



**HAL**  
open science

# Combinatorial Optimization of Stochastic Multi-objective Problems: an Application to the Flow-shop Scheduling Problem

Arnaud Liefoghe, Matthieu Basseur, Laetitia Jourdan, El-Ghazali Talbi

► **To cite this version:**

Arnaud Liefoghe, Matthieu Basseur, Laetitia Jourdan, El-Ghazali Talbi. Combinatorial Optimization of Stochastic Multi-objective Problems: an Application to the Flow-shop Scheduling Problem. Evolutionary Multi-criterion Optimization (EMO 2007), Feb 2007, Matsushima, Japan. pp.457–471, 10.1007/978-3-540-70928-2\_36 . inria-00269974

**HAL Id: inria-00269974**

**<https://inria.hal.science/inria-00269974>**

Submitted on 3 Apr 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Combinatorial Optimization of Stochastic Multi-objective Problems: an Application to the Flow-Shop Scheduling Problem

Arnaud Liefoghe, Matthieu Basseur, Laetitia Jourdan, and El-Ghazali Talbi

INRIA Futurs, Laboratoire d'Informatique Fondamentale de Lille (LIFL), CNRS  
Bât. M3, Cité Scientifique, 59655 Villeneuve d'Ascq cedex, France  
{liefogo,asseur,jourdan,talbi}@lifl.fr

**Abstract.** The importance of multi-objective optimization is globally established nowadays. Furthermore, a great part of real-world problems are subject to uncertainties due to, *e.g.*, noisy or approximated fitness function(s), varying parameters or dynamic environments. Moreover, although evolutionary algorithms are commonly used to solve multi-objective problems on the one hand and to solve stochastic problems on the other hand, very few approaches combine simultaneously these two aspects. Thus, flow-shop scheduling problems are generally studied in a single-objective deterministic way whereas they are, by nature, multi-objective and are subjected to a wide range of uncertainties. However, these two features have never been investigated at the same time.

In this paper, we present and adopt a proactive stochastic approach where processing times are represented by random variables. Then, we propose several multi-objective methods that are able to handle any type of probability distribution. Finally, we experiment these methods on a stochastic bi-objective flow-shop problem.

**Key words:** multi-objective combinatorial optimization, stochasticity, evolutionary algorithms, flow-shop, stochastic processing times

## 1 Introduction

A large part of concrete optimization problems are subject to uncertainties that have to be taken into account. Therefore, many works relate to optimization in stochastic environments (see [10] for an overview), but very few deal with the multi-objective case where Pareto dominance is used to compare solutions. Thus, Hughes [8] and Teich [18] independently suggested to extend the concept of Pareto dominance in a stochastic way by replacing the rank of a solution by its probability of being dominated; but both studies make an assumption on probability distributions. In [1], another ranking method, based on an average value per objective and on the variance of a set of evaluations, is presented. Likewise, Deb and Gupta [5] proposed to apply standard deterministic multi-objective optimizers using an average value, determined over a sample of objective vectors, for each dimension of the objective space. Finally, Basseur and Zitzler [2]

recently extended the concept of multi-objective optimization using quality indicators [21] to take stochasticity into account. However, even if existing methods are generally adaptable to the combinatorial case, most of them were only tested on continuous mathematical test functions. Thence, it is not obvious that the performances of these algorithms are similar for combinatorial and continuous problems. Furthermore, a large part of these algorithms exploits problem knowledge that may not be available in real-world applications.

The deterministic indicator-based approach [21] consists in assigning each Pareto set approximation a real value reflecting its quality, using a function  $I$  [20]. The goal is then to identify a Pareto set approximation that optimizes  $I$ . As a result,  $I$  induces a total order into the set of approximation sets in the objective space, and gives rise to a total order into the corresponding objective vectors. The interest of this perception is that no additional diversity preservation mechanisms are required, the concept of Pareto dominance not being directly used for fitness assignment. To extend this approach to the stochastic case, we must consider that every solution can be associated to an arbitrary probability distribution over the objective space.

In this paper, we propose various models to represent stochasticity for a bi-objective flow-shop scheduling problem. Then, we introduce different ways to handle uncertainty, insisting on the various technical aspects. And, we apply the resulting methods to the concrete case of a flow-shop scheduling problem with stochastic processing times, that have, to our knowledge, never been investigated in a multi-objective form. Each approach has advantages and drawbacks and is adapted from indicator-based optimization.

The paper is organized as follows. In section 2, we formulate a bi-objective flow-shop scheduling problem with stochastic processing times (SFSP). In section 3, we present three different approaches dedicated to stochastic multi-objective optimization and apply them on a SFSP. Section 4 presents experimental results. And finally, the last section draws conclusion and suggests further topics in this research area.

## **2 A bi-objective flow-shop scheduling problem with stochastic processing times**

The flow-shop is one of the numerous scheduling problems. It has been widely studied in the literature (see, for example, [6] for a survey). However, the majority of works dedicated to this problem considers it on a deterministic single-criterion form and mainly aims at minimizing the makespan, which is the completion time of the last job. Following the formulation of the deterministic model of a bi-objective flow-shop scheduling problem, this section presents various sources of uncertainty that have to be taken into account and introduces different probability distributions to model stochastic processing times. Note that, although this part focuses on the flow-shop, it can easily be generalizable to other types of problem.

