

Inverse Multi-Objective Robust Evolutionary Design

Dudy Lim¹, Yew-Soon Ong¹, Yaochu Jin², Bernhard Sendhoff² and Bu Sung Lee¹

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

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

Abstract. In this paper, we present an Inverse Multi-Objective Robust Evolutionary (IMORE) design methodology that handles the presence of uncertainty without making assumptions about the uncertainty structure. We model the clustering of uncertain events in families of nested sets using a multi-level optimization search. To reduce the high computational costs of the proposed methodology we proposed schemes for 1) adapting the step-size in estimating the uncertainty, and 2) trimming down the number of calls to the objective function in the nested search. Both offline and online adaptation strategies are considered in conjunction with the IMORE design algorithm. Design of Experiments (DOE) approaches further reduce the number of objective function calls in the online adaptive IMORE algorithm. Empirical studies conducted on a series of test functions having diverse complexities show that the proposed algorithms converge to a set of Pareto-optimal design solutions with non-dominated nominal and robustness performances efficiently.

1 Introduction

Evolutionary Algorithms (EAs) [1] are modern stochastic optimization technique that has emerged as a prominent contender for global optimization in complex engineering design. Its popularity lies in the ease of implementation and the ability to arrive close to the global optimum design. Most studies on the application of EAs to complex engineering design have focused on locating the global optimal design using deterministic computational models. However in many real-world design problems, uncertainties are present and practically impossible to avoid. In the case where a solution is very sensitive to small variations either in design variables or operating conditions, it may not be desirable to use this design. Hence optimization without taking uncertainty into considerations produces designs that should not be labeled as optimal because they are likely to perform differently when put into practice.

Various classifications of uncertainty in design optimization have been suggested over recent years [2-8]. In [2], four types of uncertainty were described. They are 1) noise in fitness function, 2) uncertainty in design and/or environmental parameters, 3) approximation errors in fitness function, 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 more knowledge can be acquired. In [6-8], uncertainty is defined as the gap between 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]. A detail analysis of deterministic optimization framework for dealing with uncertainty in linear programming and general convex programming was presented in [10]. In the context of stochastic optimization, especially evolutionary algorithms, a number of prominent new studies on handling the presence of uncertainty in engineering designs have emerged recently. In [11], 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 evaluations. The study of (1+1)-Evolutionary Strategy (ES) with isotropic normal mutations using the noisy phenotype scheme was subsequently reported in [12]. [13] considers the trade-off between robustness and the nominal performance of a potential solution using a multi-objective EA approach while [3] described a combined max-min and Baldwinian¹ trust-region optimization strategy for conservative robust design. To reduce the high computational costs of robust evolutionary design, computationally cheap local surrogate models was introduced in [3] and [14] for estimating the expected fitness and/or variance of potential solutions in place of the exact fitness functions. The success of robust evolutionary design has been shown on a series of realistic mechanical and aerodynamic problems, including 2D aerodynamic airfoil [3,18], lightweight space structures [19] or multilayer optical coating design [20].

In most of the existing schemes proposed, prior knowledge about the structure of the uncertainty, for instance, its distribution properties is assumed to be available. Hence, the quality of a solution is attainable only if the assumptions made on the structure of the uncertainty exactly reflect the actual uncertainty. In this paper, 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

¹ There are two basic strategies for using Memetic Algorithms [15][16][17]:

- Lamarckian learning forces the genotype to reflect the result of improvement in local search by placing the locally improved individual back into the population to compete for reproductive opportunities
- Baldwinian learning only alters the fitness of the individuals and the improved genotype is not encoded back into the population.

Robust Evolutionary design or IMORE in short. In contrast to conventional robust optimization, our proposed approach avoids making assumptions about the uncertainty structure in formulating the optimization process, since it can lead to erroneous designs with catastrophic consequences. A drawback of the IMORE methodology is the massive computational effort of the nested evolutionary searches involved which can be defined by the step-size and maximum fitness function calls used. To improve the efficiency of the proposed methodology, we present both offline and online strategies for adapting step-size and minimizing calls to the fitness function using 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 and our proposed IMORE methodology. Section 3 introduces the adaptation strategies for improving the computational efficiency of IMORE. To illustrate the efficacy of the adaptive IMORE, section 4 provides an empirical study on a series of test functions with diverse complexities. Further enhancements on speed-up of the adaptive IMORE algorithm using DOE methods is also presented in the section. Finally, Section 5 concludes this paper.

2 Robust Evolutionary Optimization in the Presence of Uncertainty

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

$$\begin{aligned} \text{Maximize : } & f(\mathbf{x}) \\ \text{Subject to : } & \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u \end{aligned} \quad (1)$$

where $f(\mathbf{x})$ is a scalar-valued objective function, $\mathbf{x} \in \mathfrak{R}^n$ is the vector of design variables, while \mathbf{x}_l and \mathbf{x}_u are vectors of lower and upper bounds for the design variables.

Here, our focus is on EAs for robust design optimization in the presence of uncertainty that arises in:

i) design parameters \mathbf{x}

$$f'(\mathbf{x}) = f(\mathbf{x} + \boldsymbol{\delta}) \quad (2)$$

where $\boldsymbol{\delta} = (\delta_1, \delta_2, \dots, \delta_k)$, is the noise vector in the design parameters and $f'(\mathbf{x})$ is the perturbed function value of the design vector \mathbf{x} .

ii) operating/environmental conditions

$$f'(\mathbf{x}) = f(\mathbf{x}, \mathbf{c} + \boldsymbol{\zeta}) \quad (3)$$

where $\mathbf{c} = (c_1, c_2, \dots, c_n)$, is the nominal vector of the environmental parameters and $\boldsymbol{\zeta}$ is a random vector used to model the variability in the operating conditions. Both forms of uncertainties may be treated equivalently [13].

The core mechanism of many evolutionary techniques for handling uncertainty has relied on the 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 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 theorem². The effective fitness $F(\mathbf{x})$ is then be defined as:

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

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

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

To locate a robust design solution in the presence of uncertainty in the design vector, one may consider using the GA/RS3 proposed in [11] that is outlined in Figure 1.

² Based on the central limit theorem, random samples from a given distribution with mean μ and variance σ^2 will approach a Gaussian/Normal distribution $\mathcal{N}(\mu, \sigma^2)$ when the sample size increases.

