Multi-fidelity Optimisation via Hybrid Genetic-Greedy Search

Mitra Heidari[1,3], Mohammad Golzarijalal[1,3], Ling Luo[1,3], Ellen Otte[2,3], and Uwe Aickelin[1,3]

[1] School of Computing and Information Systems, The University of Melbourne, Parkville, Australia {uwe.aickelin,ling.luo,m.golzarijalal}@unimelb.edu.au mheidari@student.unimelb.edu.au
[2] CSL Innovation, Melbourne, Australia Ellen.Otte@csl.com.au
[3] The ARC Digital Bioprocess Development Hub

Abstract. Efficiently exploring high-dimensional search spaces to identify high-quality solutions is a central challenge in optimisation, particularly when evaluating each candidate is expensive or time-consuming. In domains such as manufacturing and bioprocessing, collecting new experimental data can be costly, while low-fidelity models offer faster but less accurate approximations. To address this, we propose Multi-Fidelity Optimisation via Hybrid Genetic-Greedy Search (MFO-HGS), a novel sequential two-phase framework designed to accelerate convergence while balancing exploration and exploitation under limited high-fidelity evaluation budgets. In the first phase, a Genetic Algorithm explores the search space using low-fidelity evaluations, while the second applies a novel greedy selection mechanism based on the Optimal Computing Budget Allocation principle, which strategically assigns high-fidelity evaluations to the most informative samples. Compared to a baseline hybrid method, MFO-HGS achieves faster convergence and consistently identifies higher-quality solutions on both synthetic benchmark functions and a real bioprocessing case study. A Mann-Whitney-Wilcoxon test further confirms that the performance improvements are significant. These results demonstrate the effectiveness of MFO-HGS for efficient search space optimisation in costly evaluation settings.

Keywords: Multi-Fidelity Optimisation · Optimal Computing Budget Allocation. · Hybrid Genetic-Greedy Search· High-Dimensional Optimisation · Search Space Exploration

1 Introduction

In many industries, particularly manufacturing systems, efficiently identifying strategies that improve system performance is critical because experiments or production runs can be costly and time-consuming [25]. For instance, in bioprocessing, determining the optimal feeding schedule, nutrient concentrations, and other operational parameters can strongly influence final product yields such as

medicine concentration. In high-dimensional optimisation problems, fully evaluating all candidate solutions using real experiments or high-fidelity models can be prohibitively expensive, motivating the development of efficient computational strategies.

Multi-fidelity models provide a practical approach to address this challenge. High-fidelity (HF) models are accurate but computationally expensive, limiting the number of evaluations that can be performed. Low-fidelity (LF) models, in contrast, provide faster approximations that reasonably estimate HF outcomes, albeit with reduced accuracy. Multi-fidelity optimisation combines the strengths of both high- and low-fidelity models to achieve a balance between cost and accuracy, and has been widely applied in science and engineering since the early 2000s [7, 8]. However, not all LF data are beneficial, and combining them with HF data can sometimes worsen optimisation performance [10]. In addition, existing methods often struggle to converge rapidly in high-dimensional search spaces under limited HF evaluation budgets, leaving a gap for efficient, scalable approaches that exploit LF information effectively.

To address these challenges, we propose an innovative Multi-Fidelity Optimisation via Hybrid Genetic-Greedy Search (MFO-HGS) framework, designed to efficiently explore the search space and identify high-quality solutions under a constrained high-fidelity budget. In the first phase, a Genetic Algorithm (GA) explores the search space using LF evaluations to select promising candidates. These LF solutions are then grouped according to their performance ranks, inspired by ordinal optimisation [25]. This ranking-based grouping mitigates biases or scaling inconsistencies in LF models and simplifies the high-dimensional search space, transforming it into an ordered, lower-dimensional representation where promising regions are easier to identify. In the second phase, Optimal Computing Budget Allocation (OCBA) [2, 1, 3] is combined with a greedy search strategy [22] to efficiently allocate HF evaluations and accelerate convergence toward high-quality solutions. This cycle repeats until the total HF budget is fully exhausted, ensuring that the method balances exploration and exploitation while respecting computational constraints.

The main contributions of this research are:

1. Developing a novel sequential two-phase algorithm that efficiently searches the space and selects new samples based on prior LF knowledge, while respecting a limited HF budget.
2. Providing empirical evidence that the proposed algorithm can enhance multi-fidelity optimisation results and converge to optimal or near-optimal solutions rapidly.
3. Applying the proposed algorithm to a real case study on high-dimensional data, demonstrating its practical applicability.

