A Coevolutionary Multi-Objective Evolutionary Algorithm

Carlos A. Coello Coello and Margarita Reyes Sierra
CINVESTAV-IPN (Evolutionary Computation Group)
Departamento de Ingeniería Eléctrica, Sección de Computación
Av. IPN No. 2508. Col. San Pedro Zacatenco México D.F. 07300, MÉXICO
ccoello@cs.cinvestav.mx, mreyes@computacion.cs.cinvestav.mx

Abstract- In this paper, we propose a multi-objective evolutionary algorithm that incorporates some coevolutionary concepts. The primary design goal of the proposed approach is to reduce the total number of objective function evaluations required to produce a reasonably good approximation of the true Pareto front of a problem. The main idea of the proposed approach is to concentrate the search effort on promising regions that arise during the evolutionary process as a byproduct of a mechanism that subdivides decision variable space based on an estimate of the relative importance of each decision variable. The proposed approach is validated using several test functions taken from the specialized literature and it is compared with respect to three approaches that are representative of the state-of-the-art in evolutionary multiobjective optimization.

1 Introduction

Despite the considerable volume of research on evolutionary multiobjective optimization [4], little emphasis has been placed on certain algorithmic design aspects such as efficiency [6, 10, 3]. Additionally, the use of coevolutionary mechanisms (which have strong links to game theory [2]) has been scarce in the evolutionary multiobjective optimization literature. The main motivation of the work reported here was precisely to take advantage of some coevolutionary concepts to design a multi-objective evolutionary algorithm (MOEA) that can be more efficient (in terms of fitness function evaluations). The main idea of the proposed algorithm is to obtain information along the evolutionary process as to subdivide the search space into n subregions, and then to use a subpopulation for each of these subregions. At each generation, these different subpopulations (which evolve independently using Fonseca and Fleming's ranking scheme [8]) "cooperate" and "compete" among themselves and from these different processes we obtain a single Pareto front. Each individual contained in the Pareto optimal set has a label that indicates the subpopulation to which it belongs. These labels are used to determine which subpopulations contributed with more solutions. The size of each subpopulation is adjusted based on their contribution to the current Pareto front (i.e., subpopulations which contributed more are allowed a larger population size and viceversa). Thus, those populations contributing with more nondominated individuals have a higher reproduction probability. The proposed approach uses the adaptive grid proposed in [10] to store the nondominated vectors obtained along the evolutionary process, enforcing a more uniform distribution of such vectors along the Pareto front.

2 Statement of the Problem

We are interested in solving problems of the type:

$$\text{minimize } [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})] \quad (1)$$

subject to:

$$g_i(\vec{x}) \geq 0 \quad i = 1, 2, \ldots, m \quad (2)$$

$$h_i(\vec{x}) = 0 \quad i = 1, 2, \ldots, p \quad (3)$$

where k is the number of objective functions $f_i : R^n \to R$. We call $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ the vector of decision variables. We thus wish to determine from the set $\mathcal{F}$ of all the vectors that satisfy (2) and (3) to the vector $x_1^*, x_2^*, \ldots, x_n^*$ that are *Pareto optimal*. We say that a vector of decision variables $\vec{x}^* \in \mathcal{F}$ is *Pareto optimum* if there does not exist another $\vec{x} \in \mathcal{F}$ such that $f_i(\vec{x}) \leq f_i(\vec{x}^*)$ for every $i = 1, \ldots, k$ and $f_j(\vec{x}) < f_j(\vec{x}^*)$ for at least one j. The vectors $\vec{x}^*$ corresponding to the solutions included in the Pareto optimal set are called *nondominated*. The objective function values corresponding to the elements of the Pareto optimal set are called the *Pareto front* of the problem.

