Inverse multi-objective robust evolutionary optimization

Dudy Lim, Yew-Soon Ong, Yaochu Jin, Bernhard Sendhoff, Bu-Sung Lee

2006

Preprint:

This is an accepted article published in Genetic Programming and Evolvable Machines. The final authenticated version is available online at: https://doi.org/[DOI not available]

An Adaptive Inverse Multi-Objective Robust Evolutionary Design Optimization

Dudy Lim[1], Yew-Soon Ong[1], Yaochu Jin[2], Bernhard Sendhoff[2] and Bu Sung Lee[1]

[1]School of Computer Engineering, Nanyang Technological University
Nanyang Avenue, Singapore 639798
{dlim,asysong,ebslee}@ntu.edu.sg

[2]Honda Research Institute Europe Gmbh
Carl-Legien-Strasse 30, 63073 Offenbach
{yaochu.jin,bernhard.sendhoff}@honda-ri.de

Abstract.. Many existing works on Evolutionary Algorithm (EA) for handling uncertainty at design parameters have emphasized on introducing some prior structure of the uncertainty or noise. In this paper, we present an evolutionary design optimization that handles the presence of uncertainty with respect to the desired robust performance in mind and avoids making assumptions about the uncertainty structure, which we call an Inverse Multi-Objective Robust Evolutionary (IMORE) design optimization. In the formulation, we model the clustering of uncertain events in families of nested sets using a multi-level optimization searches within the multi-objective evolutionary search. While offering an effective approach to modeling of uncertainty in engineering design, a compelling limitation of the proposed algorithm is the massive computational efforts incurred in the nested evolutionary design searches. To mitigate this problem, we present a study to reduce the large computational cost associated with the algorithm by *1)* adapting the step size parameter and *2)* trim down the number of calls to the objective function in the nested searches. In adapting the step size, we consider both offline and online adaptive strategies in conjunction with the IMORE design algorithm. The incorporation of Design of Experiments (DOE) approaches in the online adaptive IMORE further reduces the number of objective function calls in the online adaptive IMORE algorithm significantly. Empirical study conducted on a number of test functions of diverse complexities demonstrates that the proposed adaptive algorithms are capable of converging to a set of design solutions having non-dominated nominal and robustness performances with significantly enhanced computational efficiency.

1 Introduction

Evolutionary Algorithm (EA) [1] is a modern stochastic optimization technique that has emerged as a prominent contender for global optimization in complex engineer-

ing design. Its popularity lies in the ease of implementation and the ability to arrive close to the global optimum design. Most early studies in the literature on the application of EAs to complex engineering design have mainly emphasized on locating the global optimal design using deterministic computational models. However, in many real-world design problems, uncertainties are often present and practically impossible to avoid. In the case where a solution is very sensitive to small variations either in the design variables or the operating conditions, it may not be desirable to use this design. Hence optimization without taking uncertainty into consideration generally leads to designs that should not be labeled as optimal as they are likely to perform differently when put into practice.

Various classifications of uncertainty in design optimization have been suggested over the recent years [2-8]. In [2], four types of uncertainty were described. They are 1) noise at fitness function, 2) uncertainty at design or environmental parameters, 3) approximation errors, and 4) time varying fitness function. Similar categorization can also be found in [3]. Others [4-5] classify uncertainty as either *aleatory* or *epistemic*. Aleatory uncertainty refers to naturally irreducible variability, e.g. quantities that are inherently variable over time and space. In contrast, epistemic uncertainty is caused by incomplete knowledge about the designs to be optimized and should be reducible if greater knowledge can be acquired. In [6-8], uncertainty is defined as the gap between the known and unknown facts. In this paper, we follow the categorization of uncertainty in [2] and [3]. In particular, we focus on uncertainty in the design and/or environmental parameters.

To date, many approaches exist for coping with uncertainty in complex engineering design optimization. These include the One-at-a-Time Experiments, Taguchi Orthogonal Arrays, bounds-based, fuzzy and probabilistic methods [9]. In the context of EA, a number of prominent new studies on handling the presence of uncertainty in engineering designs have emerged recently. In [10], a Genetic Algorithm with Robust Searching Scheme (GA/RS3) was introduced. In this work, a probabilistic noise vector is added to the genotype before fitness evaluation. In biological terms, this means that part of the phenotypic features of an individual is determined by the decoding process of the genotypic code of genes in the chromosomes. The study of an (1+1)-Evolutionary Strategy (ES) with isotropic normal mutations using the noisy phenotype scheme has also been reported in [11]. An evolutionary algorithm based on max-min optimization strategy using a Baldwinian trust-region framework that employs surrogate models was also recently proposed in [3] for robust design. In [12], a multi-objective approach to handling uncertainty in EA by considering the trade-off between robustness and nominal performance of a solution was discussed. These optimization methods have been successfully applied to mechanical and aerodynamic problems, including the 2D aerodynamic airfoil [3,13], lightweight space structures [14] or multilayer optical coating design [15].

In most of these schemes, often some prior knowledge about the structure of the uncertainties, such as the distribution property involved is assumed to be available. However, it is worth noting that the quality of the solution is generally attainable only when the assumptions made on the structure of the uncertainty reflect the actual uncertainty flawlessly. In this paper, instead of using sensitivity analysis, i.e., analyzing the changes in performance of a design with respect to variability in the key design

parameters, we present an evolutionary design optimization that handles the presence of uncertainty in view of the desired robust performance, which we call the Inverse Multi-Objective Robust Evolutionary (IMORE) design. In contrast to conventional robust optimization, the proposed approach avoids making assumptions about the uncertainty structure in the formulation of the optimization search process. Making assumptions about the uncertainty that are not backed up by strong evidence in evolutionary design optimization can possibly lead to erroneous designs that could have catastrophic consequences. However, the drawback of the methodology is the massive computational efforts incurred in the nested evolutionary design searches involved. Two key factors that determine the number of objective function evaluations conducted to establish the robustness of a design is the step size to define the search bounds and the number of objective function calls used for each nested search. Hence in this paper, we also present a study to reduce the large computational cost associated with the methodology by adapting the step size and trim down the number of calls to the objective function in the nested searches. In adapting the step size parameter, we consider both offline and online adaptive strategies. To reduce the number of objective function calls in a single nested search, we also consider further efficiency enhancement to the online adaptive IMORE algorithm by utilizing the Design of Experiments (DOE) sampling methods.

The rest of this paper is organized as follows. In Section 2, we provide an overview of robust evolutionary design optimization and the IMORE design methodology. Section 3 introduces the adaptive strategies to reduce the generally large computational cost associated with the proposed algorithm. To illustrate the effectiveness of the adaptive strategies, Section 4 provides an empirical study on a series of test functions of diverse complexities and introduces further enhancements to accelerate the adaptive IMORE algorithms by incorporating the DOE sampling methods. Finally, Section 5 concludes this paper.

2 Robust Evolutionary Optimization in the Presence of Uncertainty

In this section, we present a brief overview on some fundamentals of robust evolutionary design optimization in the presence of uncertainties. In particular, we consider the general bound constrained nonlinear programming problem of the form:

$$\begin{aligned} &\textbf{Maximize :} && f(x) \\ &\textbf{Subject to :} && x_l \le x \le x_u \end{aligned} \tag{1}$$

where $f(x)$ is a scalar-valued objective function, $x \in \Re^d$ is the vector of design variables, while x_l and x_u are vectors of lower and upper bounds for the design variables.

Here, our focus is on EAs for robust engineering design optimization under uncertainties that arise in:

i) design parameter x

$$f'(x) = f(x + \delta) \tag{2}$$

where $\delta = (\delta_1, \delta_2, ..., \delta_k)$, is the noise in the design parameter and *f'(x)* is the perturbed fitness of design vector *x*.

ii) operating/environmental conditions

$$f'(x) = f(x, c + \zeta) \tag{3}$$

where $c = (c_1, c_2, \ldots, c_n)$, is the nominal value of the environmental parameters and ξ is a random vector used to model the variability in the operating conditions. Both forms of uncertainties may be treated equivalently [12].

