An ALife-Inspired Evolutionary Algorithm for Dynamic Multiobjective Optimization Problems

P. Amato[1] and M. Farina[2]

[1] Soft Computing, Si-Optics and Post-Silicon Technologies Corporate R&D, STMicroelectronics, Via Remo De Feo 1, 80022 Arzano (NA), Italy
paolo.amato@st.com

[2] Soft Computing, Si-Optics and Post-Silicon Technologies Corporate R&D, STMicroelectronics, Via C. Olivetti, 2, 20041, Agrate(MI), Italy,
marco.farina@st.com

Summary. Several important applications require a time-dependent (on-line) in which either the objective function or the problem parameters or both vary with time. Several studies are available in the literature about the use of genetic algorithms for time dependent fitness landscape in single-objective optimization problems. But when dynamic multi-objective optimization is concerned, very few studies can be found. Taking inspiration from Artificial Life (ALife), a strategy is proposed ensuring the approximation of Pareto-optimal set and front in case of unpredictable parameters changes. It is essentially an ALife-inspired evolutionary algorithm for variable fitness landscape search. We describe the algorithm and test it on some test cases.

1 Introduction

Several important applications require a time-dependent (on-line) multiobjective optimization in which either the objective function or the problem parameters or both vary with time. In handling such problems, there exist not many algorithms and certainly there is a lack of test problems to adequately test a dynamic multi-objective evolutionary algorithm.

In this paper we refer to (eventually on-line) of time-varying systems, where (i) the optimal controller is time-dependent (because the system's properties are time-dependent), and (ii) several objectives have to be optimized at the same time. In [6] the authors and K. Deb gave a full formulation of the resulting multiobjective dynamic nonlinear optimization problem, and they formulated some continuous and discrete test problems where the time dependent Pareto-optimal solutions are known analytically.

Optimal design of controllers is a classical field of application for evolutionary computation and. Once closed loop stability is assured, several additional criteria for performances improvement can be considered such as maximum overshooting minimization, settling time minimization and rise time minimization, in order to design

P. Amato and M. Farina: *An ALife-Inspired Evolutionary Algorithm for Dynamic Multiobjective Optimization Problems*, Advances in Soft Computing **1**, 113–125 (2005)
www.springerlink.com

stable and powerful controllers. Several examples of such optimization procedure are available in literature in case of static design problems, that is when the optimization is to be performed off-line and when the model of the system (the plant or the device) is not time dependent. Two early examples can be found in [7] where some controllers (among which an $\mathcal{H}_2/\mathcal{H}_\infty$ one) are optimized with an EMO algorithm. Another classical application of EMO for static controllers optimization consider fuzzy rule set optimization for fuzzy controllers, some examples can be found in [8, 9].

When considering dynamic single-objective optimization problems, several studies are available in the literature [5, 10, 11, 12] about the use of genetic algorithms for. Major modifications in the operators are required for a prompt reaction to time dependent changing. Moreover, several non-GA strategies for dynamic optimization procedure for single objective problems are also proposed in the literature. But when is concerned, very few studies are available in literature [13, 14, 15].

In [16], the authors introduced artificial-life inspired algorithm for dynamic single-objective optimization problems. may be defined as a lower bound for AI following the idea that "the dumbest smart thing you can do is stay alive". This funny motto has deep meaning when ALife is considered for computational purposes [17, 18]. If life and interactions among individuals in a changing environment is itself a type of intelligence, it may be exploited for developing searching algorithms. While classical evolutionary algorithms (GA and ES) consider Darwinian evolution as a type of intelligence to be exploited [19], the proposed method uses life of individuals in a population as a basic form of intelligence and exploits this for search in a dynamic environment.

In this paper, we make an attempt to extend this approach to dynamic multi objective test cases.