## 2.1 Deterministic model

Solving the flow-shop problem consists in scheduling  $N$  jobs  $J_1, J_2, \dots, J_N$  on  $M$  machines  $M_1, M_2, \dots, M_M$ . Machines are critical resources, *i.e.* two jobs cannot be assigned to one machine simultaneously. A job  $J_i$  is composed of  $M$  consecutive tasks  $t_{i1}, t_{i2}, \dots, t_{iM}$ , where  $t_{ij}$  is the  $j^{\text{th}}$  task of the job  $J_i$ , requiring the machine  $M_j$ . A processing time  $p_{ij}$  is associated to each task  $t_{ij}$  and a job  $J_i$  must be achieved before its due date  $d_i$ . For the permutation flow-shop, the operating sequences of the jobs are identical and unidirectional on every machines.

In this study, we focus on minimizing both the makespan ( $C_{max}$ ) and the total tardiness ( $\bar{T}$ ), which are two of the most studied objectives of the literature. For each task  $t_{ij}$  to be scheduled at the time  $s_{ij}$ , we can compute the two considered objectives as follows:

$$C_{max} = \max_{i \in [1..N]} [s_{iM} + p_{iM}]. \quad (1)$$

$$\bar{T} = \sum_{i=1}^N \max(0, s_{iM} + p_{iM} - d_i). \quad (2)$$

In the Graham et al. notation [7], this problem is noted  $F/permu, d_j / (C_{max}, \bar{T})$ . Besides, the interested reader is referred to [14, 16] for a review on multi-objective scheduling.

## 2.2 Sources of uncertainty

In real-world scheduling situations, uncertainty can occur from many sources such as release or due date variations, machine breakdowns, unexpected arrival or cancellation of orders, variable processing times, ... According to the literature, in the particular case of our permutation flow-shop scheduling problem, the uncertainty mainly stem from due dates and processing times.

Firstly, in the deterministic model, the due date of a job  $J_i$  is given by a fixed number  $d_i$ . However, it looks difficult to determine it without ambiguity. So, it seems more natural to determine it using an interval  $[d_i^1, d_i^2]$  during which the human satisfaction for the completion of the job  $J_i$  decreases between  $d_i^1$  and  $d_i^2$ . Moreover, a due date  $d_i$  may change dynamically since a less important job today may be of high importance tomorrow, and vice versa. Secondly, a processing time may vary from an execution to another and some unexpected events may occur during the process. So, the processing time  $p_{ij}$  of a task  $t_{ij}$  rarely corresponds to a constant value. To conclude, it is obvious that no parameter can be regarded as an exact and precise data and that non-classical approaches are required to solve concrete scheduling problems. Thus, we decide to adopt a proactive stochastic approach where processing time values are regarded as uncertain and are represented by random variables.

## 2.3 Stochastic models

Widely studied in its single-criterion form, the stochastic flow-shop scheduling problem has, to our knowledge, never been investigated in a multi-objective

way. Furthermore, as soon as historic data about processing times are available, it seems quite easy to determine which probability distribution is associated to those parameters. Following an analysis, we propose four different general distributions a processing time may follow. Of course, a rigorous statistical analysis, based on real data, is imperative to determine the concrete and exact distribution associated to a certain processing time  $p_{ij}$  of a real-world problem.

*Uniform distribution.* A processing time  $p_{ij}$  can uniformly be included between two values  $a$  and  $b$ . Then,  $p_{ij}$  follows a uniform distribution over the interval  $[a, b]$  and its probability density function is:

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{if } x \in [a, b] \\ 0 & \text{otherwise} \end{cases} . \quad (3)$$

This kind of distribution is used to provide a simplified model of real industrial cases. For example, it has already been used by Kouvelis et al. [12].

*Exponential distribution.* A processing time  $p_{ij}$  may follow an exponential distribution  $\mathcal{E}(\lambda, a)$ . Thus, its probability density function is:

$$f(x) = \begin{cases} \lambda e^{-\lambda(x-a)} & \text{if } x \geq a \\ 0 & \text{otherwise} \end{cases} . \quad (4)$$

Exponential distributions are commonly used to model random events that may occur with uncertainty. This is typically the case when a machine is subject to unpredictable breakdowns. For example, processing times have been modeled by an exponential distribution in [3, 13].

*Normal distribution.* A processing time  $p_{ij}$  may follow a normal distribution  $\mathcal{N}(\mu, \sigma)$  where  $\mu$  stands for the mean and  $\sigma$  stands for the standard deviation, in which case its probability density function is:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) . \quad (5)$$

This kind of distribution is especially usual when human factors are observed. A process may also depend on unknown or uncontrollable factors and some parameters can be described in a vague or ambiguous way by the analyst. Therefore, processing times vary according to a normal distribution.

*Log-normal distribution.* A random variable  $X$  follows a log-normal distribution with parameters  $\mu$  and  $\sigma$  if  $\log X$  follows a normal distribution  $\mathcal{N}(\mu, \sigma)$ . Its probability density function is then:

$$f(x) = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}} \frac{1}{x} \exp\left(-\frac{1}{2}\left(\frac{\log x - \mu}{\sigma}\right)^2\right) & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} . \quad (6)$$

The log-normal distribution is often used to model the influence of uncontrolled environmental variables. In our case, a processing time  $p_{ij}$  following a log-normal distribution takes into account simultaneously the whole observed uncertainties. For example, this modeling has already been used in [4].

