Surrogate-Assisted Multi-Objective Optimization for Handling Objectives with Heterogeneous Evaluation Times: Unconstrained Problems

Balija Santoshkumar *Student Member, IEEE*, and Kalyanmoy Deb, *Fellow, IEEE*

COIN Report Number 2024005

***Abstract*—Surrogates are commonly used in single and multi-objective optimization studies for quickly evaluating objective functions which are otherwise expensive to evaluate. Starting with a set of high-fidelity evaluated solutions, optimization algorithms generate new in-fill solutions for further high-fidelity evaluations to improve the previous surrogate models towards the optimal regions of the search space. In terms of evaluating in-fill solutions, most current optimization algorithms must evaluate all objectives using high-fidelity means, irrespective of relative differences in their evaluation times. In this paper, we address the issue of handling objective functions having heterogeneous high-fidelity evaluation times and propose a framework which combines evaluation time and surrogate accuracy of each objective to decide which population members should undergo a high-fidelity evaluation and for which objectives. Our proposed approach deals with heterogeneously evaluated solutions – some objectives with high-fidelity and some with surrogates – in a generic way. Initial results on a number of two and three-objective problems, presented in this paper, show promise for approach and encourages to launch a deeper study on more complex and constrained problems.**

***Index Terms*—Heterogeneous Evaluation Times, Non-uniform Latency, Surrogate-assisted Optimization, Multi-objective Optimization, Machine Learning, Reference Vector Guided Probabilistic Dominance**

I. Introduction

One of the bottlenecks in solving real-world computationally expensive optimization problems with evolutionary algorithms (EAs) is the function calls. The function evaluations in practical problems usually come from expensive computational simulations such as Finite Element Analysis (FEA) and Computational Fluid Dynamics (CFD), each taking a different amount of time to evaluate a solution. Simpson et al. [1] showed that Ford Motor Company spends approximately 36–160 hours in a car crash-worthiness simulation. In surrogate-assisted optimization, instead of using expensive function evaluations all the time, we use surrogate models to approximate them.

In practical optimization problems, some of the objectives can come from expensive computational simulations and some of the objectives can come from analytical or closed-form solutions which take less time to evaluate. Heterogeneity in objectives is a common scenario in real-world optimization rather than a special case. In this study, we propose a generalized framework for handling heterogeneous function evaluation times in multi-objective optimization.

In the remainder of the paper, we briefly outline a few studies on the topic in Section II. The detail of proposed HET-EMO algorithm is provided in Section III. Results on two and three-objective problems are discussed in Section IV. Finally, conclusions and possible future studies are presented in Section V.

II. Existing Studies

Some studies on handling heterogeneously expensive objective functions are proposed in the literature. Among these, Allmendinger and Knowles [2] proposed a method to deal with missing objective values caused by delay caused by expensive objective functions. Later they extended [3] and proposed three strategies on how to handle delays namely: waiting, fast-first, and interleaving. Chugh et al. [4] proposed HK-RVEA for bi-objective problems with heterogeneous function evaluation times. This algorithm is a combination of K-REVA's surrogate updating mechanism and a single objective evolutionary algorithm to select training data to train surrogate model. In non-evolutionary approaches, Thomann et al. [5] proposed a multi-objective heterogeneous trust region algorithm (MHT) that interpolates the objectives with a local quadratic model. Wang et al. [6] proposed a transfer learning based approach (T-SAEA) when there is a correlation between the objectives. While most of the existing studies deal with bi-objective problems, Blank et al. [7] proposed a method for handling heterogeneous function evaluation times for more than two objectives and constraints. Mamun et al. [8] proposed an approach for handling independently evaluable objectives, although it did not consider varying latencies in objectives. Recently, Bayesian optimization approaches for handling heterogeneous evaluation times have been proposed by Buckingham et al. [9] and a review on heterogeneous objective function handling is published by Allmendinger and Knowles [10].

III. Proposed Approach

Our proposed approach consists of five key steps. We begin with an initial design of experiments N_{doe} to build surrogate

Balija Santoshkumar is with the Department of Mechanical Engineering, Michigan State University, East Lansing, MI, 48824 USA e-mail: balijasa@msu.edu.

Kalyanmoy Deb is with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, 48824 USA email: kdeb@egr.msu.edu.

models for each objective function independently. Then, we select the initial population P_0 of size N (may be $\leq N_{doe}$) by using a chosen EMO's (Evolutionary Multi-criterion Optimization algorithm, here we have chosen NSGA-III [11]) survival operator from N_{doe} solutions. Next, we run the EMO with P_0 as the initial population for the specified number of surrogate generations s without any high-fidelity evaluations. Thereafter, we obtain the final generation of survived population Q_{0+s} after the surrogate generations and combine with initial population P_0 to get the combined population R_1. For every member of R_1, we then compute a selection criteria metric ρ (defined in subsection III-D) for every objective function by combining its evaluation time and surrogate accuracy. Higher the value of ρ, higher is the probability of its chance for high-fidelity evaluation. We then evaluate the high-ρ objectives for R_1 population members by high-fidelity model. After exactly N solutions are selected, they are saved in P_1. We evaluate the selected objectives for selected P_1 members, which were not already evaluated in the past. Each surrogate function is then updated with new infill evaluations for each objective function. Note that at every survival selection phase, not all N members are high-fidelity evaluated for all objectives, as some of the previous parent members (which were already high-fidelity evaluated) may have been selected in P_1. We re-evaluate every objective function for each member of P_{t+1} using the updated surrogates for next round of surrogate generations and increment the counter t. The process is repeated until the evaluation budget ($T_{\max}$) is exhausted. At the end, we evaluate the final population for all objectives using high-fidelity evaluation for reporting purposes. The overall main algorithm is outlined in Algorithm 1. We now present the main operations of the proposed algorithm.