2 Problem Setting and Test Cases Description

A dynamic non-linear multiobjective problem can be defined in the following way

Definition 1. *Let t be the time variable,* **V** *and* **W** *be n-dimensional and M-dimensional continuous or discrete vector spaces,* **g** *and* **h** *be two functions defining inequalities and equalities constraints and* **f** *be a function from* $\mathbf{V} \times t$ *to* **W**. *A* dynamic non-linear multi-criteria (minimum) optimization problem *with M objectives is defined as:*

$$\begin{cases} \min\limits_{\mathbf{v}\in\mathbf{V}} \mathbf{f} = \{f_1(\mathbf{v},t),\ldots,f_M(\mathbf{v},t)\} \\ \text{s.t.}\ \mathbf{g}(\mathbf{v},t) \le 0,\ \mathbf{h}(\mathbf{v},t) = 0\ . \end{cases}$$

In problem 1 some variables are available for optimization (v) and some other (the time t) are imposed parameters being independent from optimization variables; both objective functions and constraints are parameter-dependent. A more general definition of the problem can be found in [15].

Definition 2. *We call* at time t $(\mathcal{S}_P(t))$ *and* at time t $(\mathcal{F}_P(t))$ *the set of Pareto-optimal solutions at time t in design domain and objective domain, respectively.*

Unlike in the single-objective optimization problems, here we are dealing with two different search spaces: decision variable space and objective space. Therefore, the following are the four possible ways a problem can dynamically change. **Type I**: The Pareto-optimal set (optimal decision variables) $\mathcal{S}_P$ changes, whereas the pareto-optimal front (optimal objective values) $\mathcal{F}_P$ does not change. **Type II**: Both $\mathcal{S}_P$ and $\mathcal{F}_P$ change. **Type III**: $\mathcal{S}_P$ does not change, whereas $\mathcal{F}_P$ changes. **Type IV**: Both $\mathcal{S}_P$ and $\mathcal{F}_P$ do not change, although the problem can dynamically change.

A straightforward extension of ZDT and DTLZ test problems developed earlier [20, 21] for two and higher objectives can be considered in order to insert time dependance factors into multiobjective optimization test cases [15]. As it is well known, ZDT and DTLZ problems provide different difficulties which may be encountered when considering real-life multiobjective optimization problems: non-concavity, discontinuity, deceptiveness, presence of local fronts, etc.

When solving dynamically changed problems, such difficulties may transform themselves from one of the above features to another with random sudden jumps or with a gradual change. A generic test problem for such a dynamic situation is presented in the following equation:

$$\min_{\mathbf{x}}(f_1(\mathbf{x}), f_2(\mathbf{x})) = (f1(\mathbf{x}_I), g(\mathbf{x}_{II}) \cdot h(\mathbf{x}_{III}, f_1, g)) \tag{1}$$

where $\mathbf{x}_I$,$\mathbf{x}_{II}$ and $\mathbf{x}_{III}$ are subsets of design variables set $\mathbf{x}$. In the above test problem, there are three functions f_1, g, and h. In the original paper, the following functions were suggested:

$$f_1(\mathbf{x}_I) = x_1,\ g(\mathbf{x}_{II}) = \sum_{x_i \in \mathbf{X}_{II}} x_i^2,\ h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^2. \tag{2}$$

Each of them can change dynamically or in combination. In dynamic multi-objective test cases the functions f_1, g and h are re-defined in terms of three new time dependent functions F, G and H.

In this paper we consider only one test case the FDA1 (see [22]):

Definition 3 (FDA1). *Type I, convex POFs*

$$\begin{cases} f1(\mathbf{x}_I) = x_1, \\ g(\mathbf{x}_{II}) = 1 + \sum_{x_i \in \mathbf{X}_{II}} (x_i - G(t))^2, \\ h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}, \\ G(t) = \sin(0.5\pi t), \quad t = \frac{1}{n_t} \lfloor \frac{\tau}{\tau_T} \rfloor, \\ \mathbf{x}_I = (x_1) \in [0, 1], \quad \mathbf{x}_{II} = (x_2, \ldots, x_n) \in [-1, 1]. \end{cases} \tag{3}$$

Here, τ is the generation counter, τ_T is the number of generation for which t remains fixed, and n_t is the number of distinct steps in t. The suggested number of variables

