Enhancing A Multiobjective Evolutionary Algorithm Through Flexible Evolution

Daniel Salazar, Blas Galván and Gabriel Winter

Institute of Intelligent Systems and Numerical Applications in Engineering (IUSIANI)
Division of Evolutionary Computation and Applications (CEANI)
University of Las Palmas de Gran Canaria, Canary Islands, Spain
daniel_salazar@cantv.net, bgalvan@step.es, gabw@step.es

Abstract. In this paper the use of a powerful single-objective optimization methodology in Multi-objective Optimization Algorithms (MOEAs) is introduced. The Flexible Evolution concepts (FE) have been recently developed and proved its efficiency gains compared with several Evolutionary Algorithms solving single-objective challenging problems. The main feature of such concepts is the flexibility to self-adapt the internal behaviour of the algorithm to optimize its search capacity. In this paper we present the first attempt to incorporate FE into MOEAs. A real coded NSGA-II algorithm was modified replacing the crossover and mutation operators with the Sampling Engine of FE. Other two FE characteristics were implemented too: The Probabilistic Control Mechanism and the Enlarged Individual's Code. The performance of the resulting algorithm has been compared with the classical NSGA-II using several test functions. The results obtained and presented show that FE_based algorithms have advantages over the classical ones, especially when optimizing highly multimodal complex functions.

1 Introduction

Evolutionary Multi-objective Optimization (EMO) is currently an area of great relevance in the scientific and technical fields due to the growing consciousness of the multiobjective nature of almost every real problem we deal with [1].

As a discipline, EMO is the conjuction of classical multiobjective optimization studied in Operation Research and Evolutionary Algorithms (EAs) which comprises Genetic Algorithms (GAs), Evolution Strategies (ES) and Evolutionary Programming.

The tremendous power of exploration shown by EAs, makes it a very atractive tool for single and multiple-objectives optimization in general, specially when the search space shows some complexity like multimodality, non continuos and/or non lineal domain, among others.

In the last twenty years, starting with Schaffer's VEGA (1984) [2], several efforts have been made in order to formulate better and more efficient MultiObjective Evolutionary Algorithms (MOEAs). The lessons learned from the first generation of MOEAs (Weighted Sum, MOGA [3], NPGA [4], NSGA[5]), with the incorporation of some succesful features in single objective optimization, like elitism, have led to develope the second generation of MOEAs, a powerful group of algorithms [SPEA [6] /SPEA2 [7], M-PAES [8], PESA [9] /PESA-II [10], NSGA-II [11]) very well known by practitioners and widely used either in academical or real problems.

However, some tools employed in single objective optimization that never have been incorporated in MOEAs still remain, in particular we refer to a recently published philosophy of problems resolution rather than a specific tool, named Flexible Evolution (FE), which demonstrated a strong search power on very hard single objective optimization problems, among other beneficial characteristics [12], [13], [14], [15].

In this paper we present the first attempt to incorporate FE into MOEAs. In this first stage, we compare the traditional GAs' real-coded sampling mechanisms (crossover and mutation operators) with FE's Sampling Engine. The results obtained show that FE has advantages over such traditional operators because it is not seized to crossover and mutation parameters. In consequence, its execution is not parameter sensitive like current MOEAs. [16], [17]. In addition, our experiments make clear that FE implementation in MOEAs produce good quality sets of solutions, accelerating the convergence to the Pareto frontier in the most complex functions tested, improving the algorithm efficiency.

