J Heuristics (2009) 15: 559–595
DOI 10.1007/s10732-008-9077-z

Evolutionary multiobjective optimization in noisy problem environments

Hamidreza Eskandari · Christopher D. Geiger

Received: 19 December 2006 / Revised: 21 February 2008 / Accepted: 22 April 2008 /
Published online: 3 June 2008

Abstract This paper presents a multiobjective evolutionary algorithm (MOEA) capable of handling stochastic objective functions. We extend a previously developed approach to solve multiple objective optimization problems in deterministic environments by incorporating a stochastic nondomination-based solution ranking procedure. In this study, concepts of stochastic dominance and significant dominance are introduced in order to better discriminate among competing solutions. The MOEA is applied to a number of published test problems to assess its robustness and to evaluate its performance relative to NSGA-II. Moreover, a new stopping criterion is proposed, which is based on the convergence velocity of any MOEA to the true Pareto optimal front, even if the exact location of the true front is unknown. This stopping criterion is especially useful in real-world problems, where finding an appropriate point to terminate the search is crucial.

Keywords Multiobjective optimization · Evolutionary algorithms · Stochastic objective function · Pareto optimality

1 Introduction

Gone are the days of only developing optimization models with a lone criterion that is to be minimized or maximized. Almost all practical problems involve more than one objective, and these objectives cannot be reduced to a common single scale such as

H. Eskandari (✉)
School of Industrial Engineering, Tarbiat Modares University, Tehran, Iran
e-mail: eskandari@modares.ac.ir

C.D. Geiger
Department of Industrial Engineering and Management Systems, University of Central Florida, 4000 Central Florida Blvd, Orlando, FL 32816, USA

cost or time. Furthermore, in many real-world cases, the objectives of interest are conflicting in nature, in that improving one objective degrades the value of one or more of the other objectives. This situation has given rise to the need to solve optimization problems with multiple, and often conflicting objectives. Generally, multiobjective optimization problems (MOPs) have the following form:

$$\begin{aligned} &\min(\max) z_i = f_i(\mathbf{x}), \quad i = 1, 2, \ldots, m, \\ &a_k \le x_k \le b_k, \quad k = 1, 2, \ldots, n, \end{aligned} \tag{1}$$

where $f_i(\mathbf{x})$ is vector of m objective functions to be optimized and solution $\mathbf{x}$ is an n-dimensional vector of decision variables that are continuous or discrete or both, *i.e.*, $\mathbf{x} = (x_1, x_2, \ldots, x_n)$.

Most traditional approaches for solving MOPs typically try to convert the MOP into a single objective optimization problem that ultimately transforms the original MOP formulation into a single objective optimization problem. Some researchers and practitioners choose to model one objective, say as a cost or profit objective, and then represent other objectives as constraints. Others choose to represent the multiplicity of objectives as a single composite weighted objective function using a vector of user-defined weights in order to produce a tractable problem. Several drawbacks of using such traditional methods include (Srinivas and Deb 1994; Deb 2001; Coello et al. 2002; Corne et al. 2003): (1) the need for appropriate selection of the weight vector, (2) missing some of the optimal solutions in the nonconvex objective space, (3) inability to easily homogenize different quantities, such as cost, quality and time, to a common unit of measure and (4) ineffective handling of uncertain (noisy) objective functions. These and other known drawbacks to traditional approaches have motivated researchers and practitioners to seek alternative techniques to find a set of Pareto optimal solutions rather than just a single solution. Hence, most multiobjective optimization solution procedures are gravitating towards using nondomination-based approaches due to the limitations of traditional multiobjective methods.

As previously mentioned, an additional complication of addressing real-world problems using traditional MOP solution methods is the degree of complexity in solving such problems; in particular, effectively handling the randomness that exists in many system components, *i.e.*, the occurrence of probabilistic events in real-world systems. In other words, stochasticity must be incorporated as an input to the MOP, thus resulting in random output performance. Therefore, when modeling and solving problems within these stochastic environments, it is necessary to consider the *expectation* (*i.e.*, the mean and variance) of the random output objective function values and not only the mean, as in deterministic environments. Little work has been done in this important area of multiobjective optimization. This undoubtedly is due to the complexities of modeling and simultaneously assessing multiple, uncertain objective function values. Undoubtedly, a multiobjective optimization algorithm capable of finding diverse set of Pareto optima and handling the uncertainty of multiple, stochastic objective functions would be greatly beneficial in solving MOPs in research and in practice. Hence, the purpose of this research is to propose such an approach that finds an evenly-distributed set of Pareto optima in a computationally-efficient manner.

In this paper, we extend a previously developed multiobjective evolutionary algorithm (MOEA) approach to rapidly converge to the true Pareto optimal front and generate a set of uniformly-distributed Pareto optima (Eskandari and Geiger 2008). Our previous work proposes a new MOEA, named *fast Pareto Genetic Algorithm* (FastPGA). We show that FastPGA quickly converges to the true Pareto front for deterministic problems, where time and resources are not a luxury and a small number of solution evaluations is desired. In this paper, we modify FastPGA and apply the new MOEA to stochastic problems.

The remaining of this paper is organized as follows. Section 2 summarizes previous work related to multiobjective evolutionary algorithms applied to stochastic optimization problems. In Sect. 3, we discuss new concepts of solution dominance for stochastic problem environments. These concepts are necessary in order to better discriminate among solutions in stochastic problem environments. Section 4 describes the proposed multiobjective evolutionary algorithm that incorporates the solution dominance concepts into its solution ranking scheme. Section 5 presents the computational study followed by the computational results and findings discussed in Sect. 6. Finally, we conclude this paper and discuss future research in Sect. 7.

2 Related work on multiobjective evolutionary algorithms under uncertainty

Several variations of multiobjective evolutionary algorithms have been developed to handle MOPs including Pareto-archived evolution strategy (PAES) of Knowles and Corne (1999), an improved version of strength Pareto EA (SPEA2) of Zitzler et al. (2001), an improved version of the nondominated sorting genetic algorithm (NSGA-II) of Deb et al. (2002), ParEGO of Knowles (2006), and FastPGA of Eskandari and Geiger (2008). Careful review of the literature reveals that only a few attempts have been made in the area of multiobjective evolutionary algorithms in stochastic environments. It is worth mentioning that several studies concerning single objective evolutionary algorithms in stochastic environments have been reported (Beyer 2000; Jin and Branke 2005). For example, Jin and Branke (2005) review and discuss the research on evolutionary optimization in the presence of uncertainties. They consider four classes of uncertainties, namely, noise in fitness functions, search for robust solutions, approximation error in the fitness function, and fitness function changing over time.

In MOPs, a few approaches have been proposed to reduce the disturbance of noise including probabilistic Pareto ranking (Hughes 2001; Teich 2001), re-evaluation of archived solutions (Buche et al. 2002), Bayesian algorithm (Fieldsend and Everson 2005), and extended averaging scheme (Singh 2003). Hughes (2001) presents a new approach for probabilistic ranking and selection for both single objective and multi-objective optimization problems accounting for uncertainties and noise present in the objective functions. Unlike the conventional ranking processes, his approach provides a statistical basis for addressing uncertainties and noise in the ranking and selection process. This study investigates how noise affects the assigned ranks within a population of solutions of an EA and finds that the proposed probabilistic ranking process outperforms the ranking processes of MOGA and NSGA in the presence of high levels of noise. Further research to employ the suggested probabilistic ranking approach in evolutionary algorithms has not been found. Teich (2001) introduces the concept of probabilistic dominance in multiobjective evolutionary algorithms when the objective values are uncertain, but constrained within certain intervals. This is an extension to the definition of Pareto-based dominance. He modifies SPEA by updating the external set in order to handle estimated objective values bounded by intervals and then applies the modified SPEA to a hardware/software partitioning problem in order to minimize execution time and cost. Both Hughes (2001) and Teich (2001) make some assumptions on the probability distribution of objective function values.

Buche et al. (2002) propose the MOEA approach based on the SPEA of Zitzler and Thiele, called noise-tolerant SPEA (NT-SPEA), capable of reducing the disturbance of noise and outliers. They incorporate three features into the NT-SPEA including domination dependent lifetime, re-evaluation of solutions and modifications in the update of the archive population. NT-SPEA is applied to the optimization of the combustion process of a stationary gas turbine in an industrial setup for simultaneous minimization of NO_x emissions and pressure fluctuations of the flame. Fieldsend and Everson (2005) introduce a Bayesian algorithm for learning the noise variance during the optimization which is used for estimating the probability of dominance in the noisy MOPs.

A large body of MOEA research focuses on algorithms that are modifications of NSGA-II (Deb et al. 2002). Singh and Minsker (2004) modified NSGA-II to deal with noisy MOPs using hypothesis test based on a student-t distribution to decide which solution is statistically dominant. They suggest if the test is inconclusive, both solutions are statistically indifferent and the solution with lower standard deviation is preferred, since it is more reliable. They applied their proposed approach to the groundwater remediation at the Umatilla Chemical Depot field. Babbar et al. (2003) suggest several modifications to the original NSGA-II ranking scheme to improve the performance of the algorithm in noisy environments. They test the modified NSGA-II on two test problems. It is worth mentioning that, in order to make a fair comparison, only real nondominated solutions rather than rank-1 frontiers at the final generation should be benchmarked. Joines et al. (2002) introduce a GA-based multiobjective simulation optimization approach using a modified version of the original NSGA-II. They apply their methodology to a real-world supply chain optimization problem with two objectives, namely, gross margin return on investment and customer service level. They find Pareto optimal solutions for different levels of customer service, which provides valuable information for the decision-maker.

Bui et al. (2004) performed comparative analyses between NSGA-II and SPEA2 in the presence of different levels of noise. They report that SPEA2 outperforms NSGA-II in the early generations whereas NSGA-II is superior during later generations irrespective of the noise level involved in the problem. In another study, Bui et al. (2005) compares the performance of five different variations of NSGA-II based upon probabilistic and resampling methods on ZDT family problems. Their experimental results indicate that resampling approach usually offers better performance in comparison to the probabilistic approach in the context of NSGA-II. Poles et al. (2004) propose a new EA for multiobjective optimization, called MOGA-II, which is different from the MOGA of Fonseca and Flemming (1993). They test the robustness of MOGA-II on noisy single-objective problems and compare its performance to that of two other algorithms.

Basseur and Zitzler (2006) propose a method for handling noise in indicator-based EA. Their approach makes no assumption on the properties of the noise including the associated distribution, bounds and general tendency. Lim et al. (2005) present an evolutionary design optimization, called inverse robust design, which handles uncertainty with respect to the desired robust performance. Similar to the work of Basser and Zitzler, inverse robust design does not assume any structure for the involved uncertainty. More recently, Goh and Tan (2007) performed extensive studies to investigate the impact of noise on the performance of a number of MOEAs. They propose three features for handling the noise including experiential learning directed perturbation, gene adaptation selection strategy and a possibilistic archiving methodology. Their comparative analysis indicates that the proposed features improve the performance of the algorithm in terms of proximity, diversity and distribution of nondominated solutions. Finally, Liefooghe et al. (2007) propose several stochastic methods to handle any type of uncertainty involved in the objective function. They applied their proposed methods to a bi-objective flow-shop scheduling problem.

It can be seen that the previous MOEA work that addresses stochastic multiobjective optimization problems is relatively limited. Moreover, to the best of our knowledge, no existing multiobjective evolutionary algorithm that is capable of effectively dealing with uncertain (noisy) objective functions has been extensively tested on a suite of test problems. One of the aims of this research is to propose an evolutionary algorithm for multiobjective optimization problems that effectively handles the existence of noisy objective function values. The proposed solution is similar to Probabilistic Pareto ranking approach suggested by Hughes, but a more sophisticated statistical differentiation among competing solutions is suggested assuming that objective function values a normal distribution structure. Prior to presenting the proposed MOEA, we first must introduce and discuss new solution dominance concepts that are needed for better discrimination among stochastic objective function values.

3 Solution dominance in stochastic multiobjective optimization problems

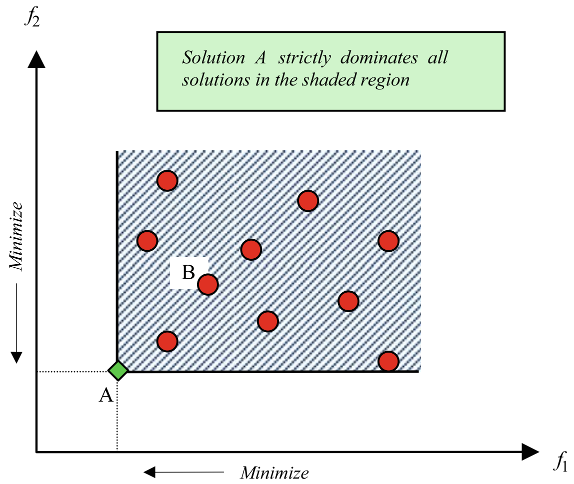
To begin, let us assume that $f_i(\mathbf{A})$ and $f_i(\mathbf{B})$ are the values of objective function i ($i \in \{1, \ldots, m\}$) for two solutions **A** and **B**, where **A** and **B** are n-dimensional vectors of the decision variables. The desire is to minimize each objective function. In a deterministic problem domain, solution **A** is said to *strictly dominate* (be better than) solution **B** if $f_i(\mathbf{A})$ is less than $f_i(\mathbf{B})$ for each objective function i. Figure 1 illustrates the concept of strict dominance graphically for an optimization problem in which $m = 2$, and the goal is to minimize both functions f_1 and f_2. In the figure, it can be seen that solution **A** strictly dominates all solutions in the shaded region, including solution **B**.

