Inverse Multi-Objective Robust Evolutionary Design Optimization in the Presence of Uncertainty

Dudy Lim Yew-Soon Ong Bu-Sung Lee
School of Computer Engineering
Nanyang Technological University
Nanyang Avenue, Singapore 639798
{dlim, asysong, ebslee}@ntu.edu.sg

ABSTRACT
In many real-world design problems, uncertainties are often present and practically impossible to avoid. Many existing works on Evolutionary Algorithm (EA) for handling uncertainty have emphasized on introducing some prior structure of the uncertainty or noise to the variable domain and conducting sensitivity analysis based on the assumed information. In this paper, we present an evolutionary design optimization that handles the presence of uncertainty with respect to the desired robust performance in mind, which we call an inverse robust design. The scheme, unlike others developed to represent uncertainty does not assume any structure of the uncertainty involved; hence it is particularly useful when there is very little information about the uncertainties available. In our 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. Empirical studies were conducted on synthetic functions to demonstrate that our algorithm converges to a set of designs with non-dominated nominal performances and robustness to the presence of uncertainties.

Categories and Subject Descriptors
G.1.6 [**Numerical Analysis**]: Optimization – *global optimization.*

General Terms
Algorithms.

Keywords
Evolutionary Algorithms, robust design optimization, design optimization in the presence of uncertainty.

GECCO'05, June 25–29, 2005, Washington, DC, USA.

1. INTRODUCTION
Uncertainties are often present and practically impossible to avoid in many real world engineering design problems. For instance, if a design is very sensitive to small geometric variations, which may arise either due to manufacturing processes, and/or in-service degradation due to erosion processes and foreign object damage, and/or drifts in 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 but rather potentially high risk designs that are likely to perform badly when put to practical use. Faced with high sensitivities to uncertainties, traditional Evolutionary Algorithms (EAs) [1] tend to display sign of over-searching since they naturally favor designs with a larger fitness value. However, in practice, the preferable design solution is probably one that may not be the globally optimum solution, but one that has a high tolerance or robustness to uncertainties. Solutions whose performances do not change much in the presence of uncertainties are often referred to as *robust designs*.

In recent years, a number of approaches have been proposed in the literature to attain *robust designs*. These include the One-at-a-Time Experiments, Taguchi Orthogonal Arrays, bounds-based, fuzzy and probabilistic methods [2]. In EA, a number of prominent new studies on handling the presence of uncertainty in engineering designs have also been made over the recent years. A noisy phenotype scheme was introduced in [3] where 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. In [4], the study of an (1+1)-Evolutionary Strategy (ES) with isotropic normal mutations using the noisy phenotype scheme has also been reported. In [5-7], uncertainty was regarded in the form of a dynamic environment where the landscape of the problem is perceived to be changing dynamically. In their work, approaches using multi-populations to facilitate exploration and exploitation were also considered. A multi-objective approach to handling uncertainty in EA was also studied in [8] where the trade-off between robustness and nominal performance of a solution was discussed. A strategy to attain robust designs with minimum variations in noise was also presented in [9] on realistic mechanical design problem.

In most of these schemes, often some prior knowledge about the structure of the uncertainties, such as the distribution property involved were assumed to be available. For instance, the uncertainty that exists in the environmental conditions and design parameters of evolutionary search are often assumed to be uniformly or normally distributed, with certain range or standard deviation. 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 many real world engineering design problems, it is often the case that very little knowledge about the structure of the uncertainty involved is available. 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. Thus, it would be wiser for one to avoid making assumptions about the structure in the formulation of the optimization search process. 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 variables, we present evolutionary design optimization that handles the presence of uncertainty in view of the desired robust performance, which we call the inverse robust design. From the desired performance, we search for solutions that guarantee a certain degree of maximum uncertainty and at the same time satisfy the desired nominal performance of the final design solution. For this purpose, we conduct series of nested multi-point local searches using the Sequential Quadratic Programming method [10] within the Non-dominated Sorting Genetic Algorithm (NSGA) [11].