A. Initialization

The proposed algorithm starts with initial sampling points N_{doe} using the Latin Hypercube Sampling (LHS) method [12]. The sampled points are evaluated, and surrogate models for each objective function are constructed independently. The evaluated points are saved for each objective in an archive A_m. The computational cost for evaluating the initial design of experiments (N_{doe}) is updated in the evaluation budget counter T. We use Kriging [13] for modeling the surrogates, since it has the ability to provide uncertainty information, as an indication of modeling error. We use Kriging with six different regressors, including constant, linear, quadratic, with and without Automatic Relevance Determination (ARD) functions independently. For surrogate model building, N_{doe} is split into training and testing sets using 5-fold cross-validation. The best-performing model from six regressors is selected based on mean absolute error (MAE), and the selected model is used for the surrogate generations of the algorithm.

For reference vector generation, we use the Riesz s-Energy method [14]. The number of reference directions W_r set equal to the population size N since the Riesz s-Energy method enables the creation of well-spaced points on the unit simplex for any N. We use NSGA-III implementation of the Pymoo [15] framework and Pysamoo [16] framework for surrogates modeling.

Algorithm 1: Heterogeneous evaluation times multi-objective optimization framework (HET-EMO)

Data: M, ET^m, T_{max}, H, N_{doe}, N, and S_{gen}
Result: Final non-dominated population P

/* initialization */
$M \leftarrow (f_1, \ldots, f_M)$
$T = 0, t = 0$
LHS for sampling N_{doe} points
for *each* $m \in M$ **do**
  Evaluate all N_{doe} solutions
  Update the computational budget T
  Save evaluated solutions into archive A_m
  Build surrogate models and select the best model
end for
Construct P_t from N_{doe} using NSGA-III survival op.
while $T \leq T_{max}$ **do**
  Run EMO for s surrogate generations with P_t
  Obtain final survivors Q_{t+s}
  $R_t = P_t \cup Q_{t+s}$
  Re-evaluate R_t with updated surrogate model
  Compute probability of domination PD for all R_t members
  $[S_t, S_{id}, S_{obj}] \leftarrow$ Select members and their objectives for high-fidelity evaluations from R_t using Algorithm 2
  Evaluate the selected objectives S_{obj} for the members in S_t, if they are not evaluated
  Update the computational budget T
  $P_{t+1} \leftarrow S_t$ and $t \leftarrow t + 1$
  for *each* $m \in M$ **do**
    Save the evaluated solutions into archive A_m
    Update the surrogate model
  end for
end while
Evaluate final population with high-fidelity evaluations

B. Surrogate Generations

In the surrogate generations stage, we use the parent population P_t as the starting population and run the selected EMO algorithm for a predefined number of surrogate generations s (= 5, chosen here) without executing any high-fidelity evaluation, This creates an offspring population of size N at the end of s surrogate-assisted EMO generations.

At this stage, we have a parent population P_t which may have been partially or completely evaluated using high-fidelity evaluations and an offspring population Q_{t+s} which are all evaluated using the current surrogate models. There are two tasks ahead of us. First, we need to select N population members from the combined $P_t \cup Q_{t+s}$ using mixed-evaluated members. Second, we need to decide the members which must be high-fidelity evaluated next in order to update the current surrogate models so the optimization can progress towards the optimal region both with more representative solutions and also by making the surrogates better. We address the latter issue first, which will then dictate a procedure for the first matter.

C. Probabilistic Dominance Metric

In order to choose solutions and specific objectives for high-fidelity evaluation, we compute a probabilistic dominance (PD) metric. Since objective values of all population members are not exact, the PD metric assesses the probability of how worse a particular solution $\mathbf{x}$ is compared with all the other solutions associated with the same reference vector (RV) $\mathbf{w}$ considering uncertainties (σ) of the solutions.

The probability that solution $\mathbf{x}$ with a predicted objective value $\mu_{\mathbf{x}}$ and uncertainty $\sigma_{\mathbf{x}}$ is worse compared to another solution $\mathbf{y}$ with prediction $\mu_{\mathbf{y}}$ and uncertainty $\sigma_{\mathbf{y}}$ for the m-th objective in the minimization sense can be formulated as follows [8], [17], [18]:

$$P^m_{\mathbf{xy}} = \Pr[f_m(\mathbf{x}) > f_m(\mathbf{y})] = \frac{1}{2}(1 + \mathrm{erf}(-z)), \quad (1)$$

where $z = (0 - (\mu_{\mathbf{x}} - \mu_{\mathbf{y}}))/\sqrt{2(\sigma^2_{\mathbf{x}} + \sigma^2_{\mathbf{y}})}$ and erf() denotes the Gaussian error function. The average probabilistic dominance metric $PD^m_{\mathbf{s}}$ for the solution $\mathbf{s}$ in the objective m, among the associated members $\pi(\mathbf{s})$ of the RV $\mathbf{w}$ can be computed as:

$$PD^m_{\mathbf{s}} = \frac{\sum_{i \in \pi(\mathbf{s}), i \neq \mathbf{s}} P^m_{\mathbf{s}i}}{|\pi(\mathbf{s})| - 1}. \quad (2)$$

For example, from Fig. 1, the average probabilistic dominance metrics PD for the solution P_2 in the objectives f_1 and f_2 become:

$$PD^1_{(P2)} = \frac{(P^1_{23} + P^1_{24})}{3 - 1}, \quad PD^2_{(P2)} = \frac{(P^2_{23} + P^2_{24})}{3 - 1}. \quad (3)$$

There are three associated members in the W_2 reference direction. Compute the PD metric for all population members of R_t for all the objective functions. We compute the probability of $\mathbf{s}$ being not worse among its associated members, as $P^m_{\mathbf{S}} = 1 - PD^m_{\mathbf{S}}$.

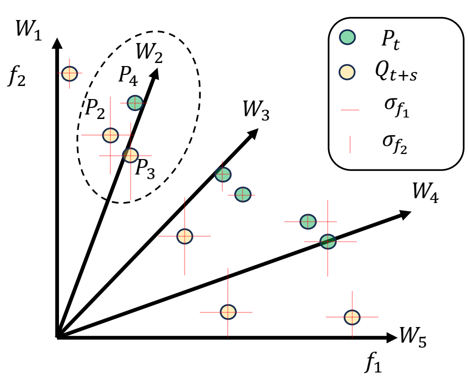


Fig. 1: Probabilistic domination computation for RV-based solutions.