In a stochastic (or noisy) multiobjective problem domain, we modify Eq. 1 to

$$\min(\max)\mu_i = E[f_{ij}(\mathbf{x})], \quad i = 1, 2, \ldots, m \text{ and } j = 1, 2, \ldots, p, \tag{2}$$

where $f_{ij}(\mathbf{x})$ is an observation corresponding to the ith objective from the jth random observation of vector $\mathbf{x}$. Thus, the strict dominance definition must be modified for stochastic multiobjective problem environments in which the objective functions

Fig. 1 Illustration of strict dominance in a deterministic problem domain



do not take on certain values but they are described with the expected values and variances (or half-widths). Assume that $f_i(\mathbf{A})$ and $f_i(\mathbf{B})$ are the stochastic outputs of two solutions **A** and **B** for ith objective function, which are output random variables due to the stochastic nature of the underlying problem. We assume that $f_{ij}(\mathbf{A})$ and $f_{ij}(\mathbf{B})$ are unbiased estimates of the ith true objective values $\mu_i(\mathbf{A})$ and $\mu_i(\mathbf{B})$, that is, $\mu_i(\mathbf{A}) = E[f_{ij}(\mathbf{A})]$ and $\mu_i(\mathbf{B}) = E[f_{ij}(\mathbf{B})]$.

In many stochastic modeling problems, it is reasonable to assume that the objective values of solutions are approximately normally-distributed. Suppose that the statistical pairs $\overline{f}_i(\mathbf{A})$, $s^2_{\mathbf{A}}$ and $\overline{f}_i(\mathbf{B})$, $s^2_{\mathbf{B}}$ are the expected values (sample averages) and variances of each objective function i for two solutions **A** and **B**, respectively. The objective function expected values and variances are calculated after a number of independent function evaluations p. Half-widths of confidence intervals of solutions **A** and **B** are calculated by

$$hw_i(\mathbf{A}) = t_{1-\alpha/2,p-1}\frac{s_{\mathbf{A}}}{\sqrt{p}} \quad \text{and} \quad hw_i(\mathbf{B}) = t_{1-\alpha/2,p-1}\frac{s_{\mathbf{B}}}{\sqrt{p}},$$

where α is the significance level ($0 \le \alpha \le 1$) and parameter $t_{1-\alpha/2,p-1}$ is the critical value for t-distribution based on $p-1$ degrees of freedom. Now, it is assumed that each objective function f_i has truncated normal distribution and is represented by its confidence interval $[\overline{f}_i - hw_i, \overline{f}_i + hw_i]$, where $(\overline{f}_i - hw_i)$ and $(\overline{f}_i + hw_i)$ are the lower and upper bounds of the interval at significance level α, respectively. Given the fact that in the stochastic models, objective functions are random and take on uncertain values, new definitions to compare two different solutions are proposed.

Definition 1 Solution **A** probabilistically dominates solution **B** with a probability of $\prod_{i=1}^{m} P(\overline{f}_i(\mathbf{A}) < \overline{f}_i(\mathbf{B}))$ if $(\overline{f}_i(\mathbf{A}) - hw_i(\mathbf{A})) < (\overline{f}_i(\mathbf{B}) + hw_i(\mathbf{B}))$ for each objective function ($i \in \{1, \dots, m\}$).

This definition is very similar to the *probabilistic dominance* definition introduced in Hughes (2001) and Teich (2001). The definition is slightly modified here by substituting truncated normal distribution represented by its confidence interval for general random distribution. Due to the uncertainty surrounding the objective function values, it is not certain that solution **A** strictly dominates solution **B**. As a result, the strict dominance definition is modified to account for this uncertainty. Further, in stochastic environments, it is necessary to know if there is a significant difference between two solutions. The following revised definition is proposed.

Definition 2 Solution **A** significantly dominates (is better than) solution **B** with a confidence level of about $(1 - m\alpha)$ if $\overline{f}_i(\mathbf{A}) + hw_i(\mathbf{A}) < \overline{f}_i(\mathbf{B}) - hw_i(\mathbf{B})$ for each objective function i $(i \in \{1, \ldots, m\})$.

If two solutions **A** and **B** with their corresponding confidence intervals are compared, three different cases can occur for calculating the probability that solution **A** dominates solution **B**, *i.e.*, $P(\mathbf{A} \succ \mathbf{B})$. First, solution **A** does not dominate solution **B** when at least one lower bound of the solution **A** confidence interval is larger than the corresponding upper bound of solution **B**. Second, solution **A** significantly dominates solution **B** when all upper bounds of the solution **A** confidence interval are less than the corresponding upper bound of solution **B**. In the third case, solution **A** probabilistically dominates solution **B** with a certain probability when all lower bounds of the solution **A** confidence intervals are less than the corresponding upper bounds of solution **B**. Therefore, the probability that solution **A** dominates solution **B** is given by

$$P(\mathbf{A} \succ \mathbf{B}) = \begin{cases} 0, & \text{if } \exists i : \overline{f}_i(\mathbf{A}) - hw_i(\mathbf{A}) \\ & \qquad > \overline{f}_i(\mathbf{B}) + hw_i(\mathbf{B}), \\ 1, & \text{if } \forall i : \overline{f}_i(\mathbf{A}) + hw_i(\mathbf{A}) \\ & \qquad < \overline{f}_i(\mathbf{B}) - hw_i(\mathbf{B}), \\ \prod_{i=1}^{m} P(\overline{f}_i(\mathbf{A}) < \overline{f}_i(\mathbf{B})), & \text{otherwise.} \end{cases} \tag{3}$$

Now, assuming that each objective function f_i follows a normal distribution with a known mean and variance, the question is how to calculate the probability $P(\overline{f}_i(\mathbf{A}) < \overline{f}_i(\mathbf{B}))$. If x and y are independent random variables, it can be proved that

$$P(x < y) = \int_{-\infty}^{\infty} f_x(t) F_y(t) dt, \tag{4}$$

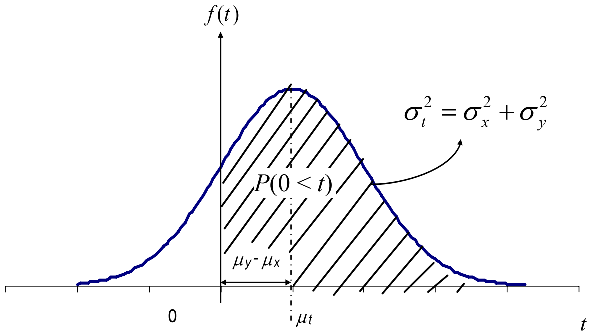
where $f_x(t)$ and $F_y(t)$ are probability density function of variable x and cumulative density function of variable y, respectively.

According to Eq. 4, we get

$$P(x < y) = \int_{-\infty}^{\infty} \left(\frac{1}{\sigma_x \sqrt{2\pi}} e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}} \int_{-\infty}^{x} \frac{1}{\sigma_y \sqrt{2\pi}} e^{-\frac{(y-\mu_y)^2}{2\sigma_y^2}} dy \right) dx. \tag{5}$$

Equation 5 does not have a closed-form expression. Therefore, an alternative approach is suggested knowing that the difference between two independent normal distributions is also normal distribution.

Fig. 2 Plot of normally-distributed random variable t



Theorem 1 *If x and y are independent normal random variables with means μ_x and μ_y $(\mu_x < \mu_y)$, and variances σ_x^2 and σ_y^2, the probability*

$$P(x < y) = 1 - Q\left(\frac{\mu_y - \mu_x}{\sqrt{\sigma_x^2 + \sigma_y^2}}\right),$$

where the Q-function is defined by $Q(x) = 1 - \Phi(x) = 1/\sqrt{2\pi} \int_x^\infty e^{-t^2} d_t$.

Proof If x and y are independent normal random variables with means μ_x and μ_y and variances σ_x^2 and σ_y^2, the probability of x being less than y is $P(x < y) = P(0 < y - x)$. Now, assuming $\mu_x < \mu_y$, the change $t = y - x$ results in $P(x < y) = P(0 < t)$, where t is a normal random variable with mean $\mu_t = \mu_y - \mu_x$ and variance $\sigma_t^2 = \sigma_x^2 + \sigma_y^2$, as shown in Fig. 2.

Now, the probability of $P(x < y) = P(0 < t)$ is

$$P(0 < t) = Q\left(\frac{0 - \mu_t}{\sigma_t}\right) = Q\left(\frac{-(\mu_y - \mu_x)}{\sqrt{\sigma_x^2 + \sigma_y^2}}\right).$$

Since $Q(-x) = 1 - Q(x)$, then

$$P(0 < t) = 1 - Q\left(\frac{\mu_y - \mu_x}{\sqrt{\sigma_x^2 + \sigma_y^2}}\right).$$

This completes the proof. □

The integral described for $Q(x)$ does not have a closed-form expression. However, an excellent closed-form approximation is suggested by Borjesson and Sundberg (1979) to estimate $Q(x)$ with an acceptable error.

It is interesting to note that although two new definitions for dominance have been suggested, it is still difficult to discriminate which solutions should be considered as nondominated at any generation. The following definition helps better identification of nondominated solutions.

Definition 3 Solution **A** stochastically dominates (is better than) solution **B** if $\overline{f}_i(\mathbf{A})$ is less than $\overline{f}_i(\mathbf{B})$ for each objective function i $(i \in \{1, \ldots, m\})$.

It is clear that if solution **A** stochastically dominates solution **B** based on the finite samples, $P(\mathbf{A} \succ \mathbf{B})$ is larger than $P(\mathbf{B} \succ \mathbf{A})$. This implies that the expectation that solution **A** is a nondominated solution in any given generation is higher that of solution **B**.

4 The stochastic Pareto genetic algorithm (SPGA)

In this paper, we extend a previously developed MOEA to address multiobjective optimization problems when objective values are uncertain. Our previous work (Eskandari and Geiger 2008) presents an MOEA named *fast Pareto Genetic Algorithm* (FastPGA) that solves computationally- and/or financially-expensive MOPs where a large number of solution evaluations is not allowed or even possible. In addition to employing specialized genetic operators that improve the MOEA's convergence behavior, FastPGA uses an intelligent population sizing strategy that dynamically adjusts the population size.

The performance of FastPGA is evaluated by comparing it against NSGA-II, considered the most widely-accepted MOEA benchmark. The results show that for a number of published test problems, FastPGA outperforms NSGA-II within a small number of solution evaluations according to (1) fast convergence to the true Pareto optimal front in the objective space, (2) close proximity to the true Pareto optimal front, and (3) diversity and even dispersion of the obtained nondominated solutions along the true Pareto optimal front. In this paper, we enhance FastPGA such that it can be applied to MOPs with noisy objective functions (*i.e.*, uncertain objective values). First, we briefly describe FastPGA. For a more detailed discussion of FastPGA, we refer the interested reader to the work of Eskandari and Geiger (2008).

4.1 Overview of the fast Pareto genetic algorithm

The logic of FastPGA is somewhat similar to the logic of other existing MOEAs except for the incorporation of a number of innovative features that improve its performance (see Fig. 3), in particular, the solution ranking strategy and a dynamic population size adjustment mechanism. Similar to other existing MOEAs, the initial population of solutions in FastPGA is created by randomly sampling each decision variable x_k within its specified range of variation. The solutions in the initial and subsequent populations are evaluated in terms of the objective functions given for each MOP.