2 Related Work

Multi-fidelity modelling has progressed from early autoregressive approaches for simple fidelity relationships [11, 15] to more advanced frameworks capable of capturing complex, nonlinear dependencies between fidelity levels [19, 4, 5].

Since HF evaluations are computationally expensive and not all LF data contribute positively to optimisation performance, multi-stage optimisation strategies have been developed to allocate computational resources more effectively. These approaches guide HF evaluations toward regions identified as promising by LF analyses. Among them, the ordinal transformation and OCBA techniques have been widely adopted in multi-fidelity optimisation [25, 24]. Ordinal transformation ranks LF solutions based on their relative performance to mitigate LF bias and simplify high-dimensional optimisation, while OCBA allocates HF evaluations across ranked groups to maximise information gain.

Surrogate-assisted approaches such as Kriging and co-Kriging have been employed to model LF-HF relationships and guide HF sampling [12], for example, in a two-stage framework where a Kriging model identifies promising LF regions, followed by a co-Kriging model that integrates LF and HF data for final objective estimation. Despite their effectiveness, surrogate-based optimization methods still face several challenges, including failed simulations and the development of gradient-enhanced and multi-fidelity surrogate models [18], while remaining a key tool for addressing complex engineering optimization problems.

Despite these advancements, efficiently accelerating convergence while maintaining robustness under limited HF evaluation budgets, especially in high-dimensional settings remains a major challenge. To address this, the present study introduces a non-surrogate, hybrid metaheuristic approach that directly explores the search space using multi-fidelity information. The proposed MFO-HGS combines the global search capability of a GA with the adaptive allocation power of OCBA. By leveraging LF information to guide exploration and HF evaluations to refine the search, MFO-HGS enhances scalability, accelerates convergence, and maintains efficiency even when the HF evaluation budget is highly constrained.

3 Methodology

This section presents the methodology of MFO-HGS, designed to efficiently explore the search space and achieve fast convergence toward optimal or near-optimal solutions under limited HF evaluations, while remaining robust in high-dimensional problems. The method consists of two sequential phases. In the first phase, a GA explores the search space using LF evaluations, which are computationally cheaper approximations of the system. The best-performing samples are retained, and the process repeats until a stopping condition, such as a predefined number of generations, is met. In the second phase, retained samples undergo ordinal transformation, where they are ranked and grouped by performance. OCBA determines the number of samples to select from each group,

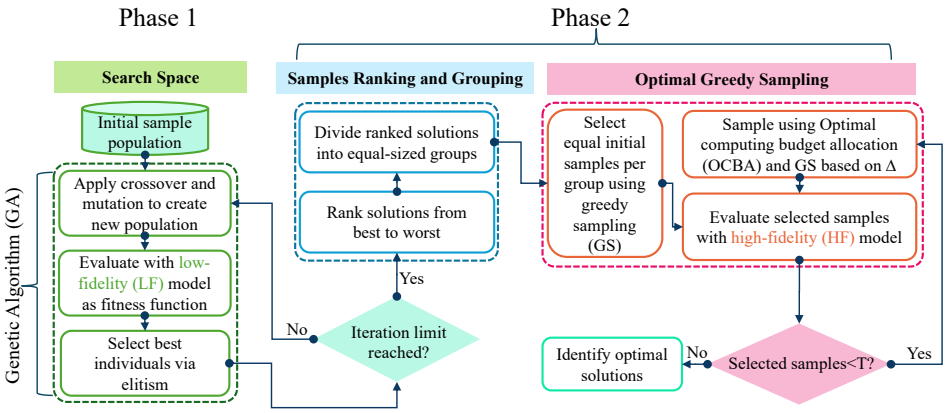

Fig. 1: Schematic diagram of the MFO-HGS algorithm. T denotes the total budget, and Δ represents the HF budget allocated per iteration.

and a greedy sampling procedure chooses candidates for HF evaluation. This cycle continues until the HF budget is exhausted [4]. Greedy search accelerates convergence in promising regions, while OCBA ensures efficient use of HF evaluations. The overall workflow is illustrated in Figure 1, with details of each phase provided in the following subsections.