The remaining of this paper is organized as follows: Section 2 provides a brief discussion on evolutionary design optimization in the presence of uncertainties. The proposed algorithm for inverse evolutionary robust design optimization in the presence of uncertainties is presented subsequently in section 3. Section 4 summarizes our empirical study on synthetically generated benchmark functions before section 5 finally concludes this paper.

2. EVOLUTIONARY DESIGN 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:

Maximize: $f(x)$

Subject to: $x_l \le x \le x_u$ (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.

Further, it is noted that the present focus is on EAs for robust engineering design optimization under uncertainties that arise in:

i) design vector x

$$F(x) = f(x+\delta) \quad (2)$$

where $\delta = (\delta_1, \delta_2, ..., \delta_k)$, is the noise in the design vector where some distribution about the uncertainty is assumed and *F(x)* is the effective fitness of design vector *x*.

ii) operating/environmental conditions

$$F(x) = f(x, c+\xi) \quad (3)$$

where $c = (c_1, c_2, ..., c_n)$, is the nominal value of the environmental parameters and ξ is a random vector used to model the variability in the operating conditions. Referring to [8], both forms of uncertainties may be treated equivalently. Hence in this study, we do not differentiate the uncertainties between design variables and operating conditions. Rather the reader is referred to [8] for greater details on the issue. However, as far as this paper is concerned, we consider uncertainties in the design variables.

The core mechanism in many existing robust schemes for solving this type of uncertainty is driven based on the effective fitness *F(x)* of the design solutions. Mathematically, the effective evaluation function $F(x)$ is generally defined as:

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

where $q(\delta)$ is a continuous density function of noise δ which is often assumed to be known *a priori*, usually a Gaussian or uniform distribution.

To locate a robust design solution in the presence of uncertainties in the design vector, one may consider using the *Noisy Phenotype Scheme* proposed in [3] 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*

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

  end for
  - apply mutation and crossover to create new population
  - perform selection of individuals.

end while

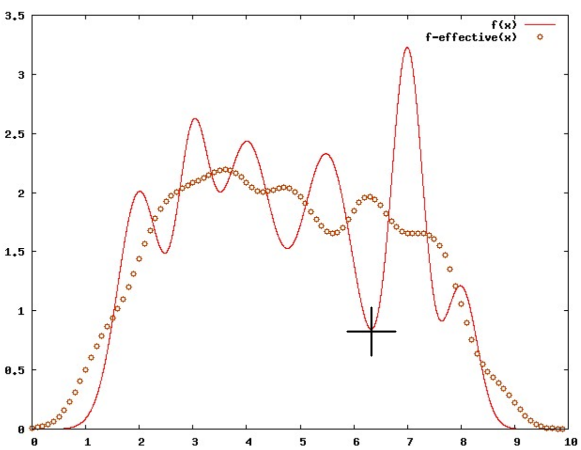
END EA

Figure 1. Noisy Phenotype Scheme.

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

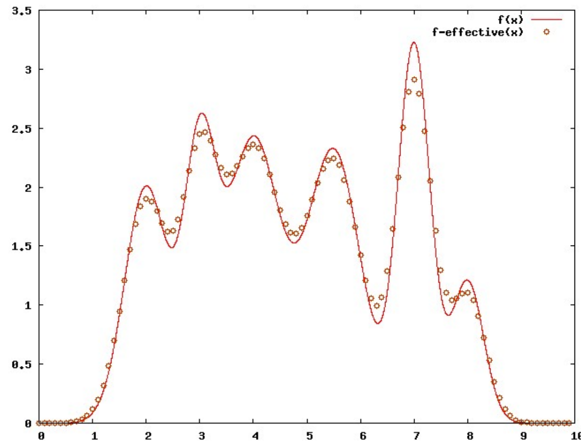
$$f(x) = 2e^{-(x-2)^2/0.32} + 2.2e^{-(x-3)^2/0.18} + 2.4e^{-(x-4)^2/0.5}$$
$$+ 2.3e^{-(x-5.5)^2/0.5} + 3.2e^{-(x-7)^2/0.18} + 1.2e^{-(x-8)^2/0.18}$$
$$\text{where } -1 \le x \le 10 \qquad (5)$$

This represents a multimodal function with a nominal global optimum located at sharp peak $x^* \in [6.5, 7.8]$ and has many other ocal 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, figures 2(a) and (b) illustrates 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 that is assumed about δ. When the range for σ is configured to be ± 1.0, the global robust 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)*. For a complete explanation of how these may be arrived at, the reader is referred to [3, 6].