In FastPGA, before fitness values are assigned to individual solutions of a population, a set of solutions $\mathbf{O}_t$ generated from the previous population $\mathbf{P}_{t-1}$ using crossover and mutation operations are combined with previous population $\mathbf{P}_{t-1}$ to form the composite population $\mathbf{CP}_t$, *i.e.*, $\mathbf{CP}_t = \mathbf{P}_{t-1} \cup \mathbf{O}_t$. FastPGA then divides the composite population of solutions into two ranks according to their solution dominance. At generation $t = 0$, the solutions are evaluated but a composite population is not created since there is no previous population. The first composite population is

Fig. 3 Logic of the fast Pareto genetic algorithm (Eskandari and Geiger 2008)

```
Initialize user decision parameters
t := 0
create initial random population P_t = {x_1^t, x_2^t, x_3^t, ...}
evaluate(P_t)
do while (stopping criterion is not met)
{
    t := t + 1
    P'_t := select(P_t-1) // select pairs of solutions for reproduction
    O_t := crossover(P'_t)
    O_t := mutate(O_t)
    evaluate(O_t)
    CP_t := P_t-1 ∪ O_t // form composite population
    rank(CP_t)
    regulate(CP_t)
    P_t := generate(CP_t)
}end do
```

created at $t = 1$. The first rank is comprised of all nondominated solutions, and the second rank is comprised of all dominated solutions. These ranks are then used to assign the individual fitness values for subsequent solution reproduction.

The fitness of the nondominated solutions in the first rank is calculated by comparing each nondominated solution with one another and assigning a fitness value, which is computed using the crowding distance approach suggested by Deb et al. (2002). Each dominated solution in the second rank is compared to all other solutions in the composite population and assigned a fitness value depending on the number of solutions it dominates and the number of solutions by which it is dominated. More specifically, the fitness assignment takes into account both dominating and dominated solutions for any dominated solution. Here, each solution in the composite population $\mathbf{CP}_t$ is assigned a strength value $S(\mathbf{x}_i)$, indicating the number of solutions it dominates, where

$$S(\mathbf{x}_i) = \left|\left\{\mathbf{x}_j \mid \forall \mathbf{x}_j \in \mathbf{CP}_t \wedge \mathbf{x}_i \succ \mathbf{x}_j \wedge j \neq i\right\}\right|. \tag{6}$$

The expression $\mathbf{x}_i \succ \mathbf{x}_j$ means solution $\mathbf{x}_i$ dominates solution $\mathbf{x}_j$. Then, the fitness value of each dominated solution is calculated by

$$F(\mathbf{x}_i) = \sum_{\mathbf{x}_i \succ \mathbf{x}_j} S(\mathbf{x}_j) - \sum_{\mathbf{x}_k \succ \mathbf{x}_i} S(\mathbf{x}_k), \quad \forall \mathbf{x}_j, \mathbf{x}_k \in \mathbf{CP}_t \wedge j \neq i \neq k. \tag{7}$$

The fitness value of each dominated solution $\mathbf{x}_i$ is equal to the summation of the strength values of all solutions it dominates minus the summation of the strength values of all solutions by which it is dominated. If most solutions do not dominate one another, it is implied that they belong to the first rank. In this case, the crowding distance operator is invoked to maintain the diversity among them.

The combination of previous generation $\mathbf{P}_{t-1}$ with generated offspring $\mathbf{O}_t$ provides an opportunity to preserve the superior solutions in the next generation and discard the inferior solutions. The number of inferior solutions discarded depends on the number of nondominated solutions obtained in the composite population $\mathbf{CP}_t$.

The number of nondominated solutions usually increases as the number of generations increases, resulting in low elitism intensity in early generations if the population size is quite large and kept fixed. Therefore, FastPGA employs a population regulation operator to dynamically adjust the size of the population to place an appropriate emphasis of elitism intensity on the nondominated solutions. FastPGA adjust the population size until it reaches a user-specified maximum. The success of FastPGA is in part due to the dynamic adjust of the population size and the conservative generation of offspring at each generation.

4.2 Ranking and fitness assignment in SPGA

The proposed MOEA, SPGA, ranks the solutions in each composite population according to stochastic dominance concepts discussed in Sect. 3 for the purpose of reproduction. First, all *stochastically nondominated* solutions are identified as the first rank, which implies that there is no solution that is stochastically better than these solutions with respect to all objectives simultaneously. All *stochastically dominated*, which comprise the second rank, are assigned an expected strength value $E[S(\mathbf{x}_i)]$, indicating the summation of the probabilities that it dominates other solutions. Equation 6 of FastPGA is modified as follows

$$E[S(\mathbf{x}_i)] = \sum_j P(\mathbf{x}_i \succ \mathbf{x}_j), \quad \forall \mathbf{x}_j \in \mathbf{CP}_t \wedge j \neq i. \tag{8}$$

The expression $\mathbf{x}_i \succ \mathbf{x}_j$ represents that solution $\mathbf{x}_i$ dominates solution $\mathbf{x}_j$. Recall that the probability $P(\mathbf{x}_i \succ \mathbf{x}_j)$ is estimated using Eq. 3. Equation 7 is then modified to compute the expected fitness value of each dominated solution as

$$E[F(\mathbf{x}_i)] = \sum_{\mathbf{x}_i > \mathbf{x}_j} E[S(\mathbf{x}_j)] - \sum_{\mathbf{x}_j > \mathbf{x}_i} E[S(\mathbf{x}_j)], \quad \forall \mathbf{x}_j \in \mathbf{CP}_t \wedge j \neq i, \tag{9}$$

where the $\mathbf{x}_i > \mathbf{x}_j$ denotes that solution $\mathbf{x}_i$ stochastically dominates solution $\mathbf{x}_j$. In other words, a fitness value is assigned to each dominated solution $\mathbf{x}_i$ is equal to the summation of the expected strength values of all solutions it stochastically dominates minus the summation of the expected strength values of all solutions by which it is stochastically dominated. Similar to FastPGA, if most solutions do not dominate one another, it is implied that they belong to the first rank and the crowding distance operator is invoked to maintain the solution diversity.

4.3 Proposed MOEA search stopping criterion—hypervolume growth rate

Different approaches have been used to stop the search process of EAs including those that consider the desired solution quality, the specific number of solution evaluations, the required computation time, and the prespecified convergence behavior (Safe et al. 2004). Originally suggested for stopping the search in a single objective evolutionary algorithm, a new stopping criterion is introduced here that considers the convergence speed towards the true Pareto optimal front while dealing with expensive, real-world optimization problems.

Here, when the hypervolume indicator (Zitzler and Thiele 1999) with the reference point of the worst objective vector enclosed by the obtained nondominated solutions does not make significant improvement within a certain number of solution evaluations,[1] called evaluation period, the search stops. For a better understanding of the proposed stopping criterion for real-world MOPs, a hypervolume growth rate is defined.

Definition 4 The hypervolume growth rate (*HGR*) is the difference of the hypervolumes at any given iterations t_1 and t_2 and is calculated as

$$HGR_t = HV_{t_2} - HV_{t_1}, \quad t_2 > t_1$$

where HV_{t_i} denotes the obtained hypervolume with the reference point of the worst objective vector at iteration t_i. The evaluation period, *i.e.*, $t_2 - t_1$, is kept fixed so that consistent hypervolume growth rate (HGR_t) is extracted as the search process proceeds. If the MOEA employs an appropriate elitism scheme, the hypervolume HV_{t_i} increases when the search proceeds. However, the hypervolume HV_{t_i} could increase until the certain point, *i.e.*, hypervolume enclosed by the true Pareto optimal front. It is expected that the hypervolume growth rate HGR_t has a decreasing function over time, but not necessarily monotonically decreasing. When HGR_t becomes less than a user-specified value implying no promising nondominated solutions are found within the evaluation period, the search stops. Careful selection of evaluation period and small HGR_t value, say HGR_{stop}, is necessary to ensure that not only a sufficient number of solution evaluations are assigned for desired convergence but also reaching to the desired HGR_{stop} value is possible within an acceptable number of solution evaluations. The evaluation period should be properly determined depending on the features of the problem under study including number of decision variables, the decision variable domains, the number of objectives, and so on.

This new stopping criterion has a few advantages over many of the existing stopping criteria, particularly when solving MOPs where each solution evaluation is computationally- and/or financially-expensive and the solution time is restricted. Firstly, it does not require knowledge about the true Pareto optimal front of the problem under study. This is often the case when addressing real-world problems. If the true Pareto optimal front is not known, determination of a sufficient number of solution evaluations for successful convergence is virtually impossible. Second, a sufficient number of solution evaluations differs from problem to problem. An appropriate number of evaluations depends on the complexity of the problem in terms of the number of decision variables, the decision variable domains, the number of objectives, the optimality characteristics, and so on. Some algorithms may converge fast to the true Pareto optimal front, but the researcher or practitioner does not know this in advance when a large number of solution evaluations is assigned at the start of the search process. Therefore, for any benchmarking and comparative analyses, the number of

[1] The phrase "solution evaluations" could be replaced by "generations" if the MOEA has a constant population size over all generations.

solution evaluations for each test problem should be set to different values to allow for convergence to the true Pareto optimal front. Finally, it is possible to evaluate this measure during the entire search process at any given iteration thus providing valuable real-time information about the convergence behavior of the algorithm.

The usefulness of the HGR_t is that it measures the convergence speed of any multiobjective optimization algorithm to the true Pareto front at any given iteration, even though the exact location of the true Pareto front is unknown to us.

5 Computational study

In this section, we present the computational study used to evaluate the performance of SPGA and its benchmarking with NSGA-II in the presence of noise. NSGA-II is widely used for comparison in many benchmarking studies. It has been reported that NSGA-II obtain good results on many (but not all) of the standard continuous problems such as ZDT family and FON problems, and it competes very well with SPEA2 in terms of convergence to the true Pareto optimal front while maintaining solution diversity (Deb et al. 2002; Zitzler et al. 2001). Some studies report that there is no significant difference between the performance of SPEA2 and NSGA-II, although SPEA2 requires significantly higher computational time (Deb et al. 2005; Bui et al. 2005; Erbas et al. 2006).

It is required to modify any standard MOEA such as NSGA-II in order to increase its robustness against the noisy objective functions. Since repeatedly sampling a solution is the simplest and most common technique in reducing the disturbance of the noise, the original NSGA-II is modified to take into account the expected value of objective functions (sample average) over a certain number of sampling, *i.e.*, $\mu_i(\mathbf{x}) = E[f_i(\mathbf{x})]$ for ith objective function. Moreover, experimental results and comparative analyses of Bui et al. (2005) indicate that resampling approach usually offers better performance in comparison to the probabilistic approach in the context of NSGA-II. In this study, the number of random samplings for both SPGA and NSGA-II is set to 15 over all experimentations. This sampling size was determined from some runs in which sampling sizes from 5 to 20 were tried. With a sampling size of 15, the most comparative results with respect to the four given different noise levels are obtained.

5.1 Control parameter settings

For both SPGA and NSGA-II, all of the parameters, except the maximum number of solution evaluations, are set to the suggested values in the original study of Deb et al. (2002). Real-parameter SBX recombination operator with $\eta_c = 15$, and a polynomial mutation operator with $\eta_m = 20$ are performed. The crossover probability of $p_c = 1$ and mutation probability of $p_m = 1/n$, where n is number of variables, are used. The selection operation is performed using the binary tournament selection scheme. In order to make better comparisons, the initial and maximum population size for SPGA is set to the population size used by Deb et al. (2002). The number of solution evaluations depends on the characteristics and complexity of the underlying problem which is extracted from using the proposed HGR_t-based stopping criterion as shown in Table 1. Note that the number of solution evaluations for each problem is different.

Table 1 Control parameter settings for SPGA and NSGA-II

Algorithm parameter	SPGA and NSGA-II					
Test problem	FON	KUR	ZDT1	ZDT2	ZDT4	ZDT6
Number of solution evaluations	1,100	3,700	10,500	11,500	14,000	15,500

Table 2 Parameter settings for noise levels

Level of noise	Variable error λ	Constant error ε
No noise	0.00	0.00
Low	0.01	0.01
Medium	0.05	0.05
High	0.10	0.10

5.2 Modeling noisy objective functions

Noise is introduced in the objective space as

$$f_i'(\mathbf{x}) = f_i(\mathbf{x}) + s_i N(0, 1),$$

where $f_i'(\mathbf{x})$ is a noisy objective function of solution $\mathbf{x}$, $f_i(\mathbf{x})$ is the real value of objective function, and s_i is the standard deviation of normal distribution of noise effect with mean zero. In most noisy GA studies, the standard deviation s_i is kept fixed over all possible values of objective functions (Bui et al. 2005; Fieldsend and Everson 2005). This assumption is not reasonable in many stochastic problem environments, particularly in the stochastic simulation context. In most stochastic real-world MOPs, the higher objective values are usually expected to have more errors than lower ones. In minimization problems, if the objective values at the beginning and middle of the search are quite large with respect to the standard deviation s_i, an employed algorithm is not challenged during the search until the objective values become relatively small so that the standard deviation significantly affects the real values of objective functions.

To model the noise in stochastic environments more accurately, it is suggested that the standard deviation s_i is composed of two components—variable error λ_i and constant error ε_i. Mathematically speaking, the s_i, over all possible objective function values, is as follows

$$s_i = \lambda_i f_i(x) + \varepsilon_i,$$

where λ_i is a coefficient that makes the standard deviation able to change corresponding to its objective value, and ε_i is the constant error along all objective values. Four levels of noise settings are implemented to evaluate the performance of the algorithms (Table 2).