3.1 Phase 1: Search space

Genetic Algorithms Genetic algorithms (GAs) are search-based optimisation methods and a subset of the broader area of evolutionary computation. In GAs, a population of size P is maintained, where each individual represents a candidate solution. New offspring are generated through crossover, and mutation [14]. For crossover, we employ a single-offspring Blend Crossover (BLX-α) [6]. Two parents are selected randomly from the population of size P, and for each decision variable, the offspring value is sampled uniformly from an extended range between the parent values:

$$x_i^{\text{child}} \sim \text{Uniform}\big[e_{1,i}, e_{2,i}\big], \quad e_{1,i} = x_{1,i} - \alpha(x_{2,i} - x_{1,i}), \quad e_{2,i} = x_{2,i} + \alpha(x_{2,i} - x_{1,i}), \quad i \in \{1, \ldots, D\}. \tag{1}$$

where $x_{1,i}$ and $x_{2,i}$ are the ith decision variables of the two parents, α controls the degree of extrapolation beyond the parent interval, and D is the total number of decision variables. Each offspring variable x_i^{child} is accepted only if it satisfies the variable bounds $x_i^L \leq x_i^{\text{child}} \leq x_i^U$; otherwise, it is resampled until feasibility is ensured. This preserves constraint satisfaction while maintaining population diversity by allowing controlled extrapolation beyond the parent range.

[4] Code for reproducing experiments is available at https://github.com/miheidari/MFO-HGS

For mutation, a uniform mutation operator is optionally applied. For each selected individual, a single decision variable is chosen randomly and replaced with a value sampled uniformly from its allowed range:

$$x_i^{\text{mut}} \sim \text{Uniform}[x_i^L, x_i^U], \quad i \in \{1, \ldots, D\}. \tag{2}$$

This operation introduces additional diversity and follows the real-coded mutation framework described in [6]. Although BLX-α crossover alone often provides sufficient exploration due to its interval-based sampling, mutation helps prevent premature convergence, particularly in small populations or constrained search spaces.

3.2 Phase 2: Optimal greedy sampling

Greedy sampling Suppose we have N samples divided into K groups. The objective function values of these samples, obtained from the previous step, need to be re-evaluated using the HF model. However, due to the limited HF budget, denoted as the total budget T, only a subset of samples can be selected for HF evaluation. At the beginning of the first iteration, an initial number of samples to be selected from each group is specified, denoted as n_{0k}. To select the most informative samples while maximising diversity in both input (feature) and output (response) spaces, we adopt the improved Greedy Sampling (iGS) approach [22]. For each group k, let the candidate sample set be $\mathcal{G}_k = \{(x_i, y_i)\}_{i=1}^{N_k}$, where $x_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$. The response values y_i are obtained from the LF model and are used to guide the sample selection process.

The first sample is chosen as the one closest to the centroid of the group in the feature space:

$$x_{k,1} = \arg\min_{x_i \in \mathcal{G}_k} \|x_i - \bar{x_k}\|_2, \quad \bar{x_k} = \frac{1}{N_k} \sum_{i=1}^{N_k} x_i. \tag{3}$$

Subsequent samples are selected iteratively to maximise diversity. For each candidate x_i not yet selected, the pairwise distances to all previously selected samples are computed in feature space ($d_{i,s}^x = \|x_i - x_s\|_2$) and response space ($d_{i,s}^y = |y_i - y_s|$). The minimum combined distance is defined as:

$$d_{i,k}^{xy} = \min_{x_s \in \mathcal{S}_k} \left(d_{i,s}^x \cdot d_{i,s}^y\right), \tag{4}$$

where $\mathcal{S}_k$ is the set of samples already selected from group k. The candidate with the largest $d_{i,k}^{xy}$ value is then chosen as the next sample:

$$(x_{\text{next},k}, y_{\text{next},k}) = \arg\max_{x_i \in \mathcal{G}_k \setminus \mathcal{S}_k} d_{i,k}^{xy}. \tag{5}$$

Optimal computing budget allocation We use the OCBA method as described in [2, 1, 3] to allocate the limited HF budget efficiently across groups.

