

Addressing Heterogeneous Evaluation Times in Constrained Multi-Objective Optimization using a Mixed-Fidelity Evaluation Technique: Proof-of-Concept Results

COIN Report Number: 2025001

Balija Santoshkumar
balijasa@msu.edu
Michigan State University
East Lansing, MI, USA

Kalyanmoy Deb
kdeb@msu.edu
Michigan State University
East Lansing, MI, USA

ABSTRACT

Most practical optimization problems involve expensive evaluation procedures for computing objective and constraint functions. To obtain reasonable and accurate solutions close to true Pareto-optimal solutions, evolutionary multi-objective optimization (EMO) algorithms create surrogate models from already-evaluated high-fidelity solutions and use them during optimization to save computational time. However, most surrogate-assisted EMO algorithms are designed to evaluate all objectives and constraints of a solution, if found worthy of a high-fidelity evaluation. Such algorithms are inefficient if objectives and constraints involve heterogeneity with orders of magnitude of difference in evaluation times. Clearly, functions with relatively small evaluation time can be high-fidelity evaluated more often to obtain an overall idea of the potential importance of the solution before deciding to spend more time on evaluating expensive functions. In this paper, we propose an EMO approach that carefully determines which constraints and objectives should be high-fidelity evaluated for every population member and suggests a mixed-fidelity survival selection procedure capable of working with low- and high-fidelity evaluated population members. Results on a number of test and engineering problems indicate the viability of such a constrained multi- and many-objective optimization algorithm and encourage further attention.

KEYWORDS

Constrained surrogate-assisted optimization, Heterogeneous evaluation, Multi-objective optimization

1 INTRODUCTION

Most practical optimization problems involve expensive evaluations of objectives and constraints over multiple iterations to find near Pareto-optimal (PO) solutions. This expensive evaluation of objectives and constraints becomes particularly challenging when solutions are evaluated using one or more third-party modeling and simulation software. To overcome this challenge, surrogate-assisted optimization (SAO) methods [4, 5, 8, 9, 15, 18, 23, 26, 31] have emerged as powerful approaches for reducing the overall computational burden. Surrogate models are mathematical representations of the original expensive objective and constraint functions that are inexpensive and easier to evaluate. These surrogate models are utilized during the majority of the optimization iterations, significantly reducing the overall computational cost.

While the concept of surrogate-assisted optimization (SAO) shows promise, evolutionary multi/many-objective optimization (EM(a)O) researchers face several challenges: (i) identifying which new solutions require high-fidelity evaluation, (ii) selecting the most appropriate surrogate modeling technique for the current high-fidelity evaluated solutions, and (iii) integrating the surrogate-assisted search with the optimization algorithm.

The first task involves assessing existing solutions that enhance the search to predict new solutions. The second task may require testing and validating different surrogate modeling methods to find the most effective one at various optimization stages. After developing surrogate models, in the third task, they must be optimized to find new (infill) solutions for further evaluation. Although evaluating surrogate models is less expensive, they may not accurately reflect the true values of the original objectives, highlighting the need for an assessment of surrogate-evaluated solutions to avoid selecting the solutions that are not corresponding to true Pareto-optimal regions.

The three tasks mentioned above need to be coordinated adaptively to ensure that progressive surrogate modeling, through new infill solutions and suitable surrogate modeling techniques, consistently move toward the near Pareto-optimal regions while minimizing the number of high-fidelity evaluations required.

In most of the existing surrogate-assisted multi/many-objective optimization (SA-EM(a)O) studies, every infill solution selected for high-fidelity evaluation is evaluated for all (M) objectives and all (J) constraints, either individually or in an aggregated manner [13]. Since the entire solution is evaluated for all objectives and constraints, this can lead to unnecessary high-fidelity evaluations for certain objectives and constraints, resulting in inefficient use of computational budget. The aspect of partial evaluation of objectives and constraints for a solution to save computational time, leading to a mixed-fidelity evaluation (MFE) of objectives and constraints of population members, has received limited attention in the current SA-EM(a)O literature [1–5, 23, 25, 27]. In this paper, we propose an MFE-EM(a)O framework extending a previous study [28] for handling constraints, that determines a high-fidelity selection criterion for each objective and constraint for each population member, considering the following aspects:

- (1) The solution being largely feasible, largely infeasible, or close to the constraint boundary considering the uncertainty in the predicted solution.

- (2) The probability of solution being non-dominated within its neighborhood in the sense of reference vectors [14, 30].
- (3) The probability of the solution being feasible.
- (4) The high-fidelity evaluation time of objectives, and
- (5) The uncertainty in the predicted solutions.

MFE-EM(a)O framework results in a population with mixed-evaluated members, in which some objectives and constraints possess surrogate-evaluated values, and some possess high-fidelity evaluated values. At the end of an optimization run, we evaluate all surviving population members with high-fidelity evaluation and present feasible non-dominated solutions.

The remainder of the paper presents the proposed mixed-fidelity evaluation based evolutionary multi- and many-objective optimization (MFE-EM(a)O) algorithm. Details of the proposed method with sketches are presented in Section 2. Section 3 presents results on multi- and many-objective test, engineering problems. Finally, Section 4 summarizes the conclusions from this study and proposes potential future research directions.

2 PROPOSED MIXED-FIDELITY EVALUATION FRAMEWORK: MFE-EM(A)O

We use the elitist evolutionary many-objective optimization algorithm, NSGA-III [14, 17], to incorporate the MFE algorithm. This approach is also compatible with other reference vector-based EM(a)O algorithms [30]. The process begins by generating a set of N reference vectors, which can be generated using the Das and Dennis method [10] or the Riesz energy method [7]. The objective of the proposed MFE-NSGA-III algorithm is to identify up to N non-dominated (ND) solutions that are as close as possible to the Pareto front (PF) of the problem, while carefully determining which objectives and constraints require high-fidelity evaluation for each selected solution.

Each member of the initial population P_{DoE} of size N_{DoE} , generated using the Latin Hypercube Sampling (LHS) method [24], is evaluated with high-fidelity for all objectives (M) and constraints (J). Assuming that the total time required to evaluate all objectives and constraints is one unit, the initial design of experiments (DoE) population P_{DoE} of size N_{DoE} consumes $T = N_{DoE}$ units of time before starting the proposed EM(a)O procedure. A maximum of T_{max} time is allowed for finding the non-dominated (ND) set, of which the final N units time is reserved for evaluating at most N final population members with high-fidelity evaluations. Thus, $(T_{max} - N)$ time is kept for MFE-EM(a)O generations.

