A Preliminary Study on Handling Uncertainty in Indicator-Based Multiobjective Optimization

Matthieu Basseur[1] and Eckart Zitzler[2]

[1] LIFL/CNRS/INRIA, Bat M3 Cité Scientifique,
Villeneuve d'Ascq 59655, France
basseur@lifl.fr

[2] Computer Engineering and Networks Laboratory, ETH Zürich,
35 Gloriastrasse, Zurich 8092, Switzerland
zitzler@tik.ethz.ee.ch

Abstract. Real-world optimization problems are often subject to uncertainties, which can arise regarding stochastic model parameters, objective functions and decision variables. These uncertainties can take different forms in terms of distribution, bound and central tendency.

In the multiobjective context, several studies have been proposed to take uncertainty into account, and most of them propose an extension of Pareto dominance to the stochastic case. In this paper, we pursue a slightly different approach where the optimization goal is defined in terms of a quality indicator, i.e., an objective function on the set of Pareto set approximations. We consider the scenario that each solution is inherently associated with a probability distribution over the objective space, without assuming a 'true' objective vector per solution. We propose different algorithms which optimize the quality indicator, and preliminary simulation results indicate advantages over existing methods such as averaging, especially with many objective functions.

1 Motivation

Knowledge about the set of Pareto-optimal solutions is useful in many applications involving multiple objectives. Therefore, considerable research, particularly in the context of evolutionary computation, has been devoted to generating methods, i.e., techniques that try to generate the entire Pareto set or approximations of it. One recent approach of this type is indicator-based multiobjective optimization [1], which has the advantage that no additional diversity preservation mechanisms are required. Zitzler and Künzli [1] have demonstrated that this approach can be superior to popular algorithms such as SPEA2 and NSGA-II, with respect to the quality indicator under consideration.

Many real-world optimization problems are subject to uncertainties and therefore this aspect needs to be accounted for. Among the different types of uncertainty one can distinguish, cf. [2], we here consider the case that the determination of the objective function values is a stochastic process, i.e., every time a solution is evaluated, a different objective vector may be returned. Such a scenario emerges,

F. Rothlauf et al. (Eds.): EvoWorkshops 2006, LNCS 3907, pp. 727–739, 2006.

e.g., if the underlying computational model involves stochastic components such as Monte Carlo simulation.

While uncertainty in the objective functions gained some attention in the single-objective context [3, 2], only few studies address this problem within a multiple criteria setting. [4] were among the first to discuss uncertainty in the light of generating methods, although they did not propose a particular multiobjective optimizer for this purpose. Several years later, [5] and [6] independently proposed stochastic extensions of Pareto dominance and suggested similar ways to integrate probabilistic dominance in the fitness assignment procedure; both studies consider special types of probability distributions. In [7], another ranking method is proposed which is based on the average value per objective and the variance of the set of evaluations. Similarly, [8] suggested to consider for each dimension the mean over a given sample of objective vectors and to apply standard multiobjective optimizers for deterministic objective functions.

In this paper, we consider different scenarios for uncertain environments and propose and investigate several techniques to integrate uncertainties within the framework of indicator-based search, based on the algorithm presented in [1]; here, we focus on the ϵ_+ quality indicator [9]. In particular, the main contributions are:

- A probabilistic model that combines quality indicators and uncertainty;
- Different algorithms to integrate this model into the optimization process;
- Preliminaries experimentations on multiobjective test functions to compare these algorithms to existing ones.

The major differences to previous studies are (i) the investigation of uncertainty in the context of indicator-based multiobjective search and (ii) the more general perspective, as knowledge about the type of underlying probability distribution is not required.

2 Proposed Model: Combining Uncertainty and Binary Indicators

2.1 Indicator-Based Multiobjective Optimization

Let X denote the search space of the optimization problem under consideration and Z the corresponding objective space. Without loss of generality, we assume that $Z = \mathbb{R}^n$ and that all n objectives are to be minimized. In the absence of uncertainty, each $\mathbf{x} \in X$ is assigned exactly one objective vector $\mathbf{z} \in Z$ on the basis of a vector function $f : X \rightarrow Z$ with $\mathbf{z} = f(\mathbf{x})$. The mapping f defines the evaluation of a solution $\mathbf{x} \in X$, and often one is interested in those solutions that are Pareto optimal with respect to f.[1] However, generating the entire set of Pareto-optimal solutions is usually infeasible, e.g., due to the complexity of the underlying problem or the large number of optima. Therefore

[1] A solution $\mathbf{x} \in X$ is Pareto optimal if and only if there exists no $\mathbf{x}' \in X$ such that (i) $f(\mathbf{x}')$ is component-wise smaller than or equal to $f(\mathbf{x})$ and (ii) $f(\mathbf{x}') \neq f(\mathbf{x})$.

in many applications, the overall goal is to identify a good approximation of the Pareto-optimal set.