Input design Variable: *x*

(a) Range of the uncertainty, σ= ± 1.0



Input design Variable: *x*

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

Figure 2. Effective fitness F(x) of the function defined in equation (5) assuming a uniform distribution for δ

In most cases, the algorithm described in Figure 1 is 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 in many realistic problems that very little knowledge about the structure 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 [3, 5, 12]. 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 possess 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 straightforward manner to solve this problem is to reformulate the robustness scheme as a constraint problem with $f(x) \ge c$ and c is the minimum acceptable nominal performance for the final design [13]. This way, any individuals that fail the constraint gets heavily penalized in the robust EA search. This approach however may not be practical since information about the perceived minimum performance may not always be available.

A more promising solution to handle the trade-off between the robustness and nominal fitness is to consider a multi-objective optimization approach [8] where a pareto front of robustness and nominal fitness can be attained. Motivated by this work, we present here an inverse robust solution based on a multi-objective evolutionary approach. In particular, we consider two objective functions, namely the *robustness* and *nominal fitness* of the design.

[1] Note that x^* represents the nominal global optimum.

[2] Note that $x^\wedge$ represents the global effective optimum.

3. INVERSE MULTI-OBJECTIVE ROBUST EVOLUTIONARY DESIGN OPTIMIZATION

To mitigate the problems identified in section 2, we present in this section an inverse multi-objective robust evolutionary 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. While it is common that designers may not possess the necessary expertise or have sufficient knowledge to identify suitable bounds of the uncertainties involve. On the contrary, it is more viable that designers have practical knowledge about the robust performance of the final design it desires.

Here, we present the proposed algorithm for Inverse Multi-Objective Robust Evolutionary design optimization (IMORE) in the presence of uncertainty. 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 [17]. 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 [18-20]. 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 subproblems of the form:

Maximize: $d(x) = f(x_i) - f^k(x)$

subject to: $x_l^k \le x \le x_u^k$ (6)

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 subproblem (or during each k iteration), the optimal solution of the k^{th} subproblem is sought. The objective of each subproblem 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

$$x_l^k = x_i - k\Delta$$

$$x_u^k = x_i + k\Delta \quad (7)$$

BEGIN IMORE (Consider a maximizing problem)

Initialization Phase:

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

Search Phase:

While (termination condition is not satisfied)

For (each individual i in the population)

- Evaluate $f(x_i)$

Repeat

- **Maximize:** $d(x) = f(x_i) - f^k(x)$

  subject to: $x_l^k \le x \le x_u^k$

  $x_l^k = x_i - k\Delta$, $x_u^k = x_i + k\Delta$
- 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_{max}^i using linear interpolation from $d\left(x_{opt}^k\right)$ for different $k\Delta$
- Nominal fitness$(x_i) = f(x_i)$
- Maximum uncertainty$(x_i) = \delta_{max}^i$

end For

- Apply standard MOEA operators to create a new population

end While

END IMORE

Figure 3. Inverse Multi-Objective Robust Evolutionary Design Algorithm in the presence of uncertainty.

It is worth noting that by conducting a sequence of local 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

$$x_l^{k+1} \le x_l^k, x_u^k \le x_u^{k+1} \rightarrow d\left(x_{opt}^k\right) \le d\left(x_{opt}^{k+1}\right) \quad (8)$$

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 is 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, labelled 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} subproblem exceeds the maximum degradation defined, i.e.

$$\left\{d^k(x) = f(x_i) - f(x_{opt}^k)\right\} > d_t \qquad (9)$$

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 standard multi-objective operators to create a new population and terminates upon convergence.