Algorithm 1 MFO-HGS Algorithm

Require: A population of size P
Ensure: Identification of optimal or near-optimal samples
for each generation **do**
    Perform crossover and mutation (Eqs. (1)–(2))
    Compute the response variable using the LF model
    Select the best population of size p using elitism and diversity
end for
Sort all input samples generated by the GA in ascending order (for minimisation) or descending order (for maximisation)
Divide samples into K equal groups
Set initial parameters: n_{0k}, Δ, and T
Select samples based on GS (Eqs. (3)–(5))
while the total number of selected samples $< T$ **do**
    Compute μ and σ, and determine the number of new samples to select from each group based on OCBA (Eqs. (6)–(8))
    Select samples based on GS (Eqs. (3)–(5))
    Compute the response variable using the HF model
    Increase the computing budget by Δ
end while
return Optimal or near-optimal samples

Suppose we have N_k samples in group k, and a total HF budget T. The performance of each group is assessed by estimating the mean and variance of the samples selected from that group. The OCBA determines the number of samples to select from each group k to maximise the probability of correct selection. As $T \to \infty$, this probability can be asymptotically maximised when

$$N_j = d_j N_r, \quad j, r \in \{1, 2, \ldots, K\},\ j \neq r \neq b, and \quad d_j = \frac{\sigma_j(\mu_b - \mu_r)}{\sigma_r(\mu_b - \mu_j)} \tag{6}$$

Here μ_b denotes the mean performance of the best group b:

$$\mu_b = \min_{j=1,\ldots,k} \mu_j \quad \text{(for minimisation; use max for maximisation)} \tag{7}$$

According to OCBA, the optimal number of samples for the best group b is:

$$N_b = \sigma_b \sqrt{\left(\sum_{j=1 \neq b}^{k} \left(\frac{N_j}{\sigma_j}\right)^2\right)} \tag{8}$$

This procedure is repeated iteratively until the total HF budget T has been fully allocated, as shown in Algorithm 1. In the algorithm, Δ denotes the incremental HF budget at each iteration.

Table 1: Synthetic functions (Syn) used in the experiments.

Name	Function	D	Bounds
Branin	$f_h(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10$	2	$x_1 \in [-5, 10]$
Ref.[20]	$f_l(x) = f_h - (A_1 + 0.5)(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2$		$x_2 \in [0, 15]$
	here, $A_1 = 0$		
Hartmann-3	$f_h(x) = -\sum_{i=1}^{4} \alpha_i \exp[-\sum_{j=1}^{3} \beta_{ij}(x_j - \gamma_{ij})^2]$	3	$x_j \in [0, 1]$
Ref.[20]	$f_l(x) = -\sum_{i=1}^{4} \alpha_i \exp[-\sum_{j=1}^{3} \beta_{ij}(x_j - 0.75\gamma_{ij}(A2 + 1))^2]$		
	$\alpha = \begin{bmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{bmatrix}$ $\beta = \begin{bmatrix} 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \end{bmatrix}$ $\gamma = \begin{bmatrix} 0.3689 & 0.1170 & 0.2673 \\ 0.4699 & 0.4387 & 0.7470 \\ 0.1091 & 0.8732 & 0.5547 \\ 0.0381 & 0.5743 & 0.8828 \end{bmatrix}$ $A_2 = 0.5$		
F-14-5	$f_h(x) = \sum_{j=1}^{D} \left(0.3 + \sin\left(\frac{16}{15}x_j - 1\right) + \sin\left(\frac{16}{15}x_j - 1\right)^2\right)$	5	$x_j \in [-1, 1]$
Ref.[26]	$f_l(x) = \sum_{j=1}^{D} \left(0.3 + \sin\left(\frac{13}{15}x_j - 1\right) + \sin\left(\frac{13}{15}x_j - 1\right)^2\right)$		
F15-6	$f_h(x) = \sum_{i=1}^{D-1} \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right)$	6	$x_i \in [0, 1]$
Ref.[26]	$f_l(x) = \sum_{i=1}^{D-1} \left(100(x_{i+1} - x_i^2)^2 + 4(x_i - 1)^4\right)$		
Michalewicz	$f_H(x) = -\sum_{i=1}^{D} \sin(x_i)\sin^{20}\left(\frac{i\,x_i^2}{\pi}\right)$	{3, 5, 8}	$x_i \in [0, \pi]$
Ref.[17, 21]	$f_L(x) = \sum_{i=1}^{D} \theta(\phi)\cos(w(\phi)x_i + b(\phi) + \pi)$		
	where, $\theta(\phi) = e^{-0.00025\phi}$, $w(\phi) = 10\pi\theta(\phi)$, $b(\phi) = 0.5\pi\theta(\phi)$		

4 Experimental Setup

4.1 Datasets

Synthetic Functions The analysis is conducted on seven synthetic functions (Syn), Branin, Hartmann, F14, F15, and Michalewicz (evaluated in 3, 5 and 8 dimensions), as presented in Table 1. For all seven functions $f_h : \omega \to \Re$ and $f_l : \omega \to \Re$ represent the HF and LF functions, respectively. The sample space ω typically defined as a hypercube, $\omega = [x_1^l, x_1^u] \times ... \times [x_D^l, x_D^u]$, where $x^l = (x_1^l, ..., x_D^l)^T$ and $x^u = (x_1^u, ..., x_D^u)^T$ represent the lower and upper bounds of ω. The parameter D shows the dimensionality of the synthetic functions. The Latin Hypercube Sampling method is employed to generate the data.