Different notions of what a good Pareto set approximation is are possible, and the definition of approximation quality strongly depends on the decision maker and the optimization scenario. We here assume that the optimization goal is given in terms of a binary quality indicator I, as proposed in [1]. A binary quality indicator, cf. [9], is a function $I : \mathcal{M}(Z) \times \mathcal{M}(Z) \rightarrow \mathbb{R}$, where $\mathcal{M}(Y)$ stands for the set of all possible multisets over Y, that can be regarded as a continuous extension of the concept of Pareto dominance to multisets of objective vectors. The value $I(A, B)$ quantifies the difference in quality between $A, B \in \mathcal{M}(Z)$. Now, if R denotes the set of Pareto-optimal solutions (or any other reference set), and $f(Y) := \{f(\mathbf{x}) \,|\, \mathbf{x} \in Y\}$, then the overall optimization goal can be defined as

$$\text{argmin}_{S \in \mathcal{M}(X)} \; I(f(S), f(R)) \tag{1}$$

2.2 Handling Uncertainty

In the following, the above optimization model will be extended to take uncertainty into account; later, we will discuss how to estimate and compute expected indicator values for uncertain environments.

As to uncertainty, the basic difference to the classical settings is that the vector function f does not represent a deterministic mapping from X to Z, but a stochastic process: every time a solution $\mathbf{x} \in X$ is evaluated using f, it may be mapped to a different objective vector $\mathbf{z} \in Z$. The higher the degree of uncertainty, the larger the variance among the objective vectors resulting from multiple, independent evaluations of $\mathbf{x}$. Thus, with each solution $\mathbf{x}$ a random variable $\mathcal{F}(\mathbf{x})$ is associated the range of which is Z; the underlying probability density function is usually unknown and may be different for other solutions.

Now, consider an arbitrary solution multiset $S = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\} \in \mathcal{M}(X)$. Based on the random variables $\mathcal{F}(\mathbf{x}_i)$ associated with the elements $\mathbf{x}_i$ of S, a corresponding random variable $\mathcal{F}(S)$ is defined for S which takes values in $\mathcal{M}(Z)$; $P(\mathcal{F}(S) = A)$ denotes the probability that (i) all members of S are mapped to elements of $A \in \mathcal{M}(Z)$ and (ii) there is at least one $\mathbf{x} \in S$ per $\mathbf{z} \in A$ for which $\mathbf{z} = f(\mathbf{x})$. Using this notation, we can now reformulate the optimization goal for uncertain environments as

$$\text{argmin}_{S \in \mathcal{M}(X)} \; E(I(\mathcal{F}(S), \mathcal{F}(R))) \tag{2}$$

where R is an arbitrary reference set from $\mathcal{M}(X)$ and $E(\cdot)$ stands for the expected value.

Note that there is a fundamental difference to other approaches, cf. [2]: we do not assume that there is a 'true' objective vector per solution which is blurred by noise; instead, we consider the scenario that each solution is inherently associated with a probability distribution over the objective space.

2.3 Estimating the Expected Indicator Value

If the probability density functions are known in advance and identical for all solutions $\mathbf{x} \in X$, then the expected value for any indicator can be computed according to

$$\begin{aligned} E(I(\mathcal{F}(S), \mathcal{F}(R))) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathrm{pdf}_{\mathcal{F}(S)\mathcal{F}(R)}(A, B) \cdot I(A, B) \; \mathrm{d}A\mathrm{d}B \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathrm{pdf}_{\mathcal{F}(S)}(A) \cdot \mathrm{pdf}_{\mathcal{F}(R)}(B) \cdot I(A, B) \; \mathrm{d}A\mathrm{d}B \end{aligned} \tag{3}$$

since $\mathcal{F}(S)$ and $\mathcal{F}(R)$ are independent from each other. Here, $\mathrm{pdf}_{\mathcal{F}(\cdot)}$ denotes the probability density function associated with the random variable $\mathcal{F}(\cdot)$.

However, in practice the underlying probability density functions are in general unknown, may vary for different solutions, and therefore can only be estimated by drawing samples. Let us assume that $\mathcal{S}(\mathbf{x}) \in \mathcal{M}(Z)$ represents a finite sample, i.e., a multiset of objective vectors, for solution $\mathbf{x}$. Now, the expected indicator value $E(I(\mathcal{F}(\mathbf{x}), \{A^*\})$ of $\mathcal{F}(\mathbf{x})$ with respect to a given set of objective vectors $\{\mathbf{z}_1^*, \ldots, \mathbf{z}_q^*\}$ can be estimated as follows

$$\hat{E}(I(\mathcal{F}(\mathbf{x}), \{\mathbf{z}_1^*, \ldots, \mathbf{z}_q^*\})) = \sum_{\mathbf{z} \in \mathcal{S}(\mathbf{x})} \frac{I(\{\mathbf{z}\}, \{\mathbf{z}_1^*, \ldots, \mathbf{z}_q^*\})}{|\mathcal{S}(\mathbf{x})|} \tag{4}$$

where $\hat{E}$ stands for the estimated expected value and $|\cdot|$ for the cardinality of a set. For a multiset S of solutions with $S = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$, the formula is

$$\hat{E}(I(\mathcal{F}(S), \{\mathbf{z}_1^*, \ldots, \mathbf{z}_q^*\}) = \sum_{\mathbf{z}_1 \in \mathcal{S}(\mathbf{x}_1)} \cdots \sum_{\mathbf{z}_m \in \mathcal{S}(\mathbf{x}_m)} \frac{I(\{\mathbf{z}_1, \ldots, \mathbf{z}_m\}, \{\mathbf{z}_1^*, \ldots, \mathbf{z}_q^*\})}{\prod_{1 \le i \le m} |\mathcal{S}(\mathbf{x}_i)|} \tag{5}$$

and if one considers a reference set R of solutions with $R = \{\mathbf{x}_1^*, \ldots, \mathbf{x}_r^*\}$, then the estimate amounts to

$$\hat{E}(I(\mathcal{F}(S), \mathcal{F}(R)) = \sum_{\mathbf{z}_1^* \in \mathcal{S}(\mathbf{x}_1^*)} \cdots \sum_{\mathbf{z}_r^* \in \mathcal{S}(\mathbf{x}_r^*)} \frac{\hat{E}(I(\mathcal{F}(S), \{\mathbf{z}_1^*, \ldots, \mathbf{z}_r^*\}))}{\prod_{1 \le i \le r} |\mathcal{S}(\mathbf{x}_i^*)|} \tag{6}$$