5.3 Termination of the search

Recall that, according to the suggested HGR_t-based stopping criterion for real-world MOPs, the search terminates when the hypervolume growth rate HGR_t becomes less than the desired small value of HGR_{stop}. In order to better determine the appropriate termination of the search in terms of the convergence speed of HGR_t, we perform a pilot study on test problems FON, ZDT1, ZDT2, ZDT4, and ZDT6. For FON problem with three decision variables, the evaluation period and HGR_{stop} are set to 200 and 0.005, respectively. For other problems with high-dimensional solution space, the evaluation period and HGR_{stop} are set to 500 and 0.001, respectively. The evaluation period (and HGR_{stop}) assigned to the ZDT problems is selected to be larger (smaller) than that of FON. This setting ensures that the MOEA is given adequate opportunity to find a new promising solution in the high-dimensional solution space.

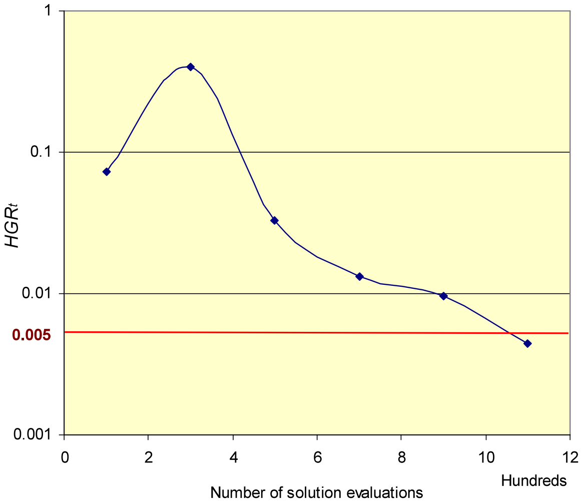
The results from the study with SPGA for FON and ZDT4 problems are shown in Fig. 4. The hypervolume growth rate HGR_t fluctuates slightly during the search for some of the test problems, but it decreases gradually for all problems. As the search process proceeds, the HGR_t becomes less than the pre-specified HGR_{stop} resulting in the termination of the search. Although the HGR_t gradually decreases through the search, it does not have monotonically decreasing behavior. The HGR_t increases at some points when promising nondominated solution(s) in the objective space is found resulting in significant increase of the hypervolume. Since the HGR_t measure provides useful information on the convergence behavior of an algorithm toward the true Pareto optimal front during the search process, it is only introduced as a measure to stop the search. It must be noted here that, since the HGR_t cannot necessarily be translated into the quality of the solutions, this convergence growth measure is not suggested as a viable algorithm performance metric.

5.4 Performance metrics

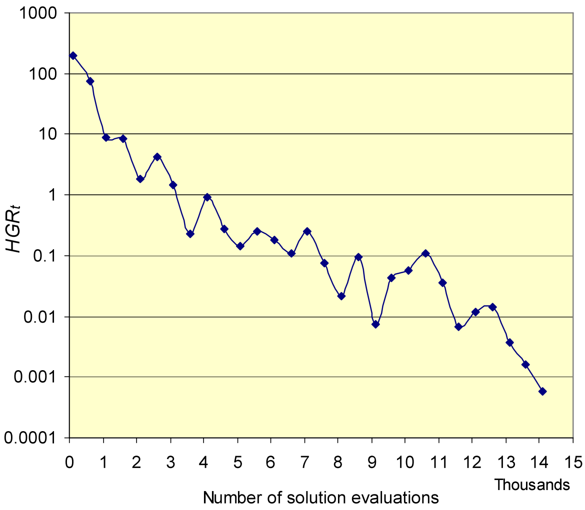
Many performance metrics have been introduced within the last decade (Knowles and Corne 2002; Zitzler et al. 2003). In this study, four performance metrics are used to measure the convergence behavior and diversity of SPGA and NSGA-II. For each test problem, a MOEA is run with 30 different random seed values, and the mean and 95% confidence interval of each metric are computed. The lower and upper bounds of the 95% confidence interval are calculated by $\overline{x} \pm t_{\alpha/2,\eta-1}s/\sqrt{p}$, where $\overline{x}$ is the sample mean, s is sample standard deviation, α is the significance level and is equal to 0.05, and p is the sample size and is equal to 30.

5.4.1 Distance from the Pareto optimal front

We calculate the distance metric using the set of $H = 500$ evenly-spaced solutions taken from the true Pareto optimal solution set in the objective space. The minimum Euclidean distance from each obtained nondominated solution to the H solutions is calculated and the average of these distances is used as the distance metric.



(a)

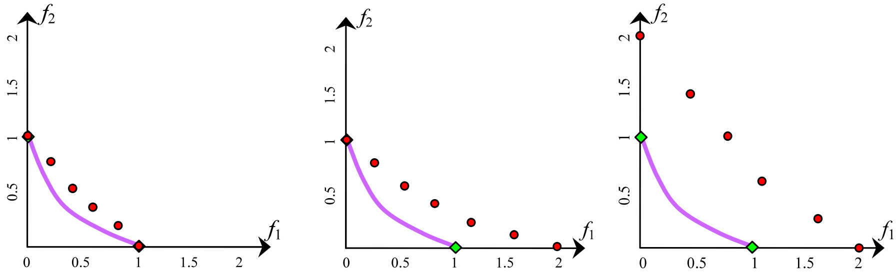


(b)

Fig. 4 The convergence speed of HGR_t with SPGA on (**a**) FON, and (**b**) ZDT4

5.4.2 *Maximum spread of nondominated solutions*

In this study, we introduce a new maximum spread metric MS to evaluate the extent of spread of the obtained nondominated solutions in the objective space. Here, the goal is to obtain a set of nondominated solutions that are widely distributed along the Pareto optimal front at the end of the search. The traditional maximum spread, say MS_{tr}, used in the literature takes into account only the extend that nondominated solutions are distributed in the objective space (the larger the better MS_{tr}) regardless



(a) MS=0 and MS_{tr}=1.41 (b) MS=0.5 and MS_{tr}=2.23 (c) MS=1 and MS_{tr}=2.82

Fig. 5 Superiority of MS over MS_{tr} in different distribution of nondominated solutions

of how far they are from the true Pareto optimal front. If the obtained nondominated solutions have an excellent diversity in terms of both spread and spacing, but they are far away from the true Pareto optimal front, their diversity is useless. To compute the maximum spread MS, the minimum Euclidean distance of the two extreme Pareto solutions of the true Pareto optimal set from the nondominated solutions, denoted by l_1 and l_H, is calculated as shown in Fig. 6. The maximum spread MS of the set of nondominated solutions is

$$MS(\mathbf{NP}_T) = \frac{l_1 + l_H}{2}.$$

Note that the distances d_p and d_q are the distances from the closest nondominated solutions, not necessarily the endpoints of nondominated solutions, to the two extreme Pareto solutions. The maximum spread metric MS returns a value in the range of $[0, \infty)$. Small values of this metric mean the nondominated solutions are well-spread. Ideally, this metric takes a value of zero. For better understanding of superiority of the proposed maximum spread metric MS over the traditional maximum spread MS_{tr}, Fig. 5 presents different situations for distribution of nondominated solutions that MS reflects reasonable values whereas MS_{tr} reflects misleading values.

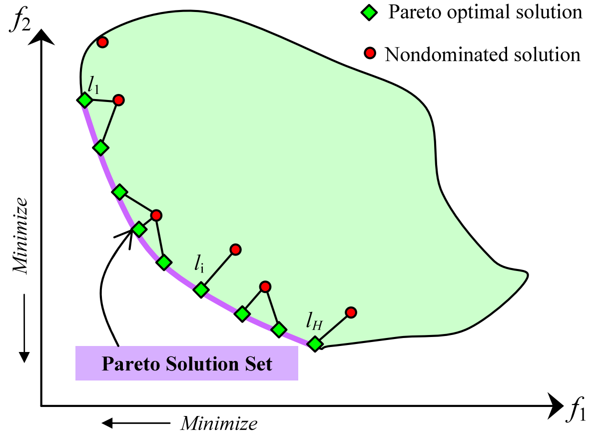
5.4.3 Delineation of the Pareto optimal front

The delineation metric, introduced in Eskandari and Geiger (2008), is used to simultaneously evaluate the extent of both convergence and diversity to a known Pareto optimal front. To calculate the delineation metric, a set of H evenly-spaced solutions from the Pareto optimal solution set must be known. The minimum Euclidean distance from each Pareto optimal solution to the obtained solutions l_i is calculated and the average of these distances is the delineation metric, or

$$\Phi(\mathbf{P}_T) = \frac{1}{H} \sum_{i=1}^{H} l_i.$$

Figure 6 shows how this metric is calculated. It is important to note that all solutions obtained by an algorithm, including those that are dominated, are considered in the

Fig. 6 Delineation metric



calculation of this metric. The delineation metric Φ returns a value in the range of $[0, \infty)$. The smaller the value of this metric, the better the Pareto optimal solutions are represented by the obtained solutions. Ideally, this metric is zero, where population size is adequately large ($\geq H$), and each H selected Pareto solution is exactly overlapped by one of the nondominated solutions. The likelihood of this happening is zero, especially when population size is smaller than H, which is the case in most applications.

5.4.4 Hypervolume

Finally, we use the hypervolume ratio (*HVR*), the proportion of the volume enclosed by the reference point **R** and true Pareto optimal front that is not covered by the nondominated solutions is of interest. *HVR* is calculated as

$$HVR(\mathbf{NP}_T) = 1 - \frac{HV(\mathbf{NP}_T)}{HV(\mathbf{PF})}.$$

PF is the set of solutions on the true Pareto optimal front. Reference point **R** is the worst objective vector for each test problem which is kept fixed through the whole search process. The reference vector **R** for FON, ZDT1, ZDT2, ZDT4, and ZDT6 are (1, 1), (10, 1), (10, 1), (300, 1), and (10, 1), respectively. *HVR* returns a value in the range [0, 1]. The smaller value of *HVR*, the better.

6 Discussion of computational results

In this section, the computational results of SPGA and NSGA-II on a suite of six test problems published in the EMO literature that have two objectives and no coupled constraints are presented. In all test problems, the functions are to be minimized.

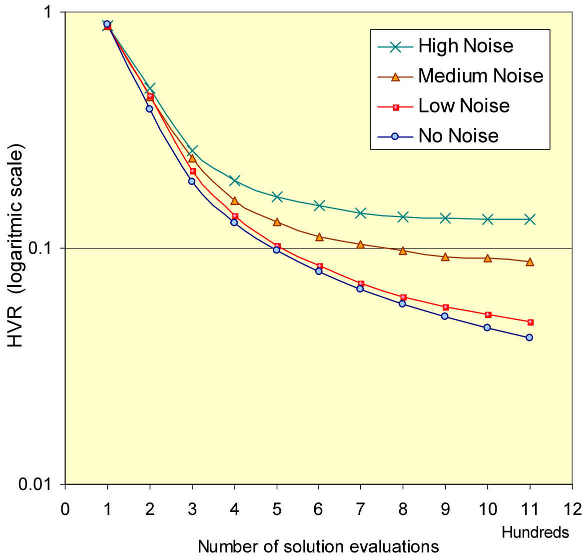
6.1 FON test problem

The first test problem, referred to as FON (Fonseca and Flemming 1993), has three decision variable and a nonconvex Pareto optimal front. The *HVR* performance metric with SPGA and NSGA-II for four different levels of noise at different iterations of search process is illustrated in Fig. 7. As expected, the noise involved in the objective function values of each algorithm reduces its convergence speed proportional to the noise level. It is observed that there is not significant difference between the performance of algorithms with no noise and low noise level situations. SPGA has faster convergence than NSGA-II at the beginning and middle of the search process making it competent algorithm when fast convergence property is desired. Figure 8 presents the mean and 95% confidence interval of four performance metrics with SPGA and NSGA-II for different noise levels at the end of the search. SPGA significantly outperforms NSGA-II with respect to distance, delineation and *HVR* for all noise levels except for high noise level. This outperformance is trivial with respect to spread metric. However, the outperformance magnitude of SPGA over NSGA-II is reduced when the noise level increases resulting in having almost similar performance at high level of noise. This comparison implies that although SPGA has better performance with lower noise levels, its performance deteriorates more with the increase of noise level on FON problem.

