

A Pareto-Evolutionary Approach for Goal and Priority Based Multi-objective Optimization Problems

Fernando Jiménez Gracia Sánchez Antonio Gómez-Skarmeta

Dept. Ingeniería de la Información y las Comunicaciones

Universidad de Murcia

Campus de Espinardo

30071 Espinardo, Murcia, Spain

e-mail: {fernan,gracia,skarmeta}@dif.um.es

Abstract

Multi-objective evolutionary algorithms for optimization have received much attention in recent literature. In this paper we propose a new Pareto-based Multi-objective Evolutionary Algorithm to solve the vector optimization problem. This algorithm uses a variant of the preselection scheme as implicit niche formation technique. In addition we propose an approach to solve goal and priority based optimization problems by using the above multi-objective evolutionary algorithm. This approach allows us to solve a wide set of optimization problems, including particular cases such as the vector optimization problem, constrained optimization, constraint satisfaction, and goal programming. Good results have been obtained for different test problems studied by other authors.

1 Introduction

The potential effectiveness of Evolutionary Algorithms (EA) in multi-objective search and optimization has been widely recognized in recent years [1, 3, 4]. In this paper we propose a Pareto-based Multi-objective EA to solve the vector optimization problem which uses a variant of the preselection scheme [2], and we also propose an approach to solve optimization problem based on goals and priorities by using the previous EA. Particular cases like the vector optimization problem, constrained optimization, constraint satisfaction, and goal programming can be solved with the proposed approach. The importance of the proposed approach arises from the separation of the technique to handle goals and priorities, and the specific implementation aspects of the optimizer (such as niche formation technique, selection mechanism, etc.). Thus, existing EAs for the vector

optimization problem can also be used to solve optimization problems based on goals and priorities by means of the proposed approach.

2 A Pareto-based evolutionary algorithm for the vector optimization problem

The multi-objective optimization problem consists of minimizing, without loss of generality, the p components of a vector function \mathbf{f} of a decision variable \mathbf{x} in a universe X , i.e.

$$\text{Minimize } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_p(\mathbf{x})) \quad (1)$$

This problem is usually called the *vector optimization problem* and it acquires importance when objective functions are non commensurable, competing or conflicting measures. The set of solutions for problem (1) is composed of all those elements of the search space for which the corresponding objective vector cannot be simultaneously improved for all components. These solutions are called *non-dominated*, *non-inferior* or *Pareto-optimal*. Given two decision vectors $\mathbf{x}, \mathbf{x}' \in X$, \mathbf{x} is said to *dominate* \mathbf{x}' if $f_i(x) \leq f_i(x')$, for all objective functions f_i , and $f_j(x) < f_j(x')$, for at least one objective function f_j . A decision vector \mathbf{x} is said to be *Pareto-optimal* if there is not a decision vector which dominates \mathbf{x} .

We use an implicit niche formation technique [2], which is a variant of the preselection scheme adapted for multi-objective optimization. The main characteristics of the EA are the following:

- The proposed algorithm is a Pareto-based multi-objective EA to solve the vector optimization problem as in (1). It has been designed to find, in a single

run, multiple non-dominated solutions in agreement with the Pareto decision strategy. The EA minimizes all objective functions.

- The EA uses a real-coded representation.
- The initial population is generated randomly with a uniform distribution within the boundaries of the search space.
- The EA uses the following variant of the preselection scheme:

In each iteration of the EA, two individuals are picked at random from the population. These individuals are crossed $nChildren$ times and children mutated resulting in $nChildren$ pairs of candidates. The best of the first group of candidates replaces the first parent, and the best of the second group replaces the second parent only if the offspring is better than the parent. An individual I is better than another individual J if I dominates J . A best individual of a collection is any individual I such that there is no other individual J which dominates I .

Note that the preselection scheme is an implicit niche formation technique to maintain diversity in the populations because an offspring replaces an individual similar to itself (one of its parents). Moreover, the preselection scheme is also an elitist strategy because the best individual in the population is replaced only by a better one.

- Two crossover operators, *uniform* and *arithmetic*, and two mutation operators, *uniform* and *non uniform*, are used [7].

3 Solving goal and priority based multi-objective optimization problems

We can use the above EA to solve other optimization problems involving goals and priorities. Goals indicate desired levels of performance in each objective and priorities are integer values which determine in which order objectives are to be optimized.

We consider the following formulation:

$$\text{Minimize } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_p(\mathbf{x})) \quad (2)$$

with goal levels $\mathbf{u} = (u_1, \dots, u_p)$ and priorities $\mathbf{p} = (p_1, \dots, p_p)$.