This approach is based on the assumption that the probability of a solution $\mathbf{x} \in X$ to be mapped to any objective vector $\mathbf{z}$ in the corresponding sample $\mathcal{S}(\mathbf{x})$ is uniformly distributed, i.e., $P(\mathcal{F}(\mathbf{x}) = \{\mathbf{z}\}) = 1/|\mathcal{S}(\mathbf{x})|$ for all $\mathbf{z} \in \mathcal{S}(\mathbf{x})$.

2.4 Computing Expected Indicator Values

Computing the expected indicator value for two multisets of solutions in the aforementioned manner is usually infeasible due to combinatorial explosion. Suppose each multiset contains 100 solutions with a sample size of 10 each, then equation 6 contains $100^{10} \cdot 100^{10} = 10^{40}$ summands. However, if particular properties of the indicator used can be exploited, then the exact calculation for $\hat{E}(\ldots)$ may become feasible. We here propose an algorithm for the (additive) ϵ-indicator [9] to compute the expected quality difference between a multiset $S \in \mathcal{M}(X)$

and a reference set R with one element only - for reference sets of arbitrary size the computation is still too expensive to be useful in practice. Later in Section 3 it will be discussed how this procedure can be integrated into an evolutionary algorithm.

For a minimization problem, the ϵ-indicator $I_{\epsilon+}$ is defined as follows:

$$\begin{aligned} I_{\epsilon+}(A,B) = \inf_{\epsilon \in \mathbb{R}}\{ \; & \forall \mathbf{z}_2 = (z_{2_1}, \ldots, z_{2_n}) \in B \; \exists \mathbf{z}_1 = (z_{1_1}, \ldots, z_{1_n}) \in A \; : \\ & \forall 1 \leq i \leq n \; : \; z_{1_i} \leq \epsilon + z_{2_i} \} \end{aligned} \tag{7}$$

It gives the minimum ϵ-value by which B can be moved in the objective space such that A is at least as good as B; a negative value implies that A is better than B in the Pareto sense. If B consists of a single objective vector $\mathbf{z}^*$, then the formula reduces to

$$I_{\epsilon+}(A, \{\mathbf{z}^*\}) = \inf_{\epsilon \in \mathbb{R}} \{\exists \mathbf{z}_1 = (z_{1_1}, \ldots, z_{1_n}) \in A \; : \forall 1 \leq i \leq n \; : \; z_{1_i} \leq \epsilon + z^*_i\} \tag{8}$$

Now, to compute $\hat{E}(I_{\epsilon+}(\mathcal{F}(S), \{\mathbf{z}^*\}))$ it is not necessary to consider all combinations of objective vectors to which the elements $\mathbf{x} \in S$ could be mapped to. Instead, one can exploit the fact that always the minimum $I_{\epsilon+}(\{\mathbf{x}\}, \{\mathbf{z}^*\}))$-value determines the actual indicator value. By sorting the objective vectors beforehand, it suffices to consider the ϵ-values in increasing order.

In detail, this works as follows. We consider all pairs $(\mathbf{x}_j, \mathbf{z}_k)$, where $\mathbf{x}_j \in S$ and $\mathbf{z}_k \in \mathcal{S}(\mathbf{x}_j)$, and sort them in increasing order regarding the indicator value $I_{\epsilon+}(\{\mathbf{z}_k\}\{\mathbf{z}^*\})$. Suppose the resulting order is $(\mathbf{x}_{j_1}, \mathbf{z}_{k_1}), (\mathbf{x}_{j_2}, \mathbf{z}_{k_2}), \ldots, (\mathbf{x}_{j_l}, \mathbf{z}_{k_l})$. Then, the estimate of the expected indicator value is

$$\begin{aligned} \hat{E}(I_{\epsilon+}(\mathcal{F}(S), \{\mathbf{z}^*\})) = \; & I_{\epsilon+}(\{\mathbf{z}_{k_1}\}, \{\mathbf{z}^*\}) \cdot P(\mathcal{F}(\mathbf{x}_{j_1}) = \mathbf{z}_{k_1}) \; + \\ & I_{\epsilon+}(\{\mathbf{z}_{k_2}\}, \{\mathbf{z}^*\}) \cdot P(\mathcal{F}(\mathbf{x}_{j_2}) = \mathbf{z}_{k_2} \,|\, \mathcal{F}(\mathbf{x}_{j_1}) \neq \mathbf{z}_{k_1}) \; + \\ & \ldots \\ & I_{\epsilon+}(\{\mathbf{z}_{k_l}\}, \{\mathbf{z}^*\}) \cdot P(\mathcal{F}(\mathbf{x}_{j_l}) = \mathbf{z}_{k_l} \,|\, \forall_{1 \leq i < l} \mathcal{F}(\mathbf{x}_{j_i}) \neq \mathbf{z}_{k_i}) \end{aligned}$$

It can be done more efficiently as soon as if all $I_{\epsilon+}$ values of the different elements of one solution are smaller than $\overline{\epsilon}$, then the remaining $I_{\epsilon+}$ values greater than $\overline{\epsilon}$, have a probability of 0. This scheme can be used, as detailed by Alg. 1.