### 3 Indicator-based evolutionary methods

This section contains a brief presentation of the indicator-based approach introduced in [21] (the interested reader will refer to this article for more details). Then, we present its extension to the stochastic case and propose three multi-objective methods that result from this extension.

#### 3.1 Indicator-based multi-objective optimization

Let us consider a generic multi-objective optimization problem defined by a decision space  $X$ , an objective space  $Z$ , and  $n$  objective functions  $f_1, f_2, \dots, f_n$ . Without loss of generality, we here assume that  $Z \subseteq \mathbb{R}^n$  and that all  $n$  objective functions are to be minimized. In the deterministic case, to each solution  $x \in X$  is assigned exactly one objective vector  $z \in Z$  on the basis of a vector function  $F : X \rightarrow Z$  with  $z = F(x) = f_1(x), f_2(x), \dots, f_n(x)$ . The mapping  $F$  defines the ‘true’ evaluation of a solution  $x \in X$ , and the goal of a deterministic multi-objective algorithm is to approximate the set of Pareto optimal solutions according to  $F^1$ . However, generating the entire set of Pareto optimal solutions is usually infeasible, due to, *e.g.*, the complexity of the underlying problem or the large number of optima. Therefore, in many applications, the overall goal is to identify a good approximation of the Pareto optimal set. The entirety of all Pareto set approximations is represented by  $\Omega$ .

Different interpretations of what a good Pareto set approximation is are possible, and the definition of approximation quality strongly depends on the decision maker preferences and the optimization scenario. As proposed in [21], we here assume that the optimization goal is given in terms of a binary quality indicator  $I : \Omega \times \Omega \rightarrow \mathbb{R}$ . Then,  $I(A, B)$  quantifies the difference in quality between two sets  $A$  and  $B \in \Omega$ , according to the decision maker preferences. So, if  $R$  denotes the set of Pareto optimal solutions, the overall optimization goal can be formulated as  $\operatorname{argmin}_{A \in \Omega} I(F(A), F(R))$ . Binary quality indicators represent a natural extension of the Pareto dominance relation and can thus directly be used for fitness assignment. Therefore, the fitness of a solution  $x$  contained in a set of solutions can be determined using the indicator values obtained by  $x$  compared to the whole set. It measures the usefulness of  $x$  according to the optimization goal.

#### 3.2 Handling stochasticity

In the stochastic case, the objective values are different each time a solution is evaluated. So, the vector function  $F$  does not represent a deterministic mapping from  $X$  to  $Z$ , because an infinite set of different objective vectors is now assigned to a solution  $x \in X$ . Note that we consider that the ‘true’ objective vector of a solution is absolutely unknown before the end of the optimization process.

---

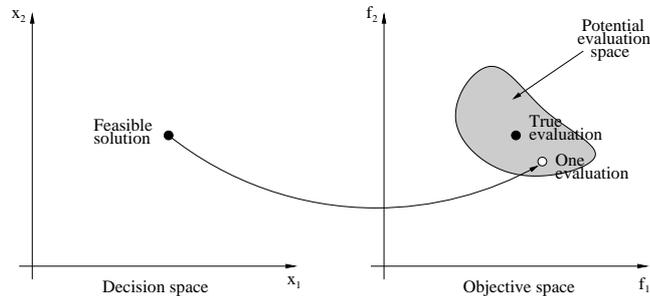
<sup>1</sup> A solution  $x_1 \in X$  is Pareto optimal if and only if there exists no  $x_2 \in X$  such that (i)  $F(x_2)$  is component-wise smaller than or equal to  $F(x_1)$  and (ii)  $F(x_2) \neq F(x_1)$ .

### 3.3 Proposed methods

To tackle the optimization of stochastic multi-objective problems, we here propose three different adaptations inspired by a multi-objective evolutionary algorithm designed for deterministic problems and recently introduced by Zitzler and Künzli [21], namely *IBEA* (Indicator-Based Evolutionary Algorithm). For each algorithm, we will use the additive  $\epsilon$ -indicator [20, 21] as the binary performance measure needed in the selection process of *IBEA*. This indicator seems to be efficient [21] and obtained significantly better results on our problem in its deterministic form than the  $I_{HD}$ -indicator (that is based on the hypervolume concept introduced in [19]). The additive  $\epsilon$ -indicator ( $I_{\epsilon+}$ ) gives the minimum  $\epsilon$ -value by which  $B$  can be moved in the objective space such that  $A$  is at least as good as  $B$ . For a minimization problem, it is defined as follows [20, 21]:

$$I_{\epsilon+}(A, B) = \inf_{\epsilon \in \mathbb{R}} \{ \forall x_2 \in B, \exists x_1 \in A : f_i(x_1) - \epsilon \leq f_i(x_2), i \in \{1, \dots, n\} \} . \quad (7)$$

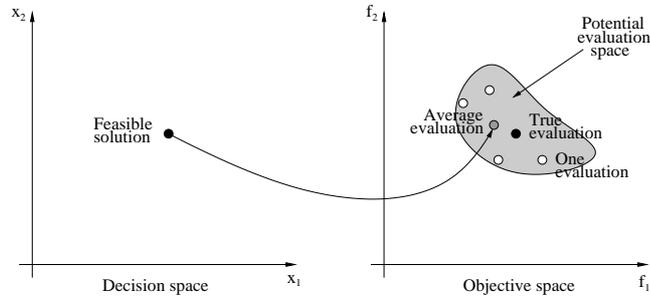
*Single evaluation-based estimate.* The first method,  $IBEA_1$ , consists in preserving the approach used in the deterministic case. A solution is evaluated only once and its fitness is estimated using this single evaluation (see fig. 1). Actually, most of the methods proceed like that since they are based on constant parameters and do not take uncertainties into account. The advantage of this method is its low computation cost, but the estimation error may be large since the evaluation used is not necessarily representative of the potential evaluation space.