A surrogate model for each objective and constraint function is constructed using N_{DoE} high-fidelity evaluations. An initial population P_0 of size N ($N \leq N_{DoE}$) is selected from the N_{DoE} solutions using EM(a)O survival operator. Then, the chosen EM(a)O algorithm (NSGA-III [14, 17] here) is executed with the initial population P_0 for t_S generations, relying solely on the surrogates, thereby avoiding any additional high-fidelity evaluations. At the end of this surrogate-assisted NSGA-III run, the final population Q_0 is combined with P_0 to form a merged population $R_0 = P_0 \cup Q_0$, where no member of Q_0 has been high-fidelity evaluated yet. The iteration counter is initialized to $t = 0$.

2.1 Mixed-Fidelity Survival Selection Procedure

Each population member s , in the combined population of R_t is then associated with one of the supplied reference vectors. To make a fair computation, all solutions for which a high-fidelity evaluation of one or more objectives and constraints were already made are re-evaluated with the respective surrogate model for executing surrogate-assisted NSGA-III run. The mixed-fidelity survival selection procedure is presented below.

Step 1: Classifying population into feasible and infeasible classes: First, the population members (s) are classified into two classes: (i) Feasible (Class I) and (ii) Infeasible (Class II) solutions, based on a statistic, defined below:

$$Z_s^j = \frac{-\mu_s^j}{\sigma_s^j}. \quad (1)$$

The terms μ_s^j and σ_s^j are prediction and associated uncertainty in prediction of j -th constraint for solution s using the Kriging surrogate model proposed elsewhere [6, 19, 21]. We use the same Kriging models as in the previous study [28]. Note that for feasible solutions, predicted constraint value $\mu_s^j \leq 0$ and $Z_s^j \geq 0$. But, due to the uncertainty of prediction with the surrogate models, we re-define a feasible solution, as follows. If for a solution s $\min_{j=1}^J (Z_s^j) \geq e$ (where e is a small negative value), s is declared feasible. We have chosen here $e = -1$ by performing a limited number of trial-and-error experiments. This means that we can include solutions having the worst constraint value up to one-sigma of its predicted value on the infeasible side as feasible solutions. Figure 1 shows (A, B , and C) as Class I members. Point D is considered a member of Class II (infeasible class).

If the number of population members in Class I is greater than N (population size), survival selection must choose best N solutions for the next generation parent population (P_{t+1}). Step 2 of the procedure achieves this task.

Step 2: Choosing N feasible solutions for next generation: A mixed-fidelity selection (MFS) metric ρ_s^m is calculated for every objective (m) of every feasible solution (s), as follows:

$$\rho_s^m = P_s^m \left(1 + \left(\frac{\sigma_s^m}{|\Delta f_m|} \right)^{\frac{1}{\eta}} \right) \left(1 + \widehat{ET}^m \right)^\alpha. \quad (2)$$

Here, P_s^m refers to the probability of solution s being non-dominated among its neighboring members (in the sense of reference vectors [14, 30]), discussed in a previous study [28]. Parameters σ_s^m and Δf_m are surrogate uncertainty and range of m -th objective in the population, respectively. The parameter $\eta = 20$ has worked well in this study. The α -update on the term for average evaluation time \widehat{ET}^m is described in Subsection 2.2. The motivation for the structure of Equation 2, also described in Subsection 2.2, comes from the successful previous study on heterogeneous evaluation of unconstrained objectives with unequal computational times. The procedure for deciding which objectives and constraints to be high-fidelity evaluated is described next.

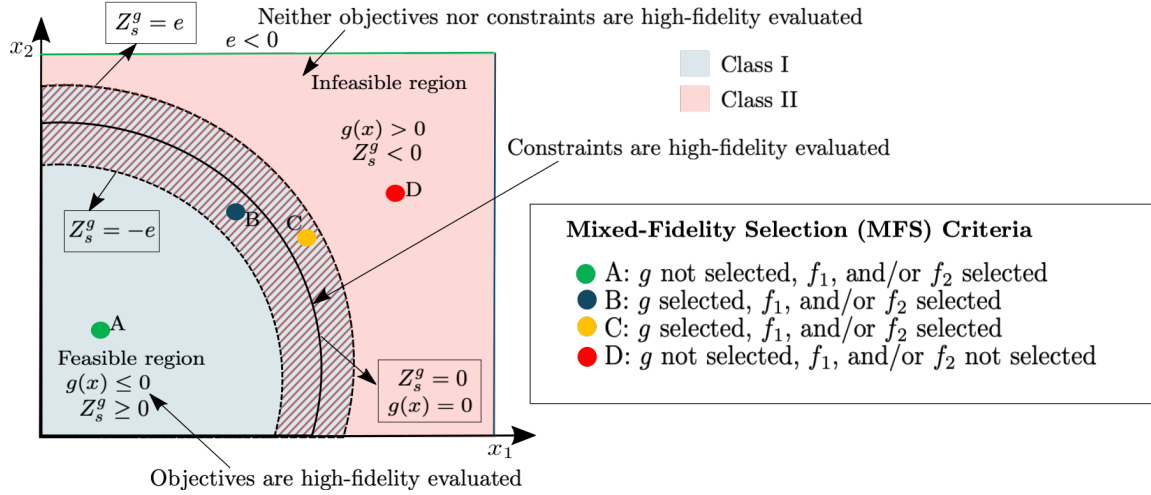


Figure 1: Mixed-fidelity selection procedure. Solutions A, B, and C are in Class I: either feasible or close to the constraint boundary, hence some of their objective and constraint functions are selected for high-fidelity evaluation based on ρ_s^m and e values. Solution D is in Class-II: infeasible and far away from the constraint boundary, hence it is not selected for either constraints or objectives for high-fidelity evaluations.

For every feasible population member (s), ρ_s^m for every objective function (m) is computed. For each reference vector w , all associated population members are identified and then the solution \bar{s} with largest ρ_s^m is chosen. If this population member has already been high-fidelity evaluated in a previous generation, it is simply included in P_{t+1} and no new high-fidelity evaluation is scheduled. Otherwise, for this solution, if any constraint function value is within plus or minus of one σ_s^j (surrogate's error) from zero, the solution is close to the constraint boundary and the constraint is considered worth evaluating with its high-fidelity model. If any constraint (j) is largely satisfied (with $Z_s^j > -e$), then there is no real need to make high-fidelity evaluation of that constraint (j). Thus, by the above process, not all J constraints for a solution \bar{s} may be chosen for high-fidelity evaluation and only the ones close to their constraint boundaries are marked for high-fidelity evaluation.