3 Coevolution

We call coevolution to a change in the genetic composition of a species (or group of species) as a response to a genetic change of another one. In a more general sense, coevolution refers to a reciprocal evolutionary change between species that interact with each other. The term "coevolution" is usually attributed to Ehrlich and Raven who published a paper on their studies performed with butterflies and plants in the mid-1960s [7]. However, several other researchers used the term before them. In fact, the original idea of coevolution was apparently introduced by Darwin [5]. The relationships between the populations of two different species A and B can be described considering all their possible types of interactions. Such interaction can be positive or negative depending on the consequences that such interaction produces on the population. Evolutionary computation researchers have developed several coevolutionary approaches in which normally two or more species relate to each other using one of the previously indicated schemes. Also, in most cases, such species evolve independently through a genetic algorithm. The key issue in these coevolutionary algorithms is that the fitness of an individual in a population depends on the individuals of a different population. There are two main classes of coevolutionary algorithms in the evolutionary computation literature:

- Those based on *competition* relationships: In this case, the fitness of an individual is the result of a series of "encounters" with other individuals (e.g., [11, 15]).
- Those based on *cooperation* relationships: In this case, the fitness of an individual is the result of a collaboration with individuals of other species (or populations) (e.g., [14, 13]).

4 Related Work

There are very few references in the literature in which coevolutionary concepts are used to solve multiobjective optimization problems. We will review the main ones in this section. Parmee & Watson [12] proposed a collaborative scheme in which they use one population to optimize each of the objective functions of a problem. The method is really created to converge to a single (ideal) trade-off solution. However, through the use of penalties the algorithm can maintain diversity in the population. These penalties relate to variability in the decision variables' values. The authors also store solutions produced during the evolutionary process so that the user can analyze the historical paths traversed by the algorithm. Barbosa and Barreto [1] proposed a cooperative approach for solving a graph layout generation problem. The approach uses two populations (a separate genetic algorithm is used for each of them and information is exchanged through a shared fitness function): a graph layout population (i.e., individuals that contain the coordinates of all vertices in the graph) and a population of weights (i.e., individuals that contain, each one, a set of weights to be applied on the different aesthetic objectives imposed on the problem). Each of the solutions produced by the system are presented to a user who ranks them based on (subjective) preferences. This ranking is used to determine fitness of the population of weights. Keerativuttitumrong et.al [9] proposed a cooperative scheme in which one population is defined for each decision variable of the problem. The evolution of each of these populations is controlled through Fonseca and Fleming's MOGA [8]. In order to evaluate an individual in any population, individuals from the other populations must be selected in order to complete a solution (this is because each population only encodes one decision variable).

5 Description of Our Approach

The main idea of our approach is to try to focus the search efforts only towards the promising regions of the search space. In order to determine what regions of the search space are promising, our algorithm performs a relatively simple analysis of the current Pareto front. The evolutionary process of our approach is divided in 4 stages. The change of stage is controlled by a certain number of generations during which we run the algorithm. Our current version equally divides the full evolutionary run into four parts (i.e., the total number of generations is divided by four), and each stage is allocated one of these four parts.

5.1 First Stage

During the first stage (first 25% of the total number of generations), the algorithm is allowed to explore all of the search space, by using a population of individuals which are selected using Fonseca and Fleming's Pareto ranking scheme [8]. Additionally, the approach uses the adaptive grid proposed by [10]. At the end of this first stage, the algorithm analyses the current Pareto front (stored in the adaptive grid) in order to determine what variables of the problem are more critical. This analysis consists of looking at the current values of the decision variables corresponding to the current Pareto front (line 6, Figure 3). This analysis is performed independently for each decision variable. The idea is to determine if the values corresponding to a certain variable are distributed along all the allowable interval or if such values are concentrated on a narrower range. When the whole interval is being used, the algorithm concludes that keeping the entire interval for that variable is important. However, if only a narrow portion is being used, then the algorithm will try to identify portions of the interval that can be discarded from the search process. As a result of this analysis, the algorithm determines whether is convenient or not to subdivide (and, in such case, it also determines how many subdivisions to perform) the interval of a certain decision variable. Each of these different regions will be assigned a different population (line 7, Figure 3). We will illustrate this process with an example. Let's suppose that our problem has two variables and that, after the analysis, the algorithm determines that it is not convenient to subdivide the interval of the first variable. Additionally, the algorithm determines that the interval of the second variable must have two subdivisions. What the algorithm does is to divide the interval of the second decision variable into three parts of equal size (i.e., add two subdivisions to the interval). The process to decide how many populations to have and to which region of the search space to assign each of them is illustrated in Figure 1.