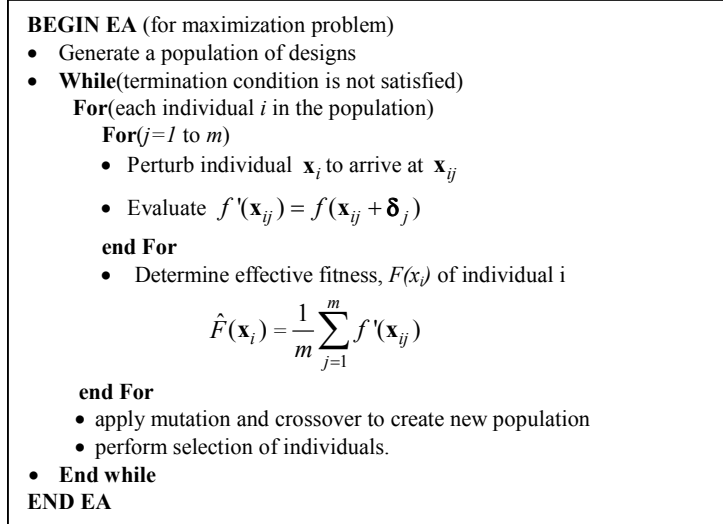


Figure 1. A Genetic Algorithm with Robust Solution Searching Scheme (GA/RS3)

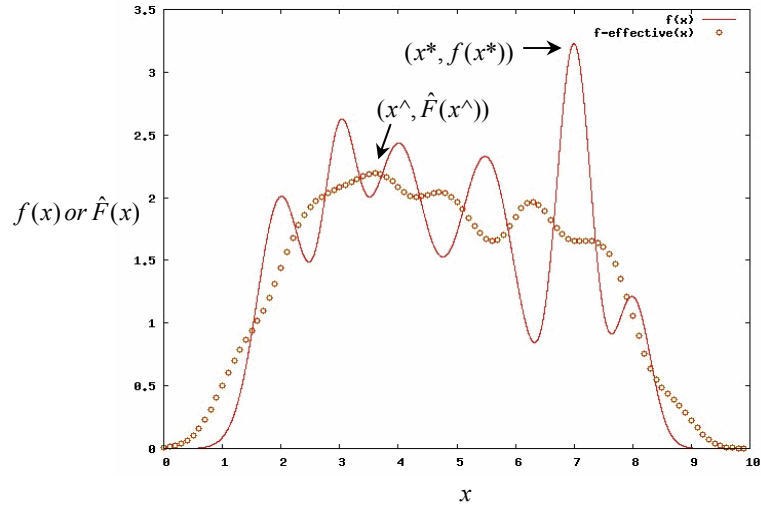
Consider the one-dimensional function depicted in Figure 2 which is given by

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

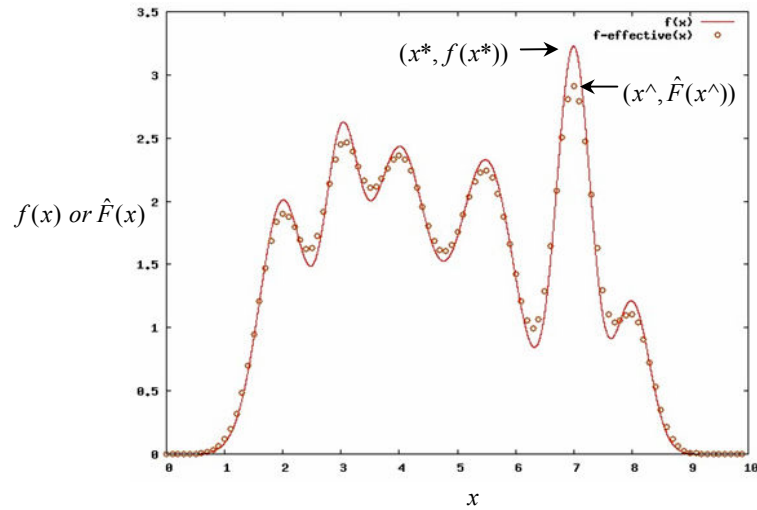
Equation (6) defines a multimodal function with a nominal global maximum located at sharp peak $x^* \in [6.5, 7.8]$ and has many other local optima located elsewhere³. The robust optimal solution that the GA/RS3 converges to is dependent on the perturbation assumed, *i.e.*, assumptions on the distribution of δ in $f(x)$. For instance, Figures 2(a) and (b) illustrate two resultant effective fitness landscapes (denoted by circles) of the one-dimensional function defined in equation (6), assuming a uniform distribution for δ with σ of ± 1.0 or ± 0.25 , respectively. Note that σ denotes the range or bound of δ . If σ is configured to ± 1.0 , the robust global maximum⁴ can be observed to be located at $x^{\wedge} \in [3.0, 4.0]$ in Figure 2(a). On the other hand, when σ is set to ± 0.25 , the global robust maximum approaches that of the nominal fitness function $f(x)$, *i.e.*, $x^{\wedge} = x^*$, see in Figure 2(b).

³ Note that x^* represents the nominal global optimum (maximum).

⁴ Note that x^{\wedge} represents the robust global optimum (maximum).



(a) Range of the uncertainty, $\sigma = \pm 1.00$



(b) Range of the uncertainty, $\sigma = \pm 0.25$

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 this section, we present a Inverse Multi-Objective Robust Evolutionary (IMORE) design optimization strategy for locating solutions with non-dominated nominal performance and robustness in the presence of uncertainties. In contrast to existing robust EA schemes [10-14], we do not make assumption on the uncertainty involved, since little knowledge about the structure of the uncertainty involved is available *a priori* in most realistic problems. Hence, instead of focusing on making any probably unjustifiable mathematical model out of the uncertainty, IMORE focus here on how a design may deteriorate in the presence of uncertainties. Further, taking into account the necessity to deal with the trade-off between robustness and nominal fitness in IMORE, we consider the bound-constrained multi-objective optimization problem of the form:

$$\begin{aligned}
 \textbf{Maximize} & : \text{Objective-1 (nominal fitness)} = f(\mathbf{x}) \\
 & \text{Objective-2 (robustness)} = d(\mathbf{x}; d_t, \Delta) \\
 \textbf{Subject to} & : \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u
 \end{aligned} \tag{7}$$

The basic steps of the proposed IMORE algorithm are outlined in Figure 3. In the first step, the maximum degradation tolerable for the final design, d_t and the step-size Δ used to conduct nested searches are initialized. A population of designs is then generated randomly or using DOE methods such as the Latin hypercube sampling or minimum discrepancy sequences [24]. Each individual in the population is first evaluated to obtain its nominal fitness. Subsequently, each individual undergoes a sequence of nested searches to establish the uncertainty or maximum variations in design parameters for the given maximum performance degradation tolerable, in the spirit of Info-Gap theory [6-8]. In particular, we solve a sequence of constrained optimization sub-problems for each chromosome of the form:

$$\begin{aligned}
 \textbf{Maximize} & : d(\mathbf{x}_i) = f(\mathbf{x}_i) - f(\mathbf{x}) \\
 \textbf{Subject to} & : \mathbf{x}_l^k \leq \mathbf{x} \leq \mathbf{x}_u^k
 \end{aligned} \tag{8}$$

where \mathbf{x}_l^k and \mathbf{x}_u^k are the appropriate bounds on the design parameters, which are updated at each k iteration based on the defined step-size, Δ .