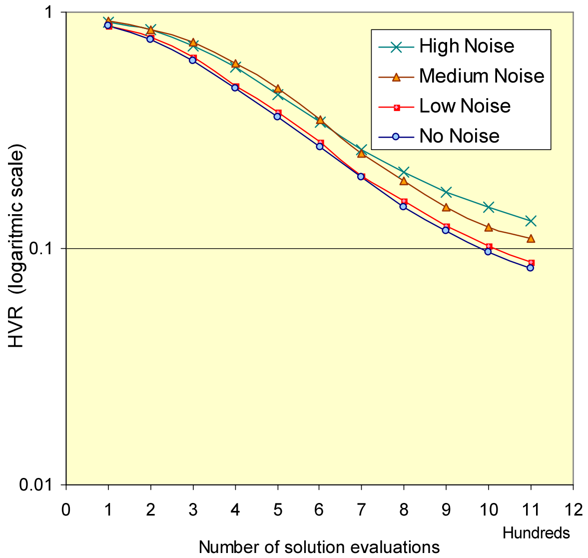
6.2 ZDT1 test problem

The test problems ZDT1, suggested by Zitzler et al. (2000), has 30 decision variables and a convex true Pareto optimal front. Figure 9 illustrates *HVR* performance metric with SPGA and NSGA-II for four different levels of noise. It is observed that there is significant difference among the performance of each algorithm in the presence of different noise levels. The convergence of each algorithm is noticeably disrupted even by the influence of low noise. SPGA has faster convergence than NSGA-II in no noise and low noise environments. This convergence superiority is not appreciable in the medium noise environment. It is seen that in high noise environment SPGA leads to premature convergence at the beginning of the search resulting in slight improvement of the performance as the search proceeds. To perceive how the noise affects the obtained population at the end of the search, Fig. 10 presents the mean and 95% confidence interval of four performance metrics with SPGA and NSGA-II for four different noise levels. SPGA significantly outperforms NSGA-II with respect to distance, delineation and *HVR* metrics for all noise levels except for high noise level. For the speared metric, the outperformance magnitude of SPGA over NSGA-II is significant in no and low noise environments while it is insignificant for medium noise environment. In the high noise environment, SPGA is fully disrupted and NSGA-II has significantly better performance for all four metrics.

Goldberg et al. (1992) has found that when dealing with noisy and uncertain objective functions, a larger population size should be considered to avoid premature convergence. In another study, Miller (1997) found that under certain assumptions there is a good approximation to estimate population size depending on the noise level. The higher the noise level, the larger the population size should be taken into



(a)



(b)

Fig. 7 *HVR* performance metric for different noise levels at different search iterations on FON. (**a**) SPGA, and (**b**) NSGA-II

account. Therefore, very low population size is one of the reasons for the inefficiency of SPGA in the presence of high level of noise. Recall that SPGA employs adaptive population sizing scheme in which population size at each iteration is the number of nondominated solutions plus the parameter $a(t)$ which is kept 20 in this study.

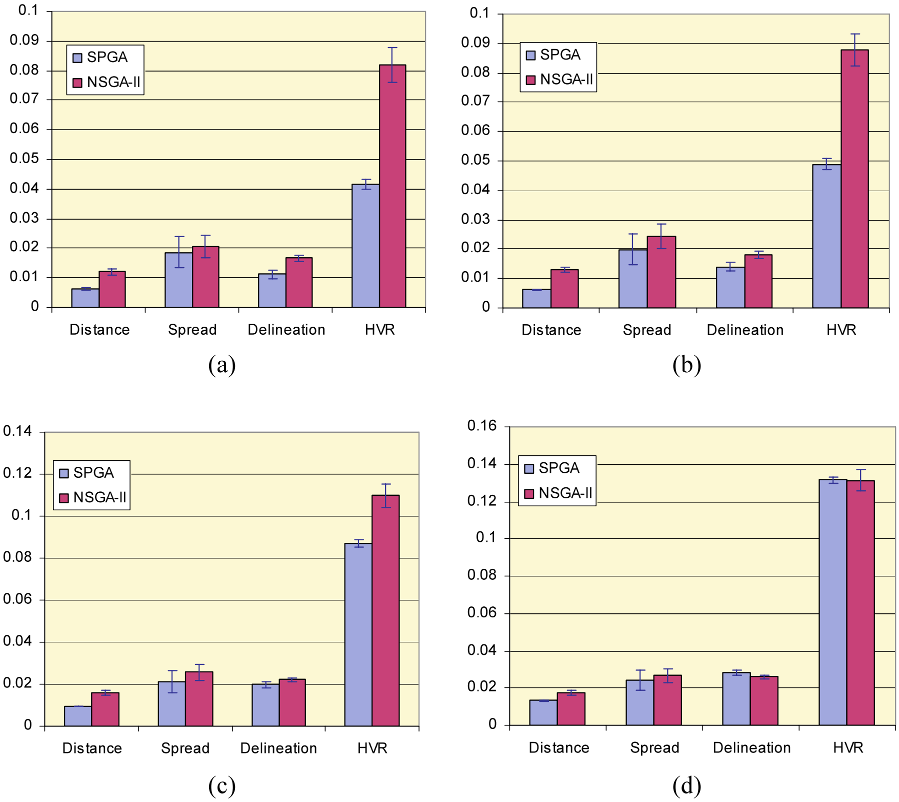
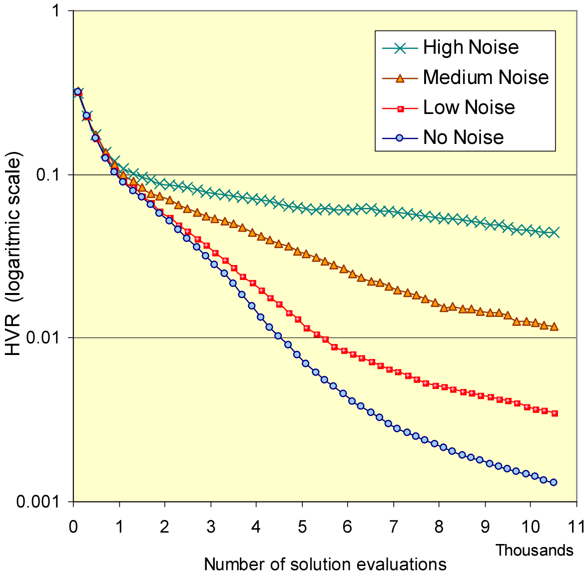
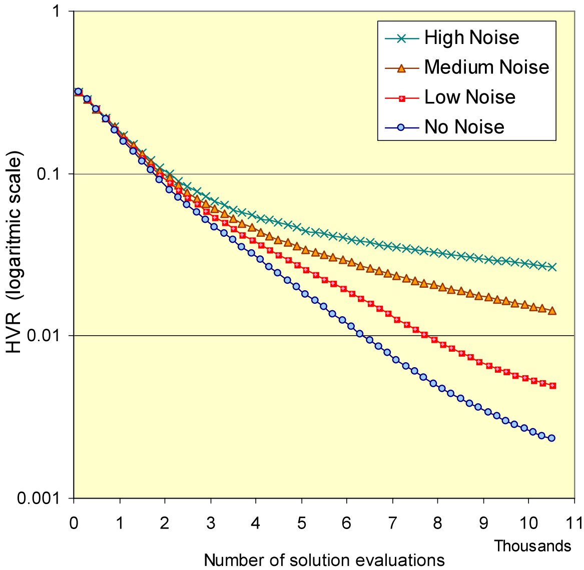


Fig. 8 Mean and 95% confidence interval of four performance metrics with SPGA and NSGA-II on FON for different noise levels. (**a**) no noise, (**b**) low noise, (**c**) medium noise and (**d**) high noise

For better understanding of convergence behavior of each algorithm in the presence of noise, Fig. 11 presents the population size, number of nondominated solutions obtained from noisy objective values (called noisy nondominated solutions), and number of nondominated solutions obtained from real objective values (called true nondominated solutions) for both SPGA and NSGA-II. Since the population size of NSGA-II is fixed (equal to 100), it is not plotted in the Fig. 11. It is clearly seen that the population size of the SPGA (Fig. 11(a)) is actually the number of noisy nondominated solutions (Fig. 11(b)) plus 20. Expectedly, the number of noisy and true nondominated solutions for each algorithm is identical in the no noise environment. It is interesting to observe that the number of noisy nondominated solutions is drastically larger than the number of true nondominated solutions for both SPGA and NSGA-II. This observation is right for each level of noisy environment. Finally, we see that number of true nondominated solutions for SPGA is larger than NSGA-II for no noise and low noise environments whereas it is larger for NSGA-II than SPGA for medium and high noise environments. It must be noted here that, while the number of nondominated solutions provides useful information on the convergence behavior of an algorithm to generate nondominated solutions during the search process, it is not



(a)



(b)

Fig. 9 *HVR* performance metric for different noise levels at different search iterations on ZDT1. (**a**) SPGA, and (**b**) NSGA-II

suggested as a viable algorithm performance metric, since the number of nondominated solutions cannot necessarily be translated into the quality of the solutions.

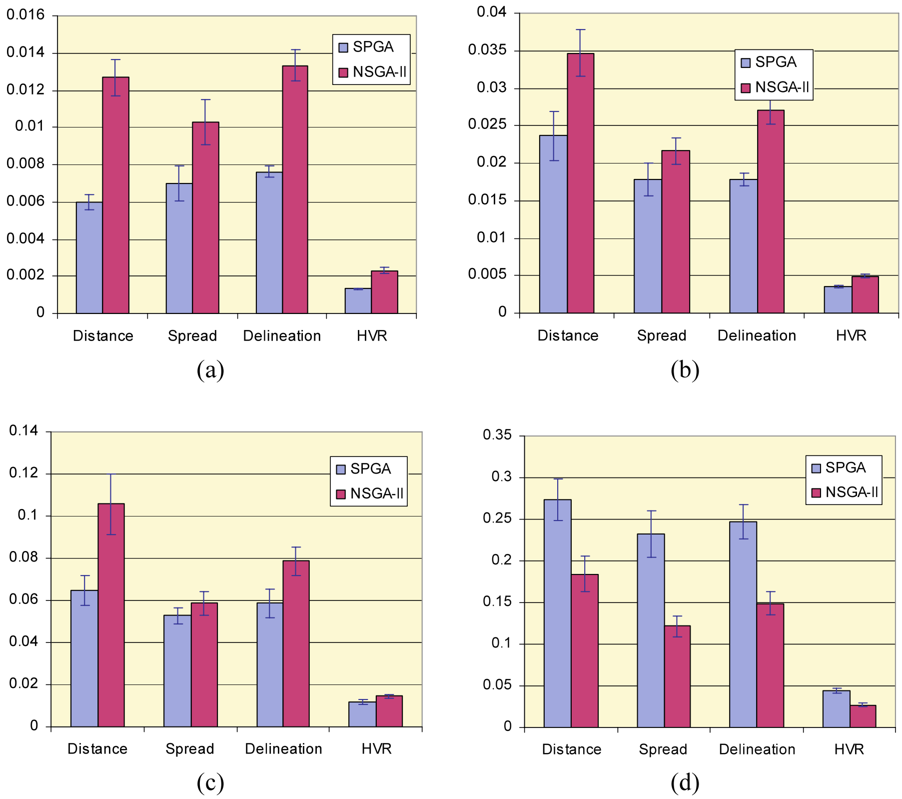
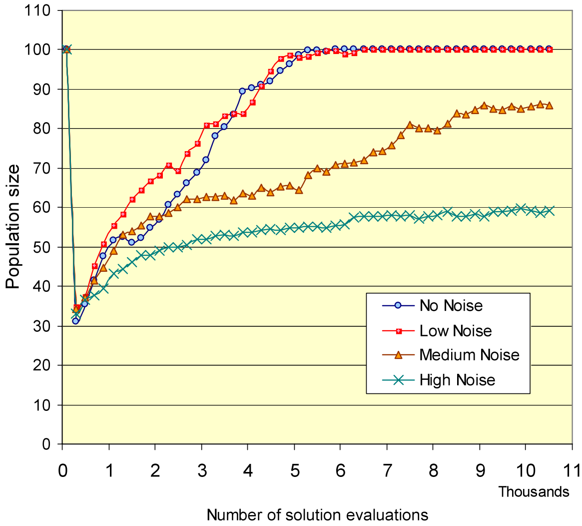


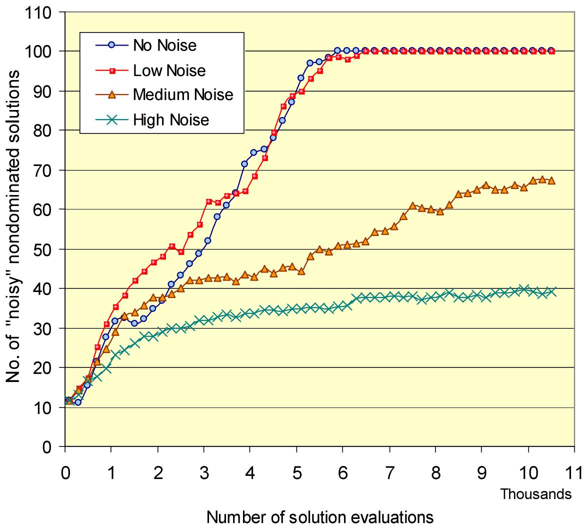
Fig. 10 Mean and 95% confidence interval of four performance metrics with SPGA and NSGA-II on ZDT1 for different noise levels. (**a**) no noise, (**b**) low noise, (**c**) medium noise and (**d**) high noise

