An Evolutionary Algorithm for Constrained Multi-objective Optimization

Fernando Jiménez **Antonio F. Gómez-Skarmeta** **Gracia Sánchez**
Departamento de Ingeniería de la Información y las Comunicaciones
Facultad de Informática, Campus de Espinardo,
30071 Murcia, Spain
Kalyanmoy Deb
Department of Mechanical Engineering, Indian Institute of Technology
Kanpur PIN 208 016, India

Abstract - The paper follows the line of the design and evaluation of new evolutionary algorithms for constrained multi-objective optimization. The evolutionary algorithm proposed (ENORA) incorporates the Pareto concept of multi-objective optimization with a constraint handling technique and with a powerful diversity mechanism to obtain multiple non dominated solutions through the simple run of the algorithm. Constraint handling is carried out in an evolutionary way and using the min-max formulation, while the diversity technique is based on the partitioning of search space in a set of radial slots along which are positioned the successive populations generated by the algorithm. A set of test problems recently proposed for the evaluation of this kind of algorithm has been used in the evaluation of the algorithm presented. The results obtained with ENORA were very good and considerably better than those obtained with algorithms recently proposed by other authors.

I. INTRODUCTION

The effectiveness of Evolutionary Algorithms (EA) in solving multi-objective optimization problems has been widely recognised in recent years. Proof of this can be seen in the growing number of special sessions and workshops on multi-objective evolutionary optimization incorporated into the framework of prestigious, international congresses and in the recent appearance of the First International Conference on Multi- Criteria Evolutionary Optimisation, held in Zurich in March 2001.

The solution to a multi-objective optimization problem is made up of a set of solutions which we call Pareto solutions. EA are characterised by the application of an in parallel search using a population of potential solutions. A multi-objective EA then incorporates the Pareto concept to identify multiple solutions through a single run of the algorithm. This practically leaves obsolete the classical tendency to aggregate the different objectives using a weight vector or similar approach to obtain a single function which is then optimised - a process requiring several executions of the EA, each with different weights, in order to identify the Pareto solution in each case. We can, therefore, conclude that it is in multi-objective optimization that Evolutionary Computation really distinguishes itself from its most direct competitors, like gradient techniques, simulated annealing or neuronal networks.

In this paper we consider, and without any loss of generality, the following constrained multi-objective optimization problem:

$$\begin{array}{lll} Minimize & f_i(\mathbf{x}), & i = 1, \ldots, n \\ s.t.: & & \\ & g_j(\mathbf{x}) \leq 0, & j = 1, \ldots, m \end{array} \qquad (1)$$

where $\mathbf{x} = (x_1, \ldots, x_p)$ is a vector of real parameters $x_k \in \Re$ belonging to a domain $[l_k, u_k]$, $k = 1, \ldots, p$, and $f_i(\mathbf{x})$, $g_j(\mathbf{x})$ are linear or non linear arbitrary functions.

The set of solutions for problem (1) is made up of all those elements of solution space for which the corresponding objectives vector cannot be simultaneously improved in all its components. These solutions are called *non dominated*, *non inferior* or *Pareto-optimals*. A solution $\mathbf{x}$ is said to be non dominated iff there exists no other $\mathbf{x}'$ such that $f_i(\mathbf{x}') \leq f_i(\mathbf{x})$, for all $i = 1, \ldots, n$, and $f_i(\mathbf{x}') < f_i(\mathbf{x})$, for at least one i.

Since Schaffer implemented the first multi-objective evolutionary algorithm (VEGA) in the mid eighties, there have been numerous authors who have worked along the same lines. Constraint handling with EA has

0-7803-7282-4/02/$10.00 ©2002 IEEE

usually been studied apart from multi-objective optimization with, *penalty methods*, *decoders* and *repair algorithms* being the most commonly used [11]. These are, nevertheless, methods which are heavily dependent on the specific problem of optimization. It is only in the last few years that greater interest has been awakened in the development of evolutionary techniques which enable optimization problems with constraints and multiple objectives to be simultaneously solved [2], [4], [5], [6], [9], [10].