**Fig. 1.**  $IBEA_1$ : a single evaluation is used to approximate the fitness of a solution.

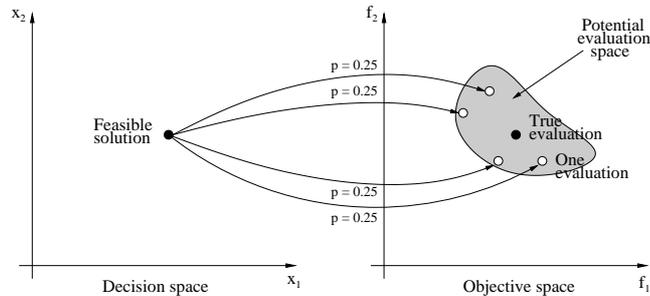
*Average estimate.* The second method, called  $IBEA_{avg}$ , follows the idea commonly used in the single-criterion form and suggested in several multi-objective studies (such as, *e.g.*, [1, 5]). It consists in doing several evaluations of the same solution, and then in calculating the average value of these evaluations on each objective function. Next, the deterministic approach is applied using those average values (see fig. 2). This method also has the advantage of having a low computation cost if the evaluation of a solution is not too expensive (what is

the case for our problem). However, losses of information may occur during the average estimate, like, e.g., the potential evaluations distribution in the research space.



**Fig. 2.**  $IBEA_{avg}$ : the average evaluation values are used to approximate the fitness of a solution.

*Probabilistic estimate.* The last method consists in estimating the fitness of a solution in a probabilistic way. Here, contrary to some other approaches [10], we do not assume that there is a ‘true’ objective vector per solution which is blurred by noise, but we consider that a probability distribution is associated to each solution on the objective space (see fig. 3). In order to allow the comparison of potential values of solutions, this extension of  $IBEA$ , called  $IBEA_{stoch}$ , consists in modifying the performance assessment procedure as proposed by Basseur and Zitzler in [2].



**Fig. 3.**  $IBEA_{stoch}$ : the fitness of a solution is estimated in a probabilistic way, a quality indicator being associated to each evaluation.

A random variable  $\mathcal{F}(x)$  is associated to each solution  $x \in X$  by the range of which is its corresponding potential evaluation space. The underlying probability

distribution is usually unknown and may differ for other solutions. Thus, in practice, for a binary quality indicator  $I$ , the fitness of a solution  $x$  is computed using an estimation of expected  $I$ -values on a finite set of evaluations. Hence, for a population  $P = \{x_1, x_2, \dots, x_m\}$  and a finite set of evaluation  $S(x)$ , the fitness of an individual  $x$  is defined as the estimated loss of quality if  $x$  is removed from the population, *i.e.*:

$$\begin{aligned} \text{Fitness}(x) &= \hat{E}(I(\mathcal{F}(P \setminus \{x\}), \mathcal{F}(P))) \\ &= \hat{E}(I(\mathcal{F}(P \setminus \{x\}), \mathcal{F}(\{x\}))) \\ &= \frac{1}{|S(x)|} \sum_{z \in S(x)} \hat{E}(I(\mathcal{F}(P \setminus \{x\}), \{z\})). \end{aligned} \quad (8)$$

To compute the estimated expected  $I_{\epsilon+}$ -value between a multiset  $A \in \Omega$  and a reference objective vector  $z^*$ , we consider all pairs  $(x_j, z_k)$  where  $x_j \in A$  and  $z_k \in S(x_j)$  and sort them in the increasing order according to the indicator values  $I_{\epsilon+}(\{z_k\}, \{z^*\})$ . Suppose the resulting order is  $(x_{j_1}, z_{k_1}), (x_{j_2}, z_{k_2}), \dots, (x_{j_l}, z_{k_l})$ , the estimated expected  $I_{\epsilon+}$ -value is then:

$$\begin{aligned} \hat{E}(I_{\epsilon+}(\mathcal{F}(A), \{z^*\})) &= I_{\epsilon+}(\{z_{k_1}\}, \{z^*\}) \times \hat{P}(\mathcal{F}(\{x_{j_1}\}) = \{z_{k_1}\}) &+ \\ &I_{\epsilon+}(\{z_{k_2}\}, \{z^*\}) \times \\ &\hat{P}(\mathcal{F}(\{x_{j_2}\}) = \{z_{k_2}\} \mid \mathcal{F}(\{x_{j_1}\}) \neq \{z_{k_1}\}) &+ \\ &\dots \\ &I_{\epsilon+}(\{z_{k_l}\}, \{z^*\}) \times \\ &\hat{P}(\mathcal{F}(\{x_{j_l}\}) = \{z_{k_l}\} \mid \forall_{1 \leq i < l} \mathcal{F}(\{x_{j_i}\}) \neq \{z_{k_i}\}). \end{aligned} \quad (9)$$

The running time complexity of an estimated expected  $I_{\epsilon+}$ -value computation is of order  $\mathcal{O}(n(Ns)^2 \log(Ns))$ , where  $N$  stands for the population size,  $n$  for the number of objectives and  $s$  for the number of evaluations per solution. Nevertheless, note that all  $l$  sums do not necessarily need to be computed and, thereby, the real computation time can be reduced.