6.3 ZDT2 test problem

The 30-decision variable problem ZDT2 has a concave Pareto optimal front. Figure 12 presents *HVR* metric with SPGA and NSGA-II for four different levels of noise. Similar to the ZDT1, it is observed that there is significant difference among the performance of each algorithm in the presence of different noise levels. Moreover, the convergence of each algorithm is noticeably disrupted even by the influence of low noise. SPGA has faster convergence than NSGA-II only in no noise environment. NSGA-II has significantly better convergence property than SPGA in the medium and high noise environments. It is seen that in both medium and high noise environments SPGA is trapped by the premature convergence at the beginning of the search resulting in slight improvement of its performance as the search proceeds. Figure 13 presents the mean and 95% confidence interval of four performance metrics with SPGA and NSGA-II for four different noise levels. SPGA significantly outperforms NSGA-II with respect to all metrics only where there is no noise involved. For the low level of noise, performance of SPGA severely reduces and the convergence improves very slowly that NSGA-II has significantly superior spread and delineation values. In the medium and high noise environment, SPGA is fully disrupted



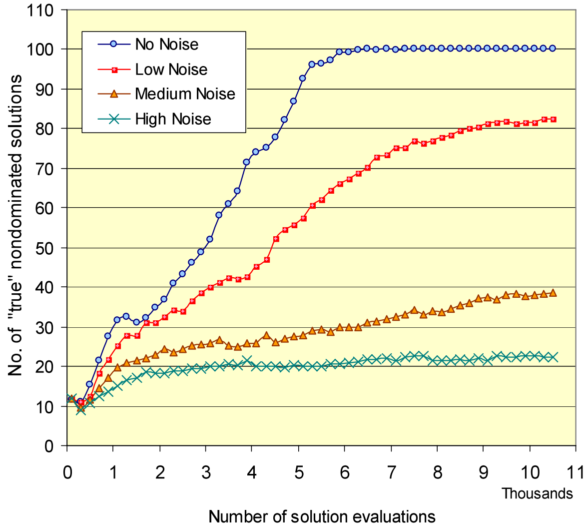
(a)



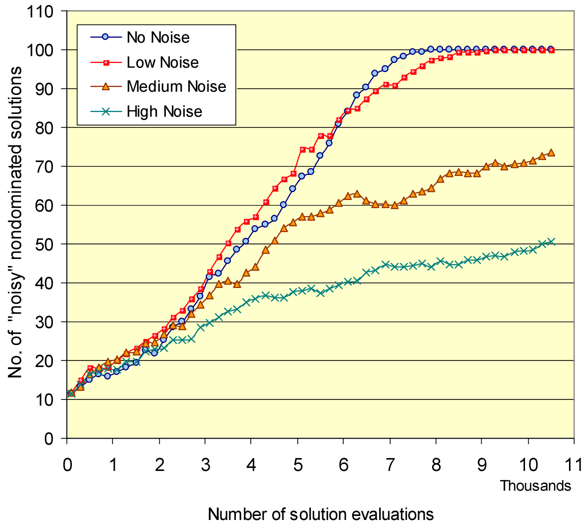
(b)

Fig. 11 Convergence behavior on ZDT1 for (**a**) population size of SPGA, (**b**) number of noisy nondominated solutions with SPGA, (**c**) number of true nondominated solutions with SPGA, (**d**) number of noisy nondominated solutions with NSGA-II, and (**e**) number of true nondominated solutions with NSGA-II

and NSGA-II has significantly better performance for all four metrics. Similar to the ZDT1, these results indicate that the performance of the SPGA is heavily affected by the noise even in the presence of low noise. Without any doubt, high intensity of elitism as a result of small population size of SPGA at the beginning of the search process plays a negative feature in the noisy environments for this algorithm and leads



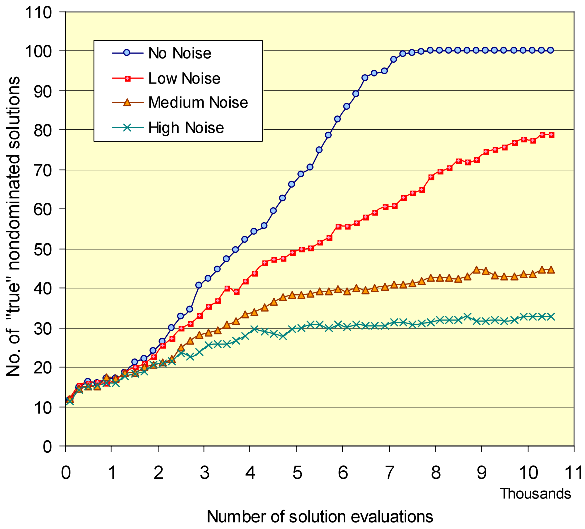
(c)



(d)

Fig. 11 (Continued)

to premature convergence. Although elitism can be employed as a convergence accelerator in the deterministic problem environments, it could be misleading in the noisy environments, since the corrupted locations of the solutions mislead the algorithm, resulting in the poor performance.



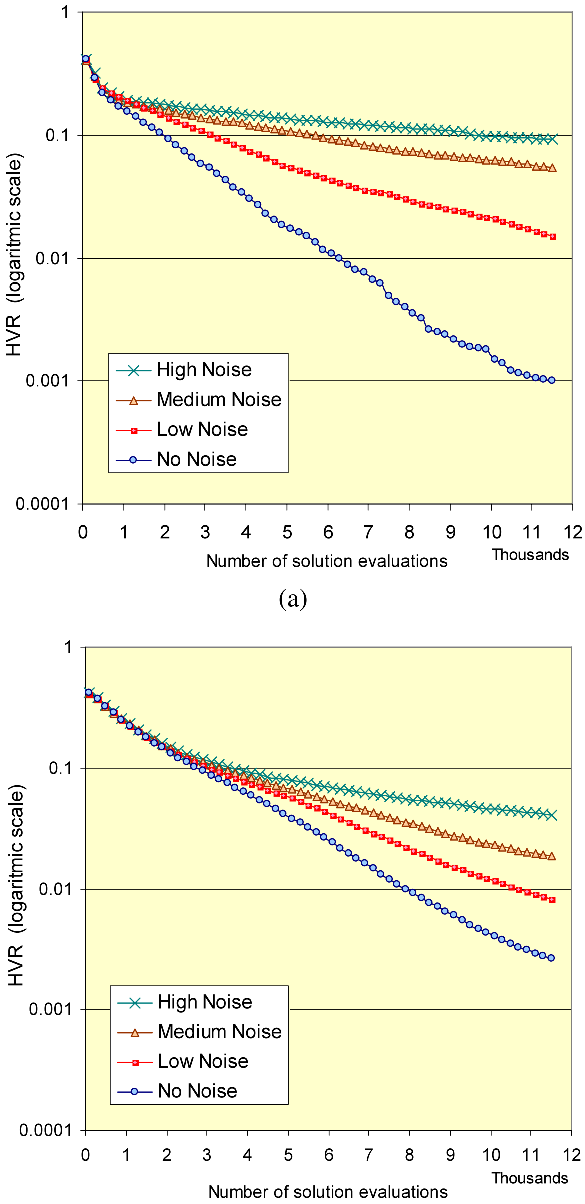
(e)

Fig. 11 (Continued)

6.4 ZDT4 test problem

The 10-decision variable test problem ZDT4 is a multi-frontal problem having a large number of local Pareto optimal fronts and a single global Pareto optimal front. Figure 14 presents the *HVR* metric with SPGA and NSGA-II for different level of noise as the search proceeds. There is a significant difference between the performance of both SPGA and NSGA-II for no noise and low level of noise. These results indicate that ZDT4 problem is very sensitive to noise and even low level of noise considerably deteriorates the convergence behavior of both algorithms. It is interesting to note that after enforcement of low noise level the deterioration of algorithms' convergence is less significant with the increase of noise level, particularly for NSGA-II. In other words, little noise disrupts the convergence of both algorithms, but the disruption does not increase significantly with the raise of noise level. This matter is better perceived that NSGA-II has almost the same *HVR* values under three different levels of noise at different iterations of search process. SPGA converges significantly faster than NSGA-II towards the true Pareto front for all different levels of noise. The superiority of SPGA reduces with the raise of noise level where not significant difference is found at the high noise level.

Figure 15 illustrates the sample distribution of obtained populations with SPGA under different levels of noise. The true Pareto optimal front, the true location of solutions, called true solutions, and corrupted location of noisy solutions, called noisy solutions are depicted. The distance and uniformity of solutions from the Pareto front increases with the raise of noise level. The noisy solutions are typically lying on the left below of true solutions. The difference between the distribution of noisy solutions and the distribution of true solutions is better observed when the noise level is



(a)

(b)

Fig. 12 *HVR* performance metric for different noise levels at different search iterations on ZDT2. (**a**) SPGA, and (**b**) NSGA-II

medium or high (Fig. 15(c) and Fig. 15(d)). On ZDT4 problem, some of the noisy solutions are actually located to the left of the Pareto front in noisy environments

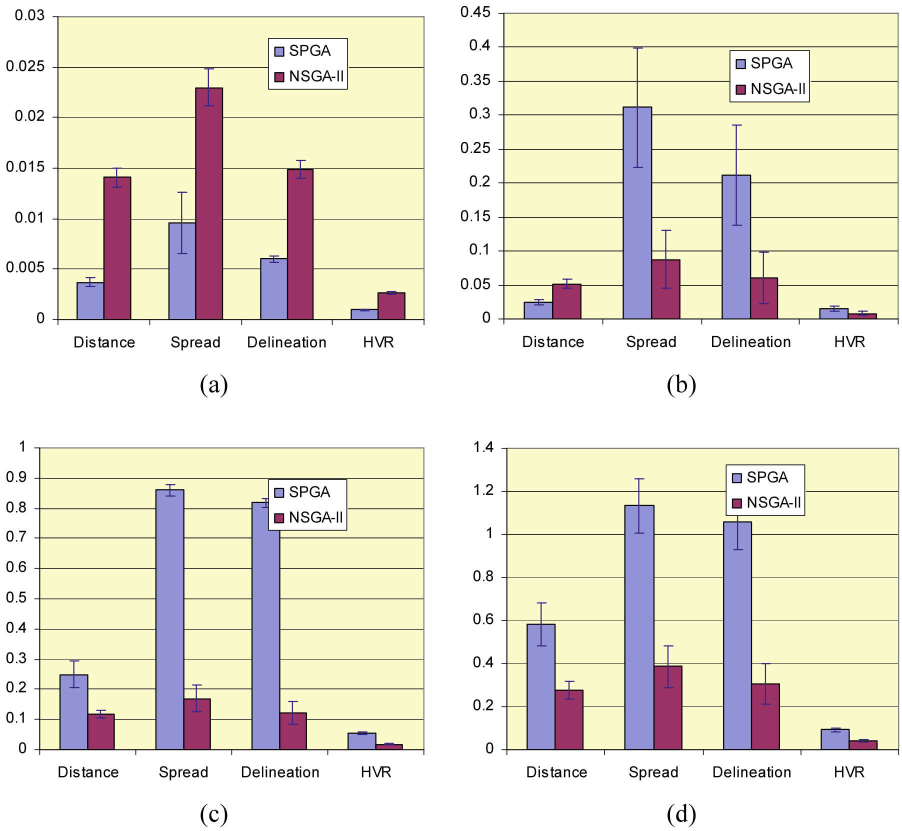
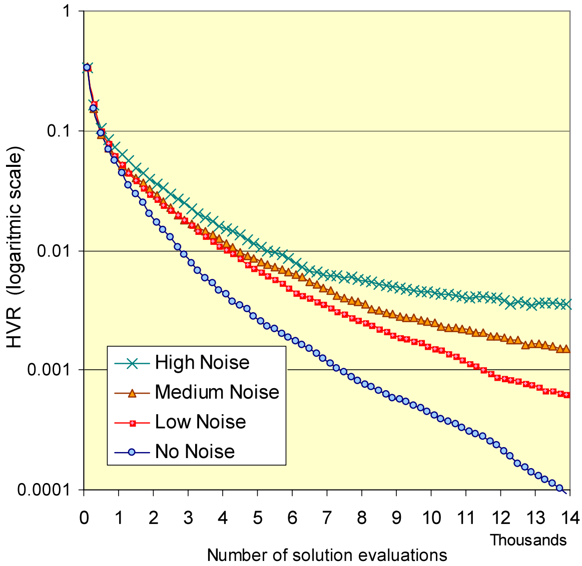


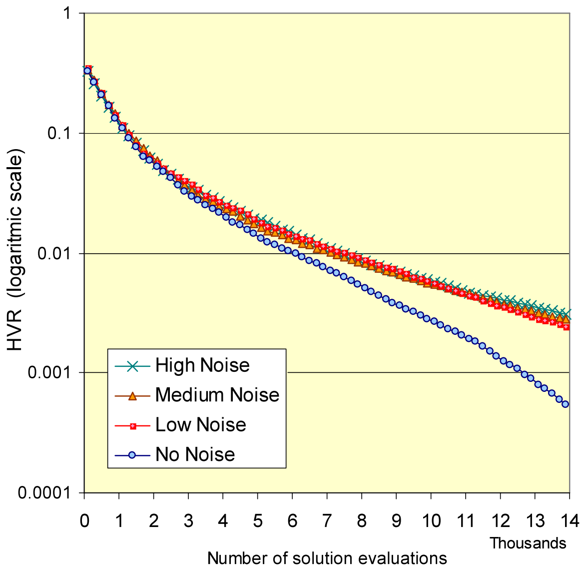
Fig. 13 Mean and 95% confidence interval of four performance metrics with SPGA and NSGA-II on ZDT1 for different noise levels. (**a**) no noise, (**b**) low noise, (**c**) medium noise and (**d**) high noise

representing how much noise corrupts the true location of solutions in the objective space.