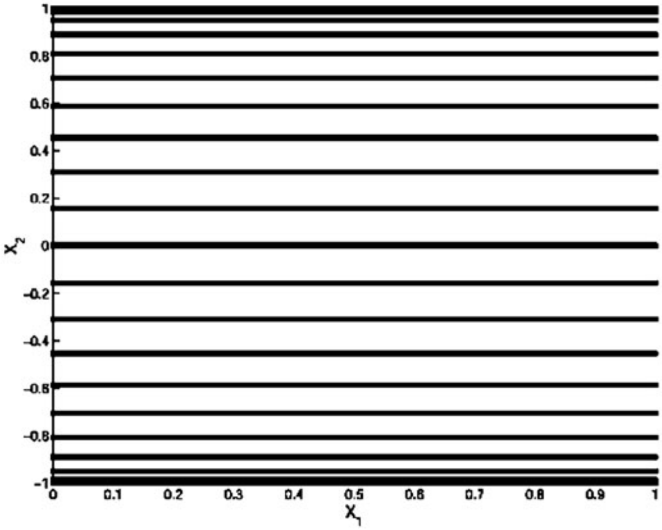

Fig. 1. $\mathcal{S}_P(t)$ for FDA1, first two decision variables, 24 time steps

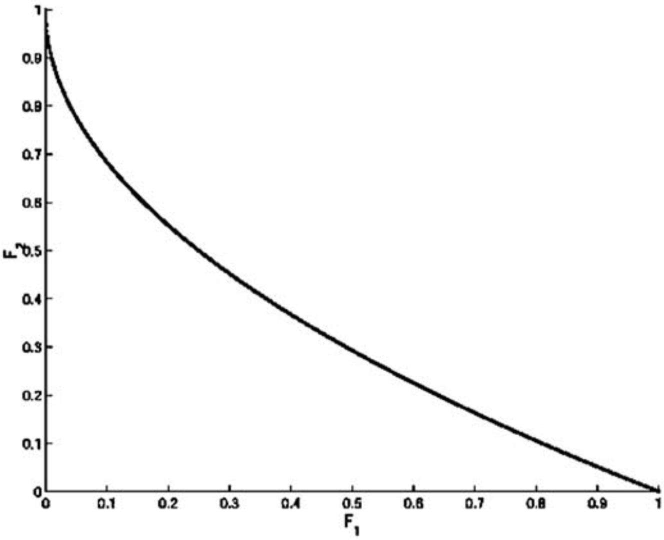

Fig. 2. $\mathcal{F}_P(t)$ for FDA1, 24 time steps

is $n = 20, \tau_T = 5$, and $n_T = 10$. In this problem the Pareto optimal front does not change, while the optimal set (in search space) suddenly change over time every τ_T iterations (as shown in Fig. 2 and Fig. 1 respectively). The task of a dynamic MOEA would be to find the same Pareto-optimal front $f_2 = 1 - \sqrt{f_1}$ every time there is a change in t.

3 Outline of the ALife-Inspired Algorithm for Dynamic Multiobjective Optimization Problems

GAs are based on the simulation of nature evolution and the exploitation of Darwinian natural selection operators; they consider coded strings as genotypes of individuals. For this reason they may be defined a low level evolution imitation, where artificial operators are considered imitating natural operators on genes. On the contrary, the proposed algorithm is based on a population level evolution: coded strings are considered as individuals interacting in a population, and artificial operators imitate interactions between individuals (like meeting, fight and reproduction). In this approach there is no a priori selection; each individual has the same probability of meeting another individual. In some cases, they will procreate two sons, which are added to the population without eliminating the parent. In other cases, they will fight and the stronger (i.e., the one that dominates in Pareto sense the other) will kill the other one. Moreover, individuals which do not encounter anybody else can reproduce in asexual way; hence a new individual (a mutation of his parent) is inserted in the population. As a consequence of all these operators, the population size is variable.

The aim of the algorithm we propose is not to definitively converge, but to be able to “sense” the changing of the Pareto optimal set or front and then automatically follow it.