Another highly important aspect of multi-objective evolutionary optimization is that of bestowing a good diversity mechanism on the algorithm. Diversity techniques in multi-objective Evolutionary Computation were originally put forward by Goldberg [7] at the end of the eighties and their importance lies fundamentally in two facts. Firstly, multiple solutions captured in a simple EA execution should cover all the Pareto-optimal fronts which make up the solution. This means that the algorithm has to search for non dominated solutions in a diversified way. Secondly, all the non dominated individuals of the population should have an equal probability of being selected, since they are all equally good. This fact may lead to the genetic drift phenomenon which causes the EA population to converge to just a small region of the solution space. Diversity techniques are, therefore, paramount in multi-objective and multi-modal optimization, and have usually been referred to as *niche formation techniques.* According to Goldberg [7], niche formation techniques can be classified into two categories: *implicit* and *explicit.* With implicit techniques, diversity is achieved through the selfsame generational substitution used by the EA, with the *pre-selection scheme* and the *crowding factor model* being the most usual. In the case of explicit techniques, *sharing function* is typically defined to determine the degree of participation in each individual of the population and this is used to degrade, as a penalty, the fitness of each individual.

For a good EA design, all these aspects (the Pareto concept, constraint handling and the diversity technique) need to be integrated along with the other components which characterise an EA (representation of solutions, initialisation, evaluation function, selection and sampling, variation operators, generational substitution mechanism). *Ranking* [5], [9], [14] and *tournament* [8], [10] are the selection methods most used in the literature, and the importance of the elitist strategy in obtaining an efficient multi-objective EA has also been recognised [15].

Lastly, it is extremely important in the evaluation of the algorithms that a suitable set of test problems is available. While the literature provides test problems within the framework of multi-objective optimization with constraints, these are generally simple, with few decision variables and with non linear constraints which are not suitable enough for use in general optimization. Very recently, Deb et al. [4] proposed a generator of test problems for multi-objective optimization with constraints. This can be configured to obtain the degree of difficulty desired through the setting of six parameters, those, in fact, which have caused serious difficulties in the most recent algorithms.

Taking into account all the above, this paper falls into the field of design and evaluation of new evolutionary algorithms for multi-objective optimization with constraints and is laid out as follows. Section 2 shows the main characteristics of the new EA for solving problems as in (1), and places emphasis on those points already highlighted, i.e. the constraint handling technique used and the diversity mechanism incorporated, in turn closely related to the generational substitution scheme proposed and, therefore, to the elitist strategy employed. Section 3 presents a set of eight test problems which have been used to evaluate the EA and which have been obtained from the paper by Deb et al. [4]. Section 4 gives the results obtained with the proposed EA for the test problems considered. Finally, section 5 provides the most important conclusions and the most promising lines of research for the future.

II. ENORA: AN EVOLUTIONARY ALGORITHM OF NON DOMINATED SORTING WITH RADIAL SLOTS

ENORA (Evolutionary NOn dominated sorting with RAdial slots) presents a classical general structure. A floating point representation is used and the initial population is obtained randomly and uniformly from the domains of the variables. An iterative process follows, in which, in each iteration, a new population is generated through selection, sampling, variation and generational substitution. The iterative process ends when the finishing condition, which consists of achieving a maximum number of generations, is fulfilled.

A. Constraint handling

The populations generated by the algorithm are made up of both feasible and unfeasible individuals. Guided by the multi-objective optimization Pareto concept, the feasible individuals evolve towards optimality, while the non-feasible individuals evolve towards feasibility guided by an evaluation function based on the *min-max* formulation. See below for details. Thus the resulting algorithm is barely dependent on the problem to be optimised since it is the evolutionary heuristics itself that is used

0-7803-7282-4/02/$10.00 ©2002 IEEE

to satisfy the constraints, unlike the repair, decoding or penalty techniques which tend to be heavily dependent on the problem.

B. Variation operators

Bearing in mind that the EA uses a floating point representation and given the coexistence of feasible and unfeasible individuals within the EA populations, the variation operators therefore act on chains (sequences) of real numbers without any consideration regarding the feasibility of new descendants. After experimenting for real parameter optimization with different variation operators proposed in the literature and with others, it was finally decided to use two cross types, *uniform* cross and *arithmetical* cross, and three types of mutation, *uniform* mutation, *non-uniform* mutation and *minimal* mutation. The first four have been studied and described in depth by other authors [11]. Minimal mutation causes a minimal change in the descendant as compared to the father, and it is especially appropriate in fine tuning real parameters. Hence it is the scheme for generating a new population in which the most innovative aspects of ENORA have their roots and which we describe below.