2. Fundamentals

2.1 Multiobjective Optimization

A multiobjective optimization problem consists in optimize a vector of functions:

$$
\begin{aligned}
&Opt(F(x) = (f_1(x), f_2(x),..., f_k(x)) \\
&\quad \text{s.t.:} \\
&\quad g_j(x) \leq 0 \quad j = 1,2,...,q \\
&\quad h_j(x) = 0 \quad j = 1,2,...,m
\end{aligned}
\qquad (1)
$$

Where x = $(x_1,x_2,...,x_n) \in X$ is the solution vector, or vector of decision variables, and X is the feasible domain. Given that the concept of optimality in single objective is not directly applicable in multiobjective problems, a clasification of the solutions is introduced in terms of Pareto optimality, according to the following definitions. In terms of minimization:

Definition1. *Pareto Optimal*: A solution vector $x^* \in X$ is Pareto Optimal solution iff

$$\neg\exists x \in X : f_i(x) \leq f_i(x^*); i = \{1,2,...,k\}$$

Definition 2: *Pareto Dominance*: A solution x^1 dominate x^2, denoted as $x^1 \succ x^2$ iff $f_i(x^1) \leq f_i(x^2) \wedge \exists j : f_i(x^1) < f_i(x^2); i, j \in \{1,2,...,k\}$. If there is no solutions which dominates x^1, then x^1 is non dominated.

Definition 3. *Pareto Set*: A set of non dominated solutions $\{x^* \mid \neg\exists x : x \succ x^*\}$ is said to be a Pareto set.

Definition 4. *Pareto Front*: the image of a Pareto Set, i.e. $\{F(x^*) \mid \neg\exists x : x \succ x^*\}$

2.2 Multiobjective Evolutionary Algorithms

In general terms, actual MOEAs based on GAs follow a sequence similar to the flow chart shown in figure 1. Notice that there are two populations: P^t wich represents the actual population during t generation, and $P_A{}^t$ which consists in a non dominated solutions archive. The state of art MOEAs keep a constant size M for the population and N for the archive. During the initialization, M individuals are generated randomly and the archive is set to empty. In each generation all non dominated individuals from both the archive and the population are selected and asigned to the archive $P_A{}^{t+1}$, after that a reproductive selection of individuals is accomplished (tipically by binary tournament) and a mating pool is filled. In this time a new population is generated following some ($\mu+\lambda$) recombination strategy. Finally, the new population replace the old one and the process is continued untill the maximum number of generation is reached. As an ouput, the non dominated solutions from the archive are reported.

It is important to mention that, in most of the cases, there are more or less non dominated solutions than N solutions available in every generation of the process. Therefore, some criteria or methods for classifying and selecting are needed in order to set the archive to N solutions. This can be achived assigning fitness values to the individuals based on the Pareto dominance combined with some other criteria, usually referred to density and distribution of the population. That is what is known as environmental selection.

It is convinient to realize that in the practice, the environmental selection constitutes the only difference between one method and another. The rest of the algorithm remains equal to a traditional EA.

2.3 Flexible Evolution

An interesting point of view to understand what FE does is through the No Free Lunch (NFL) theorems [18], [19]. This group of theorems establishes that there is no optimization method which outperforms another in the set of all optimization problems defined over a domain. In other words, given two optimization methods, the first one cannot beat the other everytime, because the features that makes it succesfull over a group of problems become less efficient than the characteristics of the second one, when they deal with a different group of problems.

At this point, the reader must be aware that the intrinsic difference between single objective optimization and multiobjective optimization makes posible that some exceptions to NFL take place [20]. Nevertheless they are referrered to the environmental selection rather than the search methods.