D. Combined Dominance and Evaluation Time Metric

For a specific objective function, a population member may have the largest probability to dominate its remaining associate members, but in this study, the evaluation time for computing the objective value and a direct estimate of the surrogate error are also important matters. We combine these factors in one combined metric ρ.

Knowing the average evaluation time (ET^m) of m-th objective, we first normalize them using the maximum evaluation time, as follows:

$$\widehat{ET}^m = \frac{ET^m}{\max(ET^1, ET^2, \ldots, ET^M)}. \quad (4)$$

This makes the every $\widehat{ET}^m$ less than or equal to one, providing a relative estimate of the heterogeneous evaluation times.

After a number of trial and error efforts and understanding the effect of each of the factors in deciding on the combined metric, we propose the following metric for a solution $\mathbf{s}$ in objective m:

$$\rho^m_{\mathbf{s}} = \left(1 + \widehat{ET}^m\right) P^m_{\mathbf{s}} \left(1 + \left(\frac{\sigma^m_{\mathbf{s}}}{|\Delta f_m|}\right)^{\frac{1}{\eta}}\right). \quad (5)$$

Here, $|\Delta f_m| = \hat{z}^{nad} - \hat{z}^*$ for the corresponding objective m, $\hat{z}^{nad}$ and $\hat{z}^*$ are the estimated nadir and ideal points respectively. The parameter $\eta > 0$ serves as a controlled factor to increase the importance of uncertainty in arriving at the metric value. We set $\eta = 6$ here, but it becomes a hyper-parameter which may need tuning for a problem class. $\sigma^m_{\mathbf{S}}$ denotes the surrogate uncertainty of solution $\mathbf{s}$ for objective m.

E. Selection of Solutions for High-fidelity Evaluations

After computing the ρ metric for the entire R_t population for all objectives, we convert the ρ metric as a 1D vector stacked RV-wise. We create a tuple $\mathcal{PD}$ that consists of ρ and their corresponding solution id and objective id for each RV. For selecting the members and their corresponding objective for high-fidelity evaluation, we go along each direction $\mathbf{w}$ and select the solution id $\bar{s}$ and its corresponding objective id $\bar{m}$ that has the maximum ρ value among all the associated members of RV $\mathbf{w}$. We store the selected member in S_t, we exactly select N unique population members. Some of the selected members might have both objectives selected for evaluation(see Fig. 2). The selected N solutions for high-fidelity evaluations are used to form P_{t+1}. The overall process is outlined in Algorithm 2 and in Fig. 2.

IV. Results

We test the proposed approach HET-NSGA-III with two other algorithms: (i) the original NSGA-III [11] in which every population member is high-fidelity evaluated for all objectives without any use of surrogates, and (ii) SA-NSGA-III, in which surrogates are used for optimization for s generations, but all offspring population members of Q_{t+s} are high-fidelity evaluated for all objectives and surrogates are then updated for

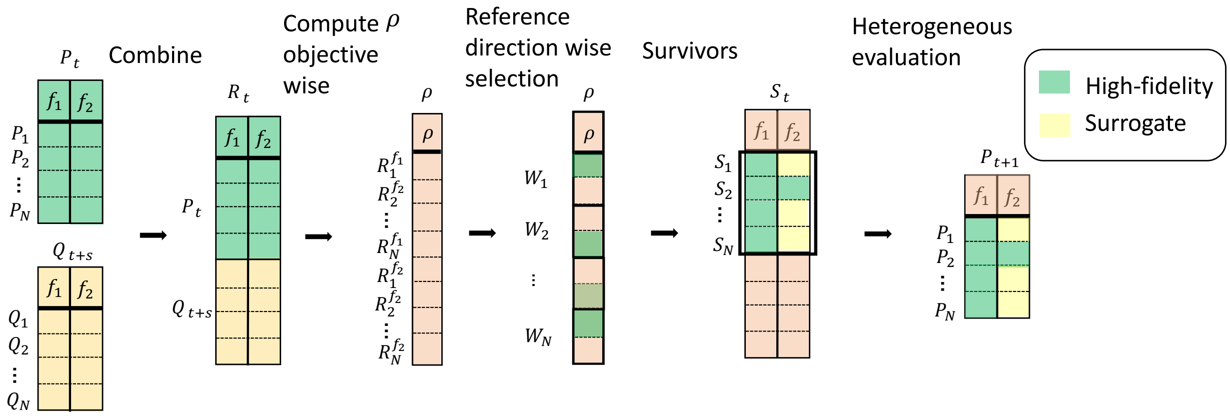


Fig. 2: Selection of solutions and their corresponding objectives for high fidelity evaluation in the initial generation.

Algorithm 2: Selection of solutions for high fidelity evaluation

Data: ET^m, $R_t(\mu, \sigma)$
Result: Selected solutions and their corresponding objectives for high-fidelity evaluations of S_t

$\rho \leftarrow \varnothing_{MN\times 1}$
$\mathcal{PD} \leftarrow \varnothing_{MN\times 4}$
$S_t \leftarrow \varnothing$, $S_{id} \leftarrow \varnothing$, $S_{obj} \leftarrow \varnothing$
for *each* $\mathbf{s} \in R_t$ **do**
  for *each* $m \in M$ **do**
    Compute $\rho_{\mathbf{s}}^m$ vector using Equation 5
    $\rho = \rho \cup \rho_{\mathbf{s}}^m$
  end for
end for
$\mathcal{PD} \equiv \{\mathbf{s}, \rho, m, \mathbf{w}\}$
while $|S_{id}| \leq N$ **do**
  $\mathcal{PD} = \mathcal{PD} \setminus S_t$
  for *each* $\mathbf{w} \in W_r$ **do**
    $[\bar{\mathbf{s}}, \bar{m}]$: $\mathrm{argmax}_{[\mathbf{s},m]\in\mathcal{PD}}\, \rho$
    $S_t = S_t \cup R_t(\bar{\mathbf{s}})$
    $S_{id} = S_{id} \cup \bar{\mathbf{s}}$
    $S_{obj} = S_{obj} \cup \bar{\mathbf{m}}$
  end for
end while

next generation. In this initial study, we test these three methods on a number of multi-objective problems: two-objective ZDT [19], two and three-objective DTLZ2 problems [20], and the crashworthiness [21] problem. The number of variables used is 10, except in the bi-objective DTLZ2, where we use 11 variables. The initial DOE is executed with $N_{doe} = 120$. and the population size N is set to 20 for bi-objective problems. For three-objective problems, N_{doe} set as 150 and population size N to 30.