C. Generating a new population

ENORA performs the following steps in the generation of a new population:

1. Two random individuals are selected.
2. Two offspring are obtained by parent crossing and mutation.
3. The offspring are inserted into the population.

Before offering a detailed description of the mechanism for inserting offspring into the new population, we will define the following terms:

- $\mathbf{f}_j$: Objective functions' value vector for the individual j.
  $$\mathbf{f}_j = (f_1^j, \ldots, f_n^j),\ j = 1, \ldots, N$$
  where N is the size of the population.
- F: The set of feasible individuals within the population.
- $\mathbf{f}_{max}$: Objective maximum values vector of the functions for the feasible individuals.
  $$\mathbf{f}_{max} = (f_1^{max}, \ldots, f_n^{max})$$
  where $f_i^{max} = \max_{j \in F}\{f_i^j\}$.
- $\mathbf{f}_{min}$: Objective minimum values vector of the functions for the feasible individuals.
  $$\mathbf{f}_{min} = (f_1^{min}, \ldots, f_n^{min})$$
  where $f_i^{min} = \min_{j \in F}\{f_i^j\}$.
- $\mathbf{h}_j$: Objective normalised values vector of the functions for the individual j.
  $$\mathbf{h}_j = (h_1^j, \ldots, h_n^j)$$
  where $h_i^j = \frac{f_i^{max} - f_i^j}{f_i^{max} - f_i^{min}}$.
- C_{in}: The set of j individuals in the population such that $\forall i = 1, \ldots, n,\ f_i^{min} \leq f_i^j \leq f_i^{max}$.
- C_{out}: The set of j individuals in the population which do not belong to C_{in}.

In this scheme the insertion of the children is the fundamental point for maintaining diversity. The individuals belonging to C_{in} are distributed in D lists $L_{in}^1, \ldots, L_{in}^D$, where $D = d^{n-1}$, with $d = \lfloor N^{\frac{1}{n-1}} \rfloor$. We also have an additional list, L_{out} for individuals belonging to C_{out}. In each list the individuals are ordered according to the *best* function, described below. The insertion is performed as follows:

1. If the individual belongs to C_{out}, it is inserted according to the *best* function into its corresponding position in the L_{out} list and the last individual of L_{out} is eliminated.
2. If the individual belongs to C_{in}, the corresponding slot t is calculated and inserted into the list L_{in}^t. The calculation of the t slot corresponding to an individual j is performed as follows:
   $$t = \sum_{i=1}^{n-1} d^{i-1} \lfloor d \frac{\alpha_i}{\pi/2} \rfloor,$$
   $$\alpha_i = \begin{cases} \frac{\pi}{2} & if\ h_i^j = 0 \\ \arctan(\frac{h_{i+1}^j}{h_i^j}) & if\ h_i^j \neq 0 \end{cases} \quad i = 1, \ldots, n-1$$
   (a) If L_{in}^t is empty, the L_{in}^r list containing a higher number of individuals is sought. If this number is greater than 1, then the last element of L_{in}^r is eliminated and the individual is inserted into L_{in}^t.
   (b) If L_{in}^t is not empty, the individual is inserted into its corresponding position according to the function *best*, and the last element of L_{in}^t is eliminated.

The function *best* is established as follows:

- A feasible individual is better than another, unfeasible one.
- One unfeasible individual is better than another if the function $\max_{j=1,\ldots,m}\{g_j(\mathbf{x})\}$ of the first is better than that of the second.
- One feasible individual is better than another if the first dominates the second.

It should be observed that we are using the *min-max* formulation to satisfy the constraints. This method has been used in multi-objective optimization [1] to minimise the relative deviations of each objective function from its individual optimum, and the best compromise solution can be obtained when objectives of equal priority are optimised. Since constraints and objectives can be treated in a similar way, and it is assumed that all constraints have equal priority, the *min-max* formulation is appropriate for satisfying constraints and is, furthermore, a technique which is independent of the problem.

It should also be noted that insertion of the new individuals is not always carried out, but only in those cases in which the new individual is better than the individual substituted and the diversity is not worsened, or when the new individual is no worse than the one substituted and diversity is improved. Thus the technique simultaneously permits optimization and conservation or improvement of the diversity. It is also an elitist technique, since individuals are never replaced by others which are worse.