For each optimization sub-problem (or during the k^{th} iteration), the optimal solution of the k^{th} sub-problem is sought. The objective of each sub-problem search is to find the worst possible performance degradation by solving a bound constrained maximization problem. After each iteration, the search bounds of the design parameters, \mathbf{x}_l^k and \mathbf{x}_u^k are updated using the step-size Δ which is given by

$$\mathbf{x}_l^k = \mathbf{x}_l - k\Delta, \quad \mathbf{x}_u^k = \mathbf{x}_u + k\Delta \tag{9}$$

By conducting a sequence of nested searches across a family of ascending nested bounds parameterized by the uncertainty vector, we arrive at a monotonically increas-

ing function of performance degradation versus uncertainty as illustrated in Figure 4 such that

$$\mathbf{x}_l^{k+1} \leq \mathbf{x}_l^k, \mathbf{x}_u^k \leq \mathbf{x}_u^{k+1} \rightarrow d(\mathbf{x}_{opt}^k) \leq d(\mathbf{x}_{opt}^{k+1}) \quad (10)$$

where \mathbf{x}_{opt}^k denotes the optimum (i.e., worst-case design vector) at the k^{th} iteration and $d(\mathbf{x}_{opt}^k) = f(\mathbf{x}_i) - f(\mathbf{x}_{opt}^k)$ is the corresponding maximum performance degradation obtained for $\mathbf{x}_l^k \leq \mathbf{x} \leq \mathbf{x}_u^k$.

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

$$\{d^k(\mathbf{x}_i) = f(\mathbf{x}_i) - f(\mathbf{x}_{opt}^k)\} > d_t \quad (11)$$


```

BEGIN IMORE (Consider a maximizing problem)
Initialization Phase:

- Initialize Maximum degradation tolerable for the final design,  $d_t$
- Initialize the step size  $\Delta$  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) =  $f(\mathbf{x}_i)$
- Objective-2 (robustness) =  $d(\mathbf{x}_i; d_t, \Delta) = \delta_{max}^i$ 
  - Repeat
    - Maximize:  $d(\mathbf{x}_i) = f(\mathbf{x}_i) - f(\mathbf{x})$
    - subject to:  $\mathbf{x}_l^k \leq \mathbf{x} \leq \mathbf{x}_u^k$  where  $\mathbf{x}_l^k = \mathbf{x}_i - k\Delta$ ,  $\mathbf{x}_u^k = \mathbf{x}_i + k\Delta$
    - Obtain  $\mathbf{x}_{opt}^k$  and  $d(\mathbf{x}_{opt}^k)$
    - Store  $d(\mathbf{x}_{opt}^k)$  and associate it with  $k\Delta$
  - until  $\{d^k(\mathbf{x}_i) = f(\mathbf{x}_i) - f(\mathbf{x}_{opt}^k)\} > d_t$
  - Estimate maximum uncertainty  $\delta_{max}^i$  using linear interpolation from  $d(\mathbf{x}_{opt}^k)$  for different  $k\Delta$

end For

- Apply standard MOEA operators to create a new population

end While
END IMORE

```

Figure 3. The IMORE design optimization algorithm.

At the end of the iterative search, the maximum uncertainty δ_{max} that a design may handle for given maximum performance degradation of d_t tolerable can be interpolated from the database of previous uncertainties and corresponding maximum performance degradations, *i.e.*, $k\Delta$ and $d^k(\mathbf{x}_i)$. This is also illustrated using 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 when the termination condition is met.

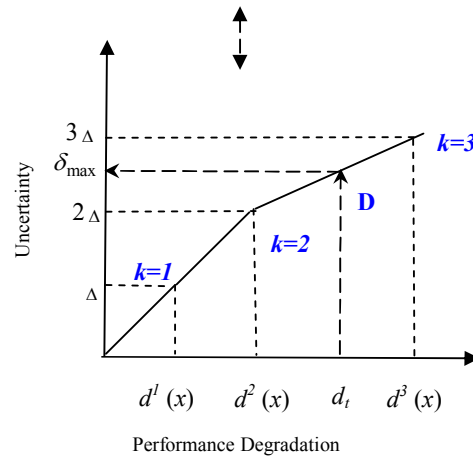
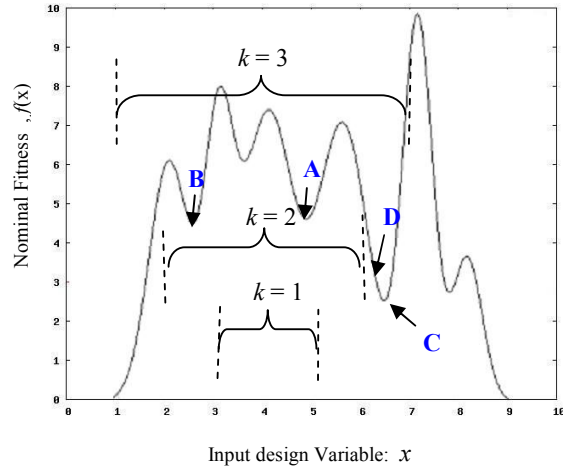


Figure 4. Steps of IMORE for $X_i=4$ and $\Delta=1$.