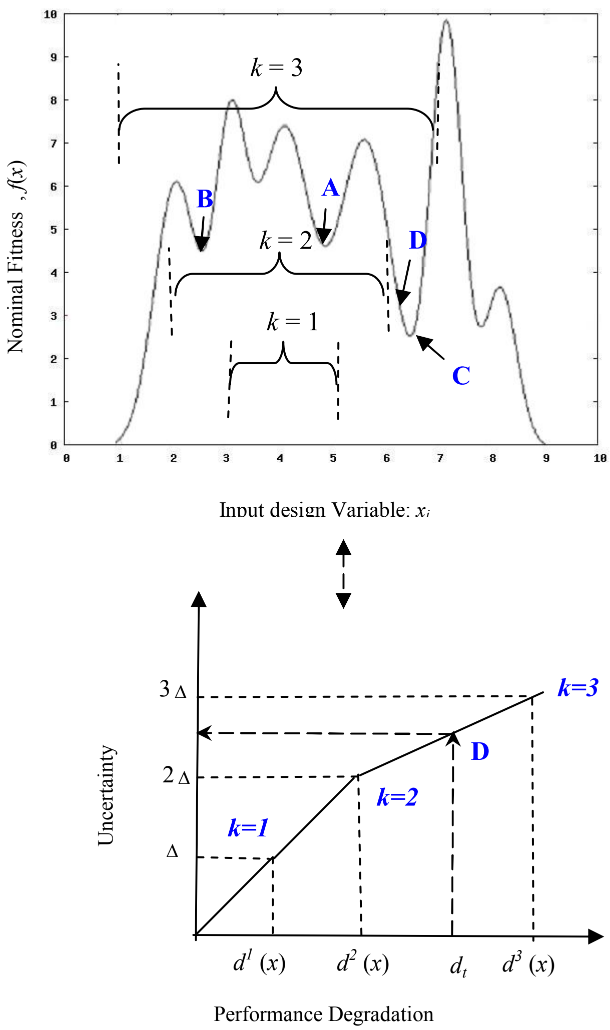


Figure 4. Steps of IMORE for x_i=4 and Δ =1.

4. EMPIRICAL STUDY

To illustrate the utility of the IMORE algorithm described in section 3, we present here an empirical study based on two synthetic one-dimensional multimodal functions. The EA was run for 100 generations with 16 bit binary-coded, linear ranking selection, mutation probability of 0.01, crossover probability of 0.9 and a population size of 100. Further, we consider here a sequence of *multi-start local* bound constrained optimization subproblems to locate the maximum certainty δ_{max} in our IMORE algorithm. In this study, we employ the Feasible Sequential Quadratic Programming (FSQP) as the local search strategy.

Test Function 1. The first test function we consider here is a one-dimensional multimodal function which is an aggregation of multiple one-dimensional Gaussian functions given in equation (10) and depicted in Figure 5. A unique property of this test function is that it contains a mixed of many sharp peaks or noisy near-global optimum solutions and rounded robust peaks in the regions $x \in [0, 4]$ and $x \in [10, 12]$, respectively. Hence it is not a simple task to identify a robust solution among them. This function is defined as:

$$\begin{aligned} f(x) = {} & e^{-(x-1)^2/0.5} + 2e^{-(x-1.25)^2/0.045} + 0.5e^{-(x-1.5)^2/0.0128} \\ & + 2e^{-(x-1.6)^2/0.005} + 2.5e^{-(x-1.8)^2/0.02} + 2.5e^{-(x-2.2)^2/0.02} \\ & + 2e^{-(x-2.4)^2/0.005} + 2e^{-(x-2.75)^2/0.045} + e^{-(x-3)^2/0.5} \\ & + 2e^{-(x-6)^2/0.32} + 2.2e^{(x-7)^2/0.18} + 2.4e^{-(x-8)^2/0.5} \\ & + 2.3e^{-(x-9.5)^2/0.5} + 3.2e^{-(x-11)^2/0.18} + 1.2e^{-(x-12)^2/0.18} \end{aligned}$$

where $-1 \le x \le 13$ (10)