III. TEST PROBLEMS

Deb et al. [4] propose a set of new test problems in which the degree and type of difficulty can be controlled by variations to a set of parameters associated to the problem. The test problems proposed have been designed to provoke two particular types of difficulty:

- Difficulty near the Pareto-optimal front.
- Difficulty in all the search space.

In the first test problem, $CTP1$, constraints only arise near the Pareto-optimal front. Part of the Pareto-optimal front is defined by the constraints :

$$CTP1 : \begin{cases} Min & f_1(\mathbf{x}) = x_1 \\ & f_2(\mathbf{x}) = g(\mathbf{x})\exp(-f_1(\mathbf{x})/g(\mathbf{x})) \\ s.t.: & f_2(\mathbf{x}) - a_j \exp(-b_j f_1(\mathbf{x})) \geq 0, \\ & j = 1, \ldots, J \end{cases}$$

where $0 \leq x_1 \leq 1$, $-5 \leq x_2, x_3, x_4 \leq 5$ and $g(\mathbf{x}) = 31 + \sum_{i=1}^{3}(x_i^2 - 10\cos(4\pi x_i))$. We have J constraints. We have considered the case $J = 2$, with values $a_1 = 0.858$, $b_1 = 0.541$, $a_2 = 0.728$ and $b_2 = 0.295$.

Another way of introducing difficulties near to the Pareto front is to include constraints in such a way that the Pareto front is feasible in some slots and unfeasible in others. In the extreme case, the Pareto front is a set of discrete points. Such a function can be described mathematically in the following way:

$$\begin{matrix} CTP2- \\ CTP7: \end{matrix} \begin{cases} Min & f_1(\mathbf{x}) = x_1 \\ & f_2(\mathbf{x}) = g(\mathbf{x})(1 - \sqrt{\frac{f_1}{g(\mathbf{x})}}) \\ s.t.: & \cos(\theta)(f_2(\mathbf{x}) - e) - \sin(\theta) f_1(\mathbf{x}) \geq \\ & a\,|\sin(b\pi(sin(\theta)\ (f_2(\mathbf{x}) - e) + cos(\theta) \\ & f_1(\mathbf{x}))^c)\ |^d \end{cases}$$

where $0 \leq x_1 \leq 1$, $-5 \leq x_2, x_3, x_4 \leq 5$ and $g(\mathbf{x}) = 31 + \sum_{i=1}^{3}(x_i^2 - 10\cos(4\pi x_i))$. The parameters chosen for the different $CTP2$ to $CTP7$ problems were those by Deb at al. [4], and which correspond to the following:

	θ	a	b	c	d	e
$CTP2$	-0.2π	0.2	10	1	6	1
$CTP3$	-0.2π	0.1	10	1	0.5	1
$CTP4$	-0.2π	0.75	10	1	0.5	1
$CTP5$	-0.2π	0.75	10	2	0.5	1
$CTP6$	0.1π	40	0.5	1	2	−2
$CTP7$	-0.05π	40	5	1	6	0

Finally, we have also taken into account the OSY problem [4], [12] which is a typical test problem and considered difficult to solve. It is, moreover, highly appropriate for testing the effectiveness of diversity techniques.

$$OSY : \begin{cases} Min & f_1(\mathbf{x}) = -(25(x_1-2)^2 + (x_2-2)^2 \\ & +(x_3-1)^2 + (x_4-4)^2 + (x_5-1)^2) \\ & f_2(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 \\ s.t.: & x_1 + x_2 - 2 \geq 0 \\ & 6 - x_1 - x_2 \geq 0 \\ & 2 - x_2 + x_1 \geq 0 \\ & 2 - x_1 + 3x_2 \geq 0 \\ & 4 - (x_3-3)^2 - x_4 \geq 0 \\ & (x_5-3)^2 + x_6 - 4 \geq 0 \end{cases}$$

with $0 \leq x_1, x_2, x_6 \leq 10$, $1 \leq x_3, x_5 \leq 5$, and $0 \leq x_4 \leq 6$.

IV. RESULTS

The ENORA algorithm has been executed for $CTP1$-$CTP7$ and OSY problems and the following parameters have been used : probability of cross 0.6 (50% uniform and 50% arithmetic), probability of mutation 0.6 (20% uniform, 30% non-uniform and 50% minimum) and population size $N = 100$. The results obtained are shown in Figure 1.