The core mechanism in many of the evolutionary techniques for handling uncertainty has very much relied on probability theory, assuming prior knowledge about the structure of the uncertainty. For example, the uncertainties, δ and/or ξ, are often assumed to have a Gaussian (normal), Cauchy, or uniform noise distribution. More often, a Gaussian noise with zero mean and variance σ^2, $N(0, \sigma^2)$ is considered, by virtue of the central limit theorem. The effective fitness *F(x)* can then be defined as:

$$F(x) = \int_{-\infty}^{\infty} f(x+\delta)\Phi(\delta)d\delta \tag{4}$$

where $\Phi(\delta)$ is the probability distribution of δ. In practice, *F(x)* is often approximated by $\hat{F}(x)$ using Monte Carlo Simulation (MCS) assuming *m* samples of noise term δ as follows:

$$\hat{F}(x) = \frac{1}{m}\sum_{i=1}^{m}\left(f(x) + \delta_i\right) \tag{5}$$

To locate a robust design solution in the presence of uncertainties in the design vector, one may consider using the GA/RS3 proposed in [10] which is outlined in Figure 1.

BEGIN EA (for maximization problem)
- Generate a population of designs
- **While**(termination condition is not satisfied)
  - **For**(each individual *i* in the population)
    - **For**(*j=1* to *m*)
      - Perturb individual to arrive at x_{ij}
      - Evaluate $f'(x_{ij}) = f(x_{ij} + \delta_j)$
    - **end For**
    - Determine effective fitness, $F(x_i)$ of individual i

$$\hat{F}(x_i) = \frac{1}{m}\sum_{j=1}^{m} f'(x_{ij})$$

  - **end For**
  - apply mutation and crossover to create new population
  - perform selection of individuals.
- **End while**

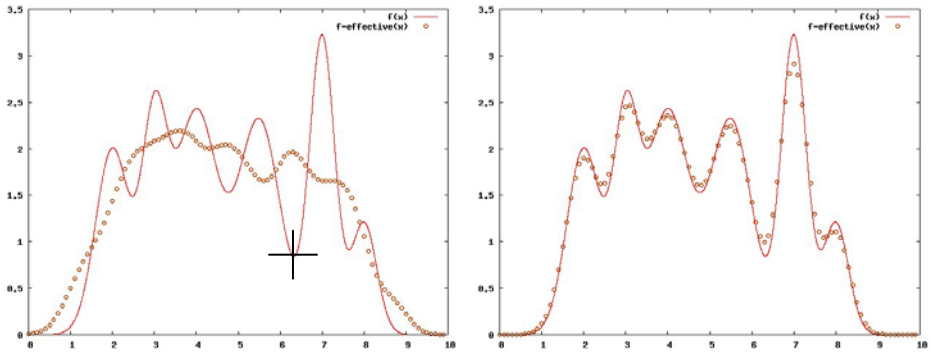
END EA

Figure 1. Genetic Algorithm with Robust Solution Searching Scheme (GA/RS3)

Consider the one-dimensional function depicted in Figure 2 and defined by

$$f(x) = 2e^{-(x-2)^2/0.32} + 2.2e^{-(x-3)^2/0.18} + 2.4e^{-(x-4)^2/0.5} + 2.3e^{-(x-5.5)^2/0.5} + 3.2e^{-(x-7)^2/0.18} + 1.2e^{-(x-8)^2/0.18}, \quad -1 \le x \le 10 \tag{6}$$

This represents a multimodal function with a nominal global optimum located at sharp peak $x^* \in [6.5,7.8]$ and has many other local optima located elsewhere[1]. The robust solution that the noisy phenotype scheme in Figure 1 converges to depends on the perturbation assumed on δ, *i.e.*, the assumption on the structure of the uncertainty, δ. By making assumption on the distribution of δ in $f(x)$, one may easily derive the respective effective fitness function, $F(x_i)$. For instance, Figure 2(a) and 2(b) illustrate the effective fitness functions for the one-dimensional function defined in equation (5) assuming a uniform distribution for δ with σ set to ± 1.0 and ± 0.25, respectively. Note that σ defines the range or bound of the uncertainty δ assumed or known beforehand. When the range for σ is configured to be ± 1.0, the robust global optimum[2] may be easily found to be located in the region $x^\wedge \in [3.0,4.0]$. On the other hand, if σ is configured to be ± 0.25, the global robust optimum approaches that of the nominal fitness function $f(x)$.



(a) Range of the uncertainty, σ= ± 1.00 **(b) Range of the uncertainty, σ= ± 0.25**

Figure 2. Effective fitness F(x) of the function defined in eq. (6) assuming a uniform distribution for δ

2.1 Inverse Multi-Objective Robust Evolutionary (IMORE) Design Optimization

In most cases, the algorithm described in Figure 1 has been shown capable of converging appropriately to the robust design solution defined by the effective fitness as long as the assumption made about the uncertainty, δ, including the type of distribution and the respective range or deviation, are known precisely. In contrast, it is often the case that in many realistic problems that very little knowledge about the structure

[1] Note that x^* represents the nominal global optimum.
[2] Note that $x^\wedge$ represents the robust global optimum.

of the uncertainty involved is available *a prior*. Besides, a major problem with many existing robust schemes in the literature is that the nominal fitness of the final design is often neglected [10, 14]. These schemes generally optimize the robustness of the final design, at the expense of nominal performance of the final design. For instance, it may be observed from Figure 2(a) that the design point x at 6.3 (marked as '+') possesses very good robustness, *i.e.*, a high effective fitness of around 1.4, but have a very poor nominal fitness of only 0.84. This implies that it is crucial to consider both the nominal performance and robustness in the design optimization search. A more promising solution to handle the trade-off between the robustness and nominal fitness is to consider a multi-objective optimization approach [12].

To mitigate the problems identified above, we present in this section an Inverse Multi-Objective Robust Evolutionary (IMORE) design optimization strategy for locating designs with non-dominated nominal performances and robustness in the presence of uncertainties. In many real world engineering design problems [12, 14-16], it is often the case that very little knowledge about the structure of the uncertainty involved is available. Hence, instead of focusing on making any probably unjustifiable mathematical model out of the uncertainty, we focus here on how a design may deteriorate in the presence of uncertainties.

The basic steps of the proposed algorithm are outlined in Figure 3. In the first step, the maximum degradation tolerable for the final design, d_t and step size Δ used to conduct nested searches are defined and initialized. Within the initialization phase, a population of designs is also created either randomly or using design of experiments techniques such as Latin hypercube sampling or minimum discrepancy sequences [19]. Each individual in the population is first evaluated to determine its nominal fitness. Subsequently, each individual then undergoes a sequence of nested searches across a family of nested search regions parameterized by the uncertainty vector in the spirit of Info-Gap theory [6-8]. The aim of the nested searches is to determine the maximum robustness that a particular design can be guaranteed to handle under the permitted maximum performance degradation of d_t defined. More specifically, during the inner search for each chromosome in an IMORE generation, we solve a sequence of bound constrained optimization sub-problems of the form:

$$\begin{aligned} &\textbf{Maximize :} \quad d(x) = f(x_i) - f^k(x) \\ &\textbf{Subject to :} \quad x_l^k \le x \le x_u^k \end{aligned} \tag{7}$$

where x_l^k and x_u^k are the appropriate bounds on the design variables, which is updated at each k iteration based on the step size defined, Δ.

