1) the electricity demand for the cooling system of the building and (2) the indoor temperature deviation from a fixed comfortable temperature reference (here, set at 24°C). The day-temperature set point profile is defined using a step function (see Fig. 9), which is updated through a 16-element set point vector, essentially the hourly temperature set points ($T = [T_1, T_2, ..., T_{16}]$) between 4 am and 8 pm. Hence, the optimal planning problem can be defined as:

$$\min_{\mathbf{T}} f = c_1 E(\mathbf{T}) + c_2 \sum_{j=1}^{16} |T_j - 24| \tag{10}$$

where, $\mathbf{T} = [T_1, T_2, \ldots, T_{16}]$ and $E(.)$ are the hourly temperature set point profile and the hourly averaged energy consumption over the studied daily 16-h period, respectively. The coefficients $c_1 = 106,300$ and $c_2 = 3,130$ are used for scaling the two terms in order to allow a reasonable single-objective formulation. Although the PSO algorithm used here can solve multi-objective problems (Tong et al. 2016) and AMR can in principle be expanded to multi-objective implementations, such explorations are outside of the scope of this paper. For solving this optimization problem, we require a model to predict the HVAC load ($E(.)$). In this paper, as illustrated in Fig. 9, the EnergyPlus (v8.9) building energy simulation software is used to estimate the average 30-min interval cooling electricity load in a prototypical large office building. The original office building model was developed in Deru et al. (2011) and DOE (2017), and modified here based on the typical meteorological conditions for New York City (New York-Central Park 725033 (TMY3)). This model is popularly used for research in computational methods for building energy prediction and operations planning (Tyler and Zhang 2015; Chen and Hong 2018).

The execution time for a single EnergyPlus simulation of this building is observed to be $\approx$ 60 s on a 2.4 GHz Intel Xeon 6148 CPU / 200 GB RAM high-performance workstation. Using such simulations as a part of an optimization process (that could require 100–1000 s of evaluation) is computationally prohibitive in the context of near real-time (or even hour-ahead) planning, which is required for applications such as thermostat automation (Corbin et al. 2013; Sharif and Hammad 2019) and effectively dispatching distributed energy storage in buildings that are subject to demand-based electricity tariffs (Sun et al. 2018; Wang et al. 2018; Ghassemi et al. 2017). Hence, the SBO techniques are popular substitutes for such computationally expensive simulations in the building energy management domain (Tian 2013; Ascione et al. 2017; Chen et al. 2017). With a typical surrogate model's (e.g., ANN's, GP's, or RBF's) nearly ten orders of magnitude smaller computing time footprint compared with the EnergyPlus simulation model (when simulating a 1-day period), a typical SBO approach can facilitate tractable decision-making toward optimal HVAC operations.

4.2 Building cooling planning problem: Results and discussion

Here, we again use COSMOS (Mehmani et al. 2018) to select the surrogate model for the building cooling problem, where the input to the surrogate model is the 16-dimensional temperature profile ($\mathbf{T}$) and the output is the hourly averaged energy consumption over a whole day (E). In this problem, for the SBO implementations, a total of 320 high-fidelity samples (evaluated using EnergyPlus) are used. The prescribed settings used in COSMOS and in the MDPSO algorithm for this problem are summarized in Appendix C.

For comparative analysis, we consider three different implementations: (1) *PSO-AMR:* performing the new AMR based SBO implemented with MDPSO, where 160 points are used to build the initial surrogate model and another 160 are added in batches for refinement during the optimization process; (2) *PSO-SBO:* performing standard surrogate-based optimization with MDPSO, where 320 samples are used to build the surrogate model that is then used to perform the optimization; and (3) *PSO-HF:* high-fidelity optimization with MDPSO, where function evaluations are performed directly using the EnergyPlus simulations. For the PSO-AMR, we use the following settings: $\tau = 3$, $p_{cr} = 0.1$, $\text{iter}_{\max} = 200$.

Figure 10a–c show the convergence histories (solid lines) and the PEMF-derived model error (dashed lines), where

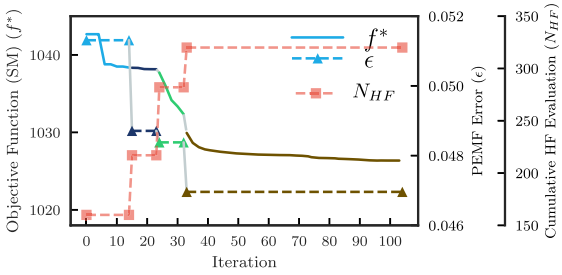
pertinent, during the three different types of optimizations. The line color changes after each refinement event in the case of the PSO-AMR plot (Fig. 10a). The relatively higher number of iterations required by PSO-AMR to converge can be attributed to the typical re-shuffling of particles' (individual best) ranks that occur after each refinement event, which re-organizes the search dynamics. As shown in Fig. 10a, the AMR technique adaptively refined the surrogate model three times during the optimization process, namely at the 15th, 24th, and 33rd iterations, by adding batches of 57, 65, and 38 infill points, respectively. The global error in the surrogate model subsequently decreased from 5.1 to 4.7%. In contrast, the global error in the surrogate model remains at ∼4.9% in the case of the PSO-SBO approach, which uses a fixed surrogate model. Significant improvement in the objective function is observable in the case of all three optimization processes; however, note that the displayed convergence histories for the PSO-AMR and PSO-SBO approaches relate to surrogate-based function values, while that of PSO-HF relate to actual simulations, and thus direct comparison of the quality of the optimum is better judged by evaluating all the three optimized designs with high-fidelity EnergyPlus simulations, which is discussed later.