After the specific feasible population member's (\bar{s}) specific objective and constraint functions are chosen for high-fidelity evaluation, the procedure is repeated for the next reference vector. This process continues until all reference vectors are considered. If less than N population members are selected by this process, it starts again with the first reference vector and its associated solution having the second-best ρ_s^m value. In case this objective function happens to be for a population member already selected for high-fidelity evaluation of its constraints and a different objective function in the previous cycle, it is not selected again for constraint evaluation, but is simply marked for high-fidelity evaluation for the second objective function. By this process, exactly N population members are selected

for P_{t+1} for which critical constraints and at least one critical objective function for each of member have been chosen for high-fidelity evaluation.

Step 3: Handling infeasible solutions for next generation:

In case the number of feasible solutions is less than N , all feasible solutions are included in P_{t+1} and the procedure of Step 2 for deciding which constraints and objectives to be high-fidelity evaluated is used for all feasible population members. The remaining population slots are filled with better infeasible solutions. If one or more constraints of each infeasible solution are violated largely (with $\max_{j=1}^J Z_s^j < e$), no high-fidelity evaluation of any constraint or objective is performed for these solutions. However, solutions having a higher probability of constraint satisfaction (PG_s) are chosen for P_{t+1} . The formulation of overall constraint satisfaction (PG_s) is defined below:

$$PG_s = \prod_{j=1}^J P(g_j(s) \leq 0), \quad (3)$$

$$\text{where, } P(g_j(s) \leq 0) = \Phi\left(\frac{Z_s^j}{\sigma_s^j}\right) = \frac{1}{2} \left(1 + \operatorname{erf}\left(\frac{Z_s^j}{\sqrt{2}\sigma_s^j}\right)\right).$$

All selected (and previously unevaluated) solutions are then high-fidelity evaluated for respective objective and constraint functions and the time counter T is incremented with the respective ET^m and ET^j evaluation times, as the case may be. These new evaluated solutions are combined with previous high-fidelity evaluated solutions and surrogate models are updated. Thereafter, the survived population is re-evaluated using newly created surrogate models and another t_S generations of surrogate-assisted NSGA-III is performed to create a new offspring population Q_{t+1} . This process is continued until $(T_{\max} - N)$, where N is the population size) units of

time is elapsed. At this point, all final N solutions are high-fidelity evaluated and feasible non-dominated solutions are reported.

Figure 1 illustrates the above-described mixed-fidelity selection (HFS) procedure. The constraint boundary $g(\mathbf{x}) = 0$ and the two other boundaries $g(\mathbf{x}) = e\sigma_x^g$ and $g(\mathbf{x}) = -e\sigma_x^g$ are also shown. Solution A is feasible ($g(A) < 0$), hence one of critical objective functions (decided by ρ_s^m metric) will be chosen for high-fidelity evaluation. But, since it is too far into the feasible region (with $Z_A^g > -e$), a high-fidelity evaluation of the constraint function may not yield any new information, hence the constraint will not be chosen for high-fidelity evaluated. For multiple constraints, all constraints that are feasible at the current point s with a large margin ($Z_s^j < e$) will not be chosen, but those that cause the current point s to fall close their boundaries ($e \leq Z_s^j \leq -e$) are chosen for high-fidelity evaluation. For the constraint in the figure, solution B is feasible ($g(B) < 0$) and it is close to the constraint boundary (with $0 < Z_B^g < -e$), hence the critical objective and the constraint will be high-fidelity evaluated for obvious reasons. Solution C, on the other hand, is infeasible ($g(C) > 0$), but the constraint violation is small ($0 > Z_C^g > e$, hence one of the critical objectives and the constraint will also be high-fidelity evaluated. However, solution D is largely infeasible ($g(D) > 0$ and $Z_D^g < e$), hence it is neither considered for high-fidelity evaluation of any of the objectives or the constraint. The step-by-step proposed MFS procedure described above implements these aspects of four different types of solutions.

Initially when the surrogate models are not precise due to a small number of points being used to model the entire search space, it is likely that prediction uncertainties σ_s of objectives and constraints will be large. This will allow many largely-violated constraints ($g_j < \sigma_s^g$, for $e = -1$) to be selected for high-fidelity evaluations. But with iterations when more high-fidelity solutions near the PO front are used for surrogate model creation, the prediction error in their neighborhood becomes small. Our proposed stochastic constraint handling method then does not allow largely violated constraints to be high-fidelity evaluated. Only when a solution lies close to a constraint boundary, the respective constraint will be allowed to be high-fidelity evaluated. Another important feature of our procedure is that if any one of the constraints is largely violated by a solution s , it is declared as Class II solution and no objective or constraint is chosen for high-fidelity evaluation for this solution. All these features are adaptive and cause a considerable saving in computational time to complete an execution of the EM(a)O run without much sacrifice on information loss.

Notice also that for unconstrained problems (without any constraint), every s is feasible. In this case, $Z_s = \infty$ and Step 3 does not get executed and the above technique simply chooses a single objective \bar{m} for high-fidelity evaluation for each solution \bar{s} at a time. This process is similar to the heterogeneous unconstrained surrogate-assisted algorithm proposed earlier and was shown to perform extremely well on unconstrained problems [28]. Thus, the previously proposed heterogeneous unconstrained surrogate-assisted optimization algorithm becomes a degenerate case of our proposed constraint handling procedure.

2.2 Motivation for Mixed-Fidelity Selection Metric

The mixed-fidelity selection (MFS) metric (ρ_s^m), proposed in Equation 2, determines a potential metric for high-fidelity evaluation of objective m for a population member s involving probability of constraint satisfaction. Intuitively, this requires a number of factors. First, the MFS metric (ρ_s^m) must be given more emphasis for quick or expensive evaluation of objectives depending on the generation counter – a matter established in a previous study in [20] for a different purpose. This is achieved by varying the α parameter as a function of elapsed time T , as follows:

$$\alpha = \frac{((T - \tau) - (T_{\max} - T))}{T_{\max} - \tau}, \quad (4)$$

where, $\tau = N_{DoE}$ budget spent on the initial design of experiments and T_{\max} is the total allowed time. It emphasizes more high-fidelity evaluation of less expensive objectives in the beginning and more expensive ones at the end.

The first term (P_s^m) is the probability of the solution s to be better with respect to m -th objective. Its computation procedure was discussed elsewhere [28]. Higher the probability, more potential is the solution and more emphasis is given for its high-fidelity evaluation.

The second term takes into account the surrogate prediction error σ_s^m . More the error with respect to the range of objective value in the population (Δf_m), the higher is the need for its high-fidelity evaluation. The parameter η controls its effect and a value around 20 or so was found to be a good number [28]. All these terms and their use in the combined equation are intuitive, but the reason for adding one to some of the terms is as follows. Among all three terms, we believe the second term signifies its potential to be better in its own neighborhood for the m -th objective, and hence, it is the more important factor. By adding one to other two terms and not to the second term, we dilute the correlation of other two terms in the computation of the overall MSF metric ρ_s^m .