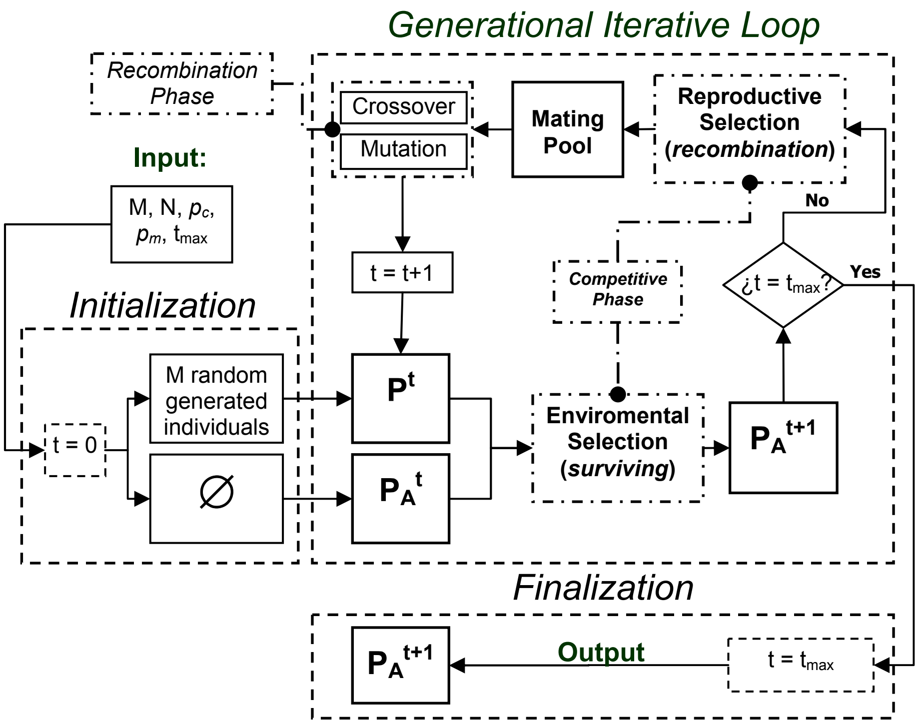

Fig. 1. General flow chrat of a 2nd generation MOEA based on genetic algorithms

Coming back to what FE does, consider now that, if we take search tools from several methods and put them together into a collection, giving an algorithm the ability to learn what tool to choose in a determined moment, the algorithm itself could take advantage of the more efficient tool in any phase of the search, accelerating the convergence and making the process more 'intelligent'. This is what FE intended to do, and it can be seen as a smart way of overcoming NFL implications, by using some kind of algorithmic intelligence.

To accomplish this task, it is necessary 1) to gather search features from all existing optimization methods, 2) Providing the algorithm with decision making procedures of when and how to use them, as well as a mechanism to identify bad decisions and to correct them, and 3) Giving some memory of what was done and how during the process. It is possible to satisfy all of these needs inside an EA following the FE general design layout [15] depicted in figure 2, but the simplest FE version of an EA can be obtained doing only three changes to our algorithm:

- Considering many well known sampling methods, collecting them in a database called the Sampling Engine. All of them will have opportunities to collaborate in the actual stage of every optimization run, depending on its demonstrated ability to find good solutions in the previous stages. Such collection can include sampling methods inspired on different sampling techniques (e.g. Genetic algorithms, Monte Carlo Simulation, etc.). In reference [15] a large study of the Sampling Engine design and performance can be found. These methods will be used by our EA instead of the traditional crossover and mutation operators, so the crossover and mutation parameters will be no longer necessary.

- A Probabilistic Control Mechanism is included, providing flexibility to adopt both different sampling strategies (e.g. How many variables will be sampled) and different sampling methods (among those included in the Sampling Engine abovementioned). Two simple if-then-else control sentences can be used as Probabilistic Control Mechanism. The first one to decide if one or all the variables of each parent individual are selected for new sampling, the second one to decide the sampling method to be used for the variable(s) each time. Both sentences will use a probability frontier (currently 0.6) for appropriate control of the process flow.

- Using an Enlarged individual's Genetic Code, in such a way that the last method used to sample each variable is recorded beside the variable values itself.

So the FE memory is provided by the Enlarged Genetic Code while the decision making is possible due to the Probabilistic Control Mechanism. The interaction among the Enlarged Genetic Code, the Probabilistic Control Mechanism and the Selection procedure of our EA make possible to identify bad decisions and correct them, because when the best individuals are selected for recombination, the sampling decision (adopted one or more generations in the past) used to obtain them (supposed good decisions) are stored in his Enlarged Genetic Code.