The NSGA-II algorithm proposed by Deb et al. [4], converges suitably in the $CTP2$, $CTP3$, $CTP5$ and $CTP6$ problems. However, for the $CTP4$ and $CTP7$ problems, NSGA-II presents difficulties both in the location of Pareto points and in the diversity. A modification of NGSA-II with controlled elitism, proposed by Deb and Goel [3], improves the results for $CTP7$ in the case of diversity, although the solutions obtained are remain at some distance from the Pareto-optimal front. While the solutions obtained for the OSY problem are on the Pareto-optimal front, they are not perfectly distributed. Deb et al. [4] did not show any results for $CTP1$.

From the results shown in [4], the algorithm proposed by Ray et al. [13] has difficulties with all the problems, both in convergence and in diversity.

Figure 1 shows the good behaviour of ENORA in all the test problems with uniformly distributed solutions being obtained on all the Pareto-optimal fronts.

V. CONCLUSIONS AND FUTURE RESEARCH

The paper presented is the result of some years of research in the field of the design and evaluation new evolutionary algorithms for multi-objective optimization with constraints. The approaches of evolutionary constrained multi-objective optimization based on ranking and tournament selection which opened the research gave satisfactory results in small test problems. The results, however, were not so satisfactory in more complicated test problems, like those treated herein. In perfecting these algorithms, the research has led to a new algorithm, the success of which has been fundamentally due to the diversity technique used. The algorithm proposed incorporates a problem independent constraints satisfaction technique together with a powerful diversity mechanism which permits the identification of multiple non dominated solutions distributed uniformly along the Pareto-optimal fronts. The experimental results in test problems designed specifically to evaluate such algorithms show a considerable improvement on those obtained to date using other recently created algorithms. As regards the main line of research in the future, work is already being carried out on the applications of ENORA in the field of the fuzzy modelling of hydrological resources. The multiobjective algorithm is used to optimise the weights and structure of a fuzzy neural network RBF (Radial Based Function) in which we consider as objectives the minimisation of the mean squared error and of the number of neurones in the neural network, and as a constraint that the mean squared error of the learning and the evaluation data do not exceed a given threshold.

References

[1] Chankong, V., Haimes, Y.Y. (1983). *Multiobjective Decision Making: Theory and Methodology. North-Holland series in Systems Science and Engineering, Andrew P. Sage (Ed.).*

[2] Coello, C.A. (2000). List of references on multi-objective evolutionary optimization. http://www.lania.mx/~ccoello/EMOO/EMOObib.html

[3] Deb, K, Goel, T. (2001). Controlled elitist non-dominated sorting genetic algorithm for better convergence. *Lectures Notes in Computer Science*, vol. 1993, pp. 67-81.

[4] Deb, K., Pratap, A., Meyarivan, T. (2001). Constrained test problems for multi-objective evolutionary optimization. *Lectures Notes in Computer Science*, vol. 1993, pp. 284-298.

[5] Fonseca, C.M., Fleming, P.J. (1998). Multi-objective optimization and multiple constraint handling with evolutionary algorihtms - Part I: A unified formulation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 28, no. 1, pp. 26-37.

[6] Fonseca, C.M., Fleming, P.J. (1998). Multi-objective optimization and multiple constraint handling with evolutionary algorihtms - Part II:Application example. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 28, no. 1, pp. 38-47.

[7] Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley.

[8] Horn, J., Nafpliotis, N. (1993). Multiobjective optimization using the niched Pareto Genetic Algorithm. IlliEAL Report no. 93005.

[9] 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.

[10] 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.

[11] Michalewicz, Z., Schoenauer, M. (1996). Evolutionary Algorithms for constrained parameter optimization problems. *Evolutionary Computation*, vol. 4, no. 1, pp. 1-32.

[12] Osyczka, A., Kundu, S. (1995). A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. *Structural Optimization*, vol 10, pp. 94-99.

[13] Ray, T., Kang, T., Chye, S. (in press). Multiobjective desing optimization by an evolutionary algorithm. *Engineering Optimization.*

[14] Srinivas, N., Deb, K. (1995). Multi-objective optimization using nondominated sorting in Genetic Algorithms. *Evolutionary Computation*, vol. 2, no. 3, pp. 221-248.