3 RESULTS

We test the proposed approach MFE-NSGA-III with two other algorithms run for an identical T_{\max} : (i) the original NSGA-III [14, 17] in which every population member is high-fidelity evaluated for all objectives and constraints without any use of surrogates, and (ii) SA-NSGA-III, in which surrogates are used for optimization for t_S generations, but all offspring population members of Q_{t+t_S} are high-fidelity evaluated for all objectives and constraints and surrogates are then updated for next generation.

In this study, we test these three methods on several multi- and many-objective problems: TNK, CTP1 [12], C2DTLZ2 [17], MW3, MW7 [22], and two engineering problems [11, 17, 29]. They cover 2-10 variables, up to 10 constraints, and 2-5 objectives.

We set population size (N) to 50 for all problems, and the number of reference directions is set equal to N . We use the maximum termination budget (T_{\max}) as 300. The number of initial design of experiments (N_{DoE}) is 100 to build surrogate models initially. We set the surrogate generations (t_S), used to generate the offspring population (Q_{t+t_S}), to 10, obtained by performing a limited number of experiments. Additionally, based trial-and-error experiments, we

set $\eta = 20$ (Equation 2) to control the emphasis on uncertainty in the surrogate modeling of objectives.

As mentioned before, we assume that the estimated total evaluation time of a solution for all objectives (M) and all constraints (J) is one unit and use relative evaluation times (ET^m) of each objective and (ET^j) of each constraint. For example, $ET^{m_1} = ET^{m_2} = ET^{j_1} = ET^{j_2} = 0.25$ mean that all objectives and constraints take more or less an identical average time to evaluate a solution. However, $ET^{m_1} = 0.7$, $ET^{m_2} = 0.1$, $ET^{j_1} = 0.15$, $ET^{j_2} = 0.05$ mean that objectives take 80% and constraints take 20% of the total evaluation time of a solution, and in which, the first objective is seven times more expensive than the second, and the first constraint is three times more expensive than the second. In the original NSGA-III and SA-NSGA-III algorithms, the parameter T corresponds directly to the number of high-fidelity solution evaluations (SE), as both methods perform high-fidelity evaluations for all objectives and all constraints for every solution. However, in MFE-NSGA-III, we increment T by ET^m each time a solution undergoes high-fidelity evaluation for the m -th objective and by ET^j for high-fidelity evaluation of the j -th constraint. We have studied various evaluation time scenarios, including cases where objectives require more time than constraints, and vice versa.

For each problem instance, we execute all algorithms 15 times using different initial populations. We select IGD^+ [16] as the performance metric. To compute IGD^+ , we normalize the population members based on the nadir and ideal points of the known PO front. In cases where the PO front is unavailable, particularly in engineering design problems, we derive an approximated PO front from the study by Tanabe et al. [29]. Since different runs of MFE-NSGA-III may record different IGD^+ values at varying T values, we employ an interpolation strategy to estimate the IGD^+ value between the two nearest T values. Additionally, because each run may generate a distinct surrogate model at a specific T , we compute the IGD^+ using high-fidelity evaluations for all algorithms to ensure consistency for reporting purposes. Moreover, MFE-NSGA-III high-fidelity evaluates all solutions at the end to present feasible non-dominated solutions. To identify statistically significant differences in algorithm performance, we apply the Wilcoxon signed-rank test with a significance level of $p = 0.05$, categorizing algorithms as statistically best, similar, or worst-performing based on the results.

To reveal the working procedure of the proposed approach (MFE-NSGA-III), we consider its performance on the TNK problem first with the assigned evaluation times as $ET^{m_1} = 0.05$ for objective f_1 , $ET^{m_2} = 0.15$ for objective f_2 , $ET^{j_1} = 0.7$ for constraint g_1 , and $ET^{j_2} = 0.1$ for constraint g_2 units of evaluation time and denote it as TNK-(0.05, 0.15 | 0.7, 0.1) in the order of evaluation time first objective to last constraint. Figure 2a presents the final solutions by three algorithms with constraint boundaries g_1 , g_2 , and the feasible region. Figure 2b shows the number of high-fidelity evaluations for f_1 , f_2 , g_1 and g_2 with respect to T in MFE-NSGA-III. This figure clearly demonstrates selective high-fidelity evaluations for objective and constraint functions. In this problem, the constraint function g_2 is less frequently evaluated using high-fidelity models because the constraint g_2 is largely inactive in the regions close to the PO front. Our proposed approach automatically detects it and does not high-fidelity evaluate g_2 for all solutions to save computational budget. Since constraints take 80% of total evaluation

time, and in this problem feasible solutions are relatively easy to find, MFE-NSGA-III high-fidelity evaluates objectives more often to get closer to the PO front. Figure 2c shows the total number of high-fidelity evaluations in all three algorithms, showing fewer overall evaluations of the constraint functions.

3.1 Two-Objective Constrained Problems

First, we consider two-objective constrained test and Welded Beam engineering problems. The median IGD^+ values of the final population for all algorithms are presented in Table 1. We have considered different relative evaluation times of objectives and constraints, with at most one of the objectives being 14 times more expensive than a constraint, and vice versa. The final generation solutions and IGD^+ convergence plots for MW3, C2DTLZ2, and CTP1 problems are presented in Figure 3.

The solutions obtained for a specific test instance of evaluation times are presented in Figures 3a, 3b, and 3c for the same overall evaluation time T_{\max} and varying relative evaluation times (ET) for objectives and constraints. Figures 3d, 3e, and 3f present the convergence of IGD^+ for the three methods, plotted against the equivalent solution evaluations T on the x-axis. While MFE-NSGA-III and SA-NSGA-III begin with 100 initial high-fidelity evaluations to build their first surrogate models, NSGA-III starts with an initial population of $N = 50$ members. The gray-colored dashed lines represent iterations where all solutions are infeasible. They are included to show the same starting point of SA-NSGA-III and MFE-NSGA-III. Colored lines indicate that at least one feasible solution exists in the respective non-dominated front. We can observe the remarkable performance of MFE-NSGA-III in IGD^+ convergence plots. MFE-NSGA-III achieves better performance by selecting solutions for high-fidelity constraint evaluation, if they are close to constraint boundaries and objective evaluation, if it is a potential non-dominated, adequately time-consuming and critically important for high-fidelity evaluation based on Equation 2. In a few problems, not all constraints may be predominantly active; in such cases, MFE-NSGA-III carefully decides for a particular solution s which constraints and objectives need high-fidelity evaluations to utilize the computational budget efficiently and go for more iterations with the saved budget to achieve superior performance, whereas most of the existing SA-EMO methods evaluate all objectives and constraints for the selected solution.