For each optimization sub-problem (or during each k iteration), the optimal solution of the k^{th} sub-problem is sought. The objective of each sub-problem search is to find the worst-case fitness function value by solving a bound constrained maximization problem. After each iteration, the design variables search bounds, x_l^k and x_u^k are updated using the step size Δ which is given by

$$x_l^k = x_i - k\Delta\,, \quad x_u^k = x_i + k\Delta \tag{8}$$

BEGIN *IMORE* (Consider a maximizing problem)

Initialization Phase:

- Initialize Maximum degradation tolerable for the final design, d_t
- Initialize the step size Δ for nested search
- Generate a population of design vectors

Search Phase:

While (termination condition is not satisfied)

For (each individual *i* in the population)

- Objective-1 (nominal fitness) = $f(x_i)$
- Objective-2 (robustness) = δ^i_{max}
- **Repeat**
  - **Maximize:** $d(x) = f(x_i) - f^k(x)$

    subject to: $x_l^k \le x \le x_u^k$ where $x_l^k = x_i - k\Delta$, $x_u^k = x_i + k\Delta$
  - Obtain x_{opt}^k and $d\left(x_{opt}^k\right)$
  - Store $d\left(x_{opt}^k\right)$ and associate it with $k\Delta$

  until $\left\{d^k(x) = f(x_i) - f(x_{opt}^k)\right\} > d_t$
- Estimate maximum uncertainty δ^i_{max} using linear interpolation from $d\left(x_{opt}^k\right)$ for different $k\Delta$

end For

- Apply standard MOEA operators to create a new population

end While

END *IMORE*

Figure 3. IMORE design optimization algorithm.

It is worth noting that by conducting a sequence of nested searches across a family of ascending nested bounds parameterized by the uncertainty vector, we arrive at a monotonic increasing function of performance degradation versus uncertainty as illustrated in Figure 4 such that

$$x_l^{k+1} \le x_l^k, x_u^k \le x_u^{k+1} \rightarrow d\left(x_{opt}^k\right) \le d\left(x_{opt}^{k+1}\right) \tag{9}$$

where x_{opt}^k denotes the optimum at the k^{th} iteration and $d\left(x_{opt}^k\right) = f(x_i) - f\left(x_{opt}^k\right)$ is the corresponding maximum performance degradation obtained for $x_l^k \le x \le x_u^k$.

In addition, the $d\left(x_{opt}^k\right)$ found and associated $k\Delta$ for each search iteration are then stored to create a database of uncertainties and corresponding performance degradations. For example, consider a design point with x_i=4 in Figure 4, labeled points A, B and C correspond to $(x_{opt}^k, f(x_{opt}^k))$ for k=1, 2 and 3 respectively and Δ set to 1. For each chromosome, the iterative searches are terminated when the optimal solution of the k^{th} sub-problem exceeds the maximum degradation defined, i.e.

$$\left\{d^k(x) = f(x_i) - f(x_{opt}^k)\right\} > d_t \tag{10}$$

At the end of the sequences of searches for a chromosome, the maximum uncertainty δ_{max} that a design may handle given a maximum performance degradation of d_t permitted can be determined by interpolating from the database of previous uncertainties and maximum performance degradations, *i.e.*, $k\Delta$ and $d^k(x)$. This is also illustrated in Figure 4 where D represents the point where a maximum performance degradation of d_t is reached and δ_{max} is the corresponding maximum uncertainty that the design guarantees to handle. The IMORE search then proceeds with the multi-objective evolutionary operators to create a new population and stops upon termination condition.

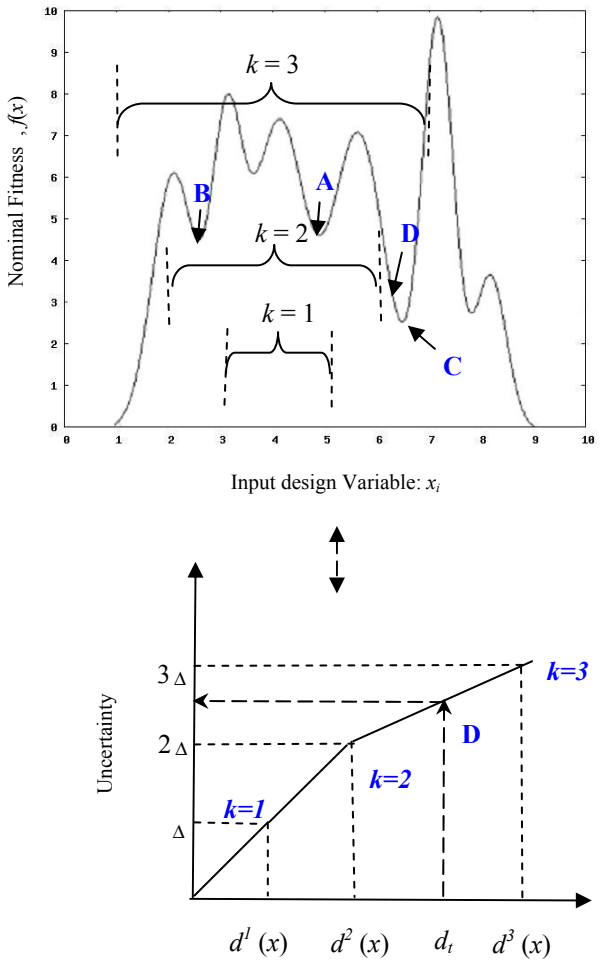


Figure 4. Steps of IMORE for x_i=4 and Δ =1.

3 Adaptive Inverse Multi-Objective Robust Evolutionary Design Optimization

In this section, we present a study on the computational complexity of the proposed IMORE algorithm and subsequently introducing possible strategies to achieve better efficiency with minimum effect on the performance of the algorithm. From the previous section, it is observable that the computational complexity for the algorithm is *O(gnkl)* where *g* is the maximum number of IMORE generations, *n* is the number of chromosomes or individuals, *k* is the average number of nested search iterations required by an individual to reach δ_{max} and *l* is the average number of function evaluations incurred in a nested search. The computational cost to locate the pareto optimum solutions can sometimes be significantly large, especially if the objective function is computationally expensive.

For every individual, δ_{max}^{approx} represents the approximated robustness fitness by IMORE, while δ_{max}^{exact} is a more accurate robustness fitness obtained from an exhaustive search. The Average Approximated Robustness (AAR) and Average Exact Robustness (AER) for an EA population become

$$\text{AAR} = \frac{1}{n}\sum_{i=1}^{n}\frac{\delta_{max\_i}^{approx}}{x_u - x_l}\times 100\% \text{ , AER} = \frac{1}{n}\sum_{i=1}^{n}\frac{\delta_{max\_i}^{exact}}{x_u - x_l}\times 100\% \tag{11}$$

where *n* is the population size, x_u and x_l are the upper and lower bound of the search space, respectively.

Besides the standard evolutionary parameters, IMORE has an additional control parameter Δ, which is inversely proportional to *k*, i.e. $k \propto \frac{1}{\Delta}$. Here, we illustrate the effect of varying Δ and *k* in the IMORE algorithm when searching for pareto-optimum solutions using the function defined in eq. (12).

$$f(x) = \sum_{i=1}^{10}\left(\sin(x)\sin^{10}\left(\frac{ix^2}{\pi}\right)\right) \text{ , } -1.5 \le x \le 3,\ d_t = 1.0 \tag{12}$$

The AAR and AER of a population for different step size Δ and hence *k* in IMORE are tabulated in Table 1. From these results, it is worth noting that the average error in the robustness accuracy across a typical population, i.e., $|\text{AAR-AER}|/\text{AER}$, varies greatly with Δ and *k*. It can be observed that the average error in robustness accuracy increases with increasing Δ but incurs a lower computational cost due to a smaller *k*. This makes good sense since a larger step size generally translates to greater interpolation errors. However, this inferiority in accuracy could also lead IMORE search convergence to false pareto-optimum solutions. On the other hand, fine step size provides a lower average error but at the expense of computational cost. Since the number of iterations, *k*, is inversely proportional to the step size, an intuitive way to reduce the search time of evolutionary optimization algorithm is to adapt a suitable balance of *k* and Δ throughout the IMORE search.

Table 1. Average Approximated Robustness (AAR) and Average Exact Robustness (AER) of an IMORE population for different step size Δ when applied on the test function in eq. (12).