Real Case Study on Bioprocessing Data To further validate the performance of our proposed optimisation method, we consider a real case study in the bioprocessing field (CS-Bio). It should be noted that while all other benchmark functions are minimisation problems, CS-Bio is a maximisation task. Obtaining HF datasets is typically challenging in this domain because it requires conducting costly and time-consuming experiments [8, 13]. Therefore, multi-fidelity models hold promise for reducing the need for extensive experimentation. For both HF and LF models, we use simulated mechanistic models developed on data collected from mammalian cell cultures, specifically Chinese Hamster Ovary (CHO-K1) cells, operated in 250-mL bioreactors for 12–14 days. A total of 1,000 HF data are obtained using the equations and parameters of a mechanistic model as

Table 2: Parameter settings used in experiments

Parameter	Value / Range	Notes
Crossover parameter (α)	0.5	Balance exploitation/exploration
Crossover rate	0.5	Fixed; higher rates costly for complex problems
Mutation rate	0.02	BLX-α ensures exploration
Number of generations (Syn)	5-30	Step of 5; optimal 20
Number of generations (CS-Bio)	5	Computationally expensive
Population size	1000	Fixed
Total budget (T)	400	Fixed
Initial samples (n_{0k})	5-20[2]	Step of 5; selected 20
Incremental budget (Δ)	5-20[2]	Step of 5; fixed at 5
Number of groups (K)	3-5	Step of 1; selected 4

explained elsewhere [9]. An additional 1,000 LF data are obtained using the same mechanistic equations, but with altered cell-growth parameters to intentionally produce biased (erroneous) predictions. For both HF and LF datasets, we considered the initial viable cell density (on day 0) and the increases in the concentrations of 20 amino acids in the bioreactor due to feed additions from days 0 to 13 as features, resulting in a total of 281 features. The concentration of monoclonal antibodies on the final day of the cell culture, representing the high-value biologic product, was used as the response variable.

4.2 Implementations and Parameter Configurations

All parameters used in our experiments are summarised in Table 2. Most parameter values are chosen based on sensitivity analyses on synthetic functions (Syn). For computationally expensive cases (e.g., CS-Bio), certain parameters, such as the number of generations, are reduced to limit computational cost.

In addition to these parameter settings, a Greedy search strategy is used for sample selection, employing Euclidean distance as its key metric. Euclidean distance measures the distance between candidate samples that are not yet selected and those that have already been chosen. It is known to be sensitive to dimensionality, and its effectiveness can decrease as the number of dimensions increases. However, this sensitivity depends on factors such as the number of dimensions and the number of samples, and it can be quantified using the following quality ratio (QR) [23]:

$$QR = \frac{d_{\max} - d_{\min}}{d_{\min}} \tag{9}$$

where $d_{\max}$ and $d_{\min}$ denote the maximum and minimum pairwise Euclidean distances between samples, $i, j \in \mathbb{D}$ and $i \neq j$, with $\mathbb{D}$ representing the dimensionality of the dataset. Since only the CS-Bio dataset has significantly higher dimensionality than the others, we measure the QR specifically for this dataset to determine whether the Euclidean distance would be meaningful. According to

[23], a smaller QR indicates poor distance quality. For the CS-bio dataset, the QR is 33.47, which is considered high; therefore, Euclidean distance is deemed appropriate for the Greedy search. Notably, as the range of each variable in the CS-bio dataset varies considerably, StandardScaler is applied to normalise the features before computing Euclidean distances.

4.3 Baseline Method: Hybrid random-greedy search

To benchmark the proposed MFO-HGS, we compare it with a baseline model, Multi-Fidelity Optimisation via Hybrid Genetic-Random Search (MFO-HRS). The MFO-HRS framework follows the same two-phase structure as MFO-HGS. The only difference lies in the second phase, where candidate samples are selected randomly rather than through the greedy sampling strategy. This setup allows for isolating and evaluating the impact of the greedy selection mechanism on convergence efficiency and optimisation performance. Most multi-fidelity methods rely on costly surrogate models assuming strong LF-HF correlations, making them impractical for high-dimensional or biased systems like the 281D CS-Bio dataset [18]. We instead use sampling-based methods, with MFO-HRS as a baseline to isolate greedy-OCBA effects.