To illustrate how the results obtained by the IMORE algorithm may be useful for robust design, we consider here a multimodal test function based on the one-dimensional "Michalewicz 2" function which is defined by:

$$f(x) = \sum_{i=1}^{10} \left(\sin(x) \sin^{10} \left(\frac{ix^2}{\pi} \right) \right), -1.5 \leq x \leq 3 \quad (12)$$

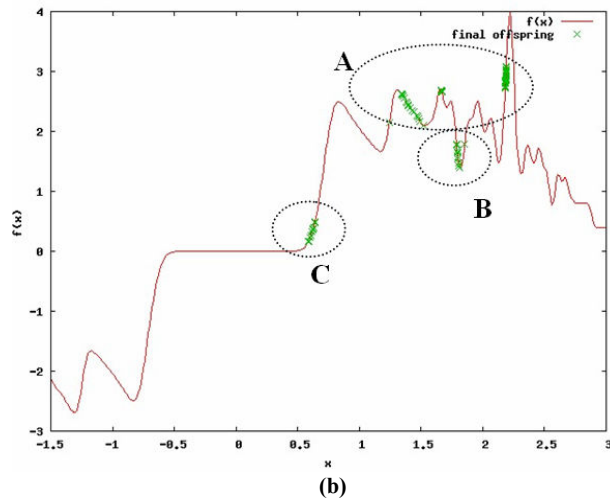
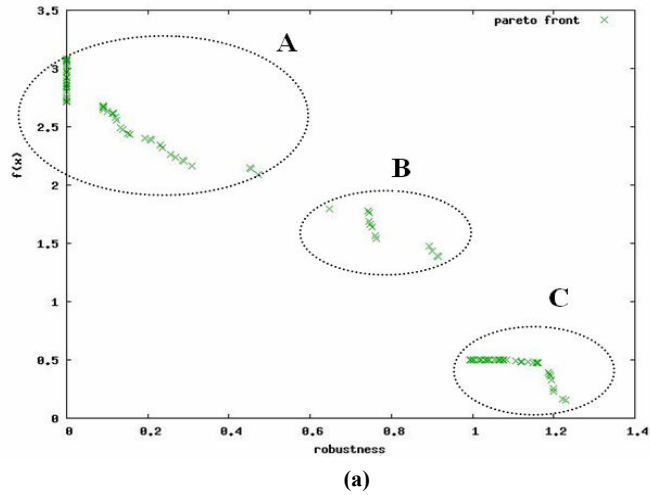


Figure 5. (a) Pareto front, (b) Corresponding offspring in (a) for the function in eq. (12).

This function contains a mixture of flat robust region having moderate nominal fitness around $x \in [-0.5, 0.5]$ and some noisy peaks with good nominal fitness around $x \in [0.5, 3]$ as depicted in Figure 5b. The solutions in the Pareto front of Figure 5a represent the diverse set of designs having non-dominated nominal performances and robustness in the presence of uncertainties. To explain the results presented in Figure 5, we cluster the solutions in the Pareto front as three separate groups. Group A consists of solutions that have excellent nominal fitness at the expense of poor robustness. On the other hand, group B are solutions that give a good balance between nominal

fitness and robustness, while the solution members of group C have poor nominal fitness but excellent robustness measure. Hence, in the case of real world engineering problem, the availability of the set of non-dominated solution can provide a wide range of option for selecting the design vectors based on the requirements on robustness and nominal fitness.

3 Adaptive Inverse Multi-Objective Robust Evolutionary Design Optimization

In this section, we present a study on the computational complexity of the proposed IMORE methodology and subsequently introducing possible strategies to achieve better efficiency with minimum impact on the performance of the algorithm. The computational complexity of the IMORE algorithm described in section 2 is $O(gnkl)$, where g is the maximum number of IMORE generations, n is the number of 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. Hence the computational costs to locate the Pareto-optimal solutions can become intractable if the objective function is computationally expensive.

For every individual, if δ_{max}^{approx} represents the approximated robustness fitness by IMORE, while δ_{max}^{exact} is a more accurate robustness fitness obtained from using an exhaustive search, the Average Approximated Robustness (AAR) and Average Exact Robustness (AER) of an EA population is then measured as:

$$AAR = \frac{1}{n} \sum_{i=1}^n \frac{\delta_{max}^{approx} - i}{x_u - x_l} \times 100\%, \quad AER = \frac{1}{n} \sum_{i=1}^n \frac{\delta_{max}^{exact} - i}{x_u - x_l} \times 100\% \quad (13)$$

where n is the population size, x_u and x_l are the upper and lower bounds 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 the pareto-optimum solutions in the "Michalewicz 2" one-dimensional function.

The AAR and AER of a population for differing 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., $|AAR-AER|/AER$, varies greatly with Δ and k . It can be observed that the average error in estimating the robustness increases with Δ , but incurs a lower computational cost due to a smaller k . This makes good sense since a larger step-size generally gives rise to greater interpolation errors. However, this inferiority in accuracy could also lead the IMORE search convergence to false Pareto-optimal solutions. On the other hand, fine step-size provides a lower average error but at the expense of higher computational costs. 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 achieve an appropriate balance between 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 itera- tions k	Average Ap- proximated Robustness AAR (%)	Average Exact Ro- bustness 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 6.

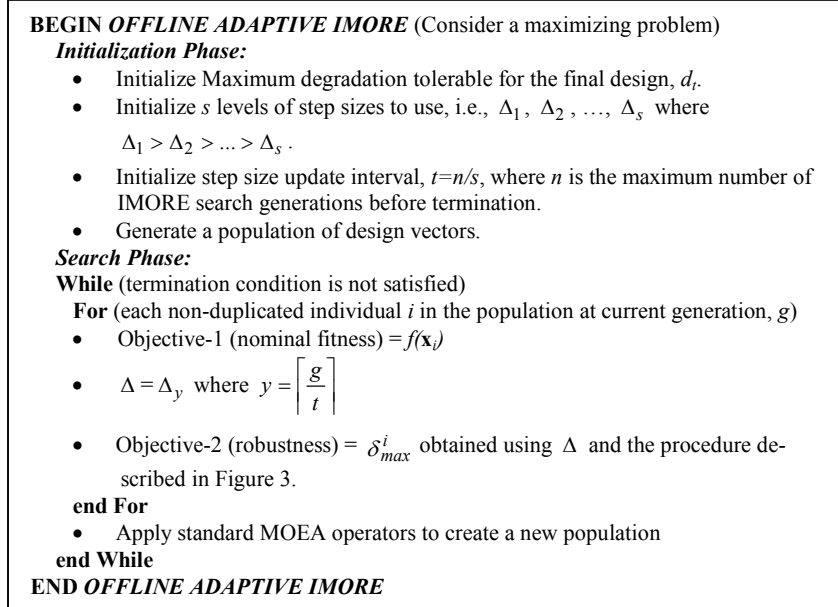


Figure 6. Offline adaptive IMORE design optimization algorithm.

In the initialization phase of the offline adaptive IMORE search, s levels of step-sizes are defined from $\Delta_1, \Delta_2, \dots, \Delta_s$. 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., $\Delta_1 > \Delta_2 > \dots > \Delta_s$.

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 adjust the step-size so that it decrements every t generations and is defined by $t=n/s$ where n is the number of IMORE search generations. 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. (8). The appropriate bounds of \mathbf{x}_l^k and \mathbf{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 counter. The offline adaptive IMORE search operates exactly like the IMORE search (as described in Figure 3) and stops when the termination criterion is reached.