Step Size Δ (%)	Average number of iterations *k*	Average Approximated Robustness AAR (%)	Average Exact Robustness AER (%)	|AAR-AER|/AER (%)
1	6	5.59	5.60	0.2
3	3	7.16	6.95	3.0
5	2	5.83	4.91	18.8
10	1	9.33	6.13	52.2

In the next subsections, we introduce the offline and online adaptive IMORE for robust design in the presence of uncertainty.

3.1 Offline Adaptive IMORE Design Optimization

In this sub-section, we present an offline adaptive IMORE optimization algorithm for robust design in the presence of uncertainty. The basic steps of the proposed adaptive algorithm are outlined in Figure 5.

In the initialization phase of the offline adaptive IMORE search, *s* levels of step sizes are defined from Δ_1, Δ_2, …, Δ_s. Since high robustness accuracy may not be exceedingly crucial during the exploration stage of the IMORE search, we consider using finer step sizes with increasing search generations, i.e., $\Delta_1 > \Delta_2 > ... > \Delta_s$.

This indicates that the number of function calls, which is proportional to *k*, can be reduced for the initial stage of the search. In particular, we use different values of step sizes that decrements every *t* generations and is defined by *t=n/s* where *n* is the number of IMORE search generations before search termination. Then for each individual during the search phase, a series of nested searches are conducted by solving a sequence of bound constrained optimization sub-problems described by eq. (7). The appropriate bounds of x_l^k and x_u^k for each nested search are defined by Δ_y, where $y = \left\lceil \frac{g}{t} \right\rceil$ and *g* is the current generation count. After which, the offline adaptive IMORE search operates exactly like the IMORE search (as described in Figure 3) and stops when the termination criterion is reached.

BEGIN ***OFFLINE ADAPTIVE IMORE*** (Consider a maximizing problem)

Initialization Phase:

- Initialize Maximum degradation tolerable for the final design, d_t.
- Initialize s levels of step sizes to use, i.e., Δ_1, Δ_2, …, Δ_s where $\Delta_1 > \Delta_2 > ... > \Delta_s$.
- Initialize step size update interval, $t=n/s$, where n is the maximum number of IMORE search generations before termination.
- Generate a population of design vectors.

Search Phase:

While (termination condition is not satisfied)

For (each non-duplicated individual i in the population at current generation, g)

- Objective-1 (nominal fitness) = $f(x_i)$
- $\Delta = \Delta_y$ where $y = \left\lceil \frac{g}{t} \right\rceil$
- Objective-2 (robustness) = δ_{max}^i obtained using Δ and the procedure described in Figure 3.

end For

- Apply standard MOEA operators to create a new population

end While

END ***OFFLINE ADAPTIVE IMORE***

Figure 5. Offline adaptive IMORE design optimization algorithm.

3.2 Online Adaptive IMORE Design Optimization

Next, we present an alternative to adapt suitable balance of k and Δ throughout the IMORE search based on an online adaptive strategy. In contrast to the offline adaptive strategy, which fixed the step sizes to be used across the various phases of the IMORE search in advance, the online adaptive strategy decides on suitable values of k and Δ using online feedback on the accuracy of the approximated robustness fitness obtained during the search. The detailed procedure of the online adaptive IMORE is described in Figure 6.

In the online adaptive IMORE algorithm studied here, we consider a straightforward toggling between two different step sizes, particularly, a fine and coarse step size value which are denoted as Δ_f and Δ_c, respectively. To start with, Δ_f, Δ_c, and the update interval, t are initialized. The IMORE search then begins with a coarse step size, Δ_c. Subsequently, the robustness fitness error of a population is assessed for every t generations in the IMORE search. The robustness fitness error for the i^{th} individual, denoted by ρ_i, is then determined by

$$\rho_i = \frac{\left|\delta_{max-1}^i - \delta_{max-2}^i\right|}{\delta_{max-1}^i} \times 100\% \tag{13}$$

where δ^i_{max-1} and δ^i_{max-2} are the robustness fitness measurements obtained by Δ_f and Δ_c, respectively. If a large robustness fitness error is found, i.e. most of the population having $\rho_i \geq 20\%$, the fine step size Δ_f will be used for the next t generation to gain better accuracy. On the other hand, if most individuals in the population having $\rho_i < 20\%$, a coarse step size Δ_c can be applied for the next t generation since the accuracy of the robustness fitness is considered sufficient. Here, we empirically decide 75% of the population size and 20% of δ^i_{max-1} to represent the whole population and the cut off value to classify an error as low, respectively.

BEGIN ***ONLINE ADAPTIVE IMORE*** (Consider a maximizing problem)

Initialization Phase:

- Initialize Maximum degradation tolerable for the final design, d_t.
- Initialize the fine and coarse step size, Δ_f and Δ_c.
- Set $\Delta = \Delta_c$.
- Generate a population of design vectors.

Search Phase:

While (termination condition is not satisfied)

- Set $\rho = 0$

For (each non-duplicated individual *i* in the population at generation *g*)

- Objective-1 (nominal fitness) = $f(x_i)$
- Objective-2 (robustness) = δ^i_{max} obtained using Δ and the procedure described in Figure 3.
- **If (**mod(*g*,*t)* == 0**)**
  - Obtain δ^i_{max-1} and δ^i_{max-2} for $\Delta = \Delta_f$ and Δ_c, respectively.
  - Obtain $\rho_i = \dfrac{\left|\delta^i_{max-1} - \delta^i_{max-2}\right|}{\delta^i_{max-1}} \times 100\%$
  - $\rho = \rho + \rho_i$

  end If

end for

- **If** (mod(*g*,*t)* == 0)
  - **If ((**Δ == Δ_c)&(75% of population having $\rho_i \geq 20\%$)), **then** $\Delta = \Delta_f$
  - **Else if ((**Δ == Δ_f)&(75% of population having $\rho_i < 20\%$)), **then** $\Delta = \Delta_c$
  - **Else** Δ is left unchanged

  End If

End While

END ***ONLINE ADAPTIVE IMORE***

Figure 6. Online adaptive IMORE design optimization algorithm.

4 Empirical Studies

To facilitate a detailed study of the IMORE algorithms, a number of test functions are created using an expansion in terms of Gaussian basis functions as follows:

$$f(X) = \sum_{i=1}^{m} \left(\beta_i \prod_{j=1}^{d} e^{\frac{-(x_j - \mu_{ij})^2}{2\sigma_i^2}} \right) \tag{14}$$

where $X=(x_1,x_2,\ldots,x_d)$ is the design vector, d is the dimension of the function, m is the number of basis functions, σ_i and β_i are the standard deviation and magnitude of the i^{th} basis function, and μ_{ij} is the centroid of the j^{th} dimension at the i^{th} basis function. The parameters of the test functions constructed are as listed in Table 2.