For simulating heterogeneous evaluation times for objectives in these test problems, we assume that the estimated total evaluation time of a solution for all objectives is $t_{\text{total}} = 30$ units of time and investigate the effect of different relative evaluation times of objectives. For example, $ET^1 = ET^2 = 15$ will mean that both objectives take more or less identical average time to evaluation a solution, however, $ET^1 = 10$ and $ET^2 = 20$ (still summing to 30 units) means that the second objective takes more average time to evaluate than the first objective. Original NSGA-III and SA-NSGA-III high-fidelity evaluates each solution for all objectives. To make a fair comparison of HET-NSGA-III with NSGA-III and SA-NSGA-III, we present intermediate results in terms of standardized solution evaluations (Γ), defined as the total elapsed units of time by an algorithm (T) divided by a complete solution evaluation time (t_{total}), or $\Gamma = T/t_{\text{total}}$. For our experiments here, $t_{\text{total}} = 30$. Thus, for NSGA-III and SA-NSGA-III, Γ is identical to the number of high-fidelity solution evaluations (SE). But for HET-NSGA-III, we increase T by ET^m every time an solution is high-fidelity evaluated for the m-th objective. At the point of interest (or reporting), we simply divide T by 30 to obtain SSE estimate.

For each problem, we run all three algorithms 15 times with different initial populations. We use the number of reference directions the same as the population size. Hyper-volume (HV) is chosen as the performance metric to compare different algorithms at specific Γ values. Since different runs of HET-NSGA-III may store the HV value at Γ values (even if they are stored at the end certain specific generations), we follow an interpolation strategy between two nearest Γ values of a reporting Γ and present the interpolated HV value. Also, since every run may have created a different surrogate model at specific Γ, for the reporting purpose, we compute the HV using high-fidelity evaluations for all three algorithms. For hyper-volume computation, we normalize the population members using known Pareto front's nadir and ideal points. The Wilcoxon signed-rank test with $p = 0.05$ is used to determine statistically better, similar, and worst performing algorithms, which are marked for each test instance with $(*)$, $(\approx)$, and $(-)$, respectively.

A. Bi-objective Problems

For the bi-objective ZDT1-3 problems, the overall evaluation budget T_{max} is set as 14,400 units (assuming a four hour (14,400 seconds) run time with a complete high-fidelity evaluation of a single solution for all objectives taking 30 seconds on an average). This results in 14,400/30 = 480 high-fidelity SEs for NSGA-III and SA-NSGA-III. For simulating different heterogeneous evaluation time combinations for two objectives, we use ET^1 as 3, 10, 12, 15, 18, 20, and 27 units and $ET^2 = 30 - ET^1$. For bi-objective DTLZ2 problem, we set the overall evaluation budget $T_{max} = 9,000$ units causing $9,000/30$ or 300 high-fidelity evaluations for NSGA-III and SA-NSGA-III. It is needless to say that HET-NSGA-III is also terminated when T_{max} units of time is elapsed. HET-NSGA-III may not spend equal number of f_1 and f_2 high-fidelity evaluations depending on the complexity of the objectives for surrogate approximations, their evaluation time and probability of dominance within their associated population members.

The median HV values of the final population for all three algorithms are shown in Table I. It is clear that the

TABLE I: Median HV values for bi-objective problems.

(ET^1, ET^2)	NSGA-III	SA-NSGA-III	HET-NSGA-III
ZDT1 $(n = 10, M = 2, T_{max} = 14,400, \Gamma = 480)$			
(3, 27)	0.21558 (-)	0.64745 (-)	**0.68532 (*)**
(27, 3)	0.21558 (-)	*0.64745* (≈)	*0.72490* (≈)
(20, 10)	0.21558 (-)	0.64745 (-)	**0.69792 (*)**
(18, 12)	0.21558 (-)	0.64745 (-)	**0.72855 (*)**
(15, 15)	0.21558 (-)	0.64745 (-)	**0.73287 (*)**
(12, 18)	0.21558 (-)	0.64745 (-)	**0.72199 (*)**
(10, 20)	0.21558 (-)	0.64745 (-)	**0.72790 (*)**
ZDT2 $(n = 10, M = 2, T_{max} = 14,400, \Gamma = 480)$			
(3, 27)	0.00000 (-)	0.07704 (-)	**0.17896 (*)**
(27, 3)	0.00000 (-)	0.07704 (-)	**0.38334 (*)**
(20, 10)	0.00000 (-)	0.07704 (-)	**0.38109 (*)**
(18, 12)	0.00000 (-)	0.07704 (-)	**0.36634 (*)**
(12, 18)	0.00000 (-)	0.07704 (-)	**0.40379 (*)**
(15, 15)	0.00000 (-)	0.07704 (-)	**0.32676 (*)**
(10, 20)	0.00000 (-)	0.07704 (-)	**0.40131 (*)**
ZDT3 $(n = 10, M = 2, T_{max} = 14,400, \Gamma = 480)$			
(3, 27)	0.21764 (-)	0.39899 (-)	**0.47680 (*)**
(27, 3)	0.21764 (-)	0.39899 (-)	**0.51690 (*)**
(20, 10)	0.21764 (-)	0.39899 (-)	**0.44733 (*)**
(18, 12)	0.21764 (-)	0.39899 (-)	**0.53610 (*)**
(15, 15)	0.21764 (-)	0.39899 (-)	**0.55070 (*)**
(12, 18)	0.21764 (-)	0.39899 (-)	**0.52437 (*)**
(10, 20)	0.21764 (-)	0.39899 (-)	**0.54070 (*)**
DTLZ2 $(n = 11, M = 2, T_{max} = 9,000, \Gamma = 300)$			
(3, 27)	0.09956 (-)	0.25042 (-)	**0.27973 (*)**
(27, 3)	0.09956 (-)	0.25042 (-)	**0.28642 (*)**
(20, 10)	0.09956 (-)	0.25042 (-)	**0.28479 (*)**
(18, 12)	0.09956 (-)	0.25042 (-)	**0.28165 (*)**
(15, 15)	0.09956 (-)	0.25108 (-)	**0.28162 (*)**
(12, 18)	0.09956 (-)	0.25108 (-)	**0.26421 (*)**
(10, 20)	0.09956 (-)	0.25042 (-)	**0.28707 (*)**
Total * / ≈ / -	0 / 0 / 28	0 / 1 / 27	27 / 1 / 0