Algorithm 1 (Estimation of the Expected ϵ-Indicator Value)

```
Input:   $S \in \mathcal{M}(X)$                              (multiset of decision vectors)
         $\mathbf{z}^* \in Z$                                  (reference objective vector)
Output:  $\hat{E}(I_{\epsilon+}(\mathcal{F}(S), \{\mathbf{z}^*\}))$    (estimate for the expectation value of $I_{\epsilon+}$)

Step 1:  Determine $\overline{\epsilon} = \min_{\mathbf{x} \in S} \max_{\mathbf{z} \in \mathcal{S}(\mathbf{x})} I_{\epsilon+}(\{\mathbf{z}\}, \{\mathbf{z}^*\})$
Step 2:  Set $L = \emptyset$. For each $\mathbf{x} \in S$ and $z \in \mathcal{S}(x)$ do:
           1. $\epsilon = I_{\epsilon+}(\{\mathbf{z}\}, \{\mathbf{z}^*\})$.
           2. If $\epsilon \leq \overline{\epsilon}$ then AppendToList$(L, (\epsilon, \mathbf{x}))$.
Step 3:  Sort $L$ in increasing order according to the $\epsilon$-values.
Step 4:  Set $\hat{E} = 0$. For each $\mathbf{x} \in X$ set $N[\mathbf{x}] = 0$.
Step 5: While $\mathsf{NotEmpty}(L)$ do:
        1. $(\epsilon', \mathbf{x}') = \mathsf{FirstElement}(L)$.
        2. $p = 1/(|\mathcal{S}(\mathbf{x}')| - N[\mathbf{x}']) \cdot \prod_{\mathbf{x} \in S} 1 - N[\mathbf{x}]/|\mathcal{S}(\mathbf{x})|$.
        3. $\hat{E} = \hat{E} + p \cdot \epsilon'$.
        4. $N[\mathbf{x}'] = N[\mathbf{x}'] + 1$ and $\mathsf{RemoveFirstElement}(L)$.
Step 6: Return $\hat{E}$.
```

3 Algorithm Design

In this section, we discuss on how to integrate algorithm 1 in multiobjective EAs in order to achieve the optimization goal defined in equation 2. The following discussion is based on [1].

In the general case, during the selection process of EAs, we are interested in the case that M solutions need to be removed from the current population, with the goal of maximizing the quality of the remaining solutions, in our case according to the binary indicator $I_{\epsilon+}$. But, clearly, this problem is NP-hard. Therefore, consider mainly the steady-state version ($M = 1$), which corresponds to an evolution strategy $ES(N + 1)$. According to a performance indicator, removing one individual can be solved optimally by deleting the solution $\mathbf{x}_w \in S$ which has the worst EIV value.

With the algorithm 1, the Estimation of the Expected ϵ-Indicator Value (EIV) is computed. Then, we are able to evaluate the quality of different populations against a reference set. As shown in equation 2.4, the algorithm can be used to evaluate the quality of a single solution against a reference set. The quality of a solution $\mathbf{x}_i \in S$ is measured by estimating EIV of $S \setminus \{\mathbf{x}_i\}$ against S, which corresponds to the lost of quality of S if we remove the solution $\mathbf{x}_i$. The general EA is detailed in algorithm 2.

Algorithm 2 (Steady-state $IBEA$ algorithm: EIV)

```
Input:  $N$    (population size)
        $G$    (maximum number of generations)
Output: $S$    (approximation set)

Step 1: Initialization: Generate an initial population $S$ of size $N$; set the generation
        counter $g$ to 0.
Step 2: Fitness assignment: Calculate fitness values of individuals in $S$, i.e., for all
        $\mathbf{x}_i \in S$ set $Fit(\mathbf{x}_i) = \hat{E}(I(\mathcal{F}(S), \mathcal{F}(\{\mathbf{x}_i\}))) = \frac{1}{|\mathcal{S}(\mathbf{x}_i)|} \sum_{\mathbf{z}_i \in \mathcal{S}(\mathbf{x}_i)} \hat{E}(I(\mathcal{F}(S), \{\mathbf{z}_i\}))$
Step 3: Environmental selection: Remove the individual $\mathbf{x}_w \in S$ with the smallest
        fitness value, i.e., $Fit(\mathbf{x}_w) \leq Fit(\mathbf{x})$ for all $\mathbf{x} \in S$.
Step 4: Termination: If $g \geq G$ then return $S$.
Step 5: Mating selection: Perform binary tournament selection on $S$.
Step 6: Variation: Apply recombination and mutation operators and insert the generated
        individual into the population $S$. Increment the generation counter
        ($g = g + 1$) and go to the Step 2.
```

Let us consider s evaluations for all $x^i \in S$. In order to realize this comparison, the epsilon values of each individual have to be sorted (see algorithm 1). Then the selection process, according to the EIV fitness assignment algorithm, has a complexity of $\theta(n(Ns)^2 log(Ns))$ (Ns solutions - Ns elements to sort for every solution to evaluate). In order to reduce this complexity, we develop two methods which approximate the ranking obtained with EIV algorithm.