3 Implementation

In order to assess the behaviour of the FE in a multiobjective algorithm, we decided to restrict our first step to the comparison of FE's Sampling Engine with the traditional GAs' sampling mechanism (crossover and mutation). Further developments will be presented in other works.

An efficient state of art MOEA, named NSGA-II, was selected to serve as a control algorithm. Thus, the experiment consists in running the original version of NSGA-II and a 'flexibilized' version of the same MOEA, denoted as NSFEA (for Non Sorted Flexible Evolutionary Algorithm) over a group of test problems, comparing the outcomes.

First, the NSGA-II was implemented according to [11] with real codification. The crossover operator employed is a one point arithmetic crossover (see table 1), whereas the mutation consists in a gaussian mutation (table 1). The selection procedure is binary tournament.

Second, the NSFEA's Sampline Engine was implemented as a slightly modified version of the one described in [15]. The reader is encouraged to review this reference for more details.

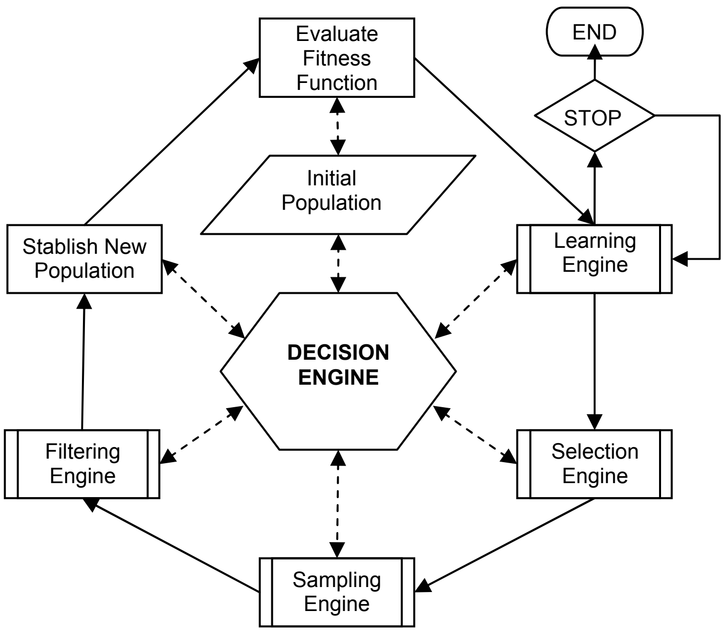

Fig. 2. Flexible Evolution general schema. The decision engine works as the brain of the search process, coordinating and interacting with peripheral processes and the engines

Table 1. Real coded recombination operators implemented in NSGA-II

Input: ▪ Individuals: x^1, x^2. ▪ Mutation rate: p_m	
One Point Arithmetic Crossover Operator (2+2)	Given a randomly cross point $\beta \in \{1,2,...,n\}$ and a crossover coefficient $\alpha \in [0,1]$ $x^{1'} = (x^1_1, x^1_2,..., x^1_{\beta-1}, \alpha x^1_\beta + (1-\alpha)x^2_\beta, x^2_{\beta+1},..., x^2_n)$ $x^{2'} = (x^2_1, x^2_2,..., x^2_{\beta-1}, \alpha x^2_\beta + (1-\alpha)x^1_\beta, x^1_{\beta+1},..., x^1_n)$
Gaussian Mutation Operator (1+1)	Let $x^{1'}$ be x^1 after applying mutation: For i = 1 to n if(an uniformed distributed number [0,1] < p_m) $x^{1'}_i = x^1_i + N(0,\sigma)$ rof
Output: ▪ Individuals: $x^{1'}$, $x^{2'}$.	

The sampling operators comprise binary operators (heuristic, geometric and arithmetic crossover) and unary operators (Gaussian and non-uniform mutation as well as rescaling operators). The selection method is the binary tournament, and follows the rule established in NSGA-II, that is, only individuals from the archive are taken to sampling. However, due to the heterogeneous nature of the operators, the sampling is executed as a (2+1) strategy, i.e. the FE Agent takes two individuals from the archive and produces one, no matters if the operators employed are unary or binary.