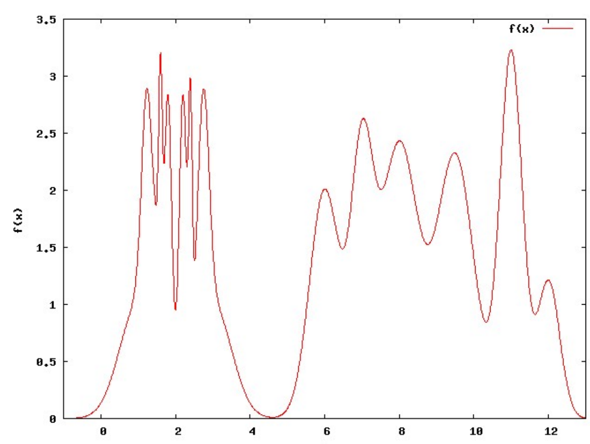


Figure 5. Test function 1.

Test Function 2. The second multimodal test function is based on the one-dimensional "Michalewicz 2" function. This test function contains a mixture of a flat and robust region with moderate nominal fitness for $x \in [-0.5, 0.5]$ and noisy peaks with good nominal fitness for $x \in [0.5, 3]$ as depicted in Figure 6 and is defined as:

$$f(x) = \sum_{i=1}^{10} \left(sin(x) sin^{10}\left(\frac{ix^2}{\pi} \right) \right), \ -1.5 \le x \le 3 \qquad (11)$$

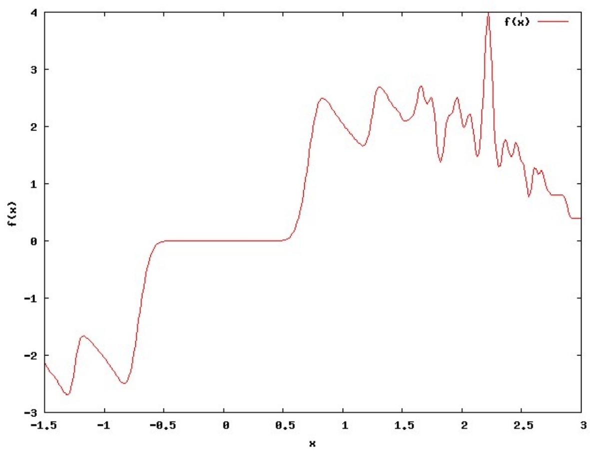


Figure 6. **Test function 2.**

In our IMORE algorithm outlined in Figure 3, it can be observed that besides the standard EA control parameters, we have introduced two additional parameters. These include 1) the maximum degradation permitted, d_t and 2) the step size for local search, Δ. d_t is user-specific and depends on the degree of robustness desired by the designer in the final design. Hence, this leaves us only with the Δ value to consider. To define a suitable value of Δ, we conduct an empirical study on the effect of IMORE for different Δs on the two test functions. In our experimental study, d_t is kept fixed at 1.0. The results obtained from the study are tabulated in Table 1. Here Δ is defined as a percentage of the search bound, *i.e.*, $x_u - x_l$. The average approximated robustness may then be defined by equation (12).

$$\frac{1}{n} \sum_{i=1}^{n} \frac{\delta_{max}^{i}}{x_u - x_l} \times 100\% \qquad (12)$$

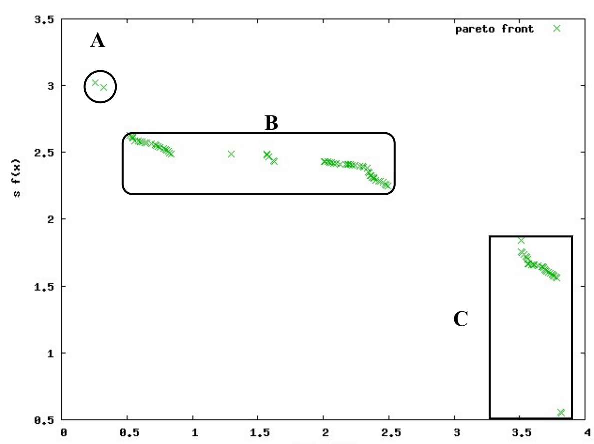
The average exact robustness is defined using the same equation (12), except that δ_{max}^{i} is now the exact robustness. From the results, it is shown that generally the average error increases with the step size, *i.e.*, the accuracy decreases with a larger step size. This makes good sense since a larger step size translates to a larger interpolation error. Like all algorithms, it is crucial to balance the accuracy desired and the computational cost incurred by the nested searches. Since a smaller step size translates to greater iterations of nested searches, *i.e.*, *k*, as a result, more function evaluations are also required. In our IMORE algorithm, it is possible to show that the computational cost as $O(kl)$, if *l* is the average number of function evaluations incurred in a single multi-start local search. Further, we consider the use of Δ =3% in all experimental studies from here onwards since this value offers good accuracy, see Table 1, the % error is lower than 0.25%.