The first approach consists in approximating the sorting step of EIV by a Bucket sort (BCK). c different values are first defined, with an uniform repartition on the definition interval $[I_{min}, I_{max}]$ of the performance indicator (we could also use the minimum and maximum values computed for the indicator). Then, during the Step 1 of algorithm 1, we compute an approximated indicator value $I_{\epsilon^+}^{BCK}$, to speed up the sorting step by the use of bucket sort. Let c_{range} be defined as $(I_{max} - I_{min})/c$ and $cell = (int)[\frac{(I_{\epsilon^+}(z,z^*) - I_{min})}{c_{range}}]$.

$$I_{\epsilon^+}^{BCK}(z, z^*) = I_{min} + cell * c_{range} \tag{9}$$

The maximum error of this approach is equal to $(I_{max} - I_{min})/c$. The algorithm complexity is in $\theta(nN^2 s(s + c))$.

The second approach consists in approximating the minimum value computed by EIV with an exponential function (Exp) applied on the different computed indicators values, as realized in [1], without uncertainty:

$$Fit(\mathbf{x}_1) = \sum_{\mathbf{z}_1 \in \mathcal{S}(\mathbf{x}_1)} \sum_{\mathbf{x}_2 \in S \backslash \{\mathbf{x}_1\}} \sum_{\mathbf{z}_2 \in \mathcal{S}(\mathbf{x}_2)} -e^{-I_{\epsilon^+}(\mathbf{z}_2, \mathbf{z}_1)/\kappa} \tag{10}$$

With one evaluation per solution, when *kappa* is close to 0, the corresponding ranking tends to be exactly a lexicographic sorting comparison between all computed indicator values. With several evaluations per solution, the probability of occurrence of each possible indicator is not considered here, but the computational complexity of the algorithm 2 is reduced to $\theta(n(Ns)^2)$.

To evaluate the different scheme proposed, we also propose two alternative algorithms. The first approach envisaged consists in approximate EIV fitness assignment function is the Averaging method (Avg). First, the average value is computed for each objective function, then the exact algorithm can be easily applied with $|\mathcal{S}(x)| = 1$, $\forall \mathbf{x} \in S$. In fact, in this case, we have the relation $\hat{E}(I(\mathcal{F}(S), \mathcal{F}(\{\mathbf{x}^*\})) = \hat{E}(I(\mathcal{F}(S), \{\mathbf{z}^*\})$, and:

$$Fit(\mathbf{x}^*) = \hat{E}(I_{\epsilon^+}(\mathcal{F}(S), \{\mathbf{z}^*\}) = min\{\mathbf{z} \in \mathcal{S}(\mathbf{x})\ I_{\epsilon^+}\{\mathbf{z}\}\{\mathbf{z}^*\}\} \tag{11}$$

Then, the algorithm complexity is in $\theta(Ns + N^2)$ (averaging step + indicator values computation).

We also implement the fitness assignment method proposed by Hughes [5], based on the Probabilistic Dominance Relation (PDR) between solutions:

$$Fit(\mathbf{x}^*) = \frac{1}{|\mathcal{S}(\mathbf{x}^*)|} * \sum_{\mathbf{z}^* \in \mathcal{S}(x^*)} \sum_{i=1}^{n} \sum_{x \in S \backslash x^*} \frac{1}{|\mathcal{S}(x)|} * \sum_{\mathbf{z} \in \mathcal{S}(x)} inf(\mathbf{z}_i, \mathbf{z}_i^*) \tag{12}$$

with $inf(\mathbf{z}_i, \mathbf{z}_i^*)$ equal to 0 (resp. 0.5, 1) if the i^{th} objective value of $\mathbf{z}$ is smaller (resp. equal, greater) than the i^{th} objective value of $\mathbf{z}^*$. The complexity of PDR fitness assignment algorithm is in $\theta(n * (Ns)^2)$.

For all the different schemes proposed in this section, with use algorithm 2. The fitness assignment step is replaced by the corresponding method. Then, for mating selection, we make a binary tournament between the solutions of S, without the deleted solution $\mathbf{x}_w$. To achieve the tournament step, we compare the solutions according to their fitness value, computed for the selection (which is not exactly the true fitness value, since one solution has been removed from the population).

4 Simulation Results

In the following, we investigate two questions concerning performance of the 5 different algorithms. First, we evaluate these algorithms for one selection step, and compare the loss of quality obtained by each method. secondly, comparison is done on entire runs on multiobjective tests functions.

We present only preliminary results. The uncertainty is defined with known bound, distribution and central tendency. Moreover, performance evaluation are realized by using the true objective vectors value of the output solutions. We would like to make evaluation based expected values as in equation 3, but it is really not feasible.

4.1 Environmental Selection

The test are done with the different approximative and exact selection methods previously described: EIV, BCK, Exp, Avg, PDR.