3.2 Online Adaptive IMORE Design Optimization

Next, we present an alternative to achieve a suitable balance of k and Δ throughout the IMORE search based on an online adaptation strategy. In contrast to the offline adaptation strategy, which fixed the step sizes to be used across the various phases of the IMORE search in advance, the online adaptation strategy decides the values of k and Δ using online feedback on the accuracy of the approximated robustness fitness throughout the search. The detailed procedure of the online adaptive IMORE is described in Figure 7.

In the online adaptive IMORE algorithm studied here, we consider a straightforward toggling between two different step-sizes, particularly, a fine and coarse step-sizes which are denoted by 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 error in estimating the robustness of the fitness is assessed for every t generations. The error of the i^{th} individual, denoted by ρ_i , is then determined by

$$\rho_i = \frac{|\delta_{\max-f}^i - \delta_{\max-c}^i|}{\delta_{\max-f}^i} \times 100\% \quad (14)$$

where $\delta_{\max-f}^i$ and $\delta_{\max-c}^i$ 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 individuals have $\rho_i \geq 20\%$, the fine step-size Δ_f will be used for the next t generations to gain a better accuracy. On the other hand, if most individuals in the population have $\rho_i < 20\%$, a coarse step-size Δ_c will be adopted for the next t generations since the accuracy of the robustness fitness is considered as adequate. Here, we empirically