### 3.4 Implementation

To implement our algorithms, we use the EO framework [11] linked to its extension dedicated to multi-objective optimization ParadisEO-MOEO<sup>2</sup>. First, it was necessary to extend this framework by defining the pareto fitness notion for stochastic problems. Then, we implemented those methods in the same way as existing deterministic methods. All these new concepts are now available within the ParadisEO-MOEO framework.

The three approaches differ the one from the others by the way their fitness function is defined. The common points of the different algorithms are:

- *Initialization*: randomly generated individuals.
- *Selection*: deterministic tournament between two randomly chosen individuals.
- *Crossover*: two-point crossover [9].
- *Mutation*: shift mutation [9].
- *Replacement*: elitist.

<sup>2</sup> ParadisEO-MOEO is available at <http://paradiseo.gforge.inria.fr>.

## 4 Simulation results

### 4.1 Benchmarks

To test our algorithms, we propose different benchmark suites<sup>3</sup> built from Taillard's instances [17]. These instances contain processing times for problems whose size varies from 20 to 500 jobs and from 5 to 20 machines.

*Deterministic bi-objective benchmarks.* First, we need to extend the Taillard's instances for the bi-objective deterministic case by adding a due date for every job. These dates were fixed using a random value chosen between  $\bar{p} \times M$  and  $\bar{p} \times (N + M - 1)$ , where  $N$  stands for the number of jobs,  $M$  for the number of machines and  $\bar{p}$  for the mean of previously generated processing times. Thus, a due date  $d_i$  lies between the average completion date of the first scheduled job and the average completion date of the last scheduled job. Moreover, in addition to Taillard's instances, we propose some instances with intermediate sizes. Each benchmark's name is composed on the same way: the first number represents the number of jobs to schedule, the second one the number of machines and the last one the index of the instance among the instances of same size.

*Stochastic bi-objective benchmarks.* To generate stochasticity on a deterministic instance, the four probability distributions a processing time may follow can be applied over initial data using a configuration file. We choose to allow to configure this uncertainty over the machines only, by specifying, for each machine, a probability distribution and its parameters or some proportions depending on its central tendency. Thus, as in real-world problems, each time stochasticity is generated on an initial deterministic instance using the same configuration file, processing times contained in the obtained stochastic instance will be different.

### 4.2 Optimization runs

For the optimization runs, we generate stochasticity over the processing times of some deterministic benchmarks. Thus, for a given instance, we carry out 10 different evaluations into which processing times follow a uniform, an exponential, a normal, a log-normal or various distributions in the following way:

- uniform distribution:  $p_{ij} \sim \mathcal{U}(a = 0.85 \times p_{ij}, b = 1.15 \times p_{ij})$ ;
- normal distribution:  $p_{ij} \sim \mathcal{N}(\mu = p_{ij}, \sigma = 0.15 \times p_{ij})$ ;
- exponential distribution:  $p_{ij} \sim \mathcal{E}(a = p_{ij}, \lambda = \frac{1}{0.15 \times p_{ij}})$ ;
- log-normal distribution:  $p_{ij} \sim \log\mathcal{N}(\mu = \log p_{ij}, \sigma = 0.15 \times \log p_{ij})$ ;
- various distributions: the distribution of the processing times differs on every machine.

---

<sup>3</sup> All benchmarks are available at <http://www.lifl.fr/~liefooga/benchmarks/>.

The population size is set to 50. For each kind of distribution, we perform 10 runs per instance and per algorithm using 10 evaluations per solution (except for  $IBEA_1$  where only the first evaluation is used). The different methods are tested using the same initial populations and the same number of generations. The crossover and mutation probabilities are set to 0.05 and 1.00 respectively. The scaling factor  $\mathcal{K}$ , required in  $IBEA_1$  [21], is set to 0.05.

### 4.3 Performance assessment

To our knowledge, no protocol fully adapted to evaluate the effectiveness of multi-objective optimization methods for stochastic problems exists by now. Consequently, we choose to revalue each final set of solutions on the reference benchmark (the one from which stochasticity was generated) and to regard this evaluation as the ‘true’ evaluation. Then, we only keep the non-dominated solutions (according to this ‘true’ evaluation) obtained by each algorithm. Therefore, we are able to apply traditional metrics, used in the deterministic case, to assess the quality of the obtained Pareto set approximations.

Here, we use two measures to compare the obtained Pareto fronts: the contribution metric [15] and the S metric [19]. The contribution metric evaluates the proportions of Pareto optimal solutions given by each front, whereas the S metric measures the size of the objective space dominated by a non-dominated set. For the contribution metric, the performance comparison is carried out using a reference set  $R$  determined by merging all the solutions found during the whole optimization runs of every algorithm into a single set and keeping only the non-dominated solutions. For the S metric, the required reference point  $Z$  is composed of the worst objective values observed on the whole optimization runs for the benchmark under consideration, multiplied by 1.1.