proposed HET-NSGA-III performs statistically better in 27 of 28 problem instances. SA-NSGA-III performs statistically equivalent to HET-NSGA-III for one instance (ZDT1 with a skewed evaluation time difference).

The obtained solutions for a specific test instance of evaluation times are shown Figs. 3a, 3b, 3c and 3g for a more comprehensive understanding of the performance. The first row of plots show that HET-NSGA-III produces near Pareto and a well-distributed set of solutions compared to other two methods with an identical overall evaluation time $T_{\max}$ for different relative ET values of objectives. The second row of plots show the growth of HV for three methods from start to end of optimization runs using Γ (equivalent solution evaluations) in the x-axis. It is remarkable that with only 480 high-fidelity evaluations HET-NSGA-III is able to find a well-distributed and well converged set of solutions in different ZDT problems. The third row of plots presents the obtained set of solutions, the HV variation with Γ over multiple runs and median HV growth for two-objective DTLZ problem. Clearly, HET-NSGA-III performs better on this problem too.

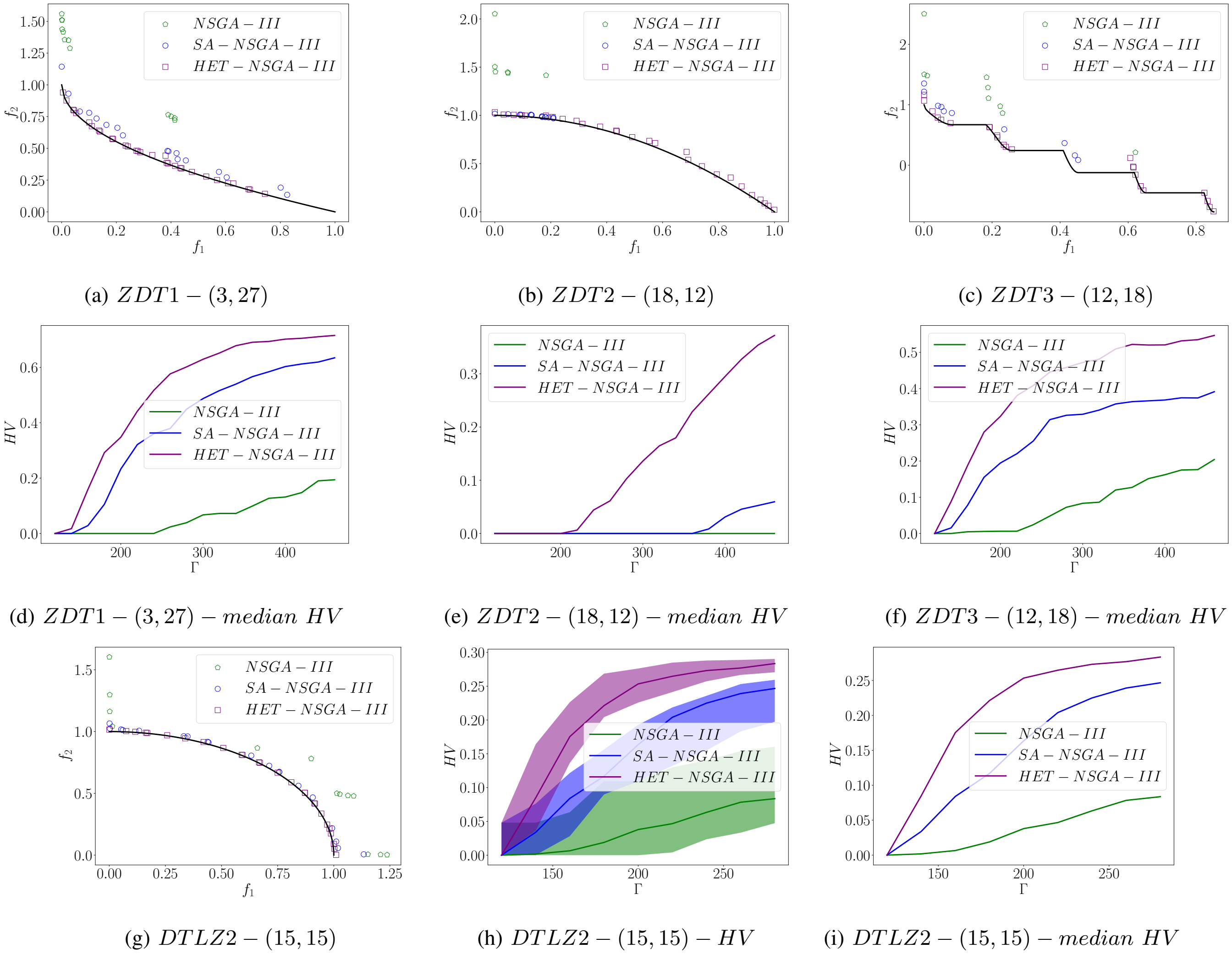
To reveal the working of the proposed combined ρ metric in HET-NSGA-III, we consider its performance on ZDT1 with $ET^1 = 3$ and $ET^2 = 27$ units of evaluation time. This setting requires f_2 evaluation time to be nine times more than that of f_1. As evident from the ZDT1 problem description, f_1 is a linear function of the first variable alone, making it also easy to model by a surrogate method. Figure 4a shows that the proposed HET-NSGA-III evaluates more solutions for f_2 than for f_1, whereas both NSGA-III and SA-NSGA-III, by design, use identical number of evaluations for both objectives. HET-NSGA-III takes advantage of more individual evaluations of f_2 to make the respective surrogate model more accurate compared to f_1 and in the process produces a much better performance at the end compared to other two algorithms. For the same problem, a generation-wise variation of evaluations for each objective is presented in Figure 4b for HET-NSGA-III. It is interesting to note that after the high-fidelity evaluations during initial DOE for f_1, the algorithm did not find much need to evaluate f_1 more often; rather, it concentrated in evaluating f_2 more almost linearly to improve the algorithm's performance. The main reason for developing the ρ metric for handling heterogeneous functions is to have such an effect.