4.4 Evaluation Metrics

Model performance is evaluated by computing the average of the best objective values obtained from the HF model over 30 independent runs, as shown in Eq. 10. In this equation, y_i denotes the best objective value from the i^{th} run, and n represents the total number of independent runs (here, $n = 30$).

$$\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n} \tag{10}$$

To statistically compare the performance of MFO-HGS and MFO-HRS methods, we employ the non-parametric Mann-Whitney-Wilcoxon (MWW) test[16]. This test evaluates whether the observed performance differences between the algorithms are statistically significant without assuming any specific distribution of the data. A result is considered statistically significant if the p-value < 0.05 and the mean difference, $\delta = \bar{y}_1 - \bar{y}_2$, exceeds 10^{-5}, where $\bar{y}_1$ and $\bar{y}_2$ are the average objective values of the two methods being compared. Positive or negative δ indicates which method performs better, while differences not meeting these thresholds are reported as Neutral. For maximisation problems, the interpretation of δ is reversed.

5 Evaluation Performance

The performance of the MFO-HGS algorithm across all synthetic functions and the CS-Bio case study outperforms that of the MFO-HRS algorithm, as shown in Table 3. For example, the global optimum of the Branin function is 0.3979,

Table 3: Comparison of MFO-HGS and MFO-HRS strategies across datasets. The values represent the average of the best objective values over 30 runs, along with ± standard deviation (σ). Bold values indicate better performance between the two methods, larger values for the maximisation problem (CS-Bio) and smaller values for all minimisation problems (all datasets except CS-Bio). The column Global Optimum reports the known global minimum for each dataset; NA indicates that the global optimum is not available for that dataset or dimension. The last three columns report the results of the MWW test.

Datasets	D	Global optimum	MFO -HGS	MFO -HRS	MWW test		
					p-value	delta	Outperforming algorithm
Syn-Branin	2	0.3979	**0.4012** ± 0.0051	0.4214 ± 0.0282	0.0002	-0.0176	MFO-HGS
Syn-Hartmann	3	-3.8628	**-3.8584** ± 0.0029	-3.8376 ± 0.0228	0.0000	-0.0249	MFO-HGS
Syn-F14	5	NA	**0.2721** ± 0.0106	0.2936 ± 0.0295	0.0000	-0.0303	MFO-HGS
Syn-F15	6	0.0000	**2.2304** ± 0.5099	2.8759 ± 0.6670	0.0000	-0.7834	MFO-HGS
Syn-Michalewicz	3	NA	**-2.1840** ± 0.2073	-2.0638 ± 0.2583	0.3671	-0.0623	Neutral
Syn-Michalewicz	5	-4.6870	**-3.0480** ± 0.3432	-2.8689 ± 0.3456	0.1223	-0.1531	Neutral
Syn-Michalewicz	8	NA	**-3.9679** ± 0.4849	-3.5864 ± 0.3064	0.0078	-0.3111	MFO-HGS
CS-Bio	281	–	**0.0500** ± 0.0000	0.0498 ± 0.0001	0.0000	0.0002	MFO-HGS

and for the 5-dimensional Michalewicz function it is -4.6870. Our MFO-HGS algorithm successfully achieves average near-optimal values of 0.4012 and -3.0480, respectively, both outperforming the MFO-HRS algorithm. For the CS-Bio case, a similar trend is observed: the MFO-HGS algorithm achieves an average best value of 0.0500, which is higher than 0.0498 obtained by MFO-HRS.

Applying the MWW test confirms that for almost all functions, MFO-HGS shows a statistically significant improvement over MFO-HRS. Only in two cases, Michalewicz functions with 3 and 5 dimensions, the differences are not statistically significant, indicating comparable performance between the two algorithms.