Table 1. Effect of step size Δ in IMORE on test functions 1 and 2.

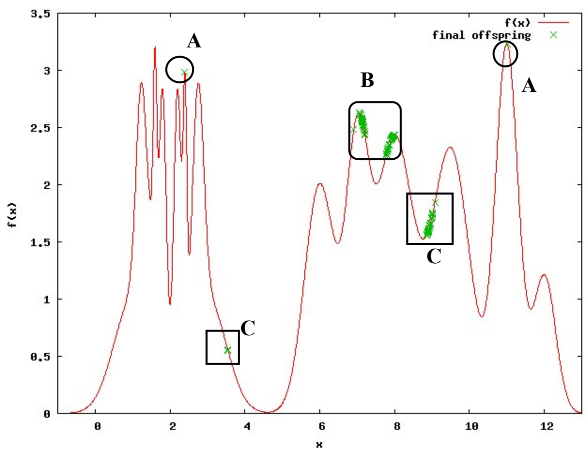
	Step Size Δ (%)	Average Approximated Robustness (%)	Average Exact Robustness (%)	Average Error (%)
Test Function 1	1	16.78	16.79	0.01
	3	*14.11*	*14.20*	*0.09*
	5	16.03	16.46	0.43
	10	14.91	15.36	0.45
Test Function 2	1	5.59	5.60	0.01
	3	*7.16*	*6.95*	*0.21*
	5	5.83	4.91	0.92
	10	9.33	6.13	3.2

Next, we consider the IMORE algorithm for optimization of functions 1 and 2. The pareto fronts obtained from the simulation runs are presented in Figures 7 and 8 for test function 1 and 2, respectively.

The solution in the pareto fronts represents a diverse set of designs having non-dominated nominal performances and robustness to the presence of uncertainties. To explain the results presented in these figures, we cluster the solutions into three separate groups in each pareto front. Group A consists of solutions having excellent nominal fitness at the expense of poor robustness. On the other hand, group B consists of solutions that are a balance trade-off between nominal fitness and robustness, while the solution members of group C have poor nominal fitness but excellent robustness measure.



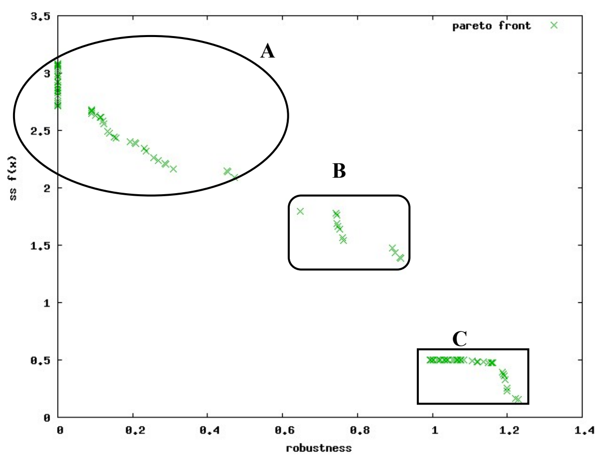
(a)