The general behaviour of the algorithm is depicted in Fig. 3. Each time an individual is considered, he can meet or not another individual according to a probability

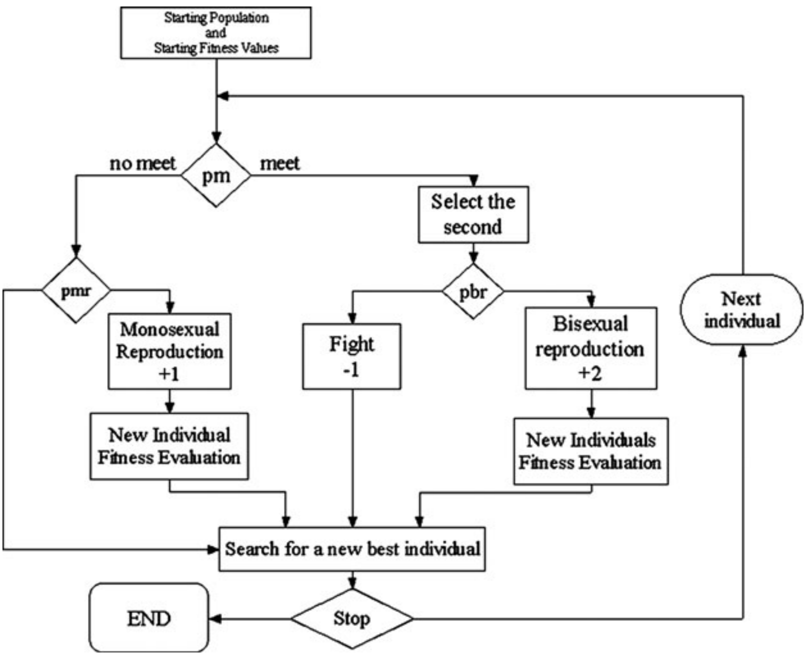

Fig. 3. Principle flowchart of the algorithm

p_m. If he meets someone else, either reproduction or competition can occur. Otherwise can take place or nothing happens. The *meeting probability* p_m is defined in terms of the actual size N_i of the population at iteration i and the maximum size $N_{\max}$ (fixed a priori) in the following way:

$$p_{\mathrm{m}} = \frac{N_i}{N_{\max}} \tag{4}$$

Thus for each individual a value r in $[0,1]$ is randomly chosen. If $r > p_{\mathrm{m}}$ meeting occurs, no meeting otherwise. In this way, when the maximum individual number is approached the meeting probability is very high and viceversa. Consequences of this will be clearer later on.

When the meeting probability is satisfied a new individual is randomly selected for meeting with the current individual. When two individuals meet *either bisexual reproduction* or *fight* can occur; the probability for bisexual reproduction p_{br} is the following:

$$p_{\mathrm{br}} = 1 - p_{\mathrm{m}} \tag{5}$$

Two new individuals are then added to the population, (for further details on bisexual reproduction see the dedicated paragraph below). If bisexual reproduction does not occur, fight is performed between the two selected individuals; the better kills the other one. This operator thus reduces the population size by one.

If meeting does not occur either asexual reproduction or nothing happens. Asexual reproduction, which is performed with probability p_{ar} (equal to p_{br}), adds a new individual to the population. More details on the operators can be found in [16].

In the actual implementation of the algorithm the bisexual reproductions is the standard random crossover operation. Thus the fight operator is the only one that involves the evaluation of the objective functions. When fight occurs, all the objective functions f_j are evaluated for both the individuals. The dominating individual in Pareto sense[1] survives, while the dominated one dies and is eliminated from the population. If nobody dominates the other, the algorithm eliminates the individual with a greater number of individual in a given neighborhood. This happens in order to preserve diversity among individuals.