Four test problems of type $Min(F(x) = (f_1(x), f_2(x)))$ were selected for benchmarking: ZDT1, ZDT4, ZDT6, FON [11]. They can be seen in table 2.

Table 2. Multiobjective test problems selected for experimentation

Name	Formulation:	Domain	Optimal set
ZDT1	$f_1(x) = x_1$ $f_2(x) = g(x)\left(1 - \sqrt{x_1 / g(x)}\right)$ $g(x) = 1 + 9\left(\sum_{i=2}^{9} x_i\right)/(n-1)$	$x_i \in [0,1]$ n = 30	$x_1 \in [0,1]$ $x_i = 0$ $i \in \{2,...,n\}$
ZDT4	$f_1(x) = x_1$ $f_2(x) = g(x)\left(1 - \sqrt{x_1 / g(x)}\right)$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^{9}\left(x_i^2 - 10\cos(4\pi x_i)\right)$	$x_1 \in [0,1]$ $x_i \in [-5,5]$; $i \in \{2,...,n\}$ n = 10	$x_1 \in [0,1]$ $x_i = 0$ $i \in \{2,...,n\}$
ZDT6	$f_1(x) = 1 - \exp(-4x_1)\sin^6(4\pi x_i)$ $f_2(x) = g(x)\left(1 - (x_1 / g(x))^2\right)$ $g(x) = 1 + 9\left[\left(\sum_{i=2}^{9} x_i\right)/(n-1)\right]^{1/4}$	$x_i \in [0,1]$ n = 10	$x_1 \in [0,1]$ $x_i = 0$ $i \in \{2,...,n\}$
FON	$f_1(x) = 1 - \exp\left(-\sum_{i=1}^{n}\left(x_i - \frac{1}{\sqrt{3}}\right)^2\right)$ $f_2(x) = 1 - \exp\left(-\sum_{i=1}^{n}\left(x_i + \frac{1}{\sqrt{3}}\right)^2\right)$	$x_i \in [-4,4]$ n = 3	$x_1 = x_2 = x_3$ $x_i \in \left[-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right]$

4 Experimentation and Results

A total of 20 run were executed for each problem and parmeter set. We use 10, 50 and 250 as maximun number of generations, whereas the population sizes M and N, were fixed to 100 and 50 respectively. For NSGA-II a crossover rate of 0.9 and a crossover coefficient α were selected. Finally, mutation rates of 0.35 (FON), 0.05 (ZDT1) and 0.20 (ZDT4, ZDT6), slightly higher than the 1/n heuristic, were chosen.

When comparing MOEAs several difficulties appear because there is no single way to measure performance in multiobjective problems. Thus, we restricted the performance assessment to convergence measure, in order to analyze the effect introduced by FE.

Figures 3 to 6 show the values of Deb's convergence metric ϒ [11]. This metric measures the mean distance of the outcomes to a Pareto optimal solutions in the objectives space. The numerical values are reported in table 3.

Table 3. Values of convergence metric ϒ.

Function	Generations	NSFEA		NSGA-II	
		Mean	Standard Deviation	Mean	Standard Deviation
FON	10	0.683136	0.0614248	0.603735	0.0658977
	50	0.651077	0.0753273	0.584358	0.0443772
	250	0.635591	0.0683103	0.603694	0.0664689
ZDT1	**Generations**	**Mean**	**Standard Deviation**	**Mean**	**Standard Deviation**
	10	1.33474	0.145961	1.04053	0.326847
	50	0.5156	0.0541808	0.356087	0.23445
	250	0.608504	0.0729707	0.380496	0.182192
ZDT4	**Generations**	**Mean**	**Standard Deviation**	**Mean**	**Standard Deviation**
	10	45.3219	7.16858	46.6258	9.65293
	50	4.86292	6.03072	17.4139	10.3511
	250	0.61577	0.215643	8.75542	9.26559
ZDT6	**Generations**	**Mean**	**Standard Deviation**	**Mean**	**Standard Deviation**
	10	4.54706	0.340028	3.51306	1.46799
	50	0.899985	0.165278	0.815201	0.173859
	250	0.848193	0.0739629	0.904961	0.137651