3.2 Results on Three and Many-Objective Constrained Problems

Next, we present results on three and five-objective C2DTLZ2 test problems and the Carside [17] engineering problem. For the three-objective C2DTLZ2 and Carside problems, the obtained final solutions with all three methods are presented in Figures 4a and 4b. Figure 4c shows the final generation objective function solutions on PCP for the five-objective C2DTLZ2 problem. Figures 4d, 4e and 4f show IGD^+ convergence plots of three-objective C2DTLZ2, Carside, and five-objective C2DTLZ2 problems respectively, clearly demonstrating the superiority of the proposed MFE-NSGA-III. The obtained median IGD^+ values of the final generation population members are presented in Table 1. The proposed MFE-NSGA-III approach demonstrates enhanced effectiveness in addressing many-objective and constrained problems by leveraging its capability

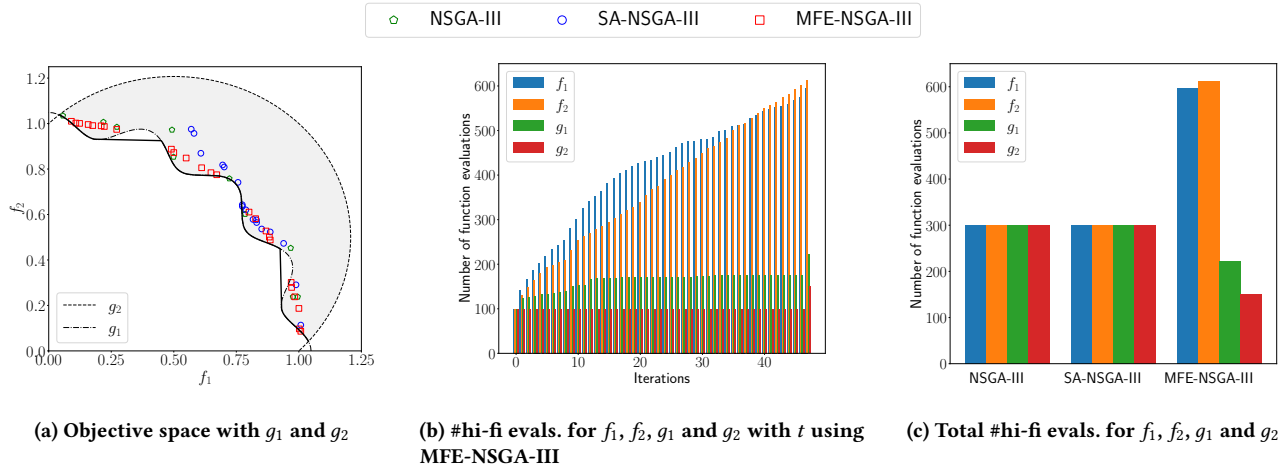


Figure 2: Obtained final solutions and number of high-fidelity evaluations recorded for f_1, f_2, g_1 and g_2 on TNK-(0.05, 0.15 | 0.7, 0.1) with $T_{\max} = 300$, MFE-NSGA-III uses less evaluation of the constraints compared to objectives to produce a better performance. Notice the less number of g_2 evaluations, as the PO set is on g_1 . Since objective evaluation is cheaper than constraints in this problem, MFE-NSGA-III assigns more objective evaluations within the stipulated overall evaluation time.

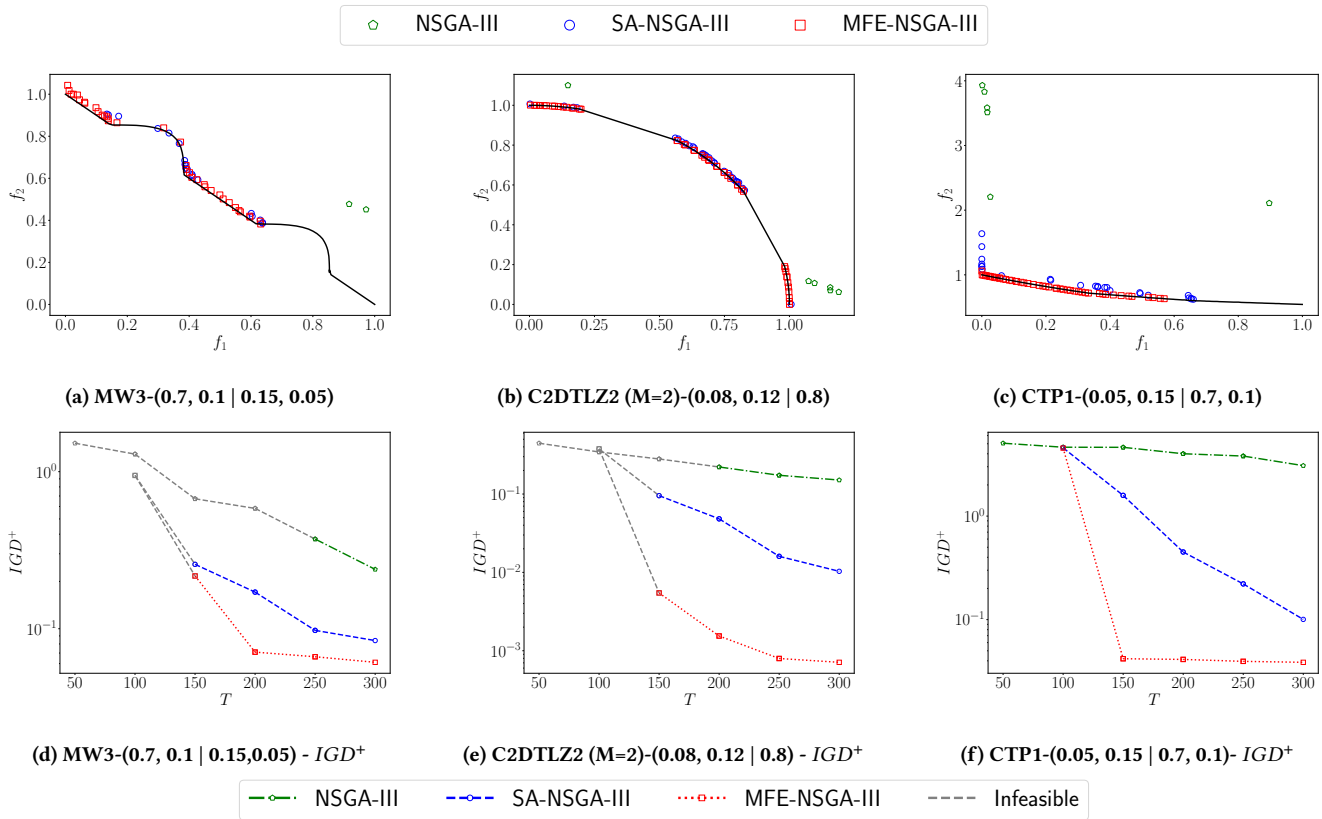


Figure 3: Comparison of algorithms for two-objective constrained problems with different relative evaluation times of objectives and constraints. In the IGD^+ plots in the second row, the gray-colored lines indicate that no feasible solutions are found after T units of time, while all colored lines correspond to populations having at least one feasible solution available for computing the IGD^+ metric. MFE-NSGA-III points are closer to the true PO set than other two algorithms and are achieved with identical $T_{\max} = 300$ units of evaluation time.