The probabilistic routing strategy leads to the population size N behavior shown in Fig. 4. As can be seen it oscillates around a probabilistic computable value satisfying the following logistic formula:

$$N_{i+1} = N_i \left(1 - 2\frac{N_i}{N_{\max}}\right) + \frac{1}{N_{\max}}\ ; \tag{6}$$

where i is the iteration index. Two limit behaviors correspond to $N \sim N_{\max} \Rightarrow p_{\mathrm{m}} \sim 1$ and $N \sim 0 \Rightarrow p_{\mathrm{m}} \sim 0$. In the former case meeting and fight always occurs and population size reduces. In the latter meeting never occurs and asexual reproduction

[1] Let $\mathbf{v}_1, \mathbf{v}$ be two candidate solutions (individuals). Then $\mathbf{v}_1$ is said to *dominate* $\mathbf{v}_2$ in the Pareto sense if and only if the following conditions hold: (i) $f_i(\mathbf{v}_1) \leq f_i(\mathbf{v}_2)$ for all $i \in \{1,2,\ldots,M\}$, (ii) $f_j(\mathbf{v}_1) < f_j(\mathbf{v}_2)$ for at least one $j \in \{1,2,\ldots,M\}$.

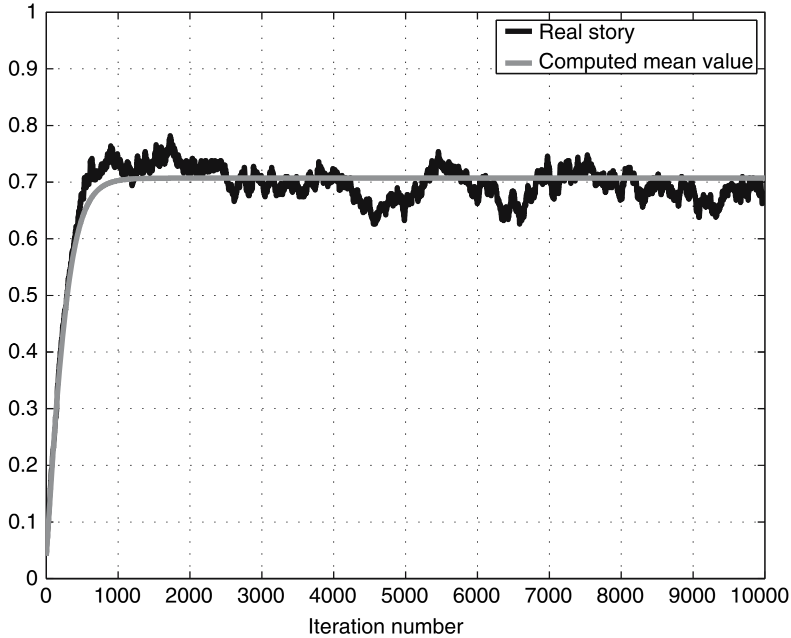

Fig. 4. Population size story compared with the computed analytical mean value

always adds one individual; consequently population size increase by one at each iteration.

The proposed strategy is supposed to run for an indeterminate time following system changing, without definitely converging towards a final optimum unless a static system is considered. For test problems a fictitious maximum iteration or generation number is imposed but it only has an obvious practical meaning. The algorithm is extremely flexible because probability threshold values are updated at each iteration.

4 Application of the ALife-Inspired Algorithm to Test Case FDA1

For the sake of clarity, we consider a problem with a two dimensional input space (X_1, X_2) and two objective functions (f_1, f_2). In this way we can easily plot the problem in both domains. Moreover we consider only two changes of the Pareto optimal set.

Since the population size is not fixed and the algorithm proceeds individual by individual, it is not properly correct to speak of "epochs". However we will use this term to signify the application of the algorithm a number of times (iterations) equal to the average size of the population.

The starting population is uniformly distributed in the search space (Fig. 5), while the analytical optimal set is the straight line $x_2 = 0$ (dotted line in Fig. 6). After 45 epochs (Figs. 7, 8), there is a good approximation of the Pareto-optimal set and front.