Figure 11a shows the optimized temperature set points given by the three approaches. It can be observed that, compared with the optimized temperature set points obtained by PSO-SBO, those given by the new PSO-AMR are relatively much closer to the optimized set points resulting from PSO-HF—with the median and maximum deviations being 0.13 °C and 0.7 °C, respectively. The nature of the AMR- or HF-optimized set point profile is intuitively reasonable given the typical daily outdoor weather patterns of cooler mornings and warmer afternoons; thus, a minor compromise in comfort by setting the temperature at slightly higher than 24 °C (i.e., closer to outdoor conditions) yields significant energy savings.

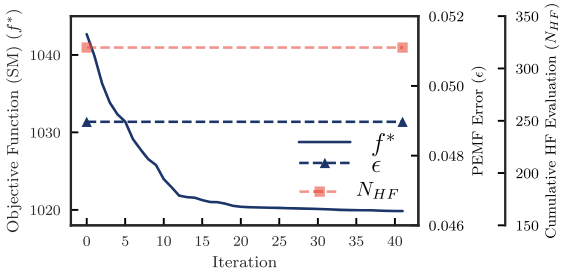
Here, the objective function value is a combination of two terms with different units and physical meanings. Thus, to allow a more intuitive comparison of results, we introduce a metric that provides a normalized understanding of the performance improvement, compared with a baseline—a *Naive* approach where the set point temperatures are all fixed at 24 °C. This metric is defined as:

$$Q = 100 \times \frac{f^*_{\text{Naive}} - f^*_{\text{K}}}{f^*_{\text{Naive}} - f^*_{\text{HF}}} \tag{11}$$

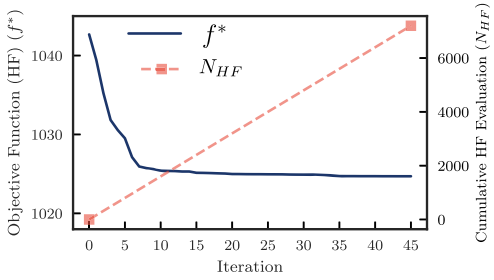
where f^*_{HF} and f^*_{K} are the optimum obtained by PSO-HF and an approach K $\in$ {PSO-SBO,PSO-AMR}, both estimated using high-fidelity simulations. Here, f^*_{Naive} is the objective function value corresponding to the *Naive* approach. This metric thus measures how close approach K



(a) PSO-AMR: convergence and SM error histories

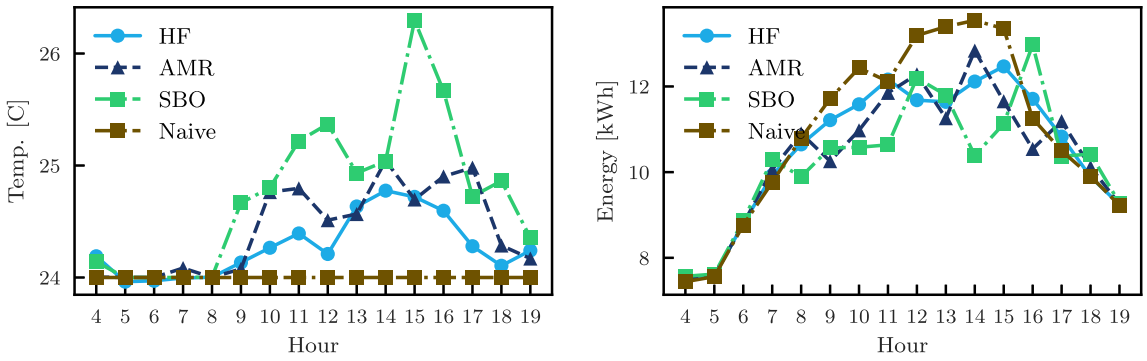


(b) PSO-SBO: convergence and SM error histories



(c) PSO-HF: convergence history

Fig. 10 Building cooling problem: convergence history and results of the three optimizations. The solid line shows the objective function (estimated value using surrogate model approximations, in (**a**) and (**b**), and the actual value in (**c**)). The dashed line with triangle marker represents the relative error of the surrogate model, which is estimated using PEMF. The dashed red line with square marker shows the cumulative number of high-fidelity function evaluations



(a) Optimized temperature set-points

(b) Optimized energy consumption

Fig. 11 Building cooling problem: results of the three optimizations

can get to the performance of the high-fidelity optimization, with 100% meaning a perfect match.

Table 4 provides a summary of the comparative results, including the Q-metric, the computed optimum objective function value, the actual objective function value at optimum given by the high-fidelity simulation, the relative error in the computed vs. actual optimum values, the cost of optimization in terms of number of high-fidelity simulation calls, and the number of surrogate model calls. It can be seen from Table 4 that compared with PSO-SBO ($Q = 55.22\%$), the optimum solution given by the new PSO-AMR ($Q = 89.55\%$) is significantly closer in performance to PSO-HF. The closeness of the actual objective function values at optimum as given by PSO-AMR and PSO-HF (1026.1 vs. 1024.7) further corroborates this observation. Thus, in terms of accuracy, PSO-AMR clearly outperforms PSO-SBO for this building cooling problem.

Table 4 shows that, as expected, a majority of function evaluations in both surrogate-based optimizations (PSO-SBO and PSO-AMR) are done using the current surrogate model. More importantly, PSO-HF requires 7680 high-fidelity function calls (EnergyPlus simulations), compared with the the total of 320 simulations used by PSO-SBO and PSO-AMR methods (for constructing and refining the surrogate). On the workstation used for this case study, this translated into a computing time of $\approx$ 6.5 h for PSO-AMR (including generation of initial samples, constructing the initial model, and running PSO-AMR with the in situ sample generation and model refinements), whereas PSO-HF required a computing time of $\approx$ 132 h (i.e., 5.5 days). The observed 20-fold computational time savings (over pure HF optimization), when taken in the context of the accuracy of the optimum solution obtained by PSO-AMR and the attractive fidelity of the SM estimation at the optimum (only 0.02% error), demonstrates the effectiveness of the AMR method in balancing fidelity and computational efficiency.

