Predator-Prey Evolution Strategy

for Solving Multi-Objective Optimization Problems

André A. Keller
Laboratoire d'Informatique Fondamentale de Lille / SMAC
Université de Lille Nord de France
Villeneuve d'Ascq,59655 France，+33(0)640123190
Andre.Keller@univ-lille1.fr

***Abstract*—This paper introduces to a particular algorithm inspired from Nature for solving multi-objective optimization problems. The predator-prey strategy is used for finding Pareto optimal fronts. This is the first essential step in the decision-making process, in which the decision makers are confronted to multiple objectives. The computations have been carried out by using software packages, such as Wolfram *Mathematica* ® and the GA-based optimization software packages GENOCOP III and NSGA II[1].**

Keywords- multi-objective optimization problem; Predator-prey strategy; Pareto-optimal front;

I. INTRODUCTION

This paper introduces to the predator-prey (PP) algorithm for solving continuous multi-objective optimization problems. Solving automated problems by using Darwinian principles originated in the fifties. The PP algorithm belongs to the class of evolution strategies by I. Rechenberg [12] and H-P. Schwefel [13], one of the three components of the evolutionary computation, besides genetic algorithms (GAs) by J.H. Holland and evolutionary programming by L.J. Fogel. The PP evolution strategy also belongs to the class of non-elitist algorithms[2]. [1, 2, 5].

II. EVOLUTIONARY OPTIMIZATION

A. Single Objective Optimization Problem

Let the nonlinear multi-dimensional bounded programming problem of the form

$$\underset{\mathbf{x}}{\min\text{imize}}\; f(\mathbf{x}),\ \mathbf{x}\in\mathbb{R}^n \text{ s.t. } \mathbf{x}\in[\underline{\mathbf{x}},\overline{\mathbf{x}}].$$

The following two-dimensional test problem, inspired from [7], is a weighted combination of sinc functions. A sinc function is defined by

$$g(\mathbf{x}) := 50\frac{\sin\left(\sqrt{x^2+y^2}\right)}{\sqrt{x^2+y^2}} - \sqrt{x^2+y^2},\ \mathbf{x}\equiv(x,y)$$

The objective function is

$$f(\mathbf{x}) = 3\,g(x+10,y+10) + 2\,g(x-5,y+5) + 1.5\,g(x-1,y+2),\ x,y\in[-20,10]$$

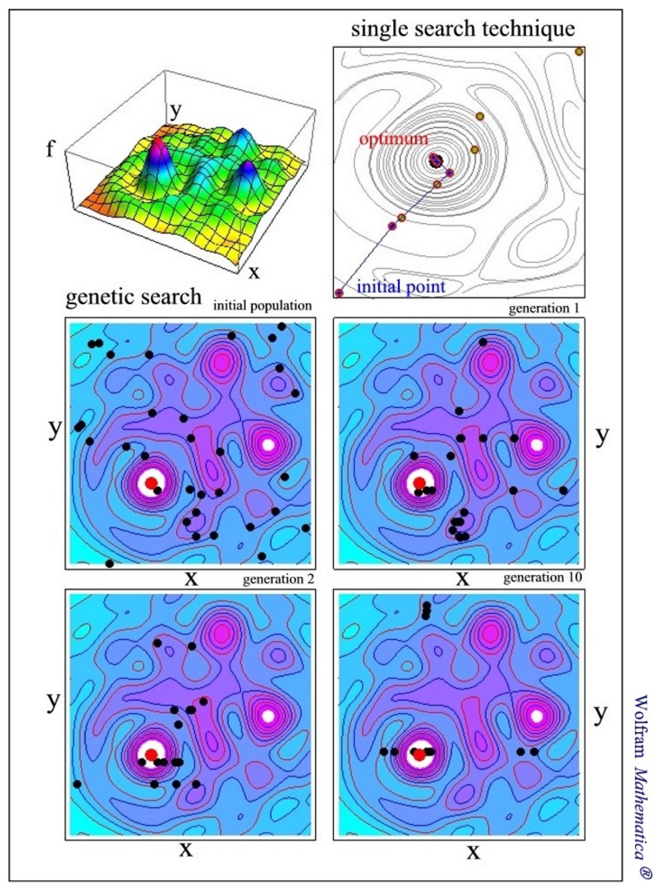

Figure 1 First ten generations.