Table 1: Median IGD^+ value comparison of algorithms for multi-and many-objective problems. The numbers in the final row indicate the number of times the respective algorithm is better, equivalent, or worse (statistically) with respect to MFE-NSGA-III. The numbers in the final column and final row indicate the number of times MFE-NSGA-III is better, equivalent, and worse than any other algorithm used in this study. IGD^+ values for statistically equivalent algorithms (with 95% confidence level) are marked in bold italics and for the best performing algorithm is marked in bold.

Problem	$(ET^{m_1} \dots ET^{m_M} \mid ET^{J_1} \dots ET^{J_J})$	NSGA-III	SA-NSGA-III	MFE-NSGA-III
C2DTLZ2 (M=2, J=1)	(0.33, 0.5 0.17)	1.5090e-1	1.0316e-2	4.5954e-3
C2DTLZ2 (M=2, J=1)	(0.08, 0.12 0.8)	1.5090e-1	1.0316e-2	6.0429e-4
C2DTLZ2 (M=2, J=1)	(0.33, 0.33 0.33)	1.5090e-1	1.0316e-2	3.8559e-3
C2DTLZ2 (M=2, J=1)	(0.12, 0.08 0.8)	1.5090e-1	1.0316e-2	5.2572e-4
C2DTLZ2 (M=2, J=1)	(0.5, 0.33 0.17)	1.5090e-1	1.0316e-2	4.5157e-3
C2DTLZ2 (M=2, J=1)	(0.4, 0.4 0.2)	1.5090e-1	1.0316e-2	3.5736e-3
MW3 (M=2, J=2)	(0.25, 0.25 0.25, 0.25)	2.3881e-1	8.3882e-2	5.0393e-2
MW3 (M=2, J=2)	(0.05, 0.15 0.7, 0.1)	2.3881e-1	8.3882e-2	7.0410e-2
MW3 (M=2, J=2)	(0.05, 0.15 0.1, 0.7)	2.3881e-1	8.3882e-2	6.6914e-2
MW3 (M=2, J=2)	(0.7, 0.1 0.15, 0.05)	2.3881e-1	8.3882e-2	6.0484e-2
MW3 (M=2, J=2)	(0.1, 0.7 0.15, 0.05)	2.3881e-1	8.3882e-2	7.7318e-2
MW7 (M=2, J=2)	(0.25, 0.25 0.25, 0.25)	2.7449e-1	9.4441e-2	1.3889e-1
MW7 (M=2, J=2)	(0.05, 0.15 0.7, 0.1)	2.7449e-1	9.4441e-2	2.4811e-1
MW7 (M=2, J=2)	(0.05, 0.15 0.1, 0.7)	2.7449e-1	9.4441e-2	2.5631e-1
MW7 (M=2, J=2)	(0.7, 0.1 0.15, 0.05)	2.7449e-1	9.4441e-2	2.4678e-1
MW7 (M=2, J=2)	(0.1, 0.7 0.15, 0.05)	2.7449e-1	9.4441e-2	2.3958e-1
CTP1 (M=2, J=2)	(0.25, 0.25 0.25, 0.25)	3.0725e+0	1.0051e-1	3.5074e-2
CTP1 (M=2, J=2)	(0.05, 0.15 0.7, 0.1)	3.0725e+0	1.0051e-1	3.8309e-2
CTP1 (M=2, J=2)	(0.05, 0.15 0.1, 0.7)	3.0725e+0	1.0051e-1	3.6374e-2
CTP1 (M=2, J=2)	(0.7, 0.1 0.15, 0.05)	3.0725e+0	1.0051e-1	2.9666e-2
CTP1 (M=2, J=2)	(0.1, 0.7 0.15, 0.05)	3.0725e+0	1.0051e-1	3.8417e-2
TNK (M=2, J=2)	(0.25, 0.25 0.25, 0.25)	7.5656e-2	1.4925e-2	8.6070e-3
TNK (M=2, J=2)	(0.05, 0.15 0.7, 0.1)	7.5656e-2	1.4925e-2	2.1107e-2
TNK (M=2, J=2)	(0.05, 0.15 0.1, 0.7)	7.5656e-2	1.4925e-2	7.7710e-3
TNK (M=2, J=2)	(0.7, 0.1 0.15, 0.05)	7.5656e-2	1.4925e-2	9.3400e-3
TNK (M=2, J=2)	(0.1, 0.7 0.15, 0.05)	7.5656e-2	1.4925e-2	8.9470e-3
Welded Beam (M=2, J=4)	(0.2, 0.3 0.15, 0.1, 0.15, 0.1)	8.1271e-2	1.3146e-1	2.7624e-2
Welded Beam (M=2, J=4)	(0.08, 0.12 0.25, 0.2, 0.15, 0.2)	8.1271e-2	1.3146e-1	1.8428e-2
Welded Beam (M=2, J=4)	(0.3, 0.2 0.15, 0.1, 0.15, 0.1)	8.1271e-2	1.3146e-1	4.4226e-2
Welded Beam (M=2, J=4)	(0.16, 0.16 0.16, 0.16, 0.16, 0.16)	8.1271e-2	1.3146e-1	2.1620e-2
C2DTLZ2 (M=3, J=1)	(0.25, 0.25, 0.25 0.25)	3.8838e-1	1.9329e-1	3.0290e-2
C2DTLZ2 (M=3, J=1)	(0.05, 0.15, 0.7 0.1)	3.8838e-1	1.9329e-1	2.5230e-2
C2DTLZ2 (M=3, J=1)	(0.05, 0.15, 0.1 0.7)	3.8838e-1	1.9329e-1	2.2998e-2
C2DTLZ2 (M=3, J=1)	(0.7, 0.1, 0.15 0.05)	3.8838e-1	1.9329e-1	2.9782e-2
C2DTLZ2 (M=3, J=1)	(0.1, 0.7, 0.15 0.05)	3.8838e-1	1.9329e-1	3.1727e-2
Carside (M=3, J=10)	(0.3, 0.2, 0.2 0.02, 0.01, 0.05, 0.01, 0.03, 0.05, 0.01, 0.05, 0.02, 0.05)	4.6211e-2	4.6281e-3	8.2246e-4
Carside (M=3, J=10)	(0.076923, ... (repeated 13 times))	4.6211e-2	4.6281e-3	1.5360e-3
Carside (M=3, J=10)	(0.2, 0.2, 0.3 0.01, 0.02, 0.05, 0.01, 0.03, 0.05, 0.01, 0.05, 0.02, 0.05)	4.6211e-2	4.6281e-3	1.7306e-3
C2DTLZ2 (M=5, J=1)	(0.2, 0.3, 0.15, 0.1, 0.15 0.1)	5.1711e-1	3.5365e-1	1.9117e-1
C2DTLZ2 (M=5, J=1)	(0.08, 0.12, 0.25, 0.15, 0.2 0.2)	5.1711e-1	3.5365e-1	1.7213e-1
C2DTLZ2 (M=5, J=1)	(0.04, 0.04, 0.04, 0.04, 0.04 0.8)	5.1711e-1	3.5365e-1	1.7463e-1
C2DTLZ2 (M=5, J=1)	(0.16, 0.16, 0.16, 0.16, 0.16, 0.16)	5.1711e-1	3.5365e-1	1.7068e-1
C2DTLZ2 (M=5, J=1)	(0.1, 0.05, 0.15, 0.1, 0.15 0.45)	5.1711e-1	3.5365e-1	2.7178e-1
Total + / = / - of 43 cases		0 / 0 / 43	0 / 10 / 33	(33 / 10 / 0)