Figure 16 presents the mean and 95% confidence interval of distance, spread and delineation metrics with SPGA and NSGA-II for four different noise levels. Since the reference vector for ZDT4 is very far away from the true Pareto, the corresponding *HVR* values are very small compared to the other metric values. Thus, the mean and 95% confidence interval of *HVR* is given in Fig. 17. SPGA significantly outperforms NSGA-II with respect to all metrics for all four noise levels, except for *HVR* in the presence of high noise. This exception implies that although SPGA has closer population than NSGA-II to true Pareto front, NSGA-II has better diversity in the presence of high noise resulting in better *HVR* value. However, the outperformance magnitude of SPGA over NSGA-II is reduced, as the noise level increases. In contrast to ZDT1 and ZDT2, the performance of SPGA in comparison to NSGA-II is not heavily affected by the raise of noise level. This observation implies that small initial population size of SPGA does not severely hurt its performance for any problem.



(a)

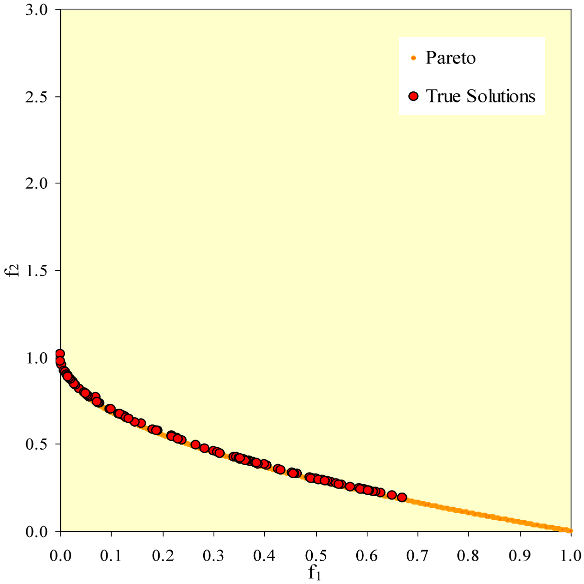


(b)

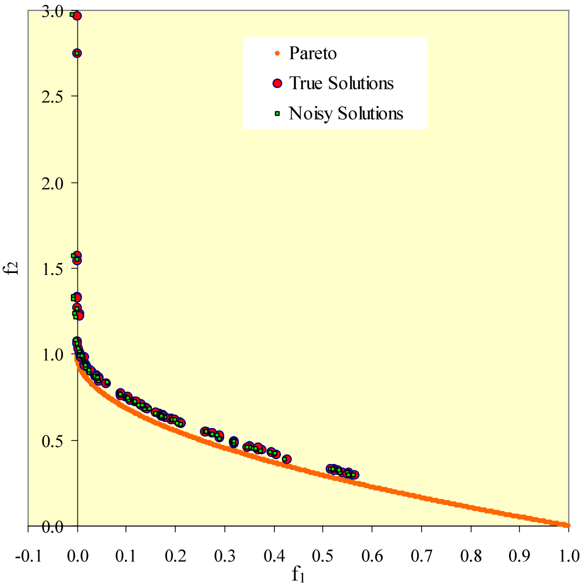
Fig. 14 *HVR* performance metric for different noise levels at different search iterations on ZDT4 (**a**) SPGA, and (**b**) NSGA-II

6.5 ZDT6 test problem

The test problem ZDT6 has 10 decision variables and a nonconvex Pareto optimal front; moreover, the density of solutions across its Pareto optimal front is non-

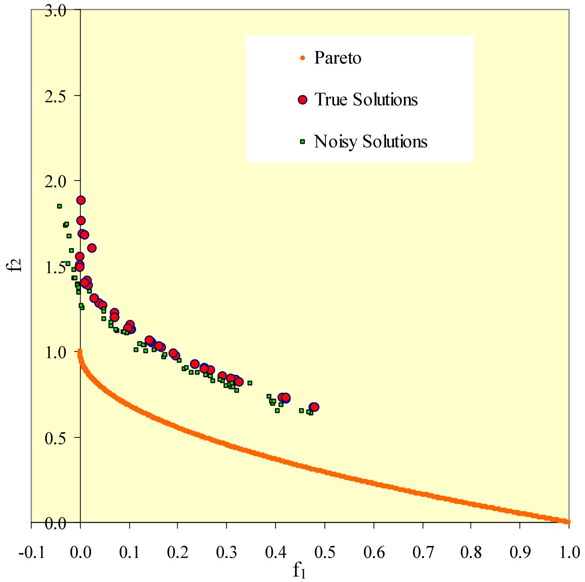


(a)

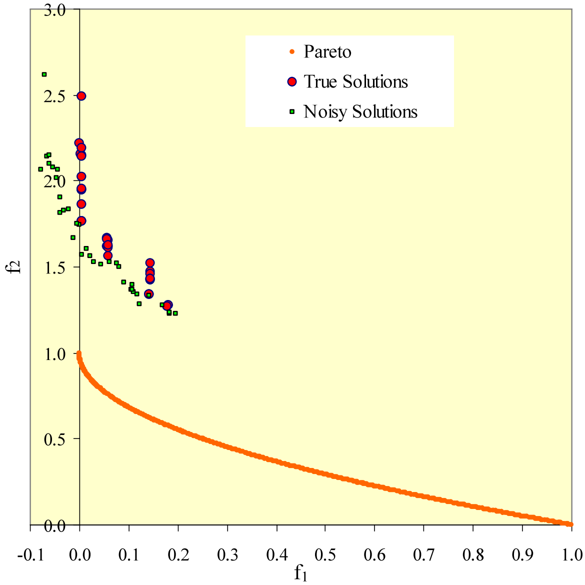


(b)

Fig. 15 Distribution of populations with SPGA on ZDT4 under different levels of noise. (**a**) no noise, (**b**) low noise, (**c**) medium noise, and (**d**) high noise



(c)



(d)

Fig. 15 (Continued)

uniform, and the density towards the Pareto optimal front gets thin. The *HVR* metric with SPGA and NSGA-II for different level of noise as the search proceeds are given in Fig. 18. There is a significant difference between the performance of each

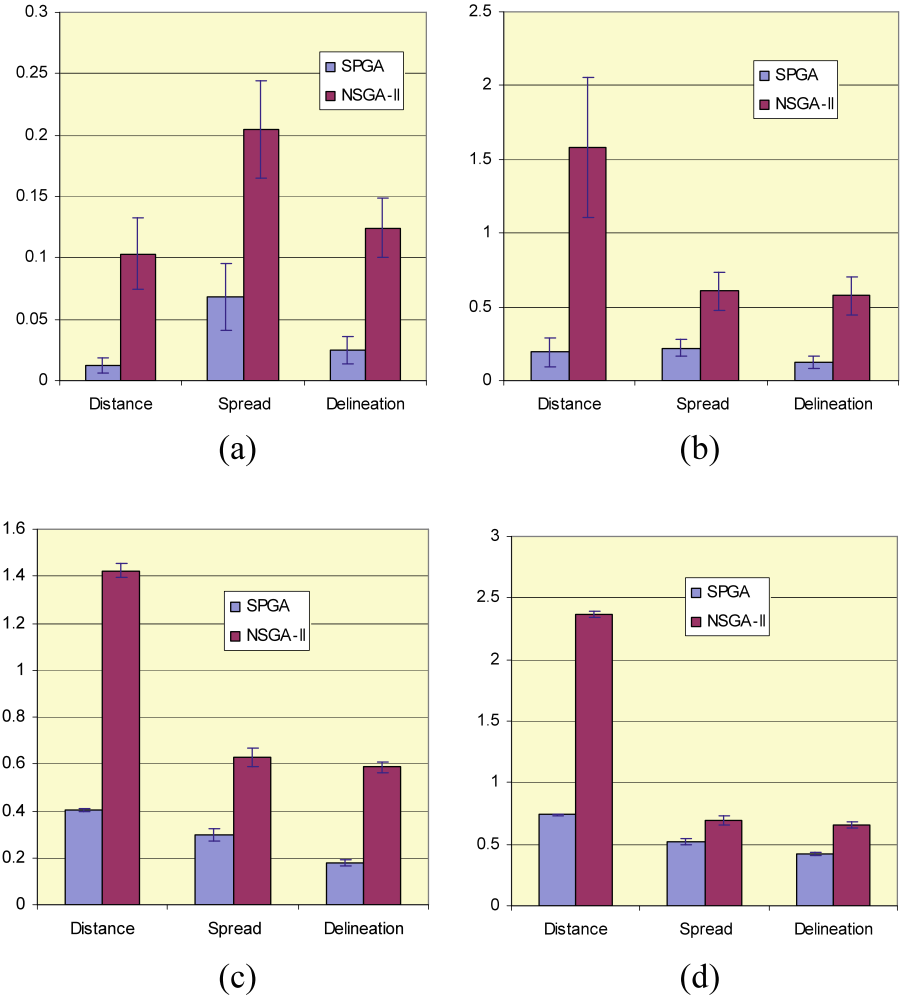


Fig. 16 Mean and 95% confidence interval of three performance metrics with SPGA and NSGA-II on ZDT4 for different noise levels. (**a**) no noise, (**b**) low noise, (**c**) medium noise and (**d**) high noise

algorithm for no noise and noisy situations. The difference among the algorithm's performance at different levels of noise is more tangible for SPGA than NSGA-II. These results indicate that ZDT6 problem is very sensitive to noise and even low level of noise considerably deteriorates the convergence behavior of both algorithms. It is interesting to note that after enforcement of low noise level the deterioration of NSGA-II convergence is less significant with the increase of noise level. In other words, NSGA-II doesn't have very different *HVR* values under three different levels of noise at different iterations of search process. SPGA converges significantly faster than NSGA-II towards the true Pareto front for all different levels of noise.

Figure 19 presents the mean and 95% confidence interval of four performance metrics with SPGA and NSGA-II for different noise levels at the end of the search.

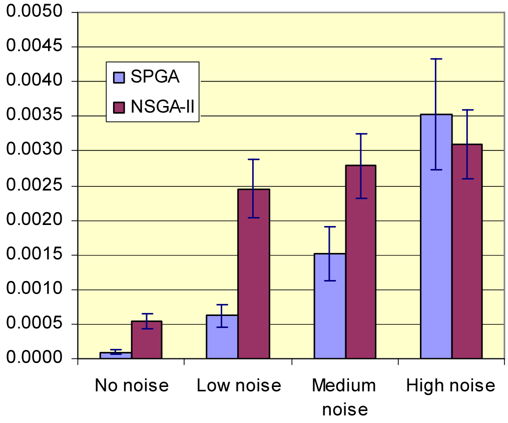


Fig. 17 Mean and 95% confidence interval of *HVR* metric with SPGA and NSGA-II on ZDT4

SPGA significantly outperforms NSGA-II with respect to all metrics at all noise levels. However, the outperformance magnitude of SPGA over NSGA-II is reduced when the noise level increases. This comparison implies that although SPGA has better performance with lower noise levels, its performance deteriorates more with the increase of noise level.