GAs are stochastic search techniques inspired from the genetic processes of biological organisms. The mechanisms consist in encoding and reproducing populations. The adequacy of these principles to real-world optimization problems has been proved [9, 11, 14]. In *Mathematica*, the real-valued GA consists in several routines : a population of chromosomes is created randomly and the genetic processes of selection, crossover and

---

[1] GENOCOP is for Genetic algorithm for Numerical Optimization of Constrained Problems. It has been developed at UNC Charlotte/USA. NSGA is for Nondominated Sorting Algorithm. It has been developed at Kanpur Genetic Algorithms Laboratory/ India.[3]

[2] In this class, we can find also the vector evaluated GA , a weight based GA, and a niched-Pareto GA.[2]

mutation are then used at each iteration, to create the next generation.

For this demonstration, the GA is ended after 10 generations . The best result we obtain is $\hat{\mathbf{x}} = (-9.1727, -9.7184)$. The exact global optimum is $\mathbf{x}^* = (-9.898, -9.966)$ (comparable to [8]). Fig.1 pictures the first ten generations.

B. *Multi-Objective Optimization Problem*

Multi-objective optimization problems (MOOPs) involve a simultaneous optimization of multiple objectives. Because of possible conflicts between objectives, trade-offs exist. The set of trade-off designs that cannot be improved without deteriorating another objectives is the Pareto set.
The MOOP takes the generalized form

$$\left.\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n}\text{imize } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_K(\mathbf{x})) \\ \text{s.t. } g_i(\mathbf{x}) \le 0,\ i = 1, \ldots, P \\ h_i(\mathbf{x}) = 0,\ i = P+1, \ldots, M \\ \mathbf{x} \in [\underline{\mathbf{x}}, \overline{\mathbf{x}}] \end{aligned}\right] .$$

Definition 1 (Dominance relation) Let $\mathbf{f}, \mathbf{g} \in \mathbb{R}^m$, then $\mathbf{f}$ dominates $\mathbf{g}$ ($\mathbf{f} \succ \mathbf{g}$), if and only if *(1)* $f_i \ge g_i$ for all $i \in \{1, \ldots, m\}$ and *(2)* $\exists j \in \{1, \ldots, m\} : f_j > g_j$.

Definition 2 (Pareto set). Let $\mathbf{F} \subseteq \mathbb{R}^m$ a set of vectors, then the Pareto set $\mathbf{F}^* \subset \mathbf{F}$ is such that $\mathbf{F}^*$ contains all non-dominated $\mathbf{g} \in \mathbf{F}$ by $\mathbf{f} \in \mathbf{F}$. Then , we may define the Pareto set as $\mathbf{F}^* = \{\mathbf{g} \in \mathbf{F} \text{ s.t. } \nexists \mathbf{f} \in \mathbf{F} : \mathbf{f} \succ \mathbf{g}\}$

The test example by [2], pp. 176-178 is with two objectives, two inequality constraints and bounds. We have

$$\left.\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^2}\text{imize } \mathbf{f}(\mathbf{x}) = \left(f_1(\mathbf{x}) \equiv x_1, f_2(\mathbf{x}) \equiv \frac{1+x_2}{x_1} \right) \\ \text{s.t. } g_1(\mathbf{x}) \equiv 6 - 9x_1 - x_2 \le 0 \\ g_2(\mathbf{x}) \equiv 1 - 9x_1 + x_2 \le 0 \\ x_1 \in [0.1, 1], x_2 \in [0, 5] \end{aligned}\right]$$

The Pareto front is approximated by using the GA. The results are illustrated in the decision variable space and in the objective space[3]. Fig.2 shows the successive approximations over 200 iterations.

The software package NSGA II for this application is a fast and elitist multi-objective evolutionary algorithm (MOEA) , sorting the population in different fronts, by using the non-domination order relation. The system uses a Pareto ranking procedure and incorporates a fitness sharing. To form the next generation, the algorithm combines the current population and its offspring with bimodal crossover and polynomial operators[4].