5.2 Second Stage

When reaching the second stage, the algorithm consists of a certain number of populations looking each at different regions of the search space. At each generation, the evolution of all the populations takes place independently and, later on, the nondominated elements from each population are sent to the adaptive grid where they "cooperate" and "compete" in order to conform a single Pareto front (line 10, Figure 3). Note that in this case we are referring to global nondominance (i.e., with respect to all the search space) and not to local nondom-

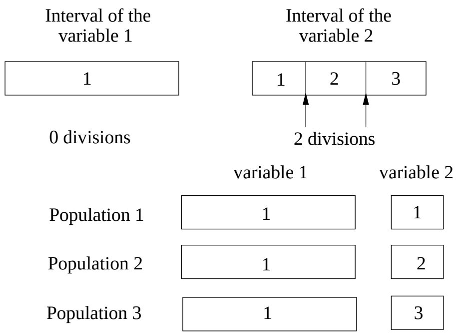

Figure 1: Mechanism used to assign regions of the search space to each population.

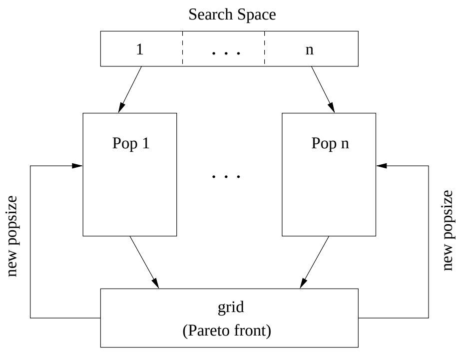

Figure 2: Graphical representation of the second stage of our algorithm.

inance. After this, we count the number of individuals that each of the populations contributed to the current Pareto front. Our algorithm is *elitist* (line 11, Figure 3), because after the first generation of the second stage, all the populations that do not provide any individual to the current Pareto front are automatically eliminated and the sizes of the other populations are properly adjusted (i.e., those populations that contributed more to the current Pareto front get their sizes proportionally increased and those who contribute less get their sizes decreased). Each population is assigned or removed individuals such that its final size is proportional to its contribution to the current Pareto front. These individuals to be added or removed are randomly generated/chosen. Thus, populations compete with each other to get as many extra individuals as possible. Note that it is, however, possible that the sizes of the populations "converge" to a constant value once their contribution to the current Pareto front does not change any longer (i.e., from one generation to the next one). Figure 2 illustrates the second stage of our algorithm.

5.3 Third Stage

During the third stage, we perform a check on the current populations in order to determine how many (and which) of them can continue (i.e., those populations which continue contributing individuals to the current Pareto front) (line 5, Figure 3). Over these (presumably good) populations, we will apply the same process from the second stage (i.e., they will be further subdivided and more populations will be created in order to exploit these "promising regions" of the search space). In order to determine the number of subdivisions that are to be used during the third stage, we repeat the same analysis as before (i.e., the analysis performed during the first stage). The individuals from the "good" populations are kept. All the good individuals are distributed across the newly generated populations. A new count is undertaken so that the algorithm can determine how many individuals are contributed by each of the new populations to the current Pareto front. Again, populations that do not contribute to the current Pareto front are eliminated. Note however, that we define a minimum population size and this size is enforced for all populations at the beginning of the third stage. After the first generation of the third stage, the size will be adjusted based on the same criteria as before (i.e., the size of populations will be modified based on their contribution to the current Pareto front).

5.4 Fourth Stage

During this stage, we apply the same procedure of the third stage in order to allow a fine-grained search.

5.5 Decision Variables Analysis

The mechanism adopted for the decision variables analysis is very simple. Given a set of values within an interval, we compute both the minimum average distance of each element with respect to its closest neighbor and the total portion of the interval that is covered by the individuals contained in the current Pareto front. Then