### 4.4 Computational results and discussion

To significantly compare all the algorithms, we choose to perform a Wilcoxon rank test for every instances and every kind of probability distribution. On each of the following results tables, the ‘T’ columns give the result of the test for a p-value lower than 5%, *i.e.*:

- according to the metric under consideration, the results of the algorithm located at the specific row are significantly better than those of the algorithm located at the specific column (+);
- according to the metric under consideration, the results of the algorithm located at the specific row are significantly worse than those of the algorithm located at the specific column (–);
- according to the metric under consideration, there is no significant difference between the results of the two algorithms ( $\equiv$ ).

Results for uniformly, exponentially, normally, log-normally and variously distributed processing times are respectively given in tables 1, 2, 3, 4 and 5.

In a general way, according to the performance metrics used,  $IBEA_{avg}$  globally outperforms  $IBEA_1$  and  $IBEA_{stoch}$  for all probability distributions (except for the first benchmark whose processing times are log-normally distributed, where  $IBEA_1$  performs significantly better according to the S metric, see table 4). However, as  $IBEA_{avg}$  uses average values, we could have expected such results for probability distributions whose central tendency is the mean (*i.e.* uniform and normal distributions), but these results are more surprising for the other distributions. For uniformly and normally distributed processing times,  $IBEA_1$  is significantly more efficient than  $IBEA_{stoch}$  according to the S metric (see tables 1 and 3). But, according to the contribution metric, there is no significant difference between these two algorithms (except for the uniform distribution where  $IBEA_{stoch}$  performs significantly better on the last benchmark, see table 1). For the exponential distribution, there is globally no significant difference between  $IBEA_1$  and  $IBEA_{stoch}$  (see table 2). Even so, according to the contribution metric,  $IBEA_{stoch}$  is more effective on the last benchmark. And, according to the S metric,  $IBEA_1$  is more effective on the second one. Finally, according to the contribution metric, for log-normally and variously distributed processing times, there is no significant difference between all the algorithms. On the contrary, according to the S metric,  $IBEA_{stoch}$  generally outperforms  $IBEA_1$  for the log-normal distribution (except for the first benchmark) whereas it outperforms  $IBEA_1$  only on the last benchmark for variously distributed processing times (see tables 4 and 5).

The poor overall effectiveness of  $IBEA_{stoch}$  can be partly explained by a low diversity among the Pareto set approximation obtained during the evaluation on the reference benchmark. Then, using an indicator that would increase the diversity in the decision space could give better results. Besides, the whole of the experimental results can be discussed as the evaluation protocol is not fully adapted to stochastic multi-objective problems.

**Table 1.** Comparison of the quality assessment values obtained by  $IBEA_1$ ,  $IBEA_{avg}$  and  $IBEA_{stoch}$  for uniformly distributed processing times using the Wilcoxon rank test.

		contribution metric				S metric			
		$IBEA_1$		$IBEA_{avg}$		$IBEA_1$		$IBEA_{avg}$	
		p-value	T	p-value	T	p-value	T	p-value	T
20_5_01	$IBEA_{avg}$	> 5 %	≡			0.002	+		
	$IBEA_{stoch}$	> 5 %	≡	0.009	-	0.002	-	0.001	-
20_5_02	$IBEA_{avg}$	0.013	+			0.002	+		
	$IBEA_{stoch}$	> 5 %	≡	0.006	-	0.005	-	0.001	-
20_10_01	$IBEA_{avg}$	0.002	+			0.001	+		
	$IBEA_{stoch}$	0.005	+	0.002	-	0.001	-	0.001	-

**Table 2.** Comparison of the quality assessment values obtained by  $IBEA_1$ ,  $IBEA_{avg}$  and  $IBEA_{stoch}$  for exponentially distributed processing times using the Wilcoxon rank test.

		contribution metric				S metric			
		$IBEA_1$		$IBEA_{avg}$		$IBEA_1$		$IBEA_{avg}$	
		p-value	T	p-value	T	p-value	T	p-value	T
20.5_01	$IBEA_{avg}$	> 5 %	≡			0.001	+		
	$IBEA_{stoch}$	> 5 %	≡	> 5 %	≡	> 5 %	≡	0.033	-
20.5_02	$IBEA_{avg}$	> 5 %	≡			> 5 %	≡		
	$IBEA_{stoch}$	> 5 %	≡	> 5 %	≡	0.002	-	0.001	-
20.10.01	$IBEA_{avg}$	0.045	+			0.001	+		
	$IBEA_{stoch}$	0.036	+	> 5 %	≡	> 5 %	≡	0.005	-

**Table 3.** Comparison of the quality assessment values obtained by  $IBEA_1$ ,  $IBEA_{avg}$  and  $IBEA_{stoch}$  for normally distributed processing times using the Wilcoxon rank test.

		contribution metric				S metric			
		$IBEA_1$		$IBEA_{avg}$		$IBEA_1$		$IBEA_{avg}$	
		p-value	T	p-value	T	p-value	T	p-value	T
20.5_01	$IBEA_{avg}$	> 5 %	≡			0.002	+		
	$IBEA_{stoch}$	> 5 %	≡	> 5 %	≡	0.021	-	0.001	-
20.5_02	$IBEA_{avg}$	> 5 %	≡			0.001	+		
	$IBEA_{stoch}$	> 5 %	≡	> 5 %	≡	0.002	-	0.001	-
20.10.01	$IBEA_{avg}$	0.004	+			0.002	+		
	$IBEA_{stoch}$	> 5 %	≡	0.004	+	0.001	-	0.001	-