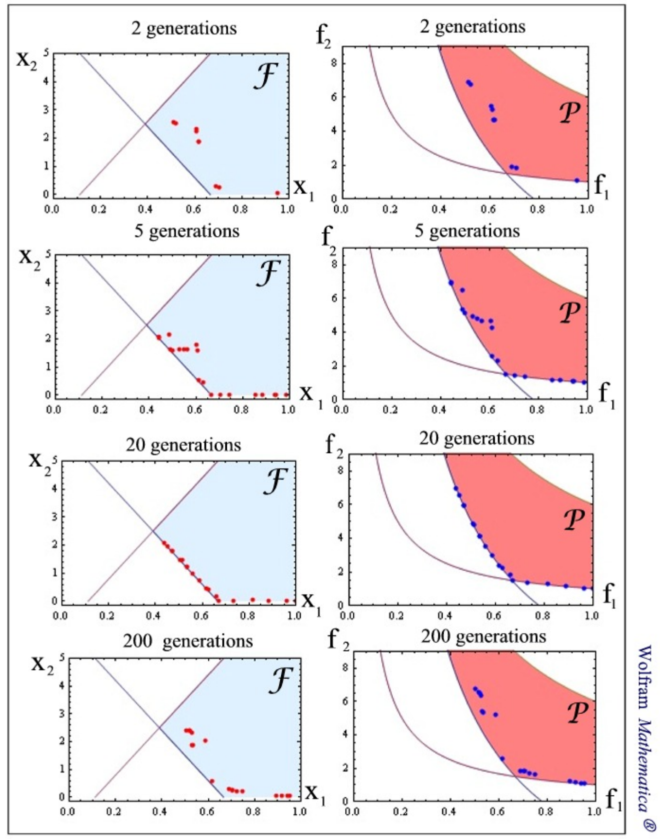

Figure 2 First 200 generations of the Deb's test example

III. PREDATOR-PREY ALGORITHM

In the Predator-Prey (PP) algorithm by Laumann et al. [10], a prey represents one feasible solution and a predator one objective. The algorithm mimics the natural phenomenon, according to which the predator eliminates the weakest prey. The PP algorithm is also used to find an approximated set of the Pareto-optimal front [4, 6].

A. *Principles*

All the preys are placed at the vertices of a toroidal grid as in Fig. 4. With this particular spatial structure, all the preys can be attained by a random walk with an equal probability. Suppose that there is a strict mapping of one predator to one objective. The predators are placed randomly on the grid search. Each predator pursues the prey within its current neighborhood and according to its own objective. Suppose that a neighborhood is defined by one step (radius one) as in Fig. 4 The principles consist in random walks and in replacing the worst solutions with mutated solutions. More precisely, a predator selects in its selection neighborhood (SN) the worst

[3] In the objective space, the lower and upper boundaries correspond respectively to $x_2 = 0$ and $x_2 = 5$. Indeed, we have $f_2 = (1 + x_2) / f_1$

[4] For this study, the package has been implemented by the author in a DOS system by using the compiler ACC from Absoft C/C++. It is also connected to *Mathematica* for preparing the inputs and analyzing the outputs.

prey which will be deleted. A reproduction neighborhood (RN) is spanned around the empty vertex. At this place, an offspring is created by mutating a randomly chosen prey in the RN.

B. Basic Algorithm

The basic algorithm for one generation may consist in the following eight steps

1. Generate a population of preys randomly in the feasible region of the optimization problem.
2. Place the preys on the vertices of a toroidal grid graph.
3. Place the predators at random on vertices of the grid graph (even in presence of a prey).
4. Assign only one objective to each predator.
5. Evaluate preys which stay in the SN and select the worst prey (the largest fitness for a minimization problem).
6. Delete the worst prey and replace it by an offspring in the RN with a higher fitness.
7. Proceed to a Gaussian mutation on the offspring.
8. Let the predators take a random walk in their neighborhood.

The expected result is to find the best approximated Pareto-optimal front. The preys with good performances w.r.t. all the objectives have more chance to survive and then to approximate the Pareto front.

C. Illustrative Problem

The following problem is drawn from [2], p. 176.

$$\left.\begin{array}{l} \min\limits_{\mathbf{x}\in\mathbb{R}^2} \text{imize } f_1(\mathbf{x}) = x_1 \\ \min\limits_{\mathbf{x}\in\mathbb{R}^2} \text{imize } f_2(\mathbf{x}) = \dfrac{1+x_2}{x_1} \\ \text{s.t. } x_1 \in [0.1,1], x_2 \in [0,5] \end{array}\right]$$

In the following, the steps of one generation will be illustrated. Fig.3 pictures, in the objective space, (f_1, f_2), a random population of 36 preys and 2 predators. These populations have to be drawn within the feasible variable space of the optimization problem.