5 Conclusion

In this paper, we presented a new surrogate-based optimization (SBO) approach called adaptive model refinement or AMR, which seeks to preserve the reliability of the search

Table 4 Optimal set point planning in cooling a large office building: results using the AMR, the standard surrogate-based optimization (SBO), and the purely high-fidelity (HF) optimization methods

Approach	PSO-AMR	PSO-SBO	PSO-HF	Naive
Q (Eq. 11)	89.55%	55.22%	100%	0%
Computed optimum objective value (f^*)	1026.29	1019.49	1024.70	–
Actual objective value at optimum (HF estimation, f^*_{HF})	1026.10	1038.10	1024.70	2076.20
Relative absolute error ($100(f^* - f^*_{HF})/f^*_{HF}$)	0.02%	1.79%	0%	–
Number of HF function evaluations (DoE+Infill)	160+160	320+0	14,720+0	1+0
Number of SM function evaluations	17,280	6,880	0	0
Selected surrogate model/kernel	Kriging Exponential	Kriging Exponential	—	—

Here, f^*_{HF} represents the HF estimation of the objective function value at the optimum obtained by the corresponding optimization method (given by the column heading)

process during optimization without compromising computational efficiency. The AMR approach can work with any major types of surrogate models (e.g., RBF, Kriging, and ANNs), and seeks to exploit the following information available in a population-based (e.g., PSO-based) optimization process: (i) stochastic measure of improvement of the population's fitness across generations, and (ii) regions of interest in the design space based on the population distribution at any given iteration. The AMR method devises and uses a hypothesis testing to determine when to add infill points to refine the model, in situ optimization. A predictive sequential cross-validation approach called PEMF is used to perform the hypothesis testing as well as to compute the number of infill points needed at each refinement event. For investigating the effectiveness of the AMR technique, we tested it on three benchmark problems (run multiple times to account for the effect of sampling and PSO's stochasticity) and a more complex practical application (optimizing the temperature set points for cooling a building). In the benchmark problems, AMR readily outperformed standard one-step SBO implementations, both in terms of the median value and the variance of the optimum over multiple runs. Further analysis with one of the benchmark problems (the camel function) demonstrated the role that the adaptive refinement process played in enabling greater response accuracy when and where needed. Parametric analyses with regard to the critical probability and the initial investment showed that the former has a notable impact on performance for sparse data sets, while increasing the initial investment (the total investment remaining the same) had a detrimental impact, likely due to the associated reduction in opportunities to add infill points.

The application problem was defined as finding the hourly optimal temperature set points (from 4am to 8pm) that minimizes the electricity consumption for the cooling system of a prototypical building in NY City and the deviation from a defined comfortable indoor temperature. High-fidelity samples were given by the state-of-the-art EnergyPlus simulations. A purely high-fidelity optimization (run until a convergence criterion is met) and standard one-step SBO was performed along with AMR for comparison (with the latter two using the same total number of high-fidelity samples). While the optimum objective function yielded by the SBO approach deviated by 45% from the solutions obtained by the high-fidelity optimization, corresponding AMR outcomes got within ∼10% of the high-fidelity solutions. This observation was further corroborated by the optimized hourly set point profile given by AMR being noticeably closer to that of the high-fidelity optimum, in comparison with that given by the standard SBO approach. In addition, AMR allowed a 20-fold reduction in computing time compared with the purely high-fidelity optimization. Overall, both the benchmark analyses and the application problem outcomes provided promising evidence of AMR's advantage over standard surrogate-based optimization, by virtue of the former's ability to adapt sample investments to the needs of the optimization search process.

An existing simplicity in the current AMR method is the explorative distance-based criterion and choice of design space bounds to determine the location of infill points at each refinement event, which in future can be replaced with a method that better balances exploration/exploitation trade-offs. A corollary direction of future investigation would be a comparison of this advanced AMR-PSO integration (with adaptive placement of infill points) with state-of-the-art multi-fidelity optimization methods such as efficient global optimization. In addition, while PSO allows tracking of the history of individual particles and thus their fitness improvement, other meta-heuristics such as evolutionary algorithms make it challenging to track individual improvement due to substantial mixing of candidate traits over generations (due to a backward tree-like heritage structure). Therefore, it would be useful to develop additional integration of AMR in the future, where it can work with evolutionary algorithms such as GA, DE, and CMA-ES. Other potential directions of extending the benefits of AMR include incorporation of samples from physics-based models of varying fidelity and application to multi-objective search processes.

Funding information Support from the National Science Foundation (NSF) Award CMMI-1642340 is gratefully acknowledged.

Compliance with ethical standards Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF.

Conflict of interest The authors declare that they have no conflict of interest.

Replication of results To aid the replication of results, data and trained models associated with the numerical experiments presented in this paper have been made available through the following public repository: https://github.com/adamslab-ub/amr-samples-metamodels-package.