The results obtained with ϒ are not conclusive, particularly due to the inability of unary metrics to characterize outcomes (for a discussion see [21] , [22]). For instance, consider the case of a MOEA which produces outcomes containing only one single solution which lies near to the Pareto optimal front, beside another method wich produces sets of several solutions, all near to Pareto optimal front, but far enough to be characterized by mean distances higher than the corresponding to the first MOEA's outcomes. In this case, ϒ values might indicate a better convergence for the first method, but if the size of the outcome is neglected, it could guide to misleading conclusions. Nevertheless, we still consider ϒ metric useful to find tendencies.

Notice that FE looks inferior in its convergence ability for FON and ZDT1 problems, but the behaviour changes with ZDT4 and ZDT6. It is important to remark that ZDT4 is a very challenging function wihich has 21^9 different local Pareto optimal fronts. Current MOEAs tend to fail in attaining the global Pareto front, because they get stuck when deal with ZDT4.

In all of the cases, the observation of figures 3 to 6 show a steeper slope of the convergence metric between 10 and 50 generations for NSFEA, suggesting an ability to accelerate convergence, at least during the first generations. Moreover, FE's performance is less variable, as can be verified comparing the standard deviation of both methods in each metric.

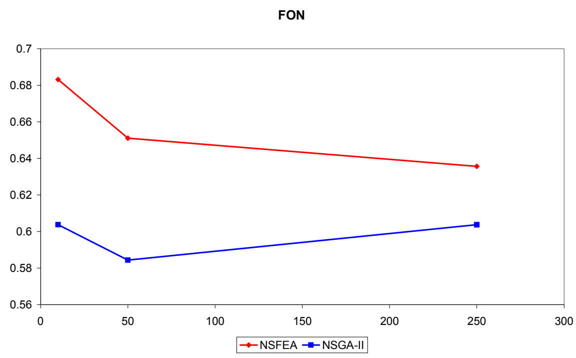

Fig. 3. Convergence metric ϒ values vs. generations number for FON function

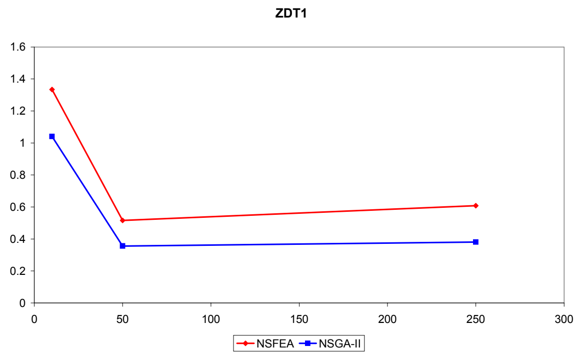

Fig. 4. Convergence metric ϒ values vs. generations number for ZDT1 function.

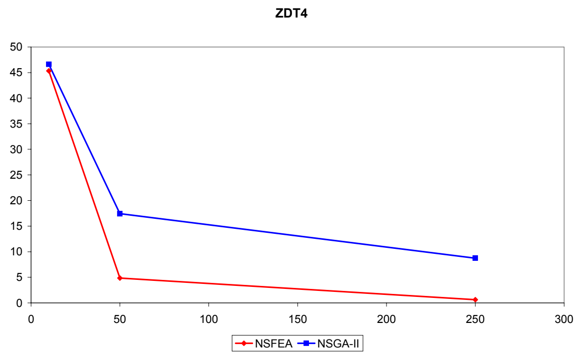

Fig. 5. Convergence metric ϒ values vs. generations number for ZDT4 function.