Table 2. Parameters used to construct the test functions based on Gaussian basis function as in eq. (14)

Test Function	Centroid $[\mu_1, \mu_2,\ldots, \mu_m]$	Standard Deviation $[\sigma_1, \sigma_2,\ldots, \sigma_m]$	Magnitude $[\beta_1, \beta_1,\ldots, \beta_m]$	Dimensionality
G1 $0\leq X\leq 13$, $d_t = 1.0$	[1, 1.25, 1.5, 1.6, 1.8, 2.2, 2.4, 2.75, 3, 6, 7, 8, 9.5, 11, 12]	[0.5, 0.15, 0.08, 0.05, 0.1, 0.1, 0.05, 0.15, 0.5, 0.4, 0.3, 0.5, 0.5, 0.3, 0.3]	[1, 2, 0.5, 1, 2.5, 2.5, 2, 2, 1, 2, 2.2, 2.4, 2.3, 3.2, 1.2]	1
G2 $0\leq X\leq 10$, $d_t = 0.5$	[(1,1), (1,3), (3,1), (3,4), (5,2)]	[0.6, 0.2, 1, 0.8, 0.6]	[0.7, 0.75, 1, 1.2, 1]	2
G5 $0\leq X\leq 10$, $d_t = 0.5$	[(4,1,6,7,8), (1,3,8,9.5,2), (8,8,2,2,5), (6,4,1.3,5,5), (5,2,9,7,8), (9,2,9,3,4.6), (6.9,3,2,8,7), (3,5,5,2,4), (4,3,5,7,3), (9,8,0.6,3,8)]	[0,3, 0.4, 1.0, 0.4, 0.6, 0.5, 0.1, 1.0, 0.2, 0.3]	[0.7, 0.75, 1, 1.2, 1, 0.6, 0.5, 0.2, 0.4, 0.1]	5
G10 $0\leq X\leq 10$, $d_t = 0.5$	[(4,1,6,7,8,3,1,1,5,6), (1,3,8,9.5,2,1,5,2,8,4), (8,8,2,2,5,3,6,4,3,5,5), (6,4,1.3,5,5,3,4,8,4,2), (5,2,9,7,8,5,2,7,4,3), (9,2,9,3,4.6,2,6,8,8,0), (6.9,3,2,8,7,5,2,7,7,3), (3,5,5,2,4,7,7,2.3,5,10), (4,3,5,7,3,3,1,2,5,2), (9,8,0.6,3,8,7,8,9,3,4,6)]	[0,3, 0.4, 1.0, 0.4, 0.6, 0.5, 0.1, 1.0, 0.2, 0.3]	[0.7, 0.75, 1, 1.2, 1, 0.6, 0.5, 0.2, 0.4, 0.1]	10

In the numerical studies, we employ a 16-bit binary coded Non-dominated Sorting Genetic Algorithm, NSGA-II [16]. Both the population size and maximum search generation permissible are configured as 100. Uniform crossover and mutation are applied at probabilities of 0.9 and 0.1, respectively. The offline adaptive IMORE is configured with four grades of step sizes having Δ_1 =10%, Δ_2 =5%, Δ_3 =3%, and Δ_4 =1%. The parameters of the online adaptive IMORE are configured as follows: Δ_f =1%, Δ_c =5%, t=20 generations. Each iteration of nested search is set with the maximum computational budget of 400 objective function calls for low dimensional problems (G1, G2, G5) and 2000 objective function calls for higher dimensional one (G10).

Further, the pareto front convergence metric (*Pc*) reported in [17] is introduced to measure the ability of the adaptive IMORE algorithms in converging to the true optimum pareto-front. This is one of the well-known metrics to evaluate the convergence towards a reference set of non-dominated solutions [18]. To determine *Pc*, a target pareto front *P** is used as the reference. For each solution *i* in the pareto front *F*, the shortest Euclidean distance d_i to *P** is sought as follows:

$$d_i = \min_{j=1}^{|P^*|} \sqrt{\sum_{k=1}^{m} \left(\frac{f^k(i) - f^k(j)}{f_u^k - f_l^k} \right)^2} \tag{15}$$

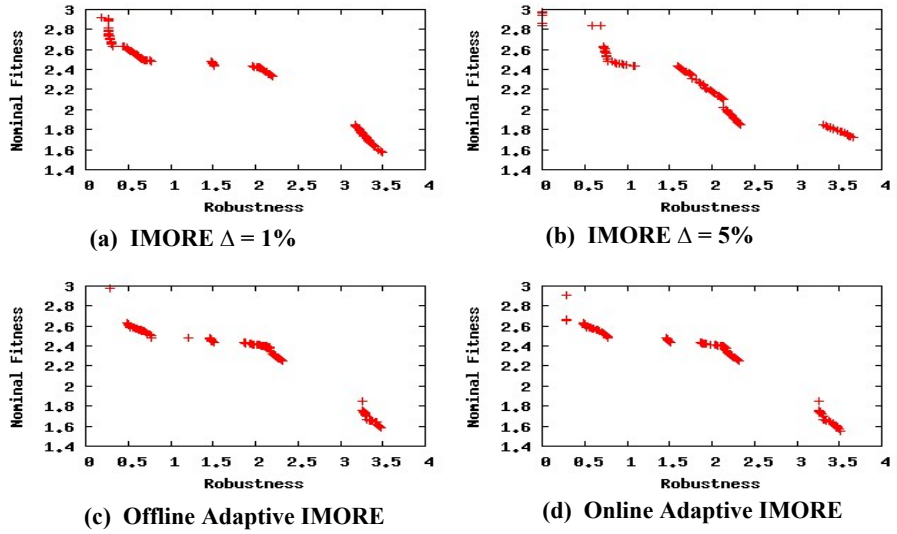
where m is the number of objectives, f_u^k and f_l^k are the upper and lower bounds of the k^{th} objective, respectively. The pareto front convergence metric, *Pc* is then calculated as the average of d_i for all design solutions in the final pareto front *F* as in eq. (16).

$$Pc(F) = \frac{\sum_{i=1}^{|F|} d_i}{|F|} \tag{16}$$

Note that a smaller *Pc* indicates a greater accuracy of convergence to the true target pareto front. Here, the target pareto front, *P** is obtained based on the IMORE with Δ =1%.

4.1 Comparative Study on IMORE and Adaptive IMORE

In this subsection, we provide an empirical study on the IMORE and adaptive IMORE algorithms on each of the abovementioned test functions. The typical pareto fronts for G1, G2, G5, and G10 using the IMORE with Δ =1% is depicted in Figures 7-10 (a) respectively. Also plotted in 7-10 (c) and (d) are the final pareto fronts for both the offline and online adaptive IMORE. It can be observed that both offline and online adaptive IMORE algorithms converge to similar pareto fronts to that obtained by the IMORE using Δ =1%. It is worth noting that the results obtained on IMORE with Δ =5% (from Figure 7-10 (b)) are also presented as references of false convergences.



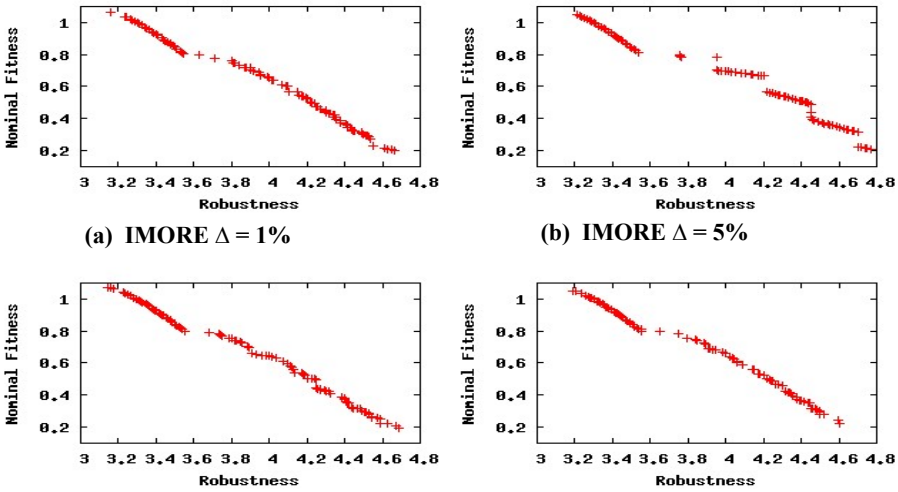
(a) IMORE Δ = 1%

(b) IMORE Δ = 5%

(c) Offline Adaptive IMORE

(d) Online Adaptive IMORE

Figure 7. Pareto fronts of test function G1 at the end of the 100th generation using IMORE for Δ =1%, Δ =5%, offline adaptive, and online adaptive IMORE



(a) IMORE Δ = 1%

(b) IMORE Δ = 5%

(c) Offline Adaptive IMORE

(d) Online Adaptive IMORE

Figure 8. Pareto fronts of test function G2 at the end of the 100th generation using IMORE for Δ =1%, Δ =5%, offline adaptive, and online adaptive IMORE