We evaluate the selection process on randomly generated Pareto population: 100 individuals are generated, with random value for each objective function. Then, we scale the values of each individual $\mathbf{x}$ to obtain $\sum_{i=1}^{n} f_i(\mathbf{x}) = n/2$. In the biobjective case, it corresponds to solutions on the diagonal $[\{1, 0\}, \{0, 1\}]$. Then, for each solution, we generate s different evaluations, by adding a random value (in an interval $[-\sigma, \sigma]$), for each objective vector. For each test, we generate 100 random populations, with 10 evaluations per solution, a uniform noise defined on the interval $[-0.05, 0.05]$, and two objective values. We evaluate the different methods by varying the number of objective functions, the sample size, or the level of uncertainty. The bucket sort was tested with $c = 50$.

Then, we first evaluate the selection process with the exact I_{ϵ^+} value computation, which determines the worst solution $\mathbf{x}_w$. Then, for each approximative fitness assignment algorithm i, we compute the worst solution $\mathbf{x}_{w_i}$. To evaluate the effectiveness of the approximation, we compute the difference, in terms of performance indicator, between the exact and the approximative approach: $I_{\epsilon^+}(S \setminus \{\mathbf{x}_{w_i}\}, S) - I_{\epsilon^+}(S \setminus \{\mathbf{x}_w\}, S)$ (smaller values are better).

The figures 1, 2 and 3 give the results obtained with the different selection methods (BCK, Exp, PDR and Avg) for 4 different number of objective functions, 3 different sample size, and 3 levels of uncertainty. The smaller values are

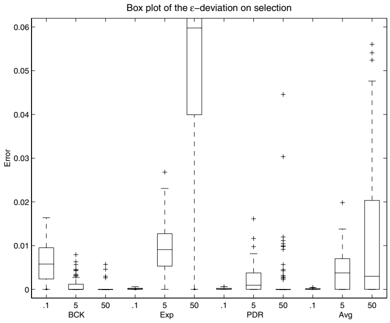

Fig. 1. Average selection error, with different levels of uncertainty

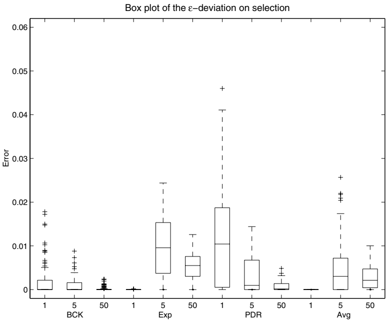

Fig. 2. Average selection error, with different number of evaluations

achieved by BCK in many cases, especially with an important level of uncertainty, many objectives, or a lot of evaluations. The other methods obtain small values only for small sample size or small level of uncertainty. The exponential approaches almost obtain the exact approach results, only when $s = 1$. In normal case, we suggest to use the exponential function approach, which is not expensive to compute and almost gives the optimal results. With uncertainty, BCK seems to be more effective.

4.2 Entire Optimization Runs

For the entire optimization runs, we consider 5 multi-objective test functions taken from the literature: ZDT1, ZDT6 [10], DTLZ2 [11], KUR [12] and COMET [13]. The number of decision variables has been fixed to 50 for all the test problems. Tests are realized by considering two different types of uncertainty: (1) an uniform-distributed random noise is applied on the evaluation functions, in a fixed interval $[-\sigma, \sigma]$; (2) a uniform-distributed random noise is applied on the

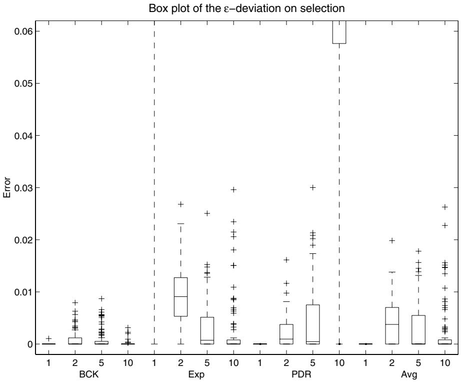

Fig. 3. Average selection error, with different number of objective functions

decision variables. The result is a variable noise, depending on the form of the objective space around the envisaged solution.

The population size N is set to 50, with $s = 5$ evaluations for each solution. Uniform repartition is applied for the two types of uncertainty (i.e. on decision or on objective space). The maximum number of generations is set to 5000. We perform 30 runs for each problem. The different methods are tested with the same initial populations. The other parameters used, such as mutation and recombination operators are those used in [1].

To evaluate the effectiveness of each method, we generate the 'true' objective vector for each solution. Then, for each approximation A, we compute $\hat{E}(I_{\epsilon^+}(R, A))$ value, where R is the reference set, determined by merging all solutions found during the experimentations, and keeping only the non-dominated evaluations. The comparison of the whole set of runs is realized using the Mann-

Table 1. Comparison of the different selection methods for the I_{ϵ^+}-indicator using the Mann-Whitley statistical test: P value, with noise on objective vectors (Z) and on decision vectors (X) - 2 objective problems. A cell 10^{-a} corresponds to a significance level in the interval $[10^{-a}, 10^{-(a-1)}]$.