**Table 4.** Comparison of the quality assessment values obtained by  $IBEA_1$ ,  $IBEA_{avg}$  and  $IBEA_{stoch}$  for log-normally distributed processing times using the Wilcoxon rank test.

		contribution metric				S metric			
		$IBEA_1$		$IBEA_{avg}$		$IBEA_1$		$IBEA_{avg}$	
		p-value	T	p-value	T	p-value	T	p-value	T
20.5_01	$IBEA_{avg}$	> 5 %	≡			0.032	-		
	$IBEA_{stoch}$	> 5 %	≡	> 5 %	≡	0.001	-	0.001	-
20.5_02	$IBEA_{avg}$	> 5 %	≡			0.001	+		
	$IBEA_{stoch}$	> 5 %	≡	> 5 %	≡	0.001	+	0.001	-
20.10.01	$IBEA_{avg}$	> 5 %	≡			0.001	+		
	$IBEA_{stoch}$	> 5 %	≡	> 5 %	≡	0.001	+	0.020	-

**Table 5.** Comparison of the quality assessment values obtained by  $IBEA_1$ ,  $IBEA_{avg}$  and  $IBEA_{stoch}$  for variously distributed processing times using the Wilcoxon rank test.

		contribution metric				S metric			
		$IBEA_1$		$IBEA_{avg}$		$IBEA_1$		$IBEA_{avg}$	
		p-value	T	p-value	T	p-value	T	p-value	T
20_5_01	$IBEA_{avg}$	> 5 %	≡			0.001	+		
	$IBEA_{stoch}$	> 5 %	≡	> 5 %	≡	0.019	-	0.001	-
20_5_02	$IBEA_{avg}$	> 5 %	≡			0.014	+		
	$IBEA_{stoch}$	> 5 %	≡	> 5 %	≡	> 5 %	≡	0.010	-
20_10_01	$IBEA_{avg}$	> 5 %	≡			0.001	+		
	$IBEA_{stoch}$	> 5 %	≡	> 5 %	≡	0.003	+	> 5 %	≡

## 5 Conclusion and perspectives

In this paper, we investigated a bi-objective flow-shop scheduling problem with stochastic processing times as well as general combinatorial optimization algorithms applied to its resolution. First, we saw that, in real-world situations, none of the parameters related to this kind of problem is deprived of uncertainty. Thus, non-deterministic models are required to take this uncertainty into account. To this end, a proactive approach, where processing times are represented by random variables, have been taken and several general stochastic models have been proposed. Next, we introduced three different indicator-based methods for stochastic multi-objective problems that are able to handle any type of uncertainty. The first method, called  $IBEA_1$ , consists in preserving the deterministic approach by computing the fitness of a solution on a single evaluation. The second method, namely  $IBEA_{avg}$ , is based on average objective values. At last, the  $IBEA_{stoch}$  method consists in estimating the quality of a solution in a probabilistic way. The latter, already investigated in [2] on continuous problems, has never been applied neither to the combinatorial case nor to the stochastic models proposed here. All these algorithms and the fitness concept for multi-objective stochastic problems are now available within the ParadisEO-MOEO framework; new methods can thus easily be implemented in order to compare their effectiveness with those presented in this paper. To test these algorithms on our stochastic flow-shop problem, we initially had to build benchmark suites, first for the deterministic bi-objective case, then for the stochastic case. According to the experimental protocol we formulated, we concluded that  $IBEA_{avg}$  was overall more efficient than  $IBEA_1$  and  $IBEA_{stoch}$ . Even so, from a purely theoretical point of view,  $IBEA_{stoch}$  seems to be more representative of the quality associated to a solution than the two other methods. In spite of that, the results it obtained are a little disappointing in comparison with the continuous case [2]; even if, in that last case, its effectiveness is especially significant for more than two objectives. This can be explained by the fact that the final solutions found by this algorithm are relatively close the ones from the others in the decision space, which generally implies, for our problem, that they are also close in the objective space. As a result, this method cannot contest the others

in term of diversity. However, all these results should be moderated. No experimental protocol fully adapted to the combinatorial optimization of stochastic multi-objective problems exists up to now. The one we proposed, although simple and fast, is not really natural and is imperfectly adapted to this kind of problem. Moreover, considering the evaluation on the deterministic benchmark as the ‘true’ evaluation advantages a lot  $IBEA_{avg}$ , especially for probability distributions whose central tendency is the mean.