7 Conclusions and future research

This research presents an MOEA approach for dealing with stochastic multiobjective optimization problems. New concepts of stochastic and significant dominance introduced in the stochastic multiobjective optimization domain which could reduce the disturbance of noise and provide a more precise measure for better discrimination among competing solutions. Computational results for a number of test problems show that the convergence speed of the algorithm is reduced in a noisy environment which is usually proportional to the magnitude of noise. SPGA outperforms NSGA-II with respect to all metrics on FON, ZDT4 and ZDT6 problems. On the other hand, compared to NSGA-II, SPGA presents poor performance on ZDT1 and ZDT2 in the presence of medium and high level of noise. Undoubtedly, high intensity of elitism as a result of small population size of SPGA at the beginning of the search process plays a negative feature in the noisy environments for this algorithm and leads to premature convergence. Generally, elitism can be employed as a convergence accelerator in the deterministic problem environments where no noise is involved in the objective values. In contrast, elitism could be deceiving in the noisy environments, since the corrupted locations of the solutions mislead the algorithm, resulting in the poor performance.

Reporting the performance metrics at the different steps of the search process can provide valuable information on the convergence behavior of any algorithm. Our results indicate that SPGA due to employment of high intensity of elitism converge faster towards true Pareto front, although in the presence of considerable noise level it performs not satisfactorily on some of test problems.

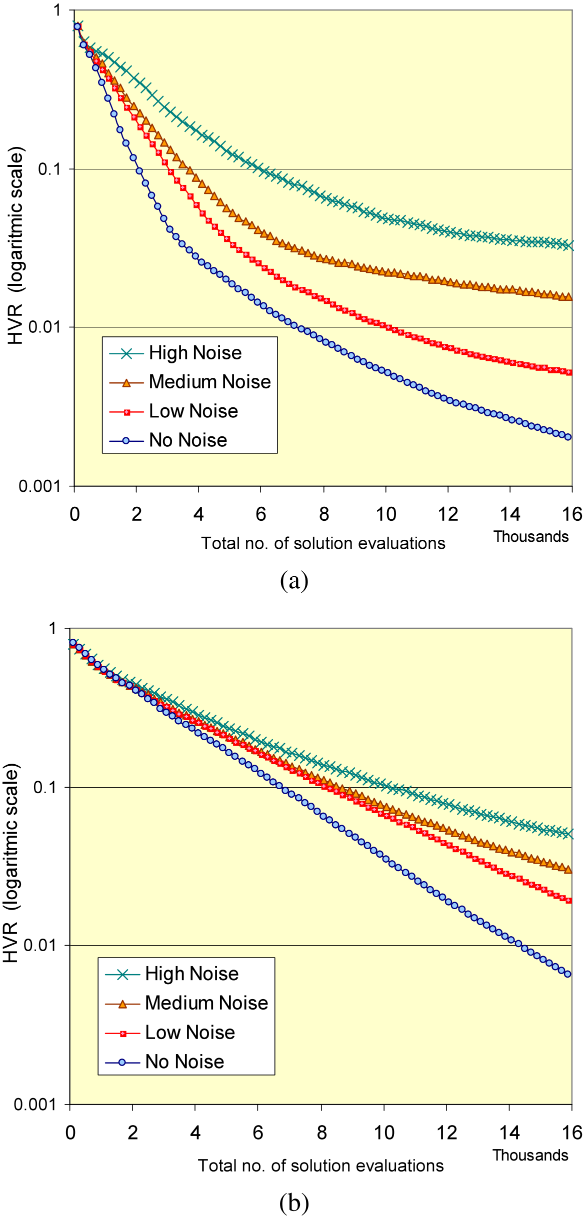


Fig. 18 *HVR* performance metric for different noise levels at different search iterations on ZDT6 (**a**) SPGA, and (**b**) NSGA-II

In this study, the number of sampling is kept fixed and quite large (15) for all solutions. Future research can study how the number of sampling can affect the performance of any algorithm and try to develop an efficient sampling scheme to find out a reasonable number of sampling for each solution. Intelligent sampling scheme

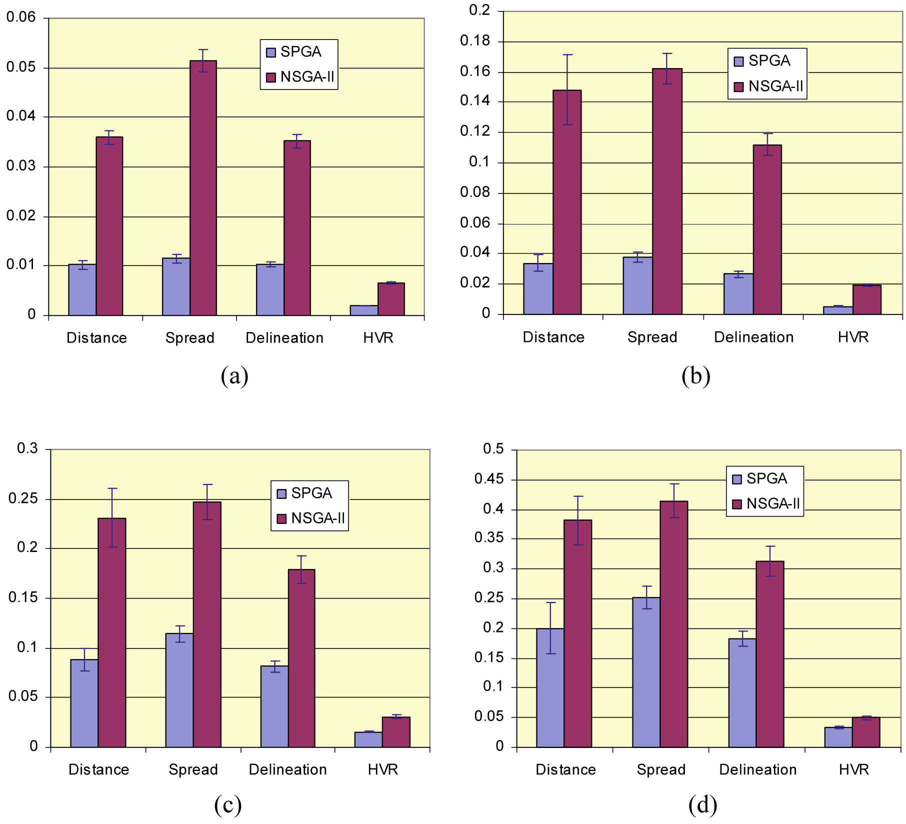


Fig. 19 Mean and 95% confidence interval of three performance metrics with SPGA and NSGA-II on ZDT4 for different noise levels. (**a**) no noise, (**b**) low noise, (**c**) medium noise and (**d**) high noise

can significantly save large number of samplings for many solutions and result in a more thorough exploration of the solution space.

References

Babbar, M., Lakshmikantha, A., Goldberg, D.E.: A modified NSGA-II to solve noisy multiobjective problems. In: 2003 Genetic and Evolutionary Computation Conference. Late-Breaking Papers, pp. 21–27. AAAI, Chicago (2003)

Basseur, M., Zitzler, E.: Handling uncertainty in indicator-based multiobjective optimization. Int. J. Comput. Intell. Res. **2**(3), 255–272 (2006)

Beyer, H.G.: Evolutionary algorithms in noisy environments: theoretical issues and guidelines for practice. Comput. Methods Appl. Mech. Eng. **186**, 239–267 (2000)

Borjesson, P.O., Sundberg, C.E.W.: Simple approximation of the error function $Q(x)$ for communications applications. IEEE Trans. Commun. **27**(3), 639–643 (1979)

Buche, D., Stoll, P., Dornberger, R., Koumoutsakos, P.: Multiobjective evolutionary algorithm for the optimization of noisy combustion processes. IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. **32**(4), 460–473 (2002)

Bui, L., Abbass, H., Essam, D., Green, D.: Performance analysis of evolutionary multi-objective optimization methods in noisy environments. In: Proceedings of the 8th Asia Pacific Symposium on Intelligent and Evolutionary Systems, pp. 29–39 (2004)

Bui, L.T., Hussein, A.A., Essam, D.: Fitness inheritance for noisy evolutionary multi-objective optimization. In: Beyer, H.-G. (ed.) Proceedings of the 2005 Genetic and Evolutionary Computation Conference, vol. 1, pp. 779–785. ACM, New York (2005)

Coello, C.A.C., Veldhuizen, D.A., Lamont, G.B.: Evolutionary Algorithms for Solving Multi-Objective Problems, 1st edn. Kluwer Academic, New York (2002)

Corne, D., Deb, K., Fleming, P., Knowles, J.: The good of the many outweighs the good of the one: evolutionary multiobjective optimization. coNNectionS **1**(1), 9–13 (2003)

Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms, 1st edn. Wiley, Chichester (2001)

Deb, K., Pratap, A., Agarval, S., Meyarivan, T.A.: Fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**, 182–197 (2002)

Deb, K., Mohan, M., Mishra, S.: Evaluating the epsilon-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions. Evol. Comput. **13**(4), 501–525 (2005)

Erbas, C., Cerav-Erbas, S., Pimentel, A.D.: Multiobjective optimization and evolutionary algorithms for the application mapping problem in multiprocessor system-on-chip design. IEEE Trans. Evol. Comput. **10**(3), 358–374 (2006)

Eskandari, H., Geiger, C.D.: A fast Pareto genetic algorithm approach for solving expensive multiobjective optimization problems. J. Heuristics **14**(3), 203–241 (2008)

Fieldsend, J.E., Everson, R.M.: Multi-objective optimization in the presence of uncertainty. In: 2005 IEEE Congress on Evolutionary Computation (CEC'2005), vol. 1, pp. 243–250. IEEE Service Center, Edinburgh (2005)

Fonseca, C.M., Flemming, P.J.: Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In: Proceedings of the Fifth International Conference on Genetic Algorithms, San Mateo, CA, pp. 416–423 (1993)

Goh, C.K., Tan, K.C.: An investigation on noisy environments in evolutionary multiobjective optimization. IEEE Trans. Evol. Comput. **11**(3), 354–381 (2007)

Goldberg, D.E., Deb, K., Clark, J.H.: Genetic algorithms, noise, and the sizing of populations. Complex Syst. **6**, 333–362 (1992)

Hughes, E.J.: Evolutionary multi-objective ranking with uncertainty and noise. In: First International Conference on Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science, vol. 1993, pp. 329–343. Springer, Berlin (2001)

Jin, Y., Branke, J.: Evolutionary optimization in uncertain environments—a survey. IEEE Trans. Evol. Comput. **9**(3), 303–318 (2005)

Joines, J., Gupta, D., Gokce, M.A., King, R.E., Kay, M.G.: Supply chain multi-objective simulation optimization. In: Proceedings of the 2002 Winter Simulation Conference, pp. 1306–1313. Institute of Electrical and Electronics Engineers, Piscataway (2002)

Knowles, J.: ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Trans. Evol. Comput. **10**(1), 50–66 (2006)

Knowles, J., Corne, D.: The Pareto archived evolution strategy: a new baseline algorithm for multiobjective optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation (CEC 1999), pp. 98–105. IEEE Service Center, Washington (1999)

Knowles, J.D., Corne, D.W.: On metrics for comparing nondominated sets. In: Proceedings of the 2002 Congress on Evolutionary Computation, vol. 1, pp. 711–716 (2002)

Liefooghe, A., Basseur, M., Jourdan, L., Talbi, E.: Combinatorial optimization of stochastic multi-objective problems: an application to the flow-shop scheduling problem. In: EMO 2006, pp. 457–471 (2007)

Lim, D., Ong, Y.S., Jin, Y., Sendhoff, B., Lee, B.S.: Inverse multi-objective robust evolutionary design. Gen. Program. Evol. Mach. **7**(4), 383–404 (2005)

Miller, B.L.: Noise, sampling, and efficient genetic algorithms. IlliGAL Report No. 97001, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL (1997)

Poles, S., Rigoni, E., Robic, T.: MOGA-II performance on noisy optimization problems. In: Proceedings of the International Conference on Bioinspired Optimization Methods and their Applications, pp. 51–62. Jozef Stefan Institute, Ljubljana (2004)

Safe, M., Carballido, J.A., Ponzoni, I., Brignole, N.B.: On stopping criteria for genetic algorithms. In: Advances in Artificial Intelligence, SBIA 2004, pp. 405–413 (2004)

Singh, A.: Uncertainty based multi-objective optimization of groundwater remediation design. Master's Thesis, University of Illinois at Urbana-Champaign (2003)

Singh, A., Minsker, B.S.: Uncertainty based multi-objective optimization of groundwater remediation at the umatilla chemical depot. In: American Society of Civil Engineers (ASCE) Environmental & Water Resources Institute (EWRI) World Water & Environmental Resources Congress 2004 & Related Symposia, Salt Lake City, UT (2004)
Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. Int. J. Evol. Comput. **2**(3), 221–248 (1994)
Teich, J.: Pareto-front exploration with uncertain objectives. In: Proceedings of the First Conference on Evolutionary Multi-Criterion Optimization (2001)
Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: empirical results. Evol. Comput. **8**(2), 173–195 (2000)
Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative study and strength Pareto approach. IEEE Trans. Evol. Comput. **3**(4), 257–271 (1999)
Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: improving the strength Pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland (2001)
Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans. Evol. Comput. **7**(2), 117–132 (2003)