		EIV		**BCK**		**Exp**		**Avg**		**PDR**	
		Z	X	Z	X	Z	X	Z	X	Z	X
DTLZ2	**EIV**			> 5%	> 5%	10^{-11}	10^{-11}	> 5%	10^{-6}	> 5%	> 5%
	BCK	10^{-4}	> 5%			10^{-11}	10^{-11}	10^{-3}	10^{-3}	> 5%	> 5%
	Exp	> 5%	> 5%	> 5%	> 5%			> 5%	> 5%	> 5%	> 5%
	Avg	> 5%	> 5%	> 5%	> 5%	10^{-11}	10^{-10}			> 5%	> 5%
	PDR	10^{-9}	10^{-4}	10^{-5}	10^{-4}	10^{-11}	10^{-11}	10^{-8}	10^{-8}		
ZDT1	**EIV**			> 5%	> 5%	10^{-4}	10^{-7}	> 5%	> 5%	10^{-5}	> 5%
	BCK	> 5%	> 5%			10^{-5}	10^{-8}	> 5%	> 5%	10^{-7}	> 5%
	Exp	> 5%	> 5%	> 5%	> 5%			> 5%	> 5%	> 5%	> 5%
	Avg	> 5%	> 5%	> 5%	> 5%	10^{-5}	10^{-7}			10^{-6}	> 5%
	PDR	> 5%	> 5%	> 5%	> 5%	> 5%	10^{-11}	> 5%	> 5%		
ZDT6	**EIV**			> 5%	> 5%	10^{-11}	> 5%	> 5%	> 5%	> 5%	> 5%
	BCK	> 5%	> 5%			10^{-11}	> 5%	> 5%	10^{-3}	> 5%	> 5%
	Exp	> 5%	10^{-5}	> 5%	10^{-3}			> 5%	10^{-7}	> 5%	10^{-2}
	Avg	10^{-2}	> 5%	10^{-4}	> 5%	10^{-11}	> 5%			> 5%	> 5%
	PDR	10^{-3}	> 5%	10^{-5}	> 5%	10^{-11}	> 5%	> 5%	10^{-3}		
KUR	**EIV**			> 5%	> 5%	> 5%	10^{-4}	> 5%	10^{-4}	> 5%	> 5%
	BCK	> 5%	> 5%			> 5%	> 5%	> 5%	10^{-2}	> 5%	> 5%
	Exp	> 5%	> 5%	> 5%	> 5%			> 5%	> 5%	> 5%	> 5%
	Avg	> 5%	> 5%	> 5%	> 5%	> 5%	> 5%			> 5%	> 5%
	PDR	> 5%	> 5%	> 5%	> 5%	> 5%	10^{-2}	> 5%	10^{-3}		
COMET	**EIV**			10^{-3}	> 5%	> 5%	10^{-11}	> 5%	> 5%	10^{-4}	10^{-11}
	BCK	> 5%	> 5%			> 5%	10^{-11}	> 5%	> 5%	> 5%	10^{-11}
	Exp	> 5%	> 5%	10^{-2}	> 5%			> 5%	> 5%	10^{-3}	10^{-10}
	Avg	> 5%	> 5%	> 5%	> 5%	> 5%	10^{-11}			10^{-2}	10^{-11}
	PDR	> 5%	> 5%	> 5%	> 5%	> 5%	> 5%	> 5%	> 5%		

Whitley statistical test, applied on the sets of $\hat{E}(I_{\epsilon^+}(R, A))$ values computed for each method.

Table 1 and 2 represents the comparison of the different selection methods for the $\hat{E}(I_{\epsilon^+})$ with the two different types of uncertainty: on objective vectors, and on decision variables. To compare the sets of runs, we use the Mann-Whitley statistical test, as described in [14]. The columns give the adjusted P value of the corresponding pairwise test that accounts for multiple testings; it equals to the lowest significance level for which the null-hypothesis (the medians are drawn from the same distribution) would still be rejected (with a significance level of 5%). A value under 5% shows that the method in the corresponding row is significantly better than the method in the corresponding column.

In many cases, the results are not significant in the bi-objective case, since the different approaches are similar, i.e. they use the same ϵ-indicator-based fitness assignment. But some conclusions could be extracted from table 1:

- The exponential approximation approach Exp, give worst results in many cases, excepted for KUR and $COMET$ instances.
- BCK and EIV obtain similar results, which shows the efficiency of BCK to approximate EIV fitness assignment method.
- Uncertainty on objective vectors: in many cases, ϵ-indicator-based approaches Avg, BCK and EIV perform significantly better than Hughes selection mechanism PDR, especially for $COMET$ and $ZDT1$ instances.
- Uncertainty on decision variables: Avg results are significantly worst than EIV, BCK and PDR, in many cases (problems $DTLZ2$, $ZDT6$ and KUR).

In table 2, we represent the results obtained for experimentations realized on $DTLZ2$ test function, with different number of objectives. This table shows a superior performance of EIV, BCK and Exp fitness assignment methods when

Table 2. Evaluation with several number of objectives to optimize: $DTLZ2$ test function, using the Mann-Whitley statistical test: P value, with noise on objective vectors (Z) and on decision vectors (X). A cell 10^{-a} corresponds to a significance level in the interval $[10^{-a}, 10^{-(a-1)}]$.