Priorities and goal levels can accommodate a whole variety of optimization problems. Besides the vector optimization problem in which all objectives have equal priority and no goal levels are given, other particular cases of the formulation (2) are:

- *Constrained Optimization*: constraints are high-priority objectives with goal levels and objectives are assigned the lowest priority with no goal levels.
- *Constraint Satisfaction*: as in constraint optimization but with no objective to be optimized.
- *Goal Programming*: objectives have goal levels which can be reached simultaneously or sequentially.

In order to solve problems as in (2), we consider the following objective vector $\mathbf{f}'(\mathbf{x}) = (f'_1(\mathbf{x}), \dots, f'_p(\mathbf{x}))$, where functions $f'_i(\mathbf{x})$ are defined as follows:

$$f'_i(\mathbf{x}) = \begin{cases} m & \text{if } h(i) \text{ and } f_i(\mathbf{x}) \leq u_i \\ f_i(\mathbf{x}) & \text{if } h(i) \\ M & \text{otherwise} \end{cases} \quad (3)$$

where m is a small enough number ($m \leq u_i, \forall i = 1, \dots, p$), M is a large enough number, and $h : \{1, \dots, p\} \rightarrow \{0, 1\}$ is a Boolean function defined as follows:

$$h(i) = \begin{cases} 1 & \text{if } f_j(\mathbf{x}) \leq u_j, \forall j : p_j > p_i \\ 0 & \text{otherwise} \end{cases}$$

Each new objective function f'_i takes the best possible value m when all objectives with lower priority and itself are reached. In other cases, if all objectives with lower priority are reached, then the new function f'_i takes the former value f_i , and finally, if some objective with lower priority is not reached, the new objective function f'_i take the worst possible value M .

Now we can use the EA described in the previous section with the objective vector $f'_i(\mathbf{x})$ to solve the former problem (2).

4 Experiments and results

We consider the following test problems in order to check out the proposed technique and the Pareto-based multi-objective EA:

- *Constrained optimization*: Test problems $G1, \dots, G11$ reported in [6, 7].
- *Unconstrained multi-objective optimization*: Test problems $F1$ and $F2$ reported in [8].
- *Constrained multi-objective optimization*: Test problem $F3$ reported in [8] and test problem reported in [5].

Table 1 summarizes the results for the constrained optimization test problems. For each test problem, the optimum, worst, best and average values (over 10 runs) are reported. We compared our results with those obtained by different evolutionary approaches for constrained optimization proposed in [6, 7].

Figures 1, 2 and 3 graphically show the non-dominated solutions obtained for the unconstrained and constrained multi-objective test problems.

The following values for the parameters of the EA were used in the simulations: population size 200, crossover probability 0.9, mutation probability 0.2, and number of children for the preselection scheme 10. All crossover and mutation operators are applied with the same probability. The EA stops when the solutions satisfy the decision maker.

5 Conclusions

This paper supposes a twofold contribution in Evolutionary Computation. In the first place we propose a new Pareto-based Multi-objective EA to solve the vector optimization problem. This algorithm uses a variant of the preselection scheme as opposed to the multi-objective EAs proposed in the literature, which usually make use of ranking or tournament selection. The present scheme is an elegant and efficient way of forming niches and, therefore, of maintaining diversity within the EA populations, and also of implementing the elitist strategy, which is so important in Multi-objective Evolutionary Computation. Furthermore, this technique is notably faster than explicit niche formation techniques, such as the use of functions sharing, and it is, therefore, suitable when a fast response is required when huge amounts of data are involved.

We also describe an approach to solve goal and priority based optimization problems, among which are included the problems of constrained optimization, constraint satisfaction and goal programming. The approach consist of transforming the goal and priority based optimization problem into a vector optimization problem, which can be solved using the Pareto-based Multi-objective EA described above, although the approach is independent of the optimizer used, i.e. any other Pareto-based Multi-objective EA could be used for the vector optimization problem. Thus the Pareto concept is used as a mechanism for satisfying constraints in constrained optimization problems and very good results are obtained compared to a wide range of evolutionary techniques for the handling of constraints, proposed by other authors. Other test problems, such as constrained and unconstrained multi-objective optimization problems, have also been tested. The results obtained are highly satisfactory and they bear out the effectiveness of the preselection technique in maintaining diversity.