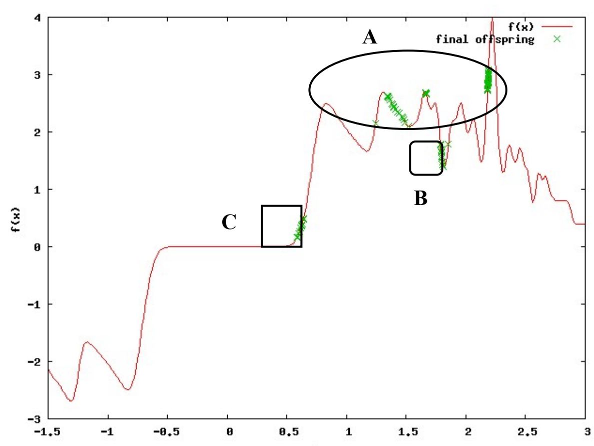
(b)

Figure 7. **(a) Pareto front at generation 100,**

(b) Corresponding offspring in (a) for test function 1.



(a)



(b)

Figure 8. (a). Pareto front at generation 100,

(b). Corresponding offspring in (a) for function 2.

5. CONCLUSION AND FUTURE WORK

In this paper, we have presented a study on inverse multi-objective robust evolutionary design optimization in the presence of uncertainty. 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 in the presence of uncertainties when applied on two synthetic functions. Most importantly, these solutions were discovered without any requirement to make possible untrue assumptions about the structure of the uncertainties involved.

In evolutionary algorithms, many thousands of calls to the objective function are often required to locate a near optimal solution. While the IMORE algorithm proposed offers an effective approach to modeling of uncertainty in engineering design, a compelling limitation of the theory is the massive computational efforts incurred in the nested evolutionary design search. The computational efforts incurred would be even more devastating if the objective function is computationally expensive which is very common in complex engineering design problems [21-22]. Nevertheless, it is worth noting here that a promising and intuitive way to reduce the search time incurred in solving the sequences of bound constrained subproblems is to replace as much as possible the computationally expensive high-fidelity analysis solvers with lower-fidelity models that are computationally less expensive. The reader is referred to [21, 22] for greater details on the algorithm available to achieve this cost savings.

6. ACKNOWLEDGMENTS

The authors would like to thank the Parallel and Distributed Computing Centre at the School of Computer Engineering, Nanyang Technological University for providing support and computing resources to this work.

7. REFERENCES

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

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

[3] 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.

[4] 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.

[5] Branke J., "Creating Robust Solutions by Means of Evolutionary Algorithms", Springer-Verlag Berlin Heidelberg, 1998.

[6] Branke J., "Evolutionary Optimization in Dynamic Environments", Kluwer Academic Publishers, 2002.

[7] Branke J., Kaußler T., Schmidt C., and Schmeck H., "A Multi-Population Approach to Dynamic Optimization

Problems", *Adaptive Computing in Design and Manufacturing*, Springer, 2000.

[8] 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

[9] Chen W., Allen J.K., Tsui K.L., and Mistree F., "A Procedure for Robust Design: Minimizing Variations caused by Noise Factors and Control Factors", *ASME Journal of Mechanical Design*, 118:478-485, 1996.

[10] Lawrence C.T. and Tits A.L., "A Computationally Efficient Feasible Sequential Quadratic Programming Algorithm", *Society for Industrial and Applied Mathematics*, Vol. 11, No. 4, pp. 1092-1118, 2001.

[11] N. Srinivas and K. Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221-248, 1994.

[12] 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.

[13] Michalewicz Z., Dasgupta D., Le Riche R.G., and Schoenauer M., "Evolutionary Algorithms for Constrained Engineering Problems", *Computers & Industrial Engineering Journal*, Vol.30, No.4, pp. 851-870, 1996.

[14] Wiesmann D., Hammel U. and Back T., "Robust Design of A Multilayer Optical Coating by Means of Evolutionary Algorithms", *IEEE Transaction on Evolutionary Computation*, Vol. 2, No. 4, pp. 162-167, 1998.

[15] 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.

[16] Padula S.L. and Li W., "Robust Airfoil Optimization in High Resolution Design Space", Hampton, Virginia: ICASE NASA Langley Research Centre, 2002.

[17] 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.

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

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

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

[21] 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", 2003.

[22] 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.