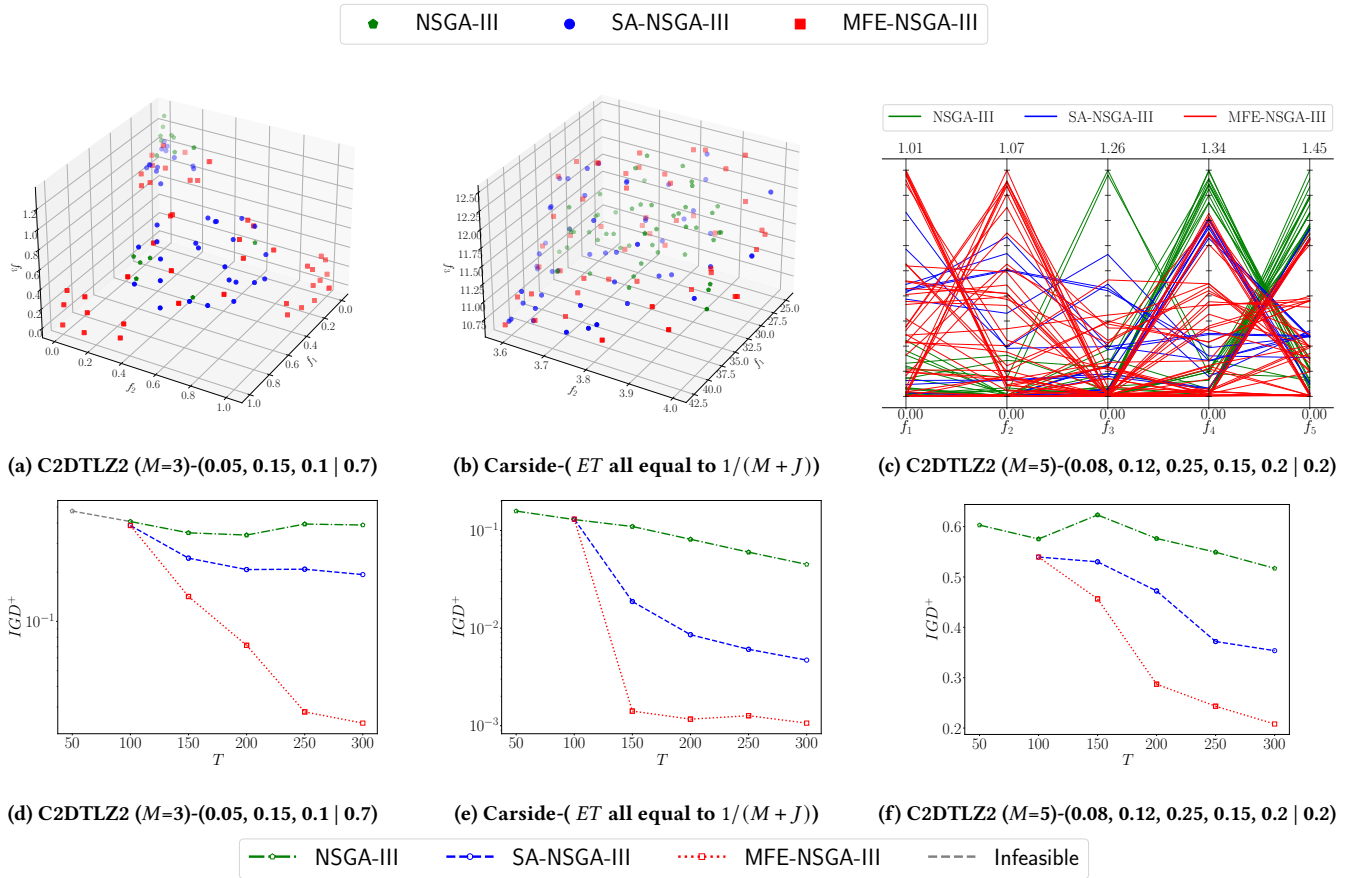


Figure 4: Comparison of algorithms for three and five-objective constrained problems with different relative evaluation times of objectives and constraints.

to selectively evaluate objectives and constraints for the selected solutions.

As presented in the table, out of 43 cases, MFE-NSGA-III performs statistically best in 33 of them and equivalent in 10 other cases with SA-NSGA-III, whereas SA-NSGA-III performs better than MFE-NSGA-III in none of the cases. These are remarkable results, clearly demonstrating the need for individualistic evaluation of objectives and constraints to arrive at a computationally efficient algorithm.

4 CONCLUSIONS

This paper has presented a mixed-fidelity evaluation algorithm to handle heterogeneous evaluation times in constrained multi-objective and many-objective optimization problems. The proposed method selects solutions for high-fidelity evaluation based on their feasibility and proximity to constraint boundaries while considering the predicted solution’s uncertainty and heterogeneous evaluation times of objectives and constraints. Extensive testing on multi- and many-objective benchmark and engineering problems has demonstrated that the proposed algorithm outperforms two other algorithms. Notably, MFE-NSGA-III achieves impressive efficiency by allowing unequal high-fidelity evaluations of objectives

and constraints, and selectively evaluating requisite objectives and constraints.