determine 75% of the population size and 20% of $\delta_{\max-f}^i$ as the cut-off values to represent the majority of the population and 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,  $d_f$ .
  • Initialize the fine and coarse step size,  $\Delta_f$  and  $\Delta_c$ .
  • 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) =  $f(\mathbf{x}_i)$ 
  • Objective-2 (robustness) =  $\delta_{\max}^i$  obtained using  $\Delta$  and the procedure de-
    scribed in Figure 3.
  • If (mod( $g,t$ ) == 0)
    ➤ Obtain  $\delta_{\max-f}^i$  and  $\delta_{\max-c}^i$  for  $\Delta = \Delta_f$  and  $\Delta_c$ , respectively.
    ➤ Obtain  $\rho_i = \frac{|\delta_{\max-f}^i - \delta_{\max-c}^i|}{\delta_{\max-f}^i} \times 100\%$ 
    ➤  $\rho = \rho + \rho_i$ 
    end If
  end for
  • If (mod( $g,t$ ) == 0)
    ➤ If (( $\Delta = \Delta_c$ ) & (75% of population having  $\rho_i \geq 20\%$ )), then  $\Delta = \Delta_f$ 
    ➤ Else if (( $\Delta = \Delta_f$ ) & (75% of population having  $\rho_i < 20\%$ )),
      then  $\Delta = \Delta_c$ 
    ➤ Else  $\Delta$  is left unchanged
    End If
  End While
END ONLINE ADAPTIVE IMORE

```

Figure 7. 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:

$$f(\mathbf{x}) = \sum_{i=1}^m \left(\beta_i \prod_{j=1}^d e^{\frac{-(x_j - \mu_{ij})^2}{2\sigma_i^2}} \right) \quad (15)$$

where $\mathbf{x}=(x_1, x_2, \dots, x_d)$ 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 constructed test functions are listed in Table 2.

Table 2. Parameters used to construct the test functions based on Gaussian basis function as in eq. (15)

Test Function	Centroid [$\mu_1, \mu_2, \dots, \mu_m$]	Standard Deviation [$\sigma_1, \sigma_2, \dots, \sigma_m$]	Magnitude [$\beta_1, \beta_2, \dots, \beta_m$]	Dimensionality
G1 $0 \leq \mathbf{x} \leq 13$, $d_i = 1.0$	[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 $0 \leq \mathbf{x} \leq 10$, $d_i = 0.5$	[(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 $0 \leq \mathbf{x} \leq 10$, $d_i = 0.5$	[(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 $0 \leq \mathbf{x} \leq 10$, $d_i = 0.5$	[(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 [21]. Both the population size and the maximum search generation allowed 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 $\Delta_1=10\%$, $\Delta_2=5\%$, $\Delta_3=3\%$, and $\Delta_4=1\%$. The parameters of the online adaptive IMORE are configured as follows: $\Delta_f=1\%$, $\Delta_c=5\%$, and $t=20$. Each iteration of nested search is set with a maximum computational budget of 400 and 2000 fitness function calls for low (G1, G2, G5) and high (G10) dimensional problems, respectively.

Further, the Pareto front convergence metric (Pc) reported in [22] is used for measuring 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 [23]. 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 calculated using equation (16).

$$d_i = \min_{j=1}^{|P^*|} \sqrt{\sum_{k=1}^m \left(\frac{f^k(i) - f^k(j)}{f_u^k - f_l^k} \right)^2} \quad (16)$$

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 the average of d_i for all design solutions in the final Pareto front F using eq. (17).

$$Pc(F) = \frac{\sum_{i=1}^{|F|} d_i}{|F|} \quad (17)$$

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 $\Delta=1\%$.

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 $\Delta=1\%$ are depicted in Figures 8-11 (a), respectively. Also plotted in Figures 8-11 (c) and (d) are the final Pareto fronts for both the offline and online adaptive IMORE. It can be observed that both the offline and online adaptive IMORE algorithms can converge approximately to the Pareto front obtained by an IMORE with $\Delta=1\%$. It is worth noting that the results obtained by IMORE with $\Delta=5\%$ (from Figure 8-11 (b)) are also presented as an indication of possible false convergences.

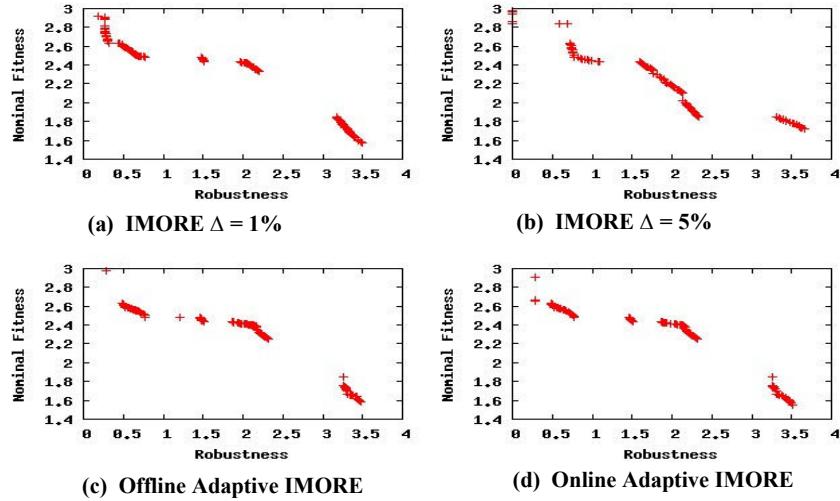


Figure 8. Pareto fronts of test function G1 at the end of the 100th generation using IMORE for $\Delta = 1\%$, $\Delta = 5\%$, offline and online adaptive IMOREs

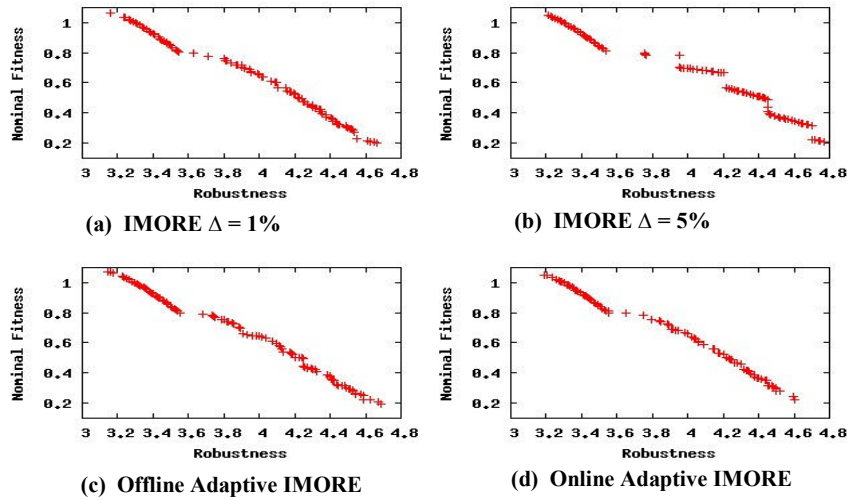


Figure 9. Pareto fronts of test function G2 at the end of the 100th generation using IMORE for $\Delta = 1\%$, $\Delta = 5\%$, offline and online adaptive IMOREs

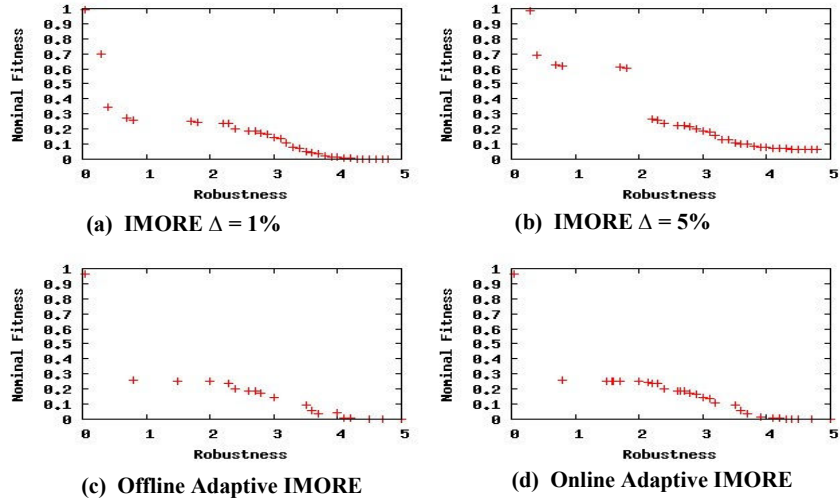


Figure 10. Pareto fronts of test function G5 at the end of the 100th generation using IMORE for $\Delta = 1\%$, $\Delta = 5\%$, offline and online adaptive IMOREs

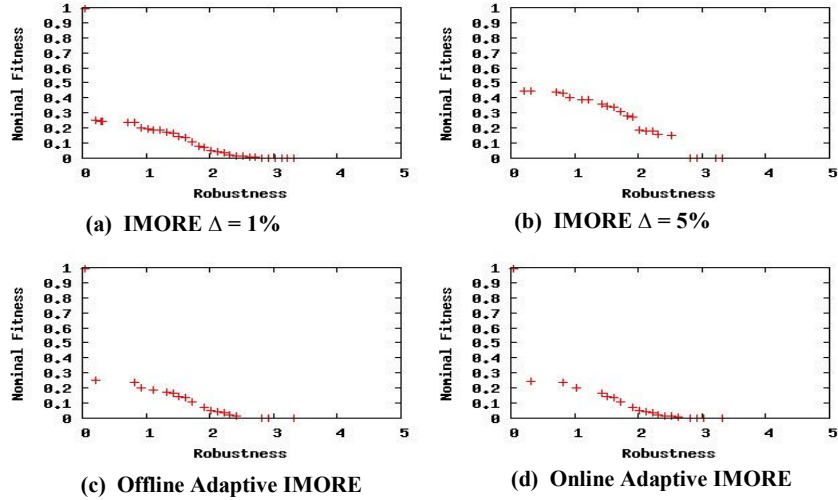


Figure 11. Pareto fronts of test function G10 at the end of the 100th generation using IMORE for $\Delta = 1\%$, $\Delta = 5\%$, offline and online adaptive IMOREs

Figure 12 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 $p \ll k$, it makes good sense that the adaptive IMORE algorithms are significantly faster than its original non-adaptive counterpart. Further, a small value for Pc in Figure 13 also indicates that both offline and online adaptive IMORE provide good convergence to the true optimal Pareto front.

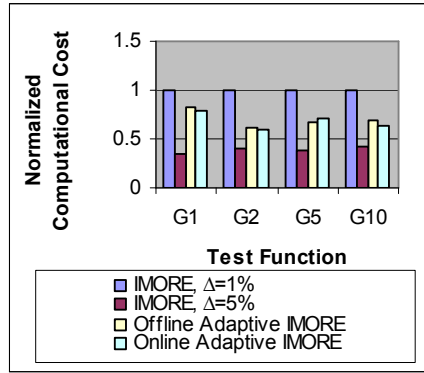