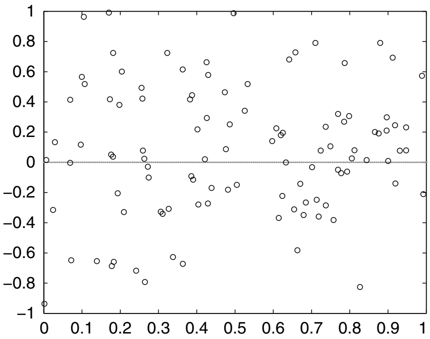

Fig. 5. Starting population in search space

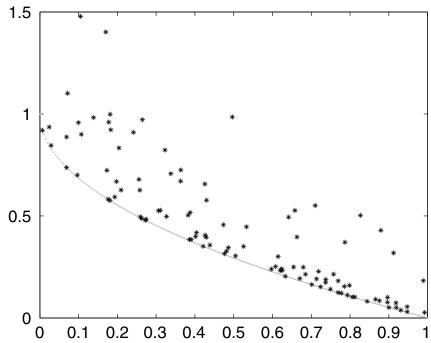

Fig. 6. Starting Population in objective space

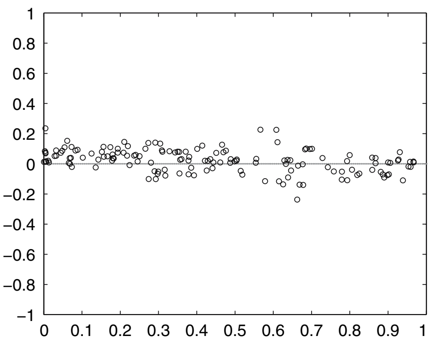

Fig. 7. Population in search space at epoch 45

As it is well known a good approximation of Pareto-optimal front (and set) requires the solution to be (i) close to the exact front (and set), and (ii) as distributed as possible on the front (and set); this two requirements are usually clashing in multiobjective evolutionary algorithms. Due to the absence of selection pressure (there is no fitness

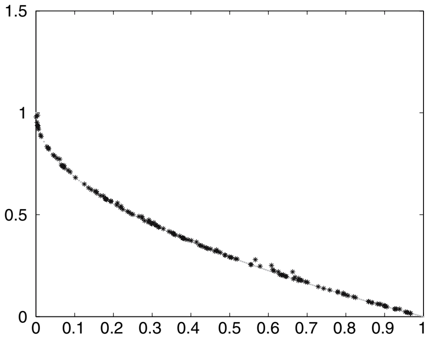

Fig. 8. Population in objective space at epoch 45

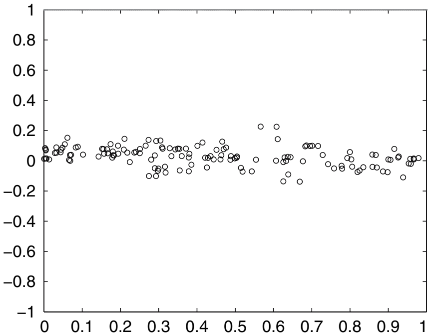

Fig. 9. Epoch 46: Sudden change of the optimal front in search space; now it becomes the straight (*dotted*) line $x_2 = 1$

based selection in the proposed algorithm) the population covers the entire front and set. Moreover the absence of fitness based selection is one of the main differences between the proposed algorithm and the evolution based multiobjective optimization algorithms. During the 46th epoch there is a sudden change of the Pareto-optimal set; now it is the straight line $x_2 = 1$. Consequently there is a big approximation error both in search and objective space (Figs. 9, 10). Finally after 100 epochs the population is again a good approximation of Pareto-optimal set and front (Figs. 11, 12). As evident the absence of fitness based selection is a drawback when speed of reaction to time-dependent changes is concerned.