The results reported by [11] indicates a convergence hierarchy between measures for NSGA-II (ϒ(FON) < ϒ(ZDT1) < ϒ(ZDT6) < ϒ(ZDT4)). The tendency shown agrees quite well with our results, allowing us to assume ZDT6 and especially ZDT4, more complicated functions than FON and ZDT1 in terms of MOEAs execution.

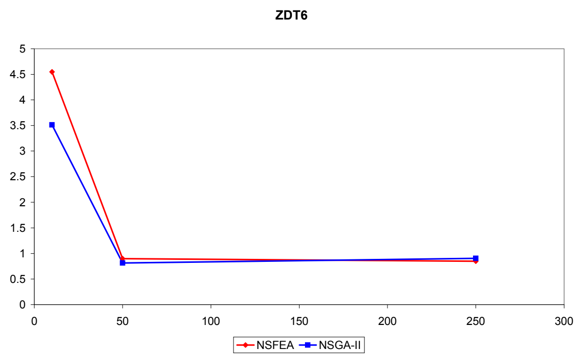

Fig. 6. Convergence metric ϒ values vs. generations number for ZDT6 function.

Therefore, assuming the convergence ability as an indicator of functions complexity, we can observe FE improves the search efficient in 'more complicated' problems, but paradogically decreases the efficcient in those not so complicated.

Apparently, the current stage of FE's Sampling Engine bias exploration over exploitation, unbalancing the search. This could possibly explain why in more complicated problems, where more exploration is required –as a ZDT4-, the search is remarkable levered by FE, whereas in those where exploitation should be favoured, FE is less efficient.

This indicates that further efforts towards balancing the exploration/exploitation FE power should be made. Additionally, new operators must be specially formulated for multiobjective problems.

Finally, some attention must be paid to the parameterless nature of FE. The results reported for NSGA-II depends on the parameter setting, so the quality of the results, and the behaviour in general, could change depending on the parameters. Common practices to achieve better results are: Parameter Tuning, doing several experiments until the best or more adequate parameter set had been found, and Parameter self-adaptation during each algorithm run. Another possibility is simply to follow heuristic rule(s) (like 1/n mutation heuristic), but sometimes those rules underachieve the optimal performance (for instance, see [16] , [17]).

FE does not use external parameters for sampling, reducing the amount of information requested for run. This makes the algorithm less sensitive and more robust than NSGA-II and MOEAs in general.

5 Conclusions

The findings of this study offered evidences on the positive effects of the introduction of FE in multiobjective optimization. In particular, we can mention:

– FE tends to accelerate the convergence, especially when the search space shows some complexity.
– FE has a remarkable ability of exploration. In contrast, its exploitation skills seen to be underdeveloped and must be levered in order to balance the performance on the Sampling Engine.

– The sampling parameterless nature of FE is a notorious advantage, because favours the robustness of the algorithm.