Figure 12. Normalized computational cost incurred for functions G1, G2, G5, and G10 in the IMORE for $\Delta=1\%$, $\Delta=5\%$, offline and online adaptive IMOREs

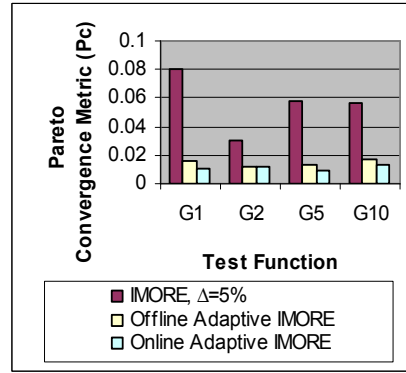


Figure 13. Pareto front convergence metric for functions G2, G5, and G10 in the IMORE for $\Delta=5\%$, offline and online adaptive IMOREs

4.2 Effect of Update Interval Setting in Online Adaptive IMORE

Now, we consider the effect of the step-size update interval, i.e., t in Figure 7, on the computational efficiency of the online adaptive IMORE. The 1) normalized computational cost (with respect to the cost incurred by the IMORE with $\Delta=1\%$), 2) the Pareto front convergence metric for several step-size update intervals, t , are presented in Figures 14 and 15, respectively. These are the average results of 20 independent runs.

It can be observed from Figure 14 that all settings of the update interval considered in the study lead to great savings in computational cost over the IMORE with $\Delta=1\%$. 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 15 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 14 and 15, $t=20$ 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 $\Delta=1\%$ 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 $t=50$, $\Delta_c=5\%$ 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 15, 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 for providing a good balance between computational cost and accurate convergence in the online adaptive IMORE.

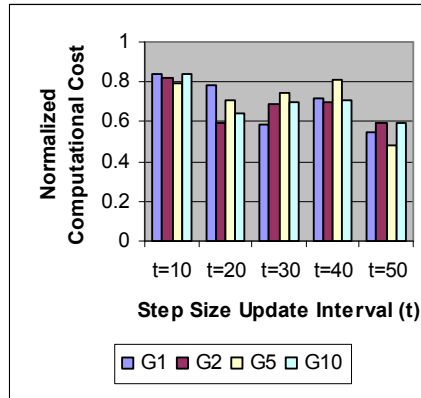


Figure 14. Normalized computational cost incurred for functions G1, G2, G5, and G10 using different t generation intervals in the online adaptive IMORE

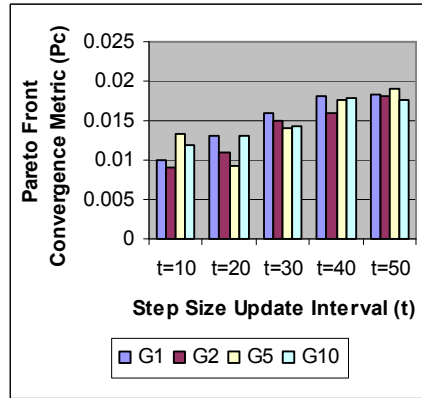


Figure 15. Pareto front convergence metric for functions G1, G2, G5, and G10 using different t generation intervals in the online adaptive IMORE

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

In our efforts 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) sampling approaches including Random Sampling (RS), Stratified Sampling (SS), and Latin Hypercube Sampling (LHS) [24-26] to generate m sampled design points as an ap-

proximation of the worst-case performance for a design in each of the k iterations (please refer to eq. (8)). 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 $m \ll l$, 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 $\Delta=1\%$) and Pareto front convergence metric for the various IMORE algorithms averaged across 20 independent runs are presented in Figures 16 and 17, respectively. The online adaptive IMORE with DOE samplings are labeled here as OAS-RS IMORE, OAS-SS IMORE and OAS-LHS IMORE.

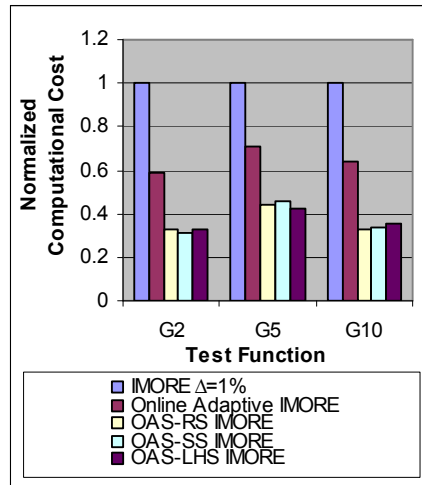


Figure 16. Normalized computational cost incurred for functions G2, G5, and G10 in IMORE, online adaptive IMORE, and OAS IMORE algorithms

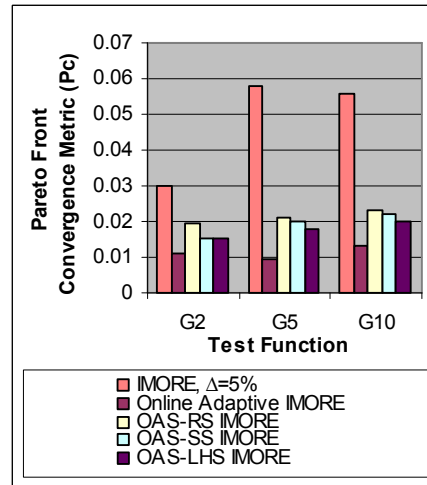


Figure 17. Pareto front convergence metric for functions G2, G5, and G10 in online adaptive and OAS IMORE algorithms

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 17, 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 17 where the Pareto front convergence accuracies, P_c , 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 17. 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 18 and 19, 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 to increment gradually with increasing search generations, and is formulated as follows:

$$m_i = m_{min} + \left[\frac{i}{g_{max}} (m_{max} - m_{min}) \right] \quad (18)$$

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 sizes. The setting for m_{min} , m_{max} , and g_{max} in the experiments are 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 16 and 17 previously. By adapting the sample size m , there is now more than 70% reduction in the computational cost. More importantly, the trade-off of this significant cost saving for convergence accuracy is minimum as shown in Figure 19, even on the higher dimensional G10 problem.

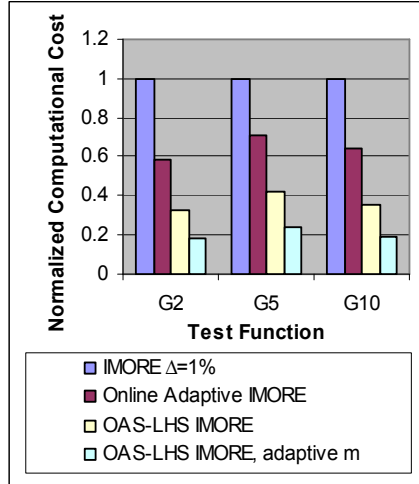


Figure 18. Normalized computational cost incurred for functions G2, G5, and G10 in online adaptive, OAS-LHS with fixed sample size, and OAS-LHS with adaptive sample size IMORE algorithms.

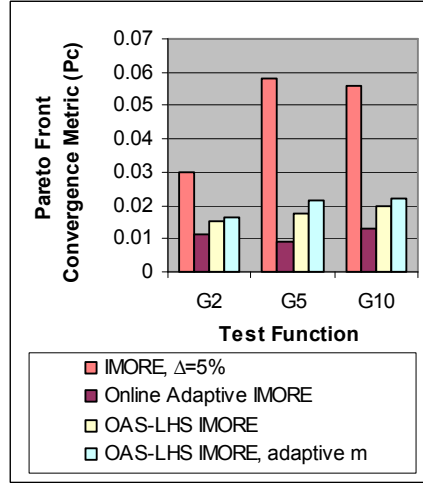


Figure 19. Pareto convergence metric for functions G2, G5, and G10 in online adaptive, OAS-LHS with fixed sample size, and OAS-LHS with adaptive sample size IMORE algorithms