The main drawback of this algorithm is that, in general, it converges slowly (in term of number of epochs). Moreover the converge velocity strongly depends on the distribution of the population in search space. In fact after the sudden change (epoch 46), the convergence to the new optimal set is much more slower than the previous one. Slowness of ALife algorithm is a known problem. For single-objective optimization problem there is a speed-up by introducing crossover operators that privilege the better parent (see [16]). However this kind of operators cannot be easily

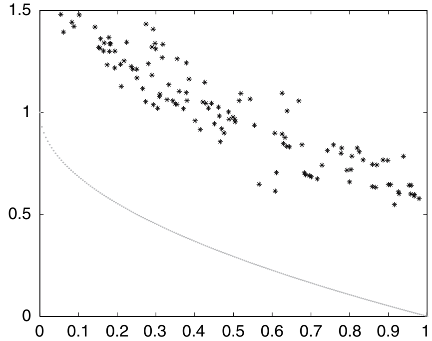

Fig. 10. Population in objective space at epoch 46

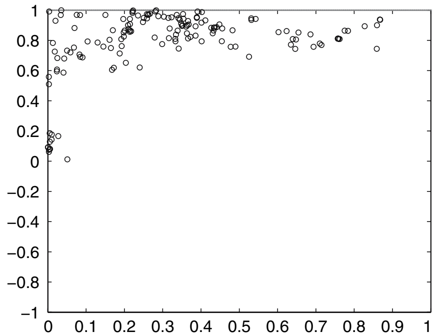

Fig. 11. Population in search space at epoch 100

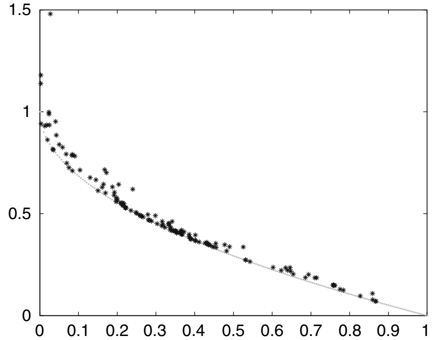

Fig. 12. Population in objective space at epoch 100

introduced in the Multiobjective algorithm, because we have to keep the diversity among the individuals (both in search and objective space).

On the other side, this algorithm has two main advantages. The first is that it is able to automatically follow the changes a dynamic Multiobjective optimization problem, without any external help.

The second one is that the evaluation of the objective function (being usually a computationally expensive task or requiring a measure on the system) is needed only when the meeting probability is satisfied and not at each iteration, as it is required for fitness based selection in an evolution based algorithm.

5 Conclusion

In this paper we introduced an ALife-inspired evolutionary algorithm for dynamical multiobjective optimization problems. Although not flawless, this algorithm is simple to implement and able to detect the change of objectives or constraints of the problem, and then to follow the Pareto-optimal set and fronts.

Our work is only at a preliminary state. Further developments may be concerned with the following considerations.

On one side, the convergence to the optimal front and set could be fastened (in terms of number of iteration) by exploiting the information about Pareto optimality. For example the worst individuals in Pareto sense could be automatically eliminated (without having to fight). Or, on the other side, the Pareto optimal individuals could gain some advantages (re-introducing in this way a kind of selection pressure).

On the other side, the number of objective function evaluation could be decreased by changing the probability route leading to a fight (the only operator that requires objective evaluation). The corresponding increasing of population size (fight is also the only operator that decrease population size) may be balanced by introducing, for example, a kind of spontaneous decay (as in ant systems) – i.e. the automatic elimination of the individuals that lived for more than a given iteration threshold.

References

1. M. A. Lee and H. Esbensen. Fuzzy/Multiobjective Genetic Systems for Intelligent Systems Design Tools and Components. In Witold Pedrycz, editor, *Fuzzy Evolutionary Computation*, pp. 57–80. Kluwer Academic Publishers, Boston, Massachusetts, 1997.
2. P.M. Reed and B.S. Minsker. Discovery & Negotiation using Multiobjective Genetic Algorithms: A Case Study in Groundwater Monitoring Design. In *Proceedings of Hydroinformatics 2002*, Cardiff, UK, 2002.
3. Kalyanmoy Deb and Tushar Goel. A Hybrid Multi-Objective Evolutionary Approach to Engineering Shape Design. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pp. 385–399. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