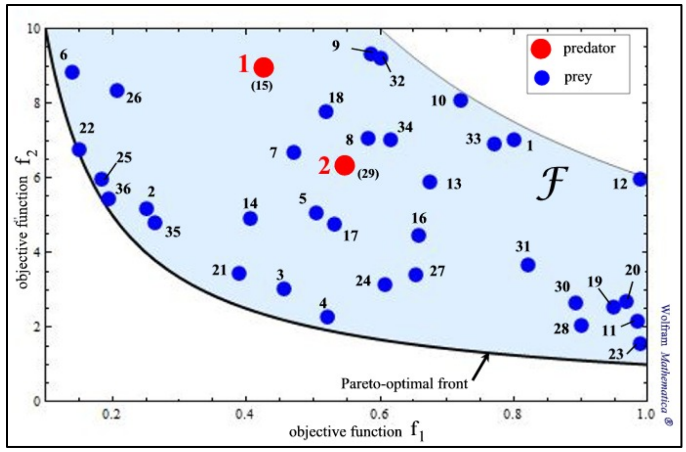

Figure 3 Random populations of predators and prey.

Consider the predator 1, to which the objective 1 has been assigned. Its position on the grid graph and its SN are shown in Fig. 4. According to Table I, prey 16 has the worst fitness $f_1(0.658, 1.936) = 0.658$ and will be suppressed from the grid graph. Around prey 16, RN is described by the set of preys $\{10,15,17,22\}$. A new prey is created by selecting randomly one of the preys belonging to the RN and by mutating it. The Gaussian mutation consists in adding to each of the coordinates a zero-mean Normal distribution , such as we get $\mathbf{y} = \mathbf{x} + \boldsymbol{\sigma}\mathcal{N}(0,1)$. Suppose that the mutation strength is $\sigma = 0.05$ for all the components. The acceptation of the new prey requires that $f_1(\mathbf{y}) < f_1(\mathbf{x})$.

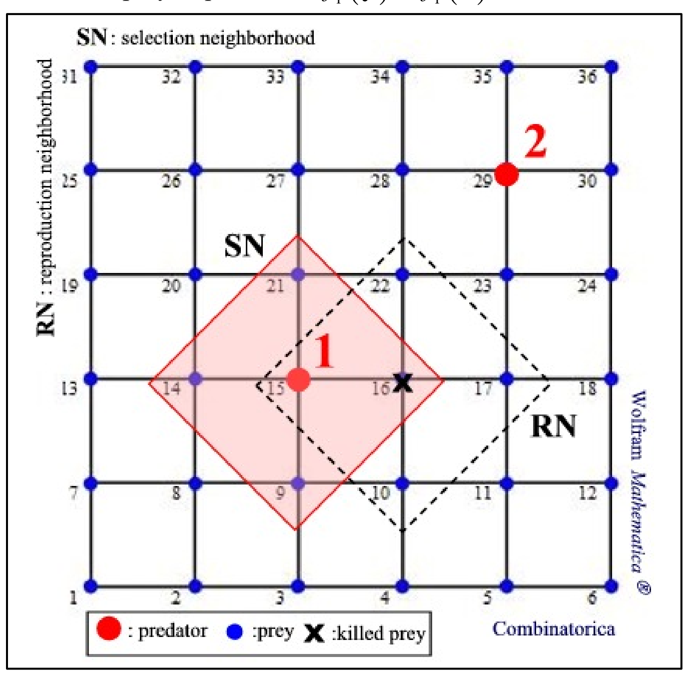

Figure 4 Toroidal spatial population structure.

The results for this first generation for the two predators are illustrated in Fig. 5.