References

1. Corne, D.W., Deb, K., Fleming, P.J., Knowles, J.D.: The Good of the Many Outweights the Good of the One: Evolutionary Multiobjective Optimizations. *coNNectionS*, **1**(1) (2003) 9-13.
2. Schaffer, J.D.: Multiple Optimization with Vector Evaluated Genetic Algorithms. Ph. D. Thesis. Vanderbilt University. Unpublished. (1984).
3. Coello, C.: A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. Knowl. Inf. Syst. **1**(3) (1999) 129-156.
4. Horn, J., Nafpliotis, N.: Multiobjective optimization using the niched pareto genetic algorithm. IlliGAL Report 93005, Illinois Genetic Algorithms Laboratory, University of Illinois. (1993).
5. Srinivas, N., Deb, K.: Multiobjective optimization Using Nondominated Sorting in Genetic Algorithms. Evolutionary Computation. **2**(3) (1994) 221-248.
6. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the stregth Pareto approach. IEEE Transactions on Evolutionary Computation **3**(4) (1999) 257-271.
7. Zitzler, E., Laummans, M., Thiele, L.: SPEA2: Improving the Stregth Pareto Evolutionary Algorithm. TIK Report No. 103. Swiss Federal Institute of Technology (ETH). Computer Engineering and Networks Laboratory (TIK). (2001).
8. Knowles, J.D., Corne, D.W.: M-PAES: A Memetic Algorithm for Multiobjective Optimization. Proceedings of the Congress on Evolutionary Computation (CEC00), IEEE Press, Piscataway, NJ. (2000) 325-332.
9. Knowles, J.D., Corne, D.W.: Approximating the nondominated front using the Pareto Archived Evolution Strategy. Evolutionary Computation **8**(2) (2000) 149-172.
10. Corne, D.W., Jerram, N.R., Knowles, J.D., and Oates, M.J.: PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001), Morgan Kaufmann Publishers (2001) 283-290.
11. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II. KanGAL Report No. 200001. Kanpur Genetic Algorithms Laboratory (KanGAL). Indian Institue of Technology (2001).
12. Winter, G., Galván, B., Cuesta, P., Alonso, S.: Flexible Evolution. In: Giannakoglu, K.C., Tsahalis, D.T., Périaux, J., Papaliouy, K.D., Fogarty, T. (eds.): Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems (EUROGEN 2001), ED. CIMNE, Spain (2001).
13. Winter, G., Galván, B., Alonso, S. González, B.: Una Propuesta de Evolución Flexible en el diseño de algoritmos evolutivos. In: Alba, E., Fernández, F.,

Gómez, J. A., Herrera, F., Hidalgo, J. I., Lanchares, J., Merelo, J. J., Sánchez, J. M. (eds.): Actas del Primer Congreso Español de Algoritmos Evolutivos y Bioinspirados AEB'02. Spain (2002) 15-22.

14. Winter, G., Galván, B., Alonso, S. González, B.: Evolving From Genetic Algorithms to Flexible Evolution Agents. In Cantú-Paz E. (ed.): Late-Breaking Papers Book of Genetic and Evolutionary Computation Conference GECCO 2002 (2002) 466-473.
15. Winter, G., Galván, B., Alonso, S. González, B., Jiménez, J.I., Greiner, D.: A Flexible Evolutionary Agent: cooperation and competition among real-coded evolutionary operators. Soft Computing. Springer-Verlag. (To Appear).
16. M. Laumanns, G. Rudolph, Schwefel, H.-P.: Mutation Control and Convergence in Evolutionary Multi-Objective Optimization. In Matousek, Osmera, P. (eds.): Proceedings of the 7th International Mendel Conference on Soft Computing (MENDEL 2001), Czech Republic (2001) 24-29.
17. Laumanns, M., Zitzler, E., Thiele, L.: On The Effects of Archiving, Elitism, and Density Based Selection in Evolutionary Multi-Objective Optimization. In Zitzler, E., et al. (eds.): Evolutionary Multi-criterion Optimization (EMO 2001), Proceedings of the First International Conference, Switzerland. Lecture Notes in Computer Science Vol. 1993, Springer. (2001) 181-196.
18. Wolpert, D. H., Macready, W. G.: No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation, **1** (1997) 67-82.
19. Corne, D.W., Knowles, J.D.: No Free Lunch and Free Leftovers Theorems for Multiobjective Optimization Problems. Evolutionary Multi-Criterion Optimization (EMO 2003), Proceedings of the Second International Conference, Portugal (2003) 327-341.
20. Corne, D.W., Knowles, J.D.: Some Multiobjective Optimizers are Better than Others. Proceedings of the IEEE Congress on Evolutionary Computation. (2003) (To appear).
21. Knowles, J.D., Corne, D.W.: On Metrics for Comparing Non-Dominated Sets. In Proceedings of the 2002 Congress on Evolutionary Computation Conference (CEC02), IEEE Press (2002) 711-716.
22. Zitzler, E., Laummans, M., Thiele, L. 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.