[15] Zitzler, E., Deb, K., Thiele, L. (1999). Comparison of multiobjective evolutionary algorithms on test function of different difficulty. *Proceeding of the Genetic and Evolutionary Computation Conference (GECCO-99), Workshop on Multi-Criterion Optimization Using Evolutionary Methods*, Orlando, Florida, USA, pp. 121-122.

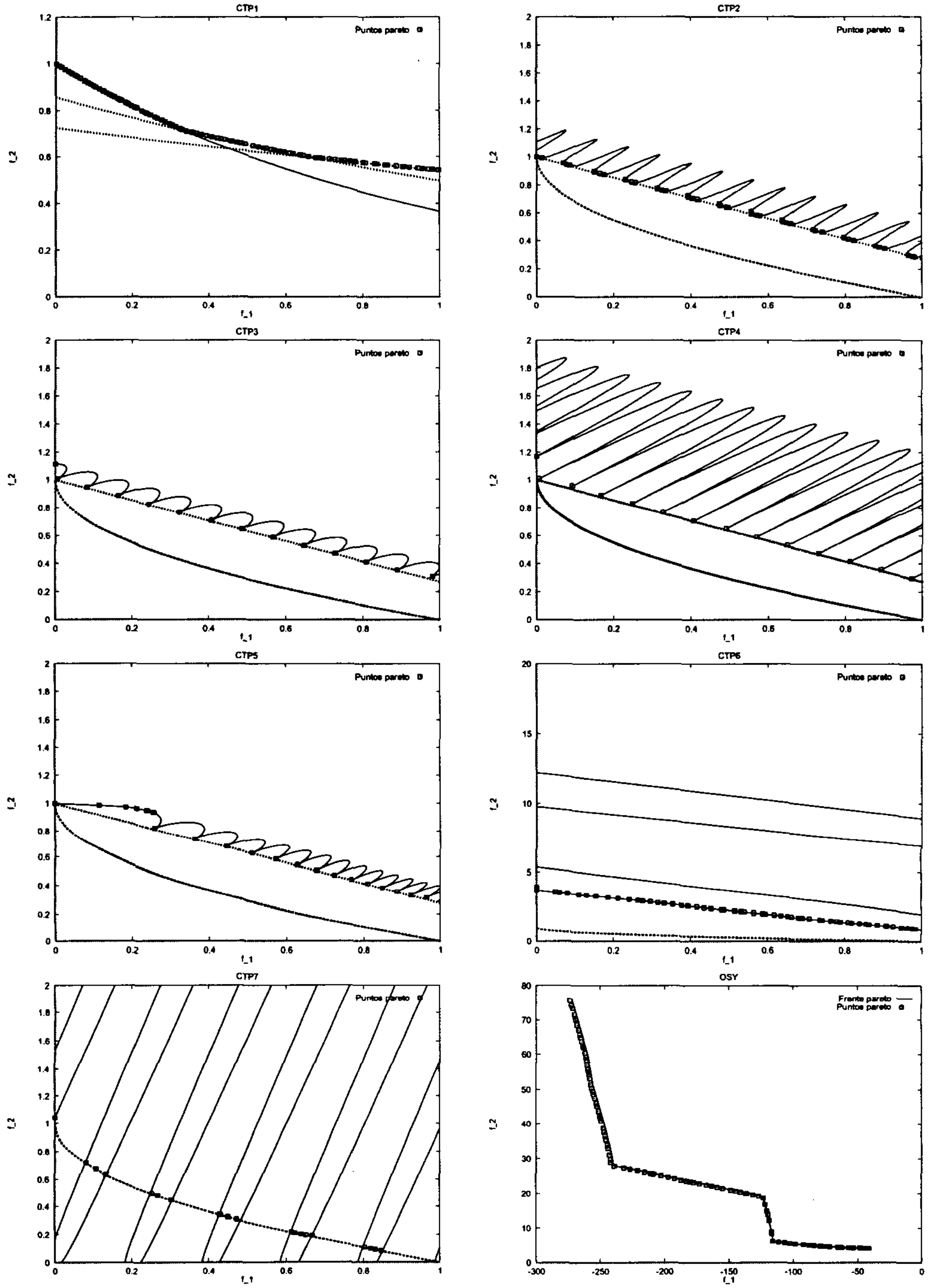

Fig. 1. Non dominated solutions obtained with ENORA for the test problems $CTP1 - CTP7$ and OSY.