Appendix A: Kernel density estimation (KDE)

KDE is a non-parametric model to estimate the probability density function of random variables. Here, it is assumed that $\Delta f = (\Delta f_1, \Delta f_2, ..., \Delta f_{N_{\text{pop}}})$ is an independent and identically distributed sample drawn from a distribution

with an unknown density $\Theta_{\Delta f}$. The kernel density estimator can then be used to determine $\Theta_{\Delta f}$, as given by:

$$\hat{\Theta}_{\Delta f}(x;\, H) = \frac{1}{N_{\text{pop}}} \sum_{i=1}^{N_{\text{pop}}} K_H(x - x_i) \tag{12}$$

Here, the kernel $K(x)$ is a symmetric probability density function, H is the bandwidth matrix which is symmetric and positive-definite, and $K_H(x) = |H|^{-1/2} K(H^{-1/2}x)$. The choice of K is not as crucial as the choice of the H estimator for the accuracy of the KDE (Epanechnikov 1969). In this article, we consider $K(x) = (2\pi)^{-d/2} \exp(-0.5x^T x)$, the standard normal throughout. The mean integrated squared error (MISE) method is used as a criterion for selecting the bandwidth matrix H (Duong and Hazelton 2003) as follows:

$$\text{MISE}(H) = \mathbb{E}\Big[\int \Big(\Theta_{\Delta f}(x;\, H) - \hat{\Theta}_{\Delta f}(x;\, H) \Big)^2 \Big] \tag{13}$$

Appendix B: Particle Swarm Optimization

Particle swarm optimization (PSO) is a population-based optimization method introduced by Kennedy and Eberhart (1995). In this method, each particle's movement is described in terms of its velocity ($\mathbf{v}_i(t)$) and its location ($\mathbf{x}_i(t)$), where i denotes the ith particle and t denotes the tth iteration. Here, we specifically exploit the MDPSO algorithm developed by Chowdhury et al. (2013), which includes explicit diversity preservation in addition to the standard PSO dynamics, in order to provide greater robustness. In MDPSO, the velocity and location of particles are updated as follows:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \tag{14}$$

$$\mathbf{v}_i(t+1) = \omega \mathbf{v}_i(t) + r_1 C_1 (\mathbf{P}_i^l(t) - \mathbf{x}_i(t)) + r_2 C_2 (\mathbf{P}^g(t) - \mathbf{x}_i(t)) + r_3 \gamma_c \hat{\nu}_i(t) \tag{15}$$

Here, $\mathbf{x}_i(t)$ and $\mathbf{v}_i(t)$ respectively denote the position and the velocity of particle i at the t^{th} iteration; ω, C_1, and C_2 represent the inertial weight, the individual search, and the global search coefficients, respectively; these are used to balance the local search (exploitation) and the global search (exploration); $\mathbf{P}_i^l(t)$ is the local leader of particle i at the t^{th} iteration, which represents the best local solution found in the motion history of particle i; $\mathbf{P}^g(t)$ is the global leader of the entire swarm at the t^{th} iteration, which is determined by comparing the local leaders of all particles; γ_c is the coefficient used to weigh the explicit diversity preservation component; $\hat{\nu}_i(t)$ is the explicit diversity preservation vector; and r_1, r_2, and r_3 are random real numbers between 0 and 1.

Appendix C: Predictive estimation of model fidelity

Predictive estimation of model fidelity (PEMF) method (Mehmani et al. 2015) can be perceived as a novel sequential implementation of k-fold cross-validation, with carefully constructed error measures that are significantly less sensitive to outliers and the DoE (compared with mean or root mean square error measures). The PEMF method predicts the error by capturing the variation of the surrogate model error with an increasing density of training points (without investing any additional test points). Algorithm 1 summarizes the PEMF method.

Algorithm 1 Predictive estimation of model fidelity (PEMF).

INPUTS: Number of sample points N, sample set (X, F);
Set Number of iterations N_{it}, indexed by t;
Set Size of intermediate training points at each iteration, n^t, where $n^t < n^{t+1}$
Set Number of heuristic subsets of training points at each iteration equal to M^t, where $\binom{M_t \leq N}{n_t}$, indexed by k.

X=Experimental Design(N)
$F \mid X$= Evaluate System (X)
$\{X\} = \{(X_i, F_i)_{i=1}^N\}$
for $t = 1, ..., N_{\text{it}}$ **do**
  for $k = 1, ..., M^t$ **do**
    Choose $\{\beta\} \subset \{X\}$, where $\#\{\beta\} = n^t$
    Define intermediate training points,$\{X^{TR}\} = \{\beta\}$
    Define intermediate test points,$\{X^{TE}\} = \{X\} - \{\beta\}$
    Construct intermediate surrogate S_k using $\{X^{TR}\}$
    Estimate median and maximum errors; $E_{med}^{k,t} = median(e_m)_{m=1,...,\#\{X^{TE}\}}$ $E_{max}^{k,t} = max(e_m)_{m=1,...,\#\{X^{TE}\}}$
  end for
  Fit distributions of the median error over all M^t combinations
  Determine the mode of the median and maximum error distributions; $E_{med}^{mo,t}$ and $E_{max}^{mo,t}$
end for
Construct a final surrogate using all N sample points
Use the estimated $E_{med}^{mo,t}$ and $E_{max}^{mo,t}$ $\forall t$, to quantify their variation with # training points (n_t) using regression functions.
Return: The modal values of the median and the maximum errors in the final surrogate; ε_{med} and ε_{max}

Appendix D: The settings of COSMOS and MDPSO

Table 5 The COSMOS and MDPSO settings for the analytical problem and application problem