The proposed MFE-NSGA-III is now ready to be applied to more challenging multi- and many-objective problems and to be compared its performance with other existing constrained SA-EMO algorithms by performing comprehensive sensitivity studies on algorithmic parameters. More efficient ρ -metric and overall surrogate evaluation assignment procedures can be developed. Nevertheless, results of this pilot study are extremely encouraging and indicate a viable direction for developing more efficient and flexible algorithms for multi- and many-objective constrained optimization problems having heterogeneous evaluation times.

REFERENCES

- [1] Richard Allmendinger, Julia Handl, and Joshua Knowles. 2015. Multiobjective optimization: When objectives exhibit non-uniform latencies. *European Journal of Operational Research* 243, 2 (2015), 497–513.
- [2] Richard Allmendinger and Joshua Knowles. 2013. ‘Hang on a minute’: Investigations on the effects of delayed objective functions in multiobjective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, Berlin, Heidelberg, 6–20.
- [3] Richard Allmendinger and Joshua Knowles. 2023. 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, Cham, 317–335.

- [4] Julian Blank and Kalyanmoy Deb. 2021. Constrained Bi-objective Surrogate-Assisted Optimization of Problems with Heterogeneous Evaluation Times: Expensive Objectives and Inexpensive Constraints. In *Evolutionary Multi-Criterion Optimization*. Springer International Publishing, Cham, 257–269.
- [5] Julian Blank and Kalyanmoy Deb. 2022. Handling constrained multi-objective optimization problems with heterogeneous evaluation times: proof-of-principle results. *Memetic Computing* 14, 2 (2022), 135–150.
- [6] Julian Blank and Kalyanmoy Deb. 2022. pysamoo: Surrogate-Assisted Multi-Objective Optimization in Python. arXiv:2204.05855 [cs.NE]
- [7] Julian Blank, Kalyanmoy Deb, Yashesh Dhebar, Sunith Bandaru, and Haitham Seada. 2020. Generating Well-Spaced Points on a Unit Simplex for Evolutionary Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation* 25, 1 (2020), 48–60.
- [8] Tinkle Chugh, Yaochu Jin, Kaisa Miettinen, Jussi Hakanen, and Karthik Sindhya. 2018. A Surrogate-Assisted Reference Vector Guided Evolutionary Algorithm for Computationally Expensive Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation* 22, 1 (2018), 129–142.
- [9] Tinkle Chugh, Karthik Sindhya, Jussi Hakanen, and Kaisa Miettinen. 2019. A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft Computing* 23 (2019), 3137–3166.
- [10] Indraneel Das and John E Dennis. 1998. Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal of Optimization* 8, 3 (1998), 631–657.
- [11] Kalyanmoy Deb. 1991. Optimal design of a welded beam via genetic algorithms. *AIAA journal* 29, 11 (1991), 2013–2015.
- [12] Kalyanmoy Deb. 2001. *Multi-objective optimization using evolutionary algorithms*. Wiley, Chichester, UK.
- [13] Kalyanmoy Deb, Rayan Hussein, Proteek Chandan Roy, and Gregorio Toscano-Pulido. 2019. A Taxonomy for Metamodeling Frameworks for Evolutionary Multi-Objective Optimization. *IEEE Transactions on Evolutionary Computation* 23, 1 (2019), 104–116.
- [14] Kalyanmoy Deb and Himanshu Jain. 2014. 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* 18, 4 (2014), 577–601.
- [15] Rayan Hussein and Kalyanmoy Deb. 2016. A Generative Kriging Surrogate Model for Constrained and Unconstrained Multi-objective Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*. Association for Computing Machinery, New York, NY, USA, 573–580.
- [16] Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima. 2015. Modified Distance Calculation in Generational Distance and Inverted Generational Distance. In *Evolutionary Multi-Criterion Optimization*. Springer International Publishing, Cham, 110–125.
- [17] Himanshu Jain and Kalyanmoy Deb. 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-dominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach. *IEEE Transactions on Evolutionary Computation* 18, 4 (2014), 602–622.
- [18] Joshua Knowles. 2006. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation* 10, 1 (2006), 50–66.
- [19] Daniel G Krige. 1951. A statistical approach to some basic mine valuation problems on the Witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy* 52, 6 (1951), 119–139.
- [20] Eric Hans Lee, Valerio Perrone, Cedric Archambeau, and Matthias Seeger. 2020. Cost-aware Bayesian Optimization. arXiv:2003.10870 [cs.LG]
- [21] Søren N Lophaven, Hans Bruun Nielsen, Jacob Sondergaard, and A Dace. 2002. A matlab kriging toolbox.
- [22] Zhongwei Ma and Yong Wang. 2019. Evolutionary Constrained Multiobjective Optimization: Test Suite Construction and Performance Comparisons. *IEEE Transactions on Evolutionary Computation* 23, 6 (2019), 972–986.
- [23] Mohammad Mohiuddin Mamun, Hemant Kumar Singh, and Tapabrata Ray. 2022. An approach for computationally expensive multi-objective optimization problems with independently evaluable objectives. *Swarm & Evolutionary Computation* 75 (2022), 101146.
- [24] Douglas C Montgomery. 2012. *Design and Analysis of Experiments* (8th edition ed.). Wiley, Hoboken, NJ, USA.
- [25] Kamrul Rahi, Hemant Singh, and Tapabrata Ray. 2024. Towards solving expensive optimization problems with heterogeneous constraint costs. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. Association for Computing Machinery, New York, NY, USA, 2032–2040.
- [26] Kamrul Hasan Rahi, Hemant Kumar Singh, and Tapabrata Ray. 2019. Investigating the use of sequencing and infeasibility driven strategies for constrained optimization. In *2019 IEEE congress on evolutionary computation (CEC)*. IEEE, Piscataway, NJ, USA, 1642–1649.
- [27] Kamrul Hasan Rahi, Hemant Kumar Singh, and Tapabrata Ray. 2021. Partial Evaluation Strategies for Expensive Evolutionary Constrained Optimization. *IEEE Transactions on Evolutionary Computation* 25, 6 (2021), 1103–1117.
- [28] Balija Santoshkumar and Kalyanmoy Deb. 2024. Surrogate-Assisted Multi-Objective Optimization for Handling Objectives with Heterogeneous Evaluation Times: Unconstrained Problems. In *2024 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, Piscataway, NJ, USA, 1–8.
- [29] Ryoji Tanabe and Hisao Ishibuchi. 2020. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing* 89 (2020), 106078.
- [30] Qingfu Zhang and Hui Li. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation* 11, 6 (2007), 712–731.
- [31] Qingfu Zhang, Wudong Liu, Edward Tsang, and Botond Virginas. 2010. Expensive Multiobjective Optimization by MOEA/D With Gaussian Process Model. *IEEE Transactions on Evolutionary Computation* 4, 3 (2010), 456–474.