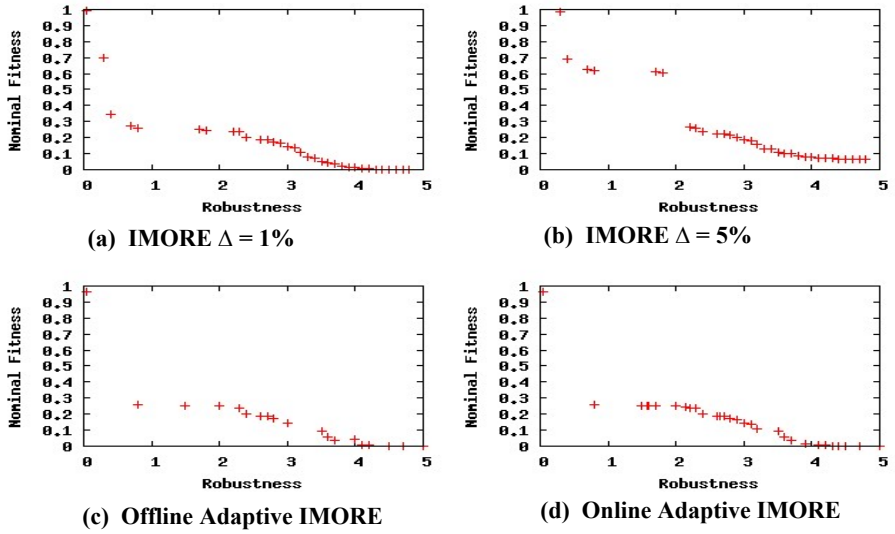


Figure 9. Pareto fronts of test function G5 at the end of the 100th generation using IMORE for Δ =1%, Δ =5%, offline adaptive, and online adaptive IMORE

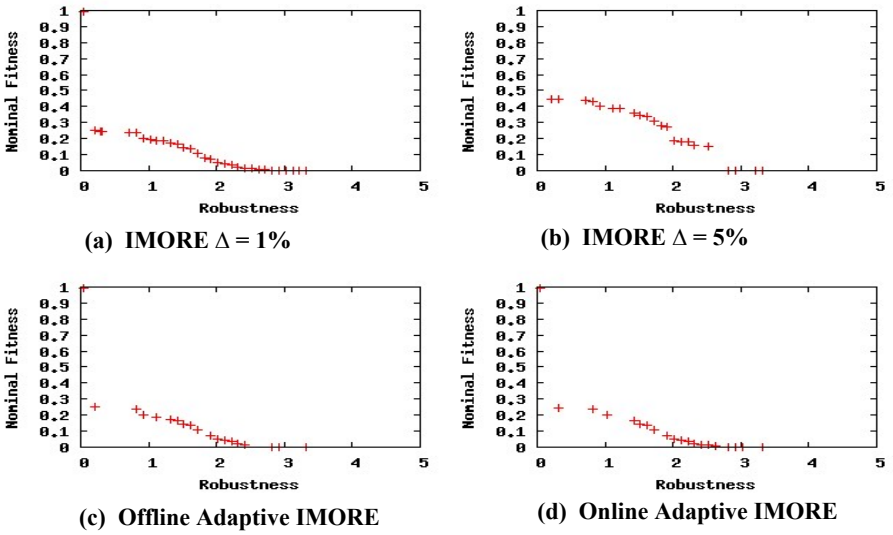


Figure 10. Pareto fronts of test function G10 at the end of the 100th generation using IMORE for Δ =1%, Δ =5%, offline adaptive, and online adaptive IMORE

Figure 11 presents the average normalized computational costs for different IMORE algorithms across 20 independent runs. Note that the computational cost incurred by both offline and online adaptive IMORE algorithms are significantly reduced in comparison to the IMORE with Δ fixed at 1%. As described earlier in section 3, the computational cost of the IMORE algorithm is *O(gnkl)*. The offline and online adaptive IMORE on the other hand incur a computational cost of *O*(*gnpl*) where *p* is the average number of nested searches required. Since $p << k$, it makes good sense that the adaptive IMORE algorithms are significantly more efficient than its original non-adaptive counterpart. Further, a small value for *Pc* in Figure 12 also indicates that both offline and online adaptive IMORE provide good convergence to the true optimal pareto front.

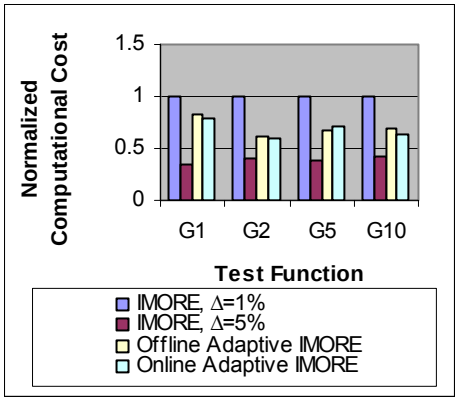


Figure 11. Normalized computational cost incurred for functions G1, G2, G5, and G10 in the IMORE for Δ =1%, Δ =5%, offline adaptive, and online adaptive IMORE

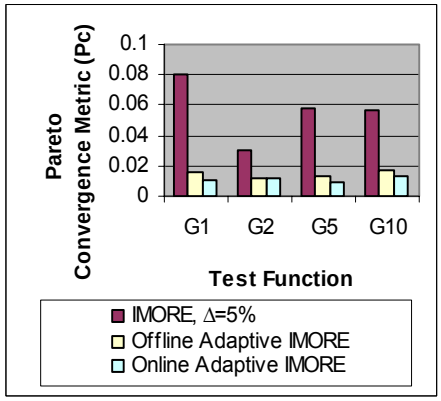


Figure 12. Pareto front convergence metric for functions G2, G5, and G10 in the IMORE for Δ =5%, offline adaptive, and online adaptive IMORE

4.2 Effect of Update Interval Setting in Online Adaptive IMORE

In this work, we also consider the effect of the step size update interval, i.e. *t* in Figure 6, on the computational efficiency of the online adaptive IMORE. The 1) normalized computational cost (with respect to the cost incurred by the IMORE with Δ =1%), 2) pareto front convergence metric for several step size update intervals, *t*, are presented in Figures 13 and 14, respectively. These are the average result of 20 independent runs.

It can be observed from Figure 13 that all settings of the update interval considered in the study lead to great savings in computational cost over the IMORE with Δ =1%. On the other hand, the final pareto front which the online adaptive IMORE converges to is highly sensitive to the configurations of the update interval, *t*. This may be observed in Figure 14 where the pareto front convergence metric is shown to

increase with t, which indicates a high dissimilarity between the final and true pareto front with increasing t. Hence, in order to avoid convergence to the false pareto front and maintain the utility of the online adaptive IMORE, appropriate values for t should be chosen. From Figure 13 and 14, t=20 appears to give an appropriate balance between computational cost and convergence accuracy in the online adaptive IMORE search. Such an outcome makes good sense and can be easily explained. At the extreme where t=1, the online adaptive IMORE has a computational cost that is equivalent to the IMORE for Δ=1% due to the overheads to determine the robustness accuracies in every search generation. Conversely, since the total computational budget is fixed at 100 generations, a lower adaptation frequency may be achieved for large values of t, (for instance, when t=50, Δ_c = 5% is used in the search for at least the initial fifty search generations and has an upper bound adaptation frequency of 1), leading to the high possibility of converging to the false optimal pareto front. In Figure 14, it may also be observed the convergence accuracies deteriorate when t increases to 40 or 50. Consequently, t at 20 (i.e., providing an adaptation frequency upper bound of 4) serves as an appropriate configuration to provide a good balance between computational cost and accurate convergence in the online adaptive IMORE.