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 the *a priori* information on the desired robustness of the final design, the algorithm has been shown capable of converging to a set of solutions that gives good nominal performances while exhibiting maximum robustness. Most importantly, these solutions were discovered without any requirement to make assumptions about the structure of the uncertainties involved. It is realized that the major drawback of the IMORE would be the massive computational cost that could incur. Hence, adaptation strategies are introduced in the 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 in every sub-problem optimization and trimming down the number of objective function calls by the 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 an enormous number of function evaluations to locate near Pareto optimal solutions, the use of IMORE can become computationally prohibitive for a 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 [27-28] 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 E. Körner at Honda Research Institute Europe, and the Parallel and Distributed Computing Centre of Nanyang Technological University for their support in this work.

References

- [1] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.
- [2] Y. Jin, J. Branke, "Evolutionary Optimization in Uncertain Environment-A Survey," *IEEE Transactions on Evolutionary Computation*, Vol. 9, No. 3, pp. 303-317, June 2005.
- [3] Y.S. Ong, P.B. Nair, K.Y. Lum, "Max-Min Surrogate Assisted Evolutionary Algorithm for Robust Aerodynamic Design," *IEEE Transactions on Evolutionary Computation*, in press, expected August 2006.
- [4] W.L. Oberkampf, et. al., "Estimation of Total Uncertainty in Modeling and Simulation," Sandia Report SAND2000-0824, 2000.
- [5] W.L. Oberkampf, J. Helton, K. Sentz, "Mathematical Representation of Uncertainty", *AIAA Proceedings of Non-Deterministic Approaches Forum*, Paper no. 2001-1645, Reston, VA, 2001.
- [6] Y. Ben-Haim, *Information Gap Decision Theory*, Academic Press: California, 2001.
- [7] Y. Ben-Haim, "Uncertainty, Probability, and Information-Gaps," *Reliability Engineering and System Safety* 85, pp. 249-266, 2004.
- [8] Y. Ben-Haim, *Robust Reliability in Mechanical Sciences*, Springer-Verlag: Berlin, 1996.
- [9] L. Huyse, "Solving Problems of Optimization Under Uncertainty as Statistical Decision Problems", *AIAA-2002-1519*, 2001.
- [10] A. Ben-Tal, and A. Nemirovski, "Robust convex optimization," *Math. Oper. Res.*, 23(4):769-805, 1998.
- [11] S. Tsutsui and A. Ghosh, "Genetic Algorithms with a Robust Solution Searching Scheme," *IEEE Transaction on Evolutionary Computation*, Vol. 1, No. 3, pp. 201-208, 1997.
- [12] D.V. Arnold and H.G. Beyer, "Local Performance of the (1+1)-ES in a Noisy Environment," *IEEE Trans. Evolutionary Computation*, Vol. 6, No. 1, pp 30-41, 2002.
- [13] Y. Jin and B. Sendhoff, "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.
- [14] I. Paenke, J. Branke, and Y. Jin, "Efficient Search for Robust Solutions by Means of Evolutionary Algorithm and Fitness Approximation," *IEEE Transactions on Evolutionary Computation*, 2006, in press, expected August 2006.
- [15] Y.S. Ong, M.H. Lim, N. Zhu and K.W. Wong, "Classification of Adaptive Memetic Algorithms: A Comparative Study," *IEEE Transactions on Systems, Man and Cybernetics - Part B*, Vol. 36, No. 1, February 2006.

- [16] Y.S. Ong and A.J. Keane, "Meta-Lamarckian Learning in Memetic Algorithm," *IEEE Transactions On Evolutionary Computation*, Vol. 8, No. 2, pp. 99-110, April 2004.
- [17] Z.Z. Zhou, Y.S. Ong, P.B. Nair, A.J. Keane and K.Y. Lum, "Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization," *IEEE Transactions On Systems, Man and Cybernetics - Part C*, in press, 2006.
- [18] L. Huyse and R.M. Lewis, "Aerodynamic Shape Optimization of Two-dimensional Airfoils Under Uncertain Operating Conditions," Hampton, Virginia: ICASE NASA Langley Research Centre, 2001.
- [19] D.K. Anthony and A.J. Keane, "Robust Optimal Design of a Lightweight Space Structure Using a Genetic Algorithm," *AIAA Journal* 41(8), pp. 1601-1604, 2003.
- [20] D. Wiesmann, U. Hammel and T. Back, "Robust Design of Multilayer Optical Coatings by Means of Evolutionary Algorithms," *IEEE Trans. Evolutionary Computation*, Vol. 2, No. 4, pp 162-167, 1998.
- [21] N. Srinivas and K. Deb., "Multi-objective Optimization Using Non-dominated Sorting in Genetic Algorithms," *Evolutionary Computation*, Vol. 2, No. 3, pp 221-248, 1994.
- [22] K. Deb, S. Jain, "Running Performance Metrics for Evolutionary Multi-objective Optimization," *Proceeding of the fourth Asia Pacific Conference on Simulated Evolution and Learning*, pp 13-20, 2002.
- [23] C. Grosan, M. Oltean, D. Dumitrescu, "Performance Metrics for Multi-objective Optimization Evolutionary Algorithms," *Proceedings of Conference on Applied and Industrial Mathematics*, Oradea, Romania, 2003.
- [24] K.T. Fang, C.X. Ma, and P. Winker, "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.
- [25] T.W. Simpson, D.K.J. Lin, C. Wei, "Sampling Strategies for Computer Experiments: Design and Analysis," *International Journal of Reliability and Applications*, 2001.
- [26] B. Tang, "Orthogonal Array-based Latin Hypercubes," *Journal of the American Statistical Association*, Vol. 88, No. 424, pp. 1392-1397, 1993.
- [27] Y.S. Ong, P.B. Nair and A.J. Keane, "Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling," *AIAA Journal*, Vol. 41, No. 4, pp 687-696, 2003.
- [28] Y. Jin, "A Comprehensive Survey of Fitness Approximation in Evolutionary Computation," *Soft Computing Journal*, Vol. 9, No. 1, pp 3-12, 2005.