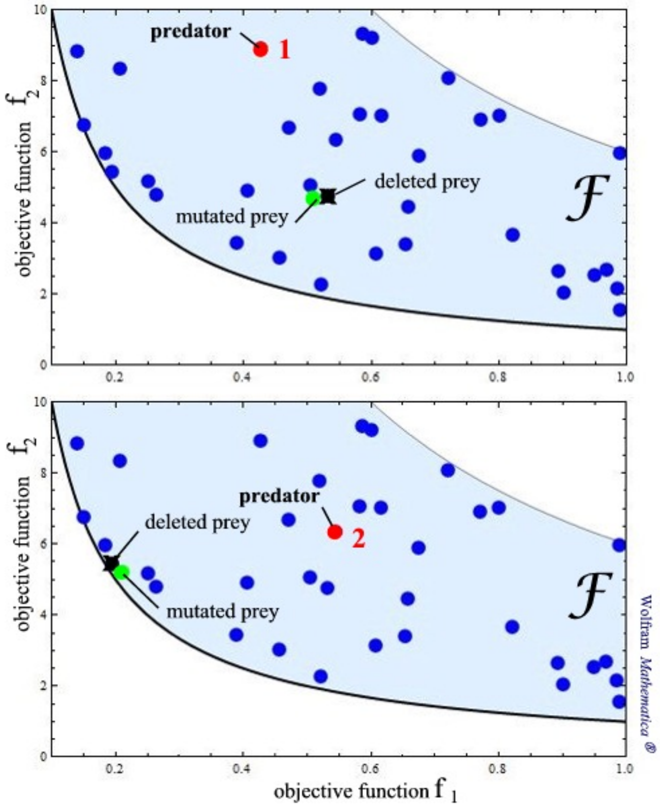

Figure 5 First generation of the PP algorithm.

The extended computation of this problem to 1,000 iterations per objective is presented by [2], p.228 for 100 preys (a 10×10 grid) with the same mutation strength. The preys well spread along the whole the Pareto-optimal front, even with more solutions in its intermediate region. However, the diversity of the outputs disappears with more than 10,000 iterations. This is one limitation of this elementary version of the PP algorithm.

IV. CONCLUSION

Further improvements of the PP evolution strategy have been proposed, such as : the implementation of a diversity-preserving operator (the problem just mentioned above with numerous iterations), the affectation of more than one objective to each predator (e.g. a weighted sum of objectives), and multiple variable mutation rates (e.g. a decreasing mutation schedule).

TABLE I. SELECTION NEIGHBORING OF PREDATOR 1

Prey	Selection neighboring at vertex 15		
	x_1	x_2	$f_1(\mathbf{x})$
9	0.587	4.464	0.587
14	0.405	0.988	0.405
16	0.658	1.936	0.658
21	0.390	0.343	0.390

REFERENCES

[1] T. Bäck, U. Hammel, and H.-P. Schwefel, "Evolutionary computation: comments and current state", IEEE Trans. on Evol. Comput., vol.1, no 1, 1997, pp.3-17.

[2] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, Chichester, UK: J. Wiley, 2001.

[3] K. Deb and U.B. Rao N,,"Investigating predator-prey algorithms for multi-objective optimization", Kanpur Genetic Algorithms Laboratory (KanGAL), Kanpur, India, 2005.

[4] D.B. Fogel,, "An introduction to simulated evolutionary optimization, IEEE Trans. on Neural Networks, vol. 5, no 1, 1994, pp. 3-14.

[5] C.M. Fonseca and P.J. Fleming, "An overview of evolutionary algorithm in multiobjective optimization", Evol. Comput., vol. 3, no 1, 1995, pp. 1-16.

[6] C. Grimme and J. Lepping, "Designing multi-objective variation operators using a predator-prey approach", Proc. Int. Conf. on Evolutionary Multi-Criterion Optimization, Berlin: Springer-Verlag, 2007, pp. 21-35.

[7] C. Jacob, Illustrating Evolutionary Computation with *Mathematica*, San Diego, CA: Academic Press, 2001.

[8] A.A. Keller, "Genetic search algorithms to fuzzy multiobjective games: a *Mathematica* implementation", 10th WSEAS Int. Conf. on Applied Computer Science (ACS'10), Morioka, Japan, 2010, pp. 351-359.

[9] J.R. Koza, Genetic Programming: on the Programming of Computers by Means of Natural Selection, Cambridge, Mass. The MIT Press, 1992.

[10] M. Laumann, G. Rudolph, and H.-P. Schwefel, "A spatial predator-prey approach to multi-objective optimization: a preliminary study", in A.E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel (eds), PPSN, Amsterdam, Hollland: Springer-Verlag, 1998, pp. 241-249.

[11] Z. Michalewicz, Genetic Algorithms+ Data Structures=Evolution Programs, 3rd ed., Berlin: Springer-Verlag, 1999.

[12] I. Rechenberg, Evolutionsstrategie Optimierung technischer Systeme nach Prinzipien der biologischen Evolution, Stuttgart, Germany: Fromann-Holzboog Verlag, 1973.

[13] H.-P. Schwefel, Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie, Basel: Birkhäuser, 1977.

[14] E. Zitler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach", IEEE Trans. on Evol. Comput., vol. 3, no 4, 1994, pp. 251-271.