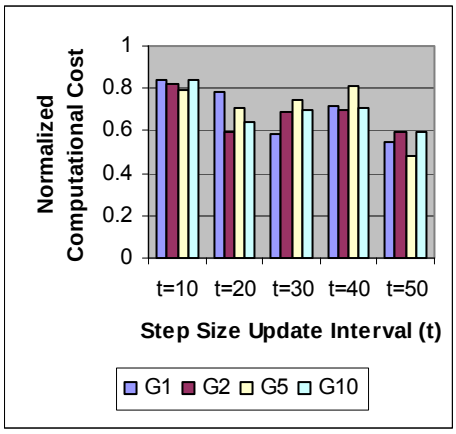


Figure 13. Normalized computational cost incurred for functions G1, G2, G5, and G10 using different t generation intervals in the online adaptive IMORE

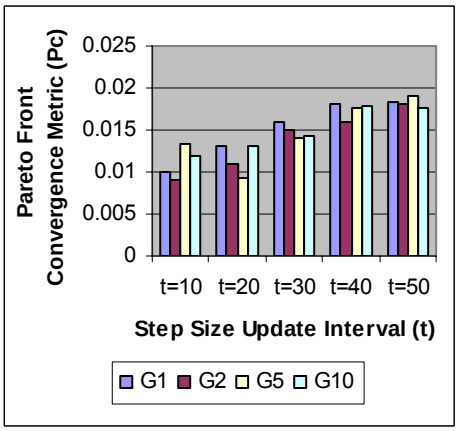


Figure 14. Pareto front convergence metric for functions G1, G2, G5, and G10 using different t generation intervals in the online adaptive IMORE

4.3 Enhancing the Computational Efficiency of Online Adaptive IMORE by Design of Experiments (DOE) Sampling Techniques

In our effort to further enhance the computational efficiency of the online adaptive IMORE, we also study the possibility of replacing the series of nested optimization sub-problems which is computationally very intensive, with well-known sampling methods. In particular, we consider the use of Design of Experiments (DOE) sam-

pling approaches including Random Sampling (RS), Stratified Sampling (SS), and Latin Hypercube Sampling (LHS) [19-21] to generate m sampled design points as an approximation of the worst-case performance for a design in each of the k iterations (please refer to eq. (7)). Here, we present only the empirical results obtained for the more complex test functions, which include G2, G5, and G10.

To generate any possible savings in computational cost using approximation via DOE approaches in the online adaptive IMORE, it is required for $m<<l$, where m and l are the number of calls to objective function required for using DOE approaches and nested optimization sub-problems, respectively. In the experimental study here, m is configured to 243 for lower dimensional problems (G2 and G5), and 1024 for higher dimensional problem G10 (since l is around 400 and 2000 for these two types of problems). Note that this guarantees a computational cost reduction of approximately 40-50% when using the DOE approaches in the IMORE. All other configurations of the online adaptive IMORE are kept the same as in section 4.1.

Consequently, our aim here is to determine whether the incorporated approximation through DOE samplings could also lead to convergence to the true pareto front. The normalized computational cost (with respect to the cost incurred by the IMORE with Δ=1%) and pareto front convergence metric for the various IMORE algorithms averaged across 20 independent runs are presented in Figures 15 and 16, respectively. The online adaptive IMORE with DOE samplings are labeled here as OAS-RS IMORE, OAS-SS IMORE and OAS-SS IMORE.

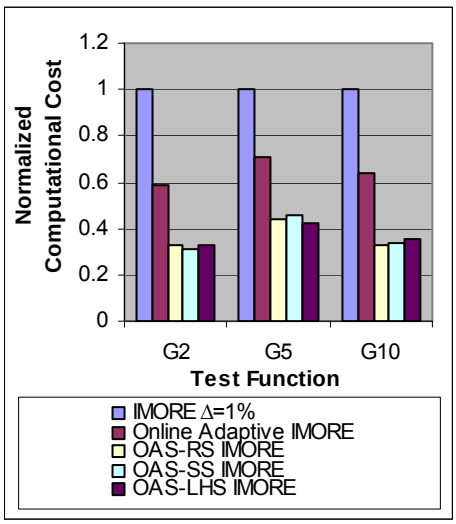


Figure 15. Normalized computational cost incurred for functions G2, G5, and G10 in IMORE, online adaptive IMORE, and OAS IMORE algorithms

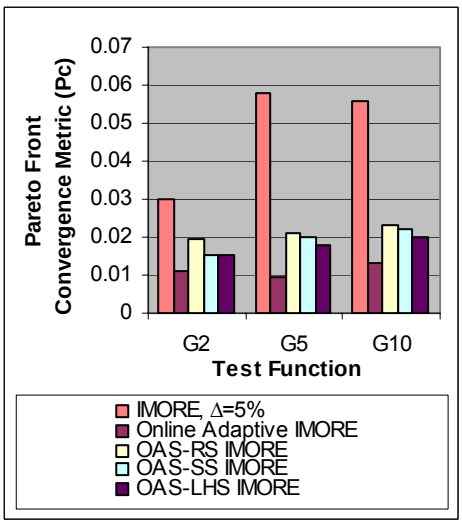


Figure 16. Pareto front convergence metric for functions G2, G5, and G10 in online adaptive and OAS IMORE algorithms

All the OAS IMORE algorithms lead to significant savings in computational costs over both the IMORE and online adaptive IMORE as expected (since *m:l* = 243:400, and 1024:2000), and arrive at approximately 40-50% increase in search efficiency compared to the online adaptive IMORE.

From Figure 16, there is a clear trade-off in convergence accuracies for reduction in computational cost when using approximation in the OAS IMORE algorithms. Nevertheless, the OAS IMORE algorithms remain to converge to the true pareto front accurately. This is indicated in Figure 16 where the pareto front convergence accuracies, *Pc*, are observed to maintain below or around 0.02 for G2, G5, and G10, which is clearly competitive to the online adaptive IMORE. In addition, among the OAS IMORE algorithms, RS results in poorer convergence accuracies compared to using SS or LHS for approximations when searching on the 2D and 5D functions considered. This is likely due to the poor coverage of the search space when using random sampling. Further, larger convergence inaccuracies to the pareto front for G5 and G10 are observed in Figure 16. It is worth noting that this is the effect of the 'curse of dimensionality' and implies the sample size used, *m*, may require to increase exponentially in order to provide a good coverage of the nested search space as the search dimension grows. To cope with the issue, a possible solution is to also adapt the sample size in the OAS IMORE search.

In Figures 17 and 18, we present the average performances of the OAS-LHS IMORE search on G2, G5, and G10 with an increasing sample size. The sample size is set to increment gradually with increasing search generations, and is formulated as follows:

$$m_i = m_{min} + \left\lceil \frac{i}{g_{max}} \left(m_{max} - m_{min} \right) \right\rceil \tag{17}$$

where m_i is the sample size at the i^{th} generation, g_{max} is the maximum total search generations before termination, m_{min} and m_{max} are the minimum and maximum sample size. In the experiment, the setting for $m_{min,}$ m_{max}, and g_{max} are as listed in Table 3.

Table 3. $m_{min,}$ m_{max}, and g_{max} used for test function G2, G5, and G10.

Test Function	m_{min}	m_{max}	g_{max}
G2	20	200	100
G5	20	200	100
G10	102	1024	100

The results are compared to those of the online adaptive and OAS-LHS IMORE as reported in Figure 15 and 16 previously. By adapting the sample size *m*, there is now more than 70% reduction in the computational cost. More importantly, the trade-off for this significant computational cost for the convergence accuracy is minimum as shown in Figure 18, even for higher dimensional problem (G10).

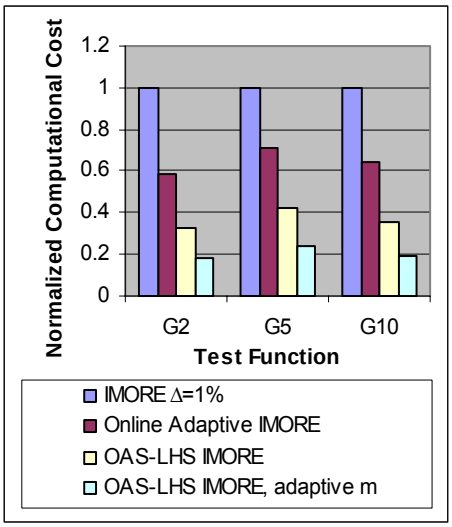


Figure 17. Normalized computational cost incurred for functions G2, G5, and G10 in online adaptive, OAS-LHS with fixed sample size, and OAS-LHS with adaptive sample size IMORE algorithms.