	Parameter	Value
PEMF	Error type	Median
	Normalization mode	Actual value
	Error type	Median
	No. of permutations (M_t)	300
	No. of training points per Iter. (n_t)	$\lfloor \max(0.05N_s, 3) \rfloor$
	No. of iterations (N_{it})	4
COSMOS	Model type	'Kriging,' 'RBF,' 'SVR'
	Kernel choices for Kriging	'Gaussian,' 'Linear,' 'Exponential,' 'Cubic,' 'Spherical'
	Kernel choices for RBF	'Linear,' 'Cubic,' 'Tps'
	Kernel choices for SVR	'Linear,' 'RBF'
MDPSO	Max. Iter.	$\text{Iter}_{\max}$
	Population size (N_{pop})	$10n_d$
	Allowed number of function calls	$\text{Iter}_{\max} \times N_{\text{pop}}$
	γ_{c0}	2
	$\gamma_{\min}$	1.0e−05
	λ_h	0.1
	ω	0.5
	$C_1(\beta_l)$	1.4
	$C_2(\beta_g)$	1.4
	N_{ter}	5

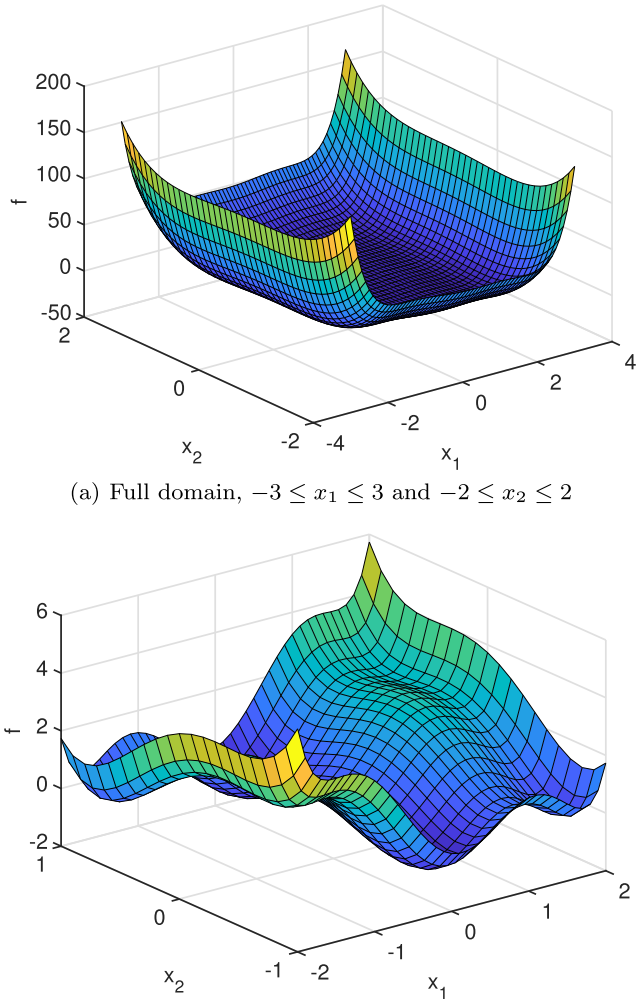
Appendix E: Description of six-hump camel back function

Figure 12 shows the six-hump camel back function, a well-known 2D test function used for benchmarking global optimization and surrogate modeling methods (Molga and Smutnicki 2005). It has six local minima, with two of them being the global minima (Molga and Smutnicki 2005). The global minima are located at $(-0.0898, 0.7126)$ and $(0.0898, -0.7126)$, and gives a minimum function value of $f(x^*) = -1.0316$.

$$f(x_1, x_2) = \left(\frac{x_1^4}{3} - 2.1x_1^2 + 4\right)x_1^2 + x_1x_2 + 4\left(x_2^2 - 1\right)x_2^2 \tag{16}$$

where $-3 \le x_1 \le 3$ and $-2 \le x_2 \le 2$.