B. Three-objective problems

For all three-objective problems, the overall evaluation budget T_{max} is set as 14,400 units, which translates 480 complete high-fidelity solution evaluations for NSGA-III and SA-NSGA-III. For simulating the heterogeneous evaluation times, we use following seven combinations of the average evaluation time for f_1, f_2 and f_3, respectively: (10, 5, 15), (15, 5, 10), (5, 10, 15), (10, 15, 5), (10, 10, 10), (5, 15, 10), (15, 10, 5). They include equal evaluation times to at most three times more evaluation time between any two objectives.

The median HV values of the final population for all three algorithms are shown in Table II. HET-NSGA-III performs better in all 21 problems, in some instances with a much better HV value. This indicates that the proposed probabilistic dominance metric and combined ρ metric concepts extend to three-objective problems as well.

The obtained solutions from the final population for a specific test instances of evaluation times are shown Figs. 5a, 5b, and 5c. In all cases, the distribution and convergence of HET-NSGA-III solutions are better. The second row of plots clearly



(a) $ZDT1-(3,27)$ (b) $ZDT2-(18,12)$ (c) $ZDT3-(12,18)$

(d) $ZDT1-(3,27)-median\ HV$ (e) $ZDT2-(18,12)-median\ HV$ (f) $ZDT3-(12,18)-median\ HV$

(g) $DTLZ2-(15,15)$ (h) $DTLZ2-(15,15)-HV$ (i) $DTLZ2-(15,15)-median\ HV$

Fig. 3: Comparison of three algorithms for two-objective problems with different relative evaluation times of objectives.

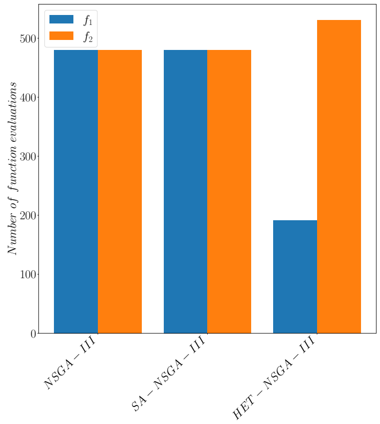
show the superiority of HET-NSGA-III in terms of growth of HV for the median run. Right from the initial generation, HET-NSGA-III evaluates the most-needed objectives more often to progress the population members towards the PO front.

C. Crashworthiness problem

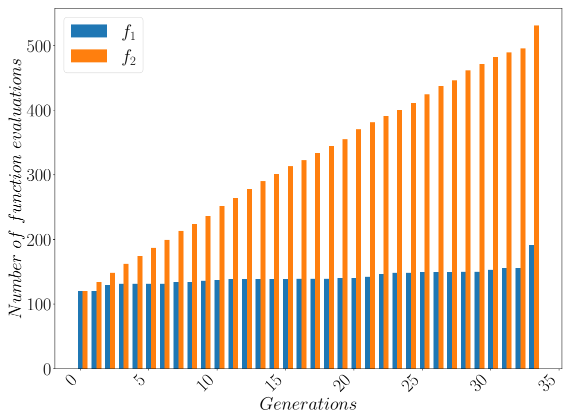
For the crashworthiness problem, we use the same parameter values as in three-objective problems. The median HV values of final generation are shown in Table III, and the obtained solutions in final generation for a specific test instance of evaluation times are shown in Fig. 6. Among seven instances, HET-NSGA-III performs statistically better in four, similar in one, and worst in two cases. The test instances where it does not perform well are when the average evaluation time f_2 is assigned low. In these cases, HET-NSGA-III evaluates f_2 relatively fewer times. The average evaluation times are arbitrarily assigned and it could be that for this problem a better accuracy in f_2 objective is important to achieve a good spread and convergence. To understand this aspect better, we plot the final populations for three instances in Fig. 6a. It clearly shows that if f_2 is not emphasized well during the optimization process, the top left part of the PF may not be found. With equal or more emphasis of f_2 through evaluation time assignments, HET-NSGA-III is able to find a well-spread Pareto solutions, thereby making the HV better, as evident from Table III.

V. Conclusions

We have proposed a surrogate-assisted evolutionary multi-objective optimization framework for handling expensive objectives with heterogeneous computational times. For each solution, each objective function is chosen for high-fidelity evaluation based on three quantities: (i) its average evaluation time, (ii) its surrogate error/uncertainty, and (iii) its probability of dominating other associated population members. Results on a number of two and three-objective test and engineering problems have revealed that our proposed HET-NSGA-III approach is able to put adequate emphasis for high-fidelity evaluations to essential objectives having more evaluation



(a) Number of total high-fidelity evaluations for f_1 and f_2.



(b) Variation of number of high-fidelity evaluations for f_1 and f_2.

Fig. 4: Statistics of number of high-fidelity evaluations for f_1 and f_2 by a single run of HET-NSGA-III on ZDT1 with (3, 27) units of evaluation times, respectively.

time, large surrogate error, and small objective value adaptively during the optimization process.