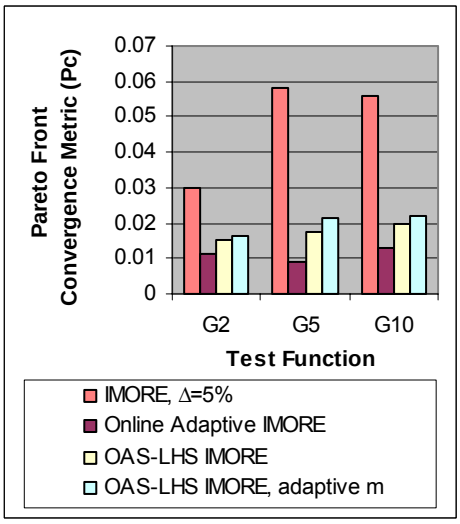


Figure 18. Pareto convergence metric for functions G2, G5, and G10 in online adaptive, OAS-LHS with fixed sample size, and OAS-LHS with adaptive sample size IMORE algorithms

5 Conclusion

In this paper, an Inverse Multi-Objective Robust Evolutionary (IMORE) algorithm for design optimization in the presence of uncertainty has been presented. Using *a prior* information on the desire robustness of the final design, the algorithm was shown capable of converging to a set of solutions that gives good nominal performances while handling maximum robustness. Most importantly, these solutions were discovered without any requirement to make possible untrue assumptions about the structure of the uncertainties involved. It is realized that the major drawback of the IMORE would be the massive computational cost possibly incurred. Hence, adaptive strategies are introduced in IMORE algorithm to reduce the massive computational efforts incurred in the nested design searches. In particular, we consider adapting the step size for determining the search bound at every sub-problem optimization and trim down the number of objective function calls by DOE sampling methods. Empirical results on diverse test functions show that the proposed adaptive IMORE algorithms provide convergence to the true pareto fronts on all the functions considered. The computational costs incurred by the adaptive IMORE algorithms are also significantly reduced.

Since the adaptive IMORE still typically requires enormous number of function evaluations to locate near pareto optimal solutions, the use of IMORE can become computationally prohibitive for the class of problems with computationally expensive objective functions. It is thus desirable to retain the appeal of inverse multi-objective robust evolutionary algorithms that can handle computationally expensive design

problems and produce high quality designs under limited computational budgets. An intriguing future work is hence to consider the use of meta-modeling strategies [22-24] in the IMORE design methodology for solving problems with computationally expensive objective functions.

Acknowledgement

This work was funded by Honda Research Institute Germany. The authors would like to thank Honda Research Institute Germany and Parallel and Distributed Computing Centre of Nanyang Technological University for their support in this work.

References

[1] Goldberg D.E., "Genetic Algorithms in Search, Optimization and Machine Learning", 1989.

[2] Jin Y., Branke J., "Evolutionary Optimization in Uncertain Environment-A Survey", *IEEE Transactions on Evolutionary Computation*, Vol. 9, No. 3, June 2005.

[3] Ong Y.S., Nair P.B., Lum K.Y., "Min-Max Surrogate Assisted Evolutionary Algorithm for Robust Aerodynamic Design", *Special Issue on Evolutionary Optimization in the Presence of Uncertainties, IEEE Transactions on Evolutionary Computation*, in press, 2005.

[4] Oberkampf W.L., et. al., "Estimation of Total Uncertainty in Modeling and Simulation", *Sandia Report SAND2000-0824*, 2000

[5] Oberkampf W.L., Helton J., Sentz K., "Mathematical Representation of Uncertainty", *AIAA Proceedings of Non-Deterministic Approaches Forum*, Paper no. 2001-1645, Reston, VA, 2001.

[6] Ben-Haim Y., "Information Gap Decision Theory", California: Academic Press, 2001.

[7] Ben-Haim Y., "Uncertainty, Probability, and Information-Gaps", Reliability Engineering and System Safety 85, pp. 249-266, 2004.

[8] Ben-Haim Y., "Robust Reliability in Mechanical Sciences", Springer-Verlag, Berlin, 1996.

[9] L. Huyse, "Solving Problems of Optimisation Under Uncertainty as Statistical Decision Problems", AIAA-2002-1519, 2001.

[10] Tsutsui S. and Ghosh A., "Genetic Algorithms with a Robust Solution Searching Scheme", *IEEE Transaction on Evolutionary Computation*, Vol. 1, No. 3, pp. 201-208, 1997.

[11] Arnold D. V. and Beyer H. G., "Local Performance of the (1+1)-ES in a Noisy Environment*", IEEE Trans. Evolutionary Computation*, Vol. 6, No. 1, pp 30-41, 2002.

[12] Jin Y. and Sendhoff B., "Trade-Off between Performance and Robustness: An Evolutionary Multiobjective Approach", *Proceedings of Second International Conference on Evolutionary Multi-criteria Optimization*. LNCS 2632, Springer, pp.237-251, Faro, 2003.

[13] Huyse L. and Lewis R.M., "Aerodynamic Shape Optimization of Two-dimensional Airfoils Under Uncertain Operating Conditions", Hampton, Virginia: ICASE NASA Langley Research Centre, 2001.

[14] Anthony D.K. and Keane A.J., "Robust Optimal Design of a Lightweight Space Structure Using a Genetic Algorithm", *AIAA Journal* 41(8), pp. 1601-1604, 2003.

[15] Wiesmann D., Hammel U. and Back T., "Robust design of multilayer optical coatings by means of evolutionary algorithms*", IEEE Trans Evolutionary Computation*, Vol. 2, No. 4, pp 162-167, 1998.

[16] N. Srinivas and K. Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, *Evolutionary Computation*, Vol. 2, No. 3, pp 221-248, 1994.

[17] Deb K., Jain S., "Running Performance Metrics for Evolutionary Multiobjective Optimization", *Proceeding of the fourth Asia Pacific Conference on Simulated Evolution and Learning*, pp 13-20, 2002.

[18] Grosan C., Oltean M., Dumitrescu D., "Performance Metrics for Multiobjective Optimization Evolutionary Algorithms", *Proceedings of Conference on Applied and Industrial Mathematics*, Oradea, Romania, 2003.

[19] Fang K.T., Ma C.X., and Winker P., "Centered L2-Discrepancy of Random Sampling and Latin Hypercube Design, and Construction of Uniform Designs", *Mathematics of Computation*, Vol. 71, No. 237, pp. 275-296, 2000.

[20] Simpson T.W., Lin D.K.J., Wei C., "Sampling Strategies for Computer Experiments: Design and Analysis", *International Journal of Reliability and Applications*, 2001.

[21] Tang B., "Orthogonal Array-based Latin Hypercubes", Journal of the American Statistical Association, Vol. 88, No. 424, pp. 1392-1397, 1993.

[22] Ong Y.S., Lum K.Y., Nair P.B., Shi D.M. and Zhang Z.K., "Global Convergence of Unconstrained and Bound Constrained Surrogate-Assisted Evolutionary Search in Aerodynamic Shape Design Solvers", *IEEE Congress on Evolutionary Computation, Special Session on Design Optimization with Evolutionary Computation*, Vol. 3, pp. 1856-1863, December 8-12, 2003, Canberra, Australia.

[23] Ong Y.S., Nair P.B. and Keane A.J., "Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling", *AIAA Journal*, Vol. 41, No. 4, pp 687-696, 2003.

[24] Jin Y., "A comprehensive survey of fitness approximation in evolutionary computation", *Soft Computing Journal*, Vol. 9, No. 1, pp 3-12, 2005.