Appendix F: Further performance illustration on the six-hump problem

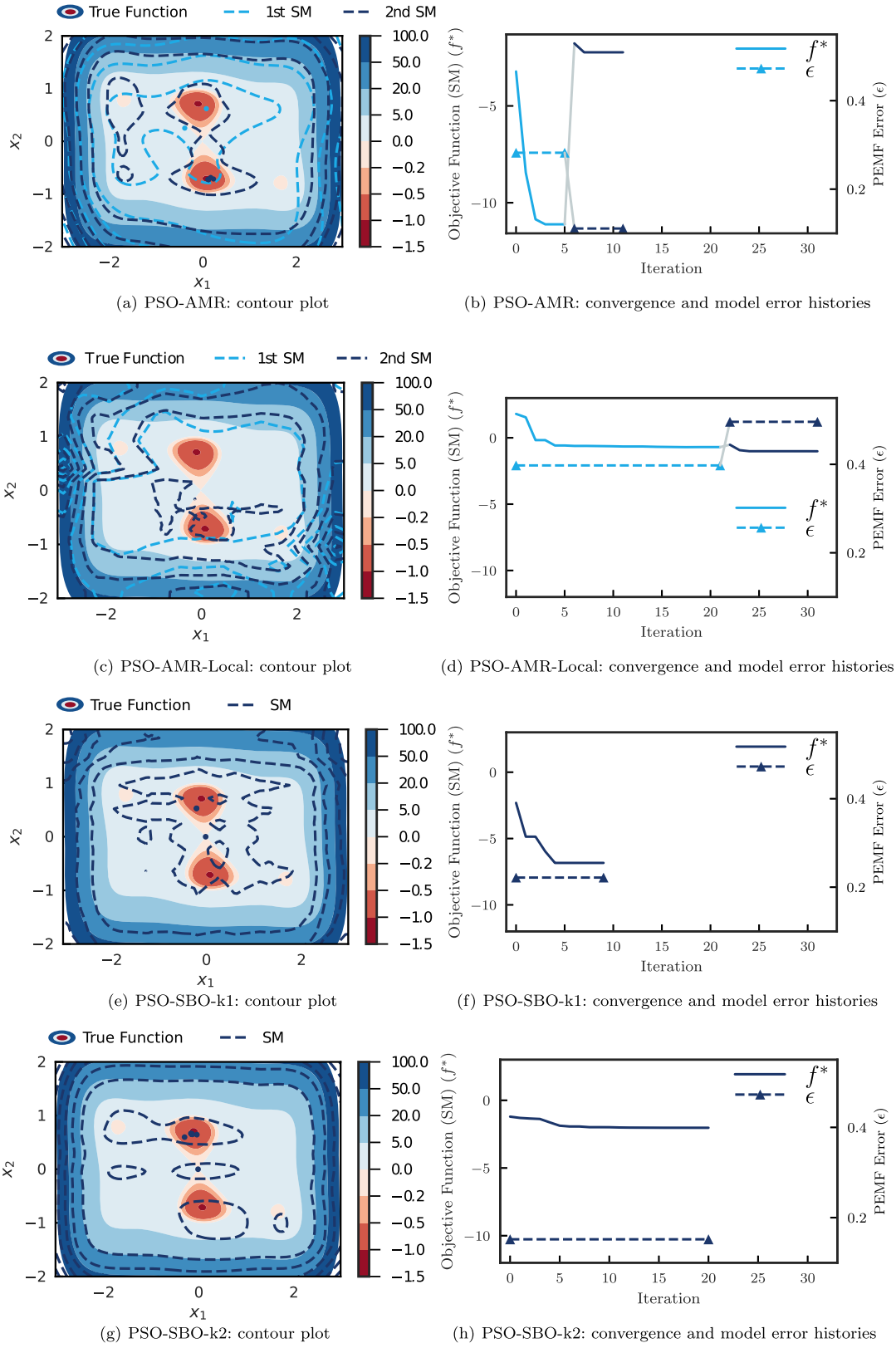


(a) Full domain, $-3 \le x_1 \le 3$ and $-2 \le x_2 \le 2$

(b) Enlarged to show the multiple local minima

Fig. 12 The response surface of the six-hump camel back function

The following illustrations in Fig. 13 respectively represent the best individual run of PSO-AMR, PSO-AMR-Local, PSO-SBO-k1, and PSO-SBO-k2.



(a) PSO-AMR: contour plot

(b) PSO-AMR: convergence and model error histories

(c) PSO-AMR-Local: contour plot

(d) PSO-AMR-Local: convergence and model error histories

(e) PSO-SBO-k1: contour plot

(f) PSO-SBO-k1: convergence and model error histories

(g) PSO-SBO-k2: contour plot

(h) PSO-SBO-k2: convergence and model error histories

Fig. 13 The selected best run of each optimization approach, for the six-hump camel back function. Left figures show filled contours and dashed line contours, respectively, representing the true function and the surrogate model (SM) approximation; in the PSO-AMR and PSO-AMR-Local cases, there are two SM contours, corresponding to the models before and after refinement. Right figures show the convergence and SM error histories over the optimization processes

References

Alexandrov N, Lewis R, Gumbert C, Green L, Newman P (1999) Optimization with variable-fidelity models applied to wing design. Tech. rep., ICASE, Institute for Computer Applications in Science and Engineering NASA Langley Research Center, Hampton, Virginia

Alexandrov NM, Dennis JE, Lewis RM, Torczon V (1998) A trust-region framework for managing the use of approximation models in optimization. Struct Optim 15(1):16–23

Ascione F, Bianco N, Stasio CD, Mauro GM, Vanoli GP (2017) Artificial neural networks to predict energy performance and retrofit scenarios for any member of a building category: a novel approach. Energy 26(118):999–1017

Audet C, Dennis JE, Moore DW, Booker A, Frank PD (2000) A surrogate-model-based method for constrained optimization. In: 8th symposium on multidisciplinary analysis and optimization. Long Beach, CA

Bichon BJ, Eldred MS, Mahadevan S, McFarland JM (2013) Efficient global surrogate modeling for reliability-based design optimization. J Mech Des 135(1):011009

Booker AJ, Dennis JE, Frank PD, Serafini DB, Torczon V, Trosset M (1999) Rigorous framework for optimization of expensive functions by surrogates. Struct Optim 17:1–13

Booker AJ, Dennis JE, Frank PD, Serafini DB, Torczon V, Trosset MW (1999b) A rigorous framework for optimization of expensive functions by surrogates. Struct Optim 17(1):1–13

Chen X, Yang H, Sun K (2017) Developing a meta-model for sensitivity analyses and prediction of building performance for passively designed high-rise residential buildings. Applied energy 194:422–439

Chen Y, Hong T (2018) Impacts of building geometry modeling methods on the simulation results of urban building energy models. Appl Energy 215:717–735

Cheng GH, Younis A, Haji Hajikolaei K, Gary Wang G (2015) Trust region based mode pursuing sampling method for global optimization of high dimensional design problems. J Mech Des 137(2):021407

Choi K, Youn BD, Yang RJ (2001) Moving least square method for reliability-based design optimization. Proc 4th world cong structural & multidisciplinary optimization

Chowdhury S, Tong W, Messac A, Zhang J (2013) A mixed-discrete particle swarm optimization algorithm with explicit diversity-preservation. Struct Multidiscip Optim 47(3):367–388

Chowdhury S, Mehmani A, Tong W, Messac A (2016) Adaptive model refinement in surrogate-based multiobjective optimization. In: 57th AIAA/ASCE/AHS/ASC structures, structural dynamics, and materials conference, p X00000. pp 0417

Clarke SM, Griebsch JH, Simpson TW (2005) Analysis of support vector regression for approximation of complex engineering analyses. Journal of Mechanical Design 127(6):1077–1087