References

- [1] Coello, C.A. (2000). List of references on multi-objective evolutionary optimization. <http://www.lania.mx/~ccoello/EMOO/EMOObib.html>
- [2] Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.
- [3] Jiménez, F., Verdegay, J.L. (1998). Constrained multi-objective optimization by evolutionary algorithms. *Procs. of the International ICSC Symposium on Engineering of Intelligent Systems (EIS'98)*, pp. 266-271. University of La Laguna, Tenerife, Spain.
- [4] Jiménez, F., Verdegay, J.L., Gómez-Skarmeta, A.F. (1999). Evolutionary techniques for constrained multi-objective optimization problems. *Proceeding of the Genetic and Evolutionary Computation Conference (GECCO-99), Workshop on Multi-Criterion Optimization Using Evolutionary Methods*, Orlando, Florida, USA, pp. 115-116.
- [5] Kita, H., Yabumoto, Y., Mori, N., Nishikawa, Y. (1996). Multi-objective optimization by means of the thermodynamical genetic algorithm. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature –PPSN IV, Lecture Notes in Computer Science*. Berlin, Germany, September, pp. 504-512. Springer-Verlag.
- [6] Koziel, S., Michalewicz, Z. (1999). Evolutionary algorithms, homomorphous mapping, and constrained parameter optimization. *Evolutionary Computation*, vol. 7, no. 1, pp. 19-44.
- [7] Michalewicz, Z., Schoenauer, M. (1996). Evolutionary Algorithms for constrained parameter optimization problems. *Evolutionary Computation*, vol. 4, no. 1, pp. 1-32.
- [8] Srinivas, N., Deb, K. (1995). Multi-objective optimization using nondominated sorting in Genetic Algorithms. *Evolutionary Computation*, vol. 2, no. 3, pp. 221-248.

<i>Problem</i>	f_{opt}	f_{worst}	f_{best}	f_{avg}	<i>best in [6]</i>	<i>best in [7]</i>
<i>G1</i>	-15	-15.0000	-15.0000	-15.0000	-14.7864	-15.000
<i>G2</i>	0.803553	0.792567	0.803556	0.800588	0.79953	0.803553
<i>G3</i>	1.0	1.0000	1.0000	1.0000	0.9997	-
<i>G4</i>	-30665.5	-30665.5	-30665.5	-30665.5	-30664.5	-30005.7
<i>G5</i>	4221.956525	5528.6536	4256.7763	4615.7970	-	5126.6653
<i>G6</i>	-6961.8	-6961.8	-6961.8	-6961.8	-6952.1	-
<i>G7</i>	24.306	25.015	24.360	24.632	24.620	25.486
<i>G8</i>	0.095825	0.095825	0.095825	0.095825	0.095825	0.095825
<i>G9</i>	680.63	680.68	680.64	680.65	680.91	680.642
<i>G10</i>	7049.33	7313.35	7051.86	7131.59	7147.9	7286.650
<i>G11</i>	0.75	0.75	0.75	0.75	0.75	0.75

Table 1: Simulation results for constrained optimization test problems.

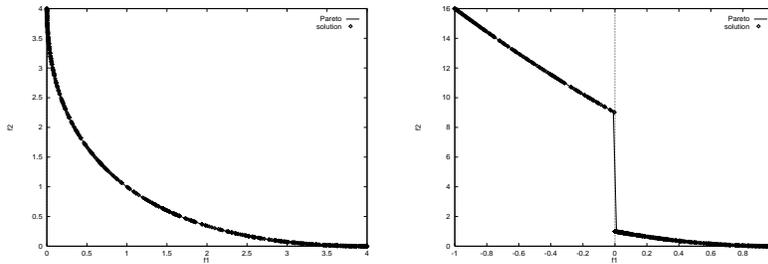


Figure 1: Non-dominated solutions in the domain of the objective functions for the test problem $F1$ (left) and $F2$ (right).

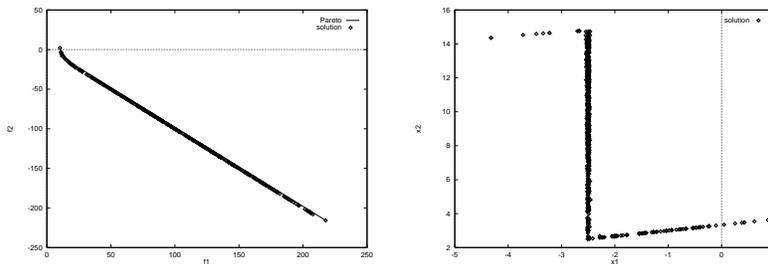


Figure 2: Non-dominated solutions in the domain of the objective functions (left) and search space (right) for the test problem $F3$.

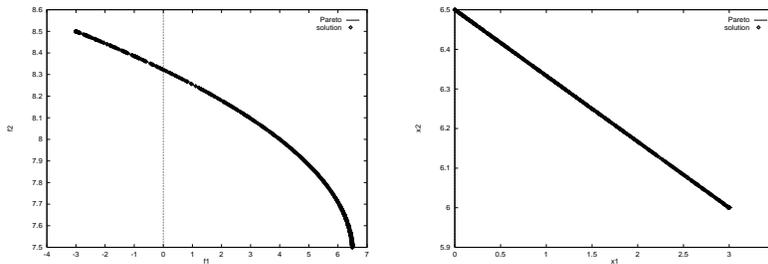


Figure 3: Non-dominated solutions in the domain of the objective functions (left) and search space (right) for the test problem reported in Kita et al. [5].