The initial results presented here are extremely encouraging and demands further studies with the proposed framework. The current framework can be extended to include constraints along with objectives. To make the approach more practical, single evaluation routine for a block of objective and constraint combinations can be considered. The crashworthiness problem raises the need for further adjusting the combined ρ metric so that the more complex objectives are automatically identified by the algorithm and are evaluated more often. Nevertheless, the proposed framework is generic, has been demonstrated to work well, and offers flexibility to develop more efficient and adaptive surrogate-assisted algorithms with heterogeneous objective/constraint evaluation times.

TABLE II: Median HV value comparison for three-objective problems.

(ET^1, ET^2, ET^3)	NSGA-III	SA-NSGA-III	HET-NSGA-III
DTLZ2 ($n = 10, M = 3, T_{max} = 14,400, \Gamma = 480$)			
(10, 5, 15)	0.243719 (-)	0.454372 (-)	**0.491989**(*)
(15, 5, 10)	0.243719 (-)	0.454372 (-)	**0.487883**(*)
(5, 10, 15)	0.243719 (-)	0.454372 (-)	**0.492393**(*)
(10, 15, 5)	0.243719 (-)	0.454372 (-)	**0.489831**(*)
(10, 10, 10)	0.243719 (-)	0.454372 (-)	**0.502489**(*)
(5, 15, 10)	0.243719 (-)	0.454372 (-)	**0.491235**(*)
(15, 10, 5)	0.243719 (-)	0.454372 (-)	**0.492404**(*)
DTLZ5 ($n = 10, M = 3, T_{max} = 14,400, \Gamma = 480$)			
(10, 5, 15)	0.057561 (-)	0.138818 (-)	**0.150756**(*)
(15, 5, 10)	0.057561 (-)	0.138818 (-)	**0.149745**(*)
(5, 10, 15)	0.057561 (-)	0.138818 (-)	**0.150895**(*)
(10, 15, 5)	0.057561 (-)	0.138818 (-)	**0.150246**(*)
(10, 10, 10)	0.057561 (-)	0.138818 (-)	**0.151179**(*)
(5, 15, 10)	0.057561 (-)	0.138818 (-)	**0.150575**(*)
(15, 10, 5)	0.057561 (-)	0.138818 (-)	**0.151458**(*)
DTLZ7 ($n = 10, M = 3, T_{max} = 14,400, \Gamma = 480$)			
(10, 5, 15)	0 (-)	0.13687 (-)	**0.360406**(*)
(15, 5, 10)	0 (-)	0.13687 (-)	**0.349183**(*)
(5, 10, 15)	0 (-)	0.13687 (-)	**0.371347**(*)
(10, 15, 5)	0 (-)	0.13687 (-)	**0.332144**(*)
(10, 10, 10)	0 (-)	0.13687 (-)	**0.371273**(*)
(5, 15, 10)	0 (-)	0.13687 (-)	**0.363476**(*)
(15, 10, 5)	0 (-)	0.13687 (-)	**0.328142**(*)
Total * / ≈ / -	0 / 0 / 0	0 / 0 / 21	21 / 0 /0

TABLE III: Median HV value comparison for the crashworthiness problem.

(ET^1, ET^2, ET^3)	NSGA-III	SA-NSGA-III	HET-NSGA-III
Crashworthiness ($T_{max} = 14,400, \Gamma = 480$)			
(10, 5, 15)	0.026269 (-)	**0.106258 (*)**	0.030322 (-)
(15, 5, 10)	0.026269 (-)	**0.106258(*)**	0.028246 (-)
(5, 10, 15)	0.026269 (-)	0.106258 (-)	**0.136039** (*)
(10, 15, 5)	0.026269 (-)	0.106258 (-)	**0.142278** (*)
(10, 10, 10)	0.026269 (-)	*0.106258 (≈)*	*0.121036 (≈)*
(5, 15, 10)	0.026269 (-)	0.106258 (-)	**0.138563** (*)
(15, 10, 5)	0.026269 (-)	0.106258 (-)	**0.141311** (*)
Total * / ≈ / -	0 / 0 / 7	2 / 1 / 4	4 / 1 / 2

REFERENCES

[1] T. W. Simpson, A. J. Booker, D. Ghosh, A. A. Giunta, P. N. Koch, and R.-J. Yang, "Approximation methods in multidisciplinary analysis and optimization: a panel discussion," *Structural and multidisciplinary optimization*, vol. 27, no. 5, pp. 302–313, 2004.

[2] R. Allmendinger and J. Knowles, "'hang on a minute': Investigations on the effects of delayed objective functions in multiobjective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2013, pp. 6–20.

[3] R. Allmendinger, J. Handl, and J. Knowles, "Multiobjective optimization: When objectives exhibit non-uniform latencies," *European Journal of Operational Research*, vol. 243, no. 2, pp. 497–513, 2015.

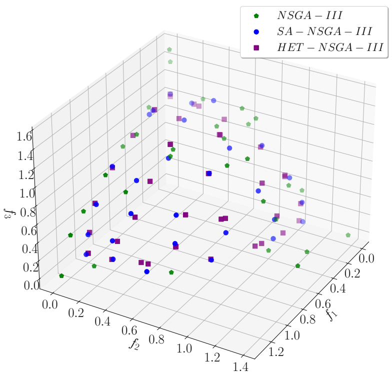
[4] T. Chugh, R. Allmendinger, V. Ojalehto, and K. Miettinen, "Surrogate-assisted evolutionary biobjective optimization for objectives with non-uniform latencies," in *Proceedings of the genetic and evolutionary computation conference*, 2018, pp. 609–616.

[5] J. Thomann and G. Eichfelder, "A trust-region algorithm for heterogeneous multiobjective optimization," *SIAM Journal on Optimization*, vol. 29, no. 2, pp. 1017–1047, 2019.