Different perspectives emerge from this work. First of all, other sources of uncertainty that processing times could be taken into account for the flow-shop problem. Furthermore, a reactive approach could be linked to our proactive approach in order to largely improve the effectiveness of all the algorithms. Additionally, as all the results obtained during our experiments reveal a weakness of convergence, hybridizing our evolutionary algorithms with local searches could be beneficial, especially to accelerate the exploration near potentially interesting solutions. Moreover, even if most of quality indicators to be used with  $IBEA$  take diversity into account, they only deal with diversity on the objective space, and not on the decision space. Then, to fill the low-level of diversity of  $IBEA_{stoch}$ , it could be interesting to create an indicator that would allow the decision maker to obtain diversified solutions in the decision space and that would not be completely focused on the Pareto dominance relation. Also, perhaps  $IBEA_{stoch}$  is simply too fine-grained compared to  $IBEA_{avg}$ , and so presents a landscape with a complex structure whereas  $IBEA_{avg}$  provides a reasonable guidance and uses stochasticity to pass over such a structure. Studying the landscape more precisely could then be helpful. Besides, the population size as well as the number of evaluations per solution are two parameters that influence a lot the effectiveness of all the algorithms. Studying more finely the way of determining them and analyzing how to make them evolve in a more efficient way during the optimization process could be profitable. Lastly, working out an experimental protocol adapted to the combinatorial optimization of stochastic multi-objective problems proves to be essential to evaluate the results in a more rigorous way. As well, this study could be extended in order to consider problems with more than two objectives.

## References

1. Babbar M., Lakshmikantha A., Goldberg D. E.: A Modified NSGA-II to Solve Noisy Multiobjective Problems. In Foster J. et al. (eds.): Genetic and Evolutionary Computation Conference (GECCO'2003). Lecture Notes in Computer Science **2723**, Springer, Chicago, Illinois, USA (2003) 21–27
2. Basseur M., Zitzler E.: Handling Uncertainty in Indicator-Based Multiobjective Optimization. International Journal of Computational Intelligence Research **2(3)** (2006) 255–272
3. Cunningham A. A., Dutta S. K.: Scheduling jobs with exponentially distributed processing times on two machines of a flow shop. Naval Research Logistics Quarterly **16** (1973) 69–81
4. Dauzère-Pérès S., Castagliola P., Lahlou C.: Niveau de service en ordonnancement stochastique. In Billaut J.-C. et al. (eds.): Flexibilité et robustesse en ordonnancement. Hermès, Paris (2004) 97–113

5. Deb K., Gupta H.: Searching for Robust Pareto-Optimal Solutions in Multi-Objective Optimization. In Coello Coello C. A. et al. (eds.): *Evolutionary Multi-Criterion Optimization, Third International Conference (EMO 2005)*. Lecture Notes in Computer Science **3410**, Springer, Guanajuato, Mexico (2005) 150–164
6. Dudek R. A., Panwalkar S. S., Smith M. L.: The Lessons of Flowshop Scheduling Research. *Operations Research* **40(1)**, INFORMS, Maryland, USA (1992) 7–13
7. Graham R. L., Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. G.: Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey. *Annals of Discrete Mathematics* **5** (1979) 287–326
8. Hughes E. J.: Evolutionary Multi-Objective Ranking with Uncertainty and Noise. In Zitzler E. et al. (eds.): *Evolutionary Multi-Criterion Optimization, First International Conference (EMO 2001)*. Lecture Notes in Computer Science **1993**, Springer, London, UK (2001) 329–343
9. Ishibuchi H., Murata T.: A Multi-Objective Genetic Local Search Algorithm and Its Application to Flowshop Scheduling. *IEEE Transactions on Systems, Man and Cybernetics* **28** (1998) 392–403
10. Jin, Y., Branke, J.: Evolutionary Optimization in Uncertain Environments - A Survey. *IEEE Transactions on Evolutionary Computation* **9** (2005) 303–317
11. Keijzer M., Merelo J. J., Romero G., Schoenauer M.: Evolving Objects: A General Purpose Evolutionary Computation Library. *Proc. of the 5th International Conference on Artificial Evolution (EA'01)*, Le Creusot, France (2001) 231–244
12. Kouvelis P., Daniels R. L., Vairaktarakis G.: Robust scheduling of a two-machine flow shop with uncertain processing times. *IIE Transactions* **32(5)** (2000) 421–432
13. Ku P. S., Niu S. C.: On Johnson's Two-Machine Flow Shop with Random Processing Times. *Operations Research* **34** (1986) 130–136
14. Landa Silva J. D., Burke E. K., Petrovic S.: An Introduction to Multiobjective Metaheuristics for Scheduling and Timetabling. In Gandibleux X. et al. (eds.): *Metaheuristics for Multiobjective Optimisation*. Lecture Notes in Economics and Mathematical Systems **535**, Springer, Berlin (2004) 91–129
15. Meunier H., Talbi E.-G., Reininger P.: A multiobjective genetic algorithm for radio network optimization. In *Proc. of the 2000 Congress on Evolutionary Computation (CEC'00)*. IEEE Press (2000) 317–324
16. T'kindt V., Billaut J.-C.: *Multicriteria Scheduling - Theory, Models and Algorithms*. Springer, Berlin, 2002.
17. Taillard E. D.: Benchmarks for Basic Scheduling Problems. *European Journal of Operational Research* **64** (1993) 278–285
18. Teich J.: Pareto-Front Exploration with Uncertain Objectives. In Zitzler E. et al. (eds.): *Evolutionary Multi-Criterion Optimization, First International Conference (EMO 2001)*. Lecture Notes in Computer Science **1993**, Springer, London, UK (2001) 314–328
19. Zitzler E., Thiele L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation* **3(4)** (1999) 257–271
20. 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(2)** (2003) 117–132
21. Zitzler E., Künzli S.: Indicator-Based Selection in Multiobjective Search. In *Parallel Problem Solving from Nature (PPSN VIII)*. Lecture Notes in Computer Science **3242**, Springer, Birmingham, UK (2004) 832–842