4. K.C. Tan, K. Sengupta, T.H. Lee, and R. Sthikannan. Autonomous Registration of Disparate Spatial Data via an Evolutionary Algorithm Toolbox. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pp. 31–36, Piscataway, New Jersey, May 2002. IEEE Service Center.
5. F. Vavak, K. A. Jukes, and T. C. Fogarty. Performance of a genetic algorithm with variable local search range relative to frequency of the environmental changes. *Genetic Programming 1998: Proceedings of the Third Annual Conference*, 1998.
6. M. Farina, P. Amato, and K. Deb. Dynamic multi-objective optimization problems: Test cases, approximations and applications. *IEEE Transactions on Evolutionary Computation*, 8(5):425–442, 2004.
7. Carlos Manuel Mira de Fonseca. *Multiobjective Genetic Algorithms with Applications to Control Engineering Problems*. PhD thesis, Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK, September 1995.
8. Jessica M. Anderson, Tessa M. Sayers, and M. G. H. Bell. Optimization of a Fuzzy Logic Traffic Signal Controller by a Multiobjective Genetic Algorithm. In *Proceedings of the Ninth International Conference on Road Transport Information and Control*, pp. 186–190, London, April 1998. IEE.
9. Anna L. Blumel, Evan J. Hughes, and Brian A. White. Fuzzy Autopilot Design using a Multiobjective Evolutionary Algorithm. In *2000 Congress on Evolutionary Computation*, volume 1, pages 54–61, Piscataway, New Jersey, July 2000. IEEE Service Center.
10. Christopher Ronnewinkel, Claus O. Wilke, and Thomas Martinetz. Genetic algorithms in time-dependent environments. In L. Kallel, B. Naudts, and A. Rogers, editors, *Theoretical Aspects of Evolutionary Computing*, pp. 263–288, Berlin, 2000. Springer.
11. J. Branke. Evolutionary approaches to dynamic optimization problems – A survey. *Juergen Branke and Thomas Baeck editors: Evolutionary Algorithms for Dynamic Optimization Problems*, 13:134–137, 1999.
12. J.J. Grefenstette. Evolvability in dynamic fitness landscapes: A genetic algorithm approach. *Proc. Congress on Evolutionary Computation (CEC99) Washington DC IEEE press*, pp. 2031–2038, 1999.
13. Zafer Bingul, Ali Sekmen, and Saleh Zein-Sabatto. Adaptive Genetic Algorithms Applied to Dynamic Multi-Objective Problems. In Cihan H. Dagli, Anna L. Buczak, Joydeep Ghosh, Mark Embrechts, Okan Ersoy, and Stephen Kercel, editors, *Proceedings of the Artificial Neural Networks in Engineering Conference (ANNIE'2000)*, pp. 273–278, New York, 2000. ASME Press.
14. Kazuo Yamasaki. Dynamic Pareto Optimum GA against the changing environments. In *2001 Genetic and Evolutionary Computation Conference. Workshop Program*, pp. 47–50, San Francisco, California, July 2001.
15. M. Farina, K. Deb, and P. Amato. Dynamic multiobjective optimization problems: Test cases, approximation and applications. *To be published in the Proceedings of EMO'2003*, 2003.
16. P. Amato, M. Farina, G. Palma, and D. Porto. An alife-inspired evolutionary algorithm for adaptive control of time-varying systems. In *Proceedings of the EUROGEN2001 Conference, Athens, Greece, September 19-21, 2001*, pp. 227–222. International Center for Numerical Methods in Engineering (CIMNE), Barcelona, Spain, March 2002.
17. C.G. Langton. *Artificial life: an overview*. MIT Press, 1995.
18. C. Adami. *Introduction to Artificial life*. Springher-Verlag, 1998.
19. M. Mitchell and S. Forrest. Genetic algorithms and artificial life. *Santa Fe Institute Working Paper 93-11-072*. (to appear in Artificial Life).
20. Kalyanmoy Deb. Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. *Evolutionary Computation*, 7(3):205–230, Fall 1999.

21. Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable Test Problems for Evolutionary Multi-Objective Optimization. Technical Report 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.
22. Marco Farina, Alessandro Bramanti, and Paolo Di Barba. A GRS Method for Pareto-Optimal Front Identification in Electromagnetic Synthesis. *IEE Proceedings–Science, Measurement and Technology*, 2002. (In Press).