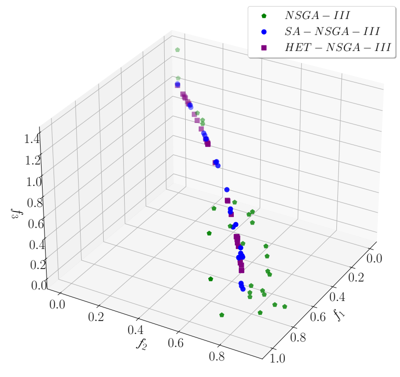
[6] X. Wang, Y. Jin, S. Schmitt, and M. Olhofer, "Transfer learning for gaussian process assisted evolutionary bi-objective optimization for objectives with different evaluation times," in *Proceedings of the 2020 genetic and evolutionary computation conference*, 2020, pp. 587–594.

[7] J. Blank and K. Deb, "Handling constrained multi-objective optimization problems with heterogeneous evaluation times: proof-of-principle results," *Memetic Computing*, vol. 14, no. 2, pp. 135–150, 2022.

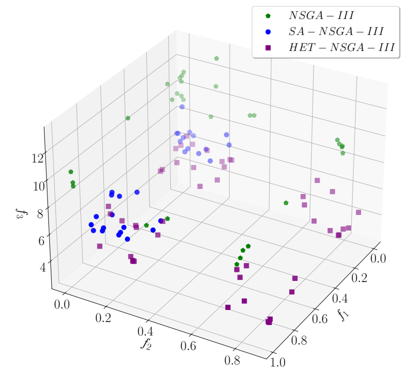
[8] M. M. Mamun, H. K. Singh, and T. Ray, "An approach for computationally expensive multi-objective optimization problems with independently



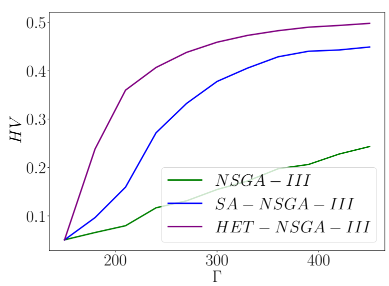
(a) $DTLZ2 - (10, 10, 10)$



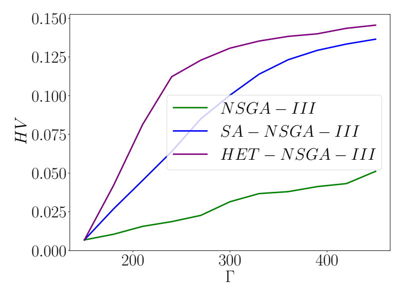
(b) $DTLZ5 - (15, 10, 5)$



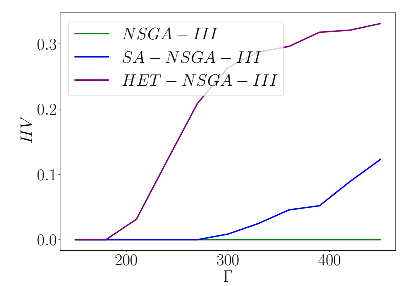
(c) $DTLZ7 - (15, 5, 10)$



(d) $DTLZ2 - (10, 10, 10) - median\ HV$

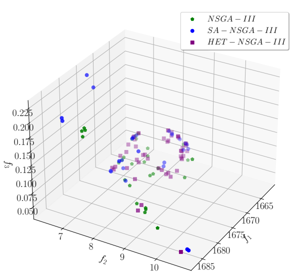


(e) $DTLZ5 - (15, 10, 5) - median\ HV$

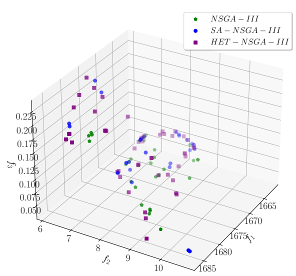


(f) $DTLZ7 - (15, 5, 10) - median\ HV$

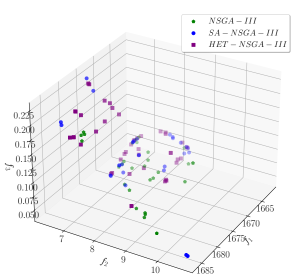
Fig. 5: Comparison of three algorithms for three-objective problems with different relative evaluation times of objectives.



(a) $(15, 5, 10)$



(b) $(5, 15, 10)$



(c) $(10, 10, 10)$

Fig. 6: Obtained solutions for 3 instances of crashworthiness.

evaluable objectives," *Swarm & Evol. Compt.*, vol. 75, p. 101146, 2022.

[9] J. M. Buckingham, S. R. Gonzalez, and J. Branke, "Bayesian optimization of multiple objectives with different latencies," *arXiv preprint arXiv:2302.01310*, 2023.

[10] R. Allmendinger and J. Knowles, "Heterogeneous objectives: state-of-the-art and future research," in *Many-Criteria Optimization and Decision Analysis: State-of-the-Art, Present Challenges, and Future Perspectives*. Springer, 2023, pp. 317–335.

[11] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[12] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 42, no. 1, pp. 55–61, 2000.

[13] D. G. Krige, "A statistical approach to some basic mine valuation problems on the witwatersrand," *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 52, no. 6, pp. 119–139, 1951.

[14] J. Blank, K. Deb, Y. Dhebar, S. Bandaru, and H. Seada, "Generating wellspaced points on a unit simplex for evolutionary many-objective optimization," *IEEE Trans. on Evol. Computation*, in press.

[15] J. Blank and K. Deb, "pymoo - Multi-objective Optimization in Python," https://pymoo.org.

[16] ——, "pysamoo: Surrogate-assisted multi-objective optimization in python," 2022.

[17] E. J. Hughes, "Evolutionary multi-objective ranking with uncertainty and noise," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2001, pp. 329–343.

[18] R. V. Hogg, J. W. McKean, and A. T. Craig, *Introduction to mathematical statistics*. Pearson, 2019.

[19] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation Journal*, vol. 8, no. 2, pp. 125–148, 2000.

[20] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proceedings of the Congress on Evolutionary Computation (CEC-2002)*, 2002, pp. 825–830.

[21] X. Liao, Q. Li, W. Zhang, and X. Yang, "Multiobjective optimization for crash safety design of vehicle using stepwise regression model," *Struc. and Multidisc. Optimization*, vol. 35, pp. 561–569, 2008.