Corbin CD, Henze GP, May-Ostendorp P (2013) A model predictive control optimization environment for real-time commercial building application. J Build Perform Simul 5(3):159–174

Deru M, Field K, Studer D, Benne K, Griffith B, Torcellini P, Liu B (2011) US Department of Energy commercial reference building models of the national building stock. Tech. rep., Department of Energy

Dixon LCW, Price R (1989) Truncated newton method for sparse unconstrained optimization using automatic differentiation. J Optim Theory Appl 60(2):261–275

DOE (2017) Commercial prototype building models. energycodes.gov/development/commercial

Duan Q, Sorooshian S, Gupta V (1992) Effective and efficient global optimization for conceptual rainfall-runoff models. Water Res Res 28(4):1015–1031

Duong T, Hazelton M (2003) Plug-in bandwidth matrices for bivariate kernel density estimation. Nonparametr Stat 15(1):17–30

Epanechnikov VA (1969) Non-parametric estimation of a multivariate probability density. Theory Prob Appli 14(1):153–158

Fernández-Godino MG, Park C, Kim NH, Haftka RT (2016) Review of multi-fidelity models. arXiv:160907196

Forrester A, Keane A (2009) Recent advances in surrogate-based optimization. Prog Aerosp Sci 45(1-3):50–79

Forrester A, Sobester A, Keane A (2008) Engineering design via surrogate modelling: a practical guide. Wiley, New York

Ghassemi P, Zhu K, Chowdhury S (2017) Optimal surrogate and neural network modeling for day-ahead forecasting of the hourly energy consumption of university buildings. In: ASME 2017 international design engineering technical conferences and computers and information in engineering conference, American Society of Mechanical Engineers, pp V02BT03A026–V02BT03A026

Glantz SA, Slinker BK, Neilands TB (1990) Primer of applied regression and analysis of variance, vol 309. McGraw-Hill, New York

Gräning L, Jin Y, Sendhoff B (2007) Individual-based management of meta-models for evolutionary optimization with application to three-dimensional blade optimization. In: Evolutionary computation in dynamic and uncertain environments, Springer, pp 225–250

Griewank AO (1981) Generalized descent for global optimization. J Optim Theory Appl 34(1):11–39

Hansen N (2006) The cma evolution strategy: a comparing review. In: Towards a new evolutionary computation, Springer, pp 75–102

Hardy RL (1971) Multiquadric equations of topography and other irregular surfaces. J Geophys Res 76:1905–1915

Hennig P, Schuler CJ (2012) Entropy search for information-efficient global optimization. J Mach Learn Res 13:1809–1837

Tyler HoytAE, Zhang H (2015) Extending air temperature setpoints: simulated energy savings and design considerations for new and retrofit buildings. Build Environ 88:89–96

Jakobsson S, Patriksson M, Rudholm J, Wojciechowski A (2010) A method for simulation based optimization using radial basis functions. Optim Eng 11(4):501–532

Jin R, Chen W, Simpson TW (2001) Comparative studies of metamodelling techniques under multiple modelling criteria. Struct Multidiscip Optim 23(1):1–13

Jin Y (2005) A comprehensive survey of fitness approximation in evolutionary computation. Soft Comput 9(1):3–12

Jin Y (2011) Surrogate-assisted evolutionary computation: recent advances and future challenges. Swarm Evol Comput 1(2):61–70

Jin Y, Sendhoff B (2004) Reducing fitness evaluations using clustering techniques and neural network ensembles. In: Genetic and evolutionary computation, GECCO, vol 2004, pp 688–699

Jin Y, Olhofer M, Sendhoff B (2002) A framework for evolutionary optimization with approximate fitness functions. IEEE Trans Evol Comput 6(5):481–494

Jones D, Schonlau M, Welch W (1998) Efficient global optimization of expensive black-box functions. J Glob Optim 13(4):455–492

Keane A, Nair P (2005) Computational approaches for aerospace design: the pursuit of excellence. Wiley, New York

Keane AJ (2006) Statistical improvement criteria for use in multiobjective design optimization. AIAA J 44(4):879–891

Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: IEEE international conference on neural networks, , vol IV. IEEE, Piscataway, pp 1942–1948

Kleijnen JP, Beers WV, Nieuwenhuyse IV (2012) Expected improvement in efficient global optimization through bootstrapped kriging. J Global Optim 54(1):59–73

Kourakos G, Mantoglou A (2009) Pumping optimization of coastal aquifers based on evolutionary algorithms and surrogate modular neural network models. Adv Water Resour 32(4):507–521

Liu Y, Ghassemi P, Chowdhury S, Zhang J (2018) Surrogate based multi-objective optimization of j-type battery thermal management system. In: ASME 2018 international design engineering technical conferences and computers and information in engineering conference, American society of mechanical engineers digital collection

Lulekar S, Ghassemi P, Chowdhury S (2018) Cfd-based analysis and surrogate-based optimization of bio-inspired surface riblets for aerodynamic efficiency. In: 2018 Multidisciplinary Analysis and Optimization Conference, p 3107

March A, Willcox K, Wang Q (2011) Gradient-based multifidelity optimisation for aircraft design using Bayesian model calibration. Aeronaut J 115(1174):729–738

Marduel X, Tribes C, Trepanier JY (2006) Variable-fidelity optimization: efficiency and robustness. Optim Eng 7(4):479–500

Marmin S, Chevalier C, Ginsbourger D (2015) Differentiating the multipoint expected improvement for optimal batch design. In: International workshop on machine learning, Optimization and big data. Springer, pp 37–48