MFO-HGS also converges faster to near-optimal values than MFO-HRS. For instance, in Syn-F-14, Syn-F-15, and CS-Bio, MFO-HGS converges when the computation budget is approximately 100, whereas MFO-HRS does not reach near-optimal values even after consuming the full budget of 400, as shown in Fig-

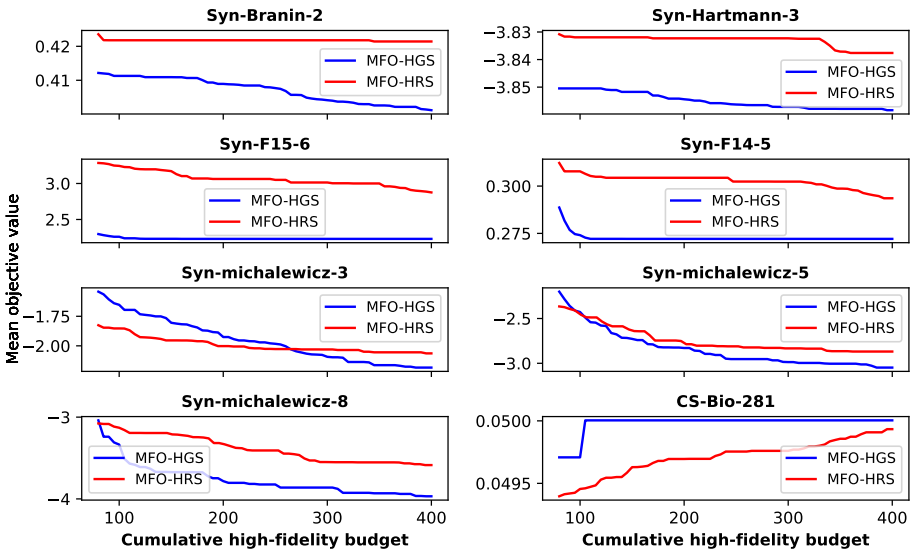

Fig. 2: Convergence of MFO-HGS vs. MFO-HRS: The X-axis shows the computation budget, with a total budget of 400, an initial budget of 20 per group, and a budget of 5 per iteration. The Y-axis shows the average of the best objective values, as evaluated by the HF model, over 30 independent runs. Note that, except for CS-Bio, which is a maximisation problem, all other functions are minimisation problems.

ure 2. For other functions, such as Syn-Hartmann and Syn-Michalewicz, MFO-HGS converges around a budget of 300, while MFO-HRS again fails to approach optimal values within a budget of 400. These analyses demonstrate that MFO-HGS can be highly useful when the cost of collecting HF data is high, as it can save computational resources. Moreover, it remains robust and efficient in high-dimensional problems, maintaining consistent convergence without performance degradation.

6 Conclusions

We propose MFO-HGS, a novel sequential two-phase multi-fidelity optimisation framework that efficiently explores high-dimensional search spaces and achieves fast convergence under a limited HF evaluation budget. In the first phase, a GA explores the search space using LF evaluations to identify high-potential candidates. In the second phase, samples are ordered and grouped via ordinal transformation, and a novel greedy-OCBA selection mechanism is used to assign HF evaluations selectively.

Statistical evaluation using the Mann-Whitney-Wilcoxon test shows that MFO-HGS consistently outperforms the random-based MFO-HRS across synthetic benchmarks and the CS-Bio dataset, achieving faster convergence and

solutions closer to the global optimum. While the Greedy search requires pairwise distance computations, the GA's focused exploration and ordinal transformation preserve ranking consistency, maintaining a high quality ratio (33.47 on CS-Bio) even in high-dimensional spaces. Future work will extend the framework to multi-objective optimisation, where ordinal transformation can project multiple objectives into a single-dimensional ranking for efficient search.

Acknowledgments. This research was supported under the Australian Research Council's Industrial Transformation Research Program (ITRP) funding scheme (project number IH210100051). The ARC Digital Bioprocess Development Hub is a collaboration between The University of Melbourne, University of Technology Sydney, RMIT University, CSL Innovation Pty Ltd, Cytiva (Global Life Science Solutions Australia Pty Ltd) and Patheon Biologics Australia Pty Ltd.