		EIV		**BCK**		**Exp**		**Avg**		**PDR**	
		Z	X	Z	X	Z	X	Z	X	Z	X
n=5	**EIV**			$> 5\%$	$> 5\%$	10^{-8}	10^{-9}	10^{-2}	10^{-7}	10^{-11}	10^{-11}
	BCK	$> 5\%$	$> 5\%$			10^{-9}	10^{-9}	10^{-2}	10^{-7}	10^{-11}	10^{-11}
	Exp	$> 5\%$	$> 5\%$	$> 5\%$	$> 5\%$			$> 5\%$	$> 5\%$	10^{-11}	10^{-10}
	Avg	$> 5\%$	$> 5\%$	$> 5\%$	$> 5\%$	10^{-6}	10^{-2}			10^{-11}	10^{-11}
	PDR	$> 5\%$	$> 5\%$	$> 5\%$	$> 5\%$	$> 5\%$	$> 5\%$	$> 5\%$	$> 5\%$		
n=10	**EIV**			$> 5\%$	$> 5\%$	10^{-2}	10^{-7}	10^{-2}	10^{-7}	10^{-11}	10^{-11}
	BCK	$> 5\%$	$> 5\%$			$> 5\%$	10^{-8}	10^{-2}	10^{-8}	10^{-11}	10^{-11}
	Exp	$> 5\%$	$> 5\%$	$> 5\%$	$> 5\%$			$> 5\%$	$> 5\%$	10^{-11}	10^{-11}
	Avg	$> 5\%$	$> 5\%$	$> 5\%$	$> 5\%$	$> 5\%$	$> 5\%$			10^{-11}	10^{-11}
	PDR	$> 5\%$	$> 5\%$	$> 5\%$	$> 5\%$	$> 5\%$	$> 5\%$	$> 5\%$	$> 5\%$		

the number of objective functions is growing. *Exp* is dominated by *EIV* and/or *BCK* in several cases, especially for the 5 objectives instance.

5 Discussion

In this paper, we propose a method for handling uncertainty in indicator-based evolutionary algorithm. Our approach tries to make no assumption about distribution, bounds and general tendency of the uncertainty. We propose the algorithm *EIV*, which computes the exact expected value of ϵ-indicator. In order to apply this algorithm to environmental selection in EAs, we propose several algorithms which approximate the results obtained by *EIV*, which select the best possible solutions during environmental selection, according to the ϵ-indicator performance metric. We have made several experimentations. First, we consider the goal to minimize the loss in quality during environmental selection: *BCK* give a good approximation of *EIV* selection, which is more time consuming. Then, we made some experimentations on classical tests functions. Our proposed method give interesting results with an increasing number of objective functions. This are preliminary results. More experimentations are needed to evaluate the different approaches on different problems and uncertainties. The first experimentation done for environmental selection process, with different levels of uncertainties, number of objective functions and sample size, show that the quality of the selected individuals, according to the I_{ϵ^+} indicator, is improved by the use of *EIV* or *BCK* fitness assignment method. We can expect that entire runs results will show the same tendency, but further experimentations has to be done. We also need to define new comparison test, which involve a set of expected objective vectors values, without knowledge about the true objective vector.

References

1. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Proc. 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII), Birmingham, UK (2004) 832–842
2. Jin, Y., Branke, J.: Evolutionary optimization in uncertain environments - a survey. IEEE Transactions on evolutionary computation **9** (2005) 303–317
3. Arnold, D.V.: A comparison of evolution strategies with other direct search methods in the presence of noise. Computational Optimization and Applications **24** (2003) 135–159
4. Horn, J., Nafpliotis, N.: Multiobjective optimization using the niched pareto genetic algorithm. Technical report, University of Illinois, Urbana-Champaign, Urbana, Illinois, USA (1993)
5. Hughes, E.: Evolutionary multi-objective ranking with uncertainty and noise. In: EMO'01: Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization, London, UK, Springer-Verlag (2001) 329–343
6. Teich, J.: Pareto-front exploration with uncertain objectives. In: Conference on Evolutionary Multi-Criterion Optimization (EMO'01). Volume 1993 of Lecture Notes in Computer Science (LNCS). (2001) 314–328

7. Babbar, M., Lakshmikantha, A., Goldberg, D.E.: A modified NSGA-II to solve noisy multiobjective problems. In et al., E.C., ed.: Genetic and Evolutionary Computation Conference (GECCO'2003), late breaking papers. Volume 2723 of Lecture Notes in Computer Science., Chicago, Illinois, USA, Springer (2003) 21–27
8. Deb, K., Gupta, H.: Searching for robust pareto-optimal solutions in multiobjective optimization. In Coello Coello, C.A., Aguirre, A.H., Zitzler, E., eds.: Conference on Evolutionary Multi-Criterion Optimization (EMO'05). Volume 3410 of Lecture Notes in Computer Science (LNCS)., Guanajuato, Mexico, Springer-Verlag (2005) 150–164
9. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation **7** (2003) 117–132
10. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation **8** (2000) 173–195
11. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multi-objective optimization. Technical report, TIK Report Nr. 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich (2001)
12. Kursawe, F.: A variant of evolution strategies for vector optimization. In Schwefel, H.P., Männer, R., eds.: Parallel Problem Solving from Nature, Springer (1991) 193–197
13. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multi-objective optimization. In Abraham, A., et al., eds.: Evolutionary Computation Based Multi-Criteria Optimization: Theoretical Advances and Applications. Springer (2004) To appear.
14. Knowles, J.D., Thiele, L., Zitzler, E.: A tutorial on the performance assessment of stochastive multiobjective optimizers. Technical Report TIK-Report No. 214, Computer Engineering and Networks Laboratory, ETH Zurich (2005)