McKay M, Conover W, Beckman R (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21(2):239–245

Meeker W, Hahn G, Escobar L (2017) Statistical intervals: a guide for practitioners and researchers. Wiley Series in Probability and Statistics, Wiley. https://books.google.com/books?id=y3o0DgAAQBAJ

Mehmani A, Chowdhury S, Messac A (2015a) Adaptive switching of variable-fidelity models in population-based optimization algorithms. In: 16th AIAA/ISSMO multidisciplinary analysis and optimization conference, p 3233

Mehmani A, Chowdhury S, Messac A (2015) Predictive quantification of surrogate model fidelity based on modal variations with sample density. Struct Multidiscip Optim 52(2):353–373

Mehmani A, Chowdhury S, Messac A (2016) Variable-fidelity optimization with in-situ surrogate model refinement. In: ASME 2015 international design engineering technical conferences and computers and information in engineering conference, American society of mechanical engineers digital collection

Mehmani A, Chowdhury S, Meinrenken C, Messac A (2018) Concurrent surrogate model selection (cosmos): optimizing model type, kernel function, and hyper-parameters. Struct Multidiscip Optim 57(3):1093–1114

Molga M, Smutnicki C (2005) Test functions for optimization needs. Apr, 101

Moore RA, Romero DA, Paredis CJ (2011) A rational design approach to gaussian process modeling for variable fidelity models. In: ASME 2011 International design engineering technical conferences (IDETC). Washington, DC

Pelikan M (2005) Hierarchical Bayesian optimization algorithm. In: Hierarchical Bayesian optimization algorithm, Springer, pp 105–129

Peng L, Liu L, Long T, Guo X (2014) Sequential rbf surrogate-based efficient optimization method for engineering design problems with expensive black-box functions. Chin J Mech Eng 27(6):1099–1111

Rai R (2006) Qualitative and quantitative sequential sampling,. PhD thesis, University of Texas Texas, Austin, USA

Regis RG (2014) Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. Eng Optim 46(2):218–243

Regis RG, Shoemaker CA (2013) Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization. Eng Optim 45(5):529–555

Robinson TD, Eldred MS, Willcox KE, Haimes R (2008) Surrogate-based optimization using multifidelity models with variable parameterization and corrected space mapping. AIAA J 46(11):2814–2822

Rodriguez JF, Perez VM, Padmanabhan D, Renaud JE (2001) Sequential approximate optimization using variable fidelity response surface approximations. Struct Multidiscip Optim 22(1):24–34

Sharif SA, Hammad A (2019) Developing surrogate ann for selecting near-optimal building energy renovation methods considering energy consumption, lcc and lca. J Buil Eng 25:100790

Simpson T, Korte J, Mauery T, Mistree F (2001) Kriging models for global approximation in simulation-based multidisciplinary design optimization. AIAA J 39(12):2233–2241

Simpson T, Toropov V, Balabanov V, Viana F (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, p 5802

Snoek J, Larochelle H, Adams RP (2012) Practical bayesian optimization of machine learning algorithms. In: Advances in neural information processing systems, pp 2951–2959

Sugiyama M (2006) Active learning in approximately linear regression based on conditional expectation of generalization error. J Mach Learn Res 7:141–166

Sun M, Chang CL, Zhang J, Mehmani A, Culligan P (2018) Break-even analysis of battery energy storage in buildings considering time-of-use rates. In: IEEE green technologies conference (GreenTech), pp 95–99

Tajbakhsh SD, del Castillo E, Rosenberger JL (2013) A fully Bayesian approach to the efficient global optimization algorithm. PhD thesis, Pennsylvania State University Working Paper

Tanabe R, Fukunaga AS (2014) Improving the search performance of shade using linear population size reduction. In: 2014 IEEE congress on evolutionary computation (CEC). IEEE, pp 1658–1665

Tian W (2013) A review of sensitivity analysis methods in building energy analysis. Renew Sust Energ Rev 20:411–419

Tong W, Chowdhury S, Messac A (2016) A multi-objective mixed-discrete particle swarm optimization with multi-domain diversity preservation. Struct Multidiscip Optim 53(3):471–488

Toropov VV, Schramm U, Sahai A, Jones RD, Zeguer T (2005) Design optimization and stochastic analysis based on the moving least squares method. 6th World Congress of Structural and Multidisciplinary Optimization

Ulmer H, Streichert F, Zell A (2004) Evolution strategies with controlled model assistance. In: Congress on evolutionary computation, 2004. CEC2004. IEEE, vol 2, pp 1569–1576

Viana FA, Haftka RT, Watson LT (2013) Efficient global optimization algorithm assisted by multiple surrogate techniques. J Glob Optim 56(2):669–689

Wang Y, Song Z, De Angelis V, Srivastava S (2018) Battery life-cycle optimization and runtime control for commercial buildings demand side management: a New York City case study. Energy 165:782–791

Wild SM, Regis RG, Shoemaker CA (2008) Orbit: optimization by radial basis function interpolation in trust-regions. SIAM J Sci Comput 30(6):3197–3219

Williams B, Loeppky JL, Moore LM, Macklem MS (2011) Batch sequential design to achieve predictive maturity with calibrated computer models. Reliab Eng Syst Saf 96(9):1208–1219

Yao W, Chen X, Huang Y, van Tooren M (2014) A surrogate-based optimization method with rbf neural network enhanced by linear interpolation and hybrid infill strategy. Optim Methods Softw 29(2):406–429

Yegnanarayana B (2004) Artificial neural networks. PHI Learning Pvt. Ltd.

Zhang J, Chowdhury S, Messac A (2012) An adaptive hybrid surrogate model. Struct Multidiscip Optim 46(2):223–238