References

1. Chen, C.H., Lee, L.H.: Stochastic simulation optimization: an optimal computing budget allocation, vol. 1. World scientific (2011)
2. Chen, C.H., Lin, J., Yücesan, E., Chick, S.E.: Simulation budget allocation for further enhancing the efficiency of ordinal optimization. Discrete Event Dynamic Systems **10**, 251–270 (2000)
3. Chen, W., Gao, S., Chen, C.H., Shi, L.: An optimal sample allocation strategy for partition-based random search. IEEE Transactions on Automation Science and Engineering **11**(1), 177–186 (2013)
4. Cutajar, K., Pullin, M., Damianou, A., Lawrence, N., González, J.: Deep gaussian processes for multi-fidelity modeling. arXiv preprint arXiv:1903.07320 (2019)
5. Durantin, C., Rouxel, J., Désidéri, J.A., Glière, A.: Multifidelity surrogate modeling based on radial basis functions. Structural and Multidisciplinary Optimization **56**, 1061–1075 (2017)
6. Eshelman, L.J., Schaffer, J.D.: Real-coded genetic algorithms and interval-schemata. In: Foundations of genetic algorithms, vol. 2, pp. 187–202. Elsevier (1993)
7. Fernández-Godino, M.G.: Review of multi-fidelity models. Advances in Computational Science and Engineering **1**(4), 351–400 (2023)
8. Golzarijalal, M., Aickelin, U., Otte, E.: Addressing small data challenges in biopharmaceutical development and manufacturing: A mini review of multi-fidelity techniques. Biotechnology and Bioengineering (2026)
9. Golzarijalal, M., Yatipanthalawa, B.S., Martin, G.J., Gras, S.L., Otte, E., Aickelin, U.: A generalizable modelling framework based on genome-scale dynamic flux balance analysis for cho fed-batch culture and in-silico dataset generation. Available at SSRN 5591701
10. Heidari, M., Luo, L., Golzarijalal, M., Otte, E., Aickelin, U.: Efficient selection of low-fidelity data for multi-fidelity surrogate models. In: Pacific Rim International Conference on Artificial Intelligence. pp. 475–491. Springer (2025)
11. Kennedy, M.C., O'Hagan, A.: Predicting the output from a complex computer code when fast approximations are available. Biometrika **87**(1), 1–13 (2000)

12. Kenny, A., Ray, T., Singh, H.K.: An iterative two-stage multifidelity optimization algorithm for computationally expensive problems. IEEE Transactions on Evolutionary Computation **27**(3), 520–534 (2022)
13. Khuat, T.T., Bassett, R., Otte, E., Grevis-James, A., Gabrys, B.: Applications of machine learning in biopharmaceutical process development and manufacturing: Current trends, challenges, and opportunities. arXiv preprint arXiv:2310.09991 (2023)
14. Lambora, A., Gupta, K., Chopra, K.: Genetic algorithm-a literature review. In: 2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon). pp. 380–384. IEEE (2019)
15. Le Gratiet, L., Garnier, J.: Recursive co-kriging model for design of computer experiments with multiple levels of fidelity. International Journal for Uncertainty Quantification **4**(5) (2014)
16. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. The annals of mathematical statistics pp. 50–60 (1947)
17. Molga, M., Smutnicki, C.: Test functions for optimization needs. Test functions for optimization needs **101**(48), 32 (2005)
18. Palar, P.S., Liem, R.P., Zuhal, L.R., Shimoyama, K.: On the use of surrogate models in engineering design optimization and exploration: The key issues. In: Proceedings of the genetic and evolutionary computation conference companion. pp. 1592–1602 (2019)
19. Perdikaris, P., Raissi, M., Damianou, A., Lawrence, N.D., Karniadakis, G.E.: Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **473**(2198), 20160751 (2017)
20. Toal, D.J.: Some considerations regarding the use of multi-fidelity kriging in the construction of surrogate models. Structural and Multidisciplinary Optimization **51**, 1223–1245 (2015)
21. Wang, H., Jin, Y., Doherty, J.: A generic test suite for evolutionary multifidelity optimization. IEEE Transactions on Evolutionary Computation **22**(6), 836–850 (2017)
22. Wu, D., Lin, C.T., Huang, J.: Active learning for regression using greedy sampling. Information Sciences **474**, 90–105 (2019)
23. Xia, S., Xiong, Z., Luo, Y., Zhang, G., et al.: Effectiveness of the euclidean distance in high dimensional spaces. Optik **126**(24), 5614–5619 (2015)
24. Xu, J., Zhang, S., Huang, E., Chen, C.H., Lee, L.H., Celik, N.: Mo2tos: Multi-fidelity optimization with ordinal transformation and optimal sampling. Asia-Pacific Journal of Operational Research **33**(03), 1650017 (2016)
25. Xu, s., Zhang, s., Huang, s., Chen, s.H., Lee, s.H., Celik, s.: Efficient multi-fidelity simulation optimization. In: Proceedings of the Winter Simulation Conference 2014. pp. 3940–3951. IEEE (2014)
26. Xu, Y., Song, X., Zhang, C.: Hierarchical regression framework for multi-fidelity modeling. Knowledge-Based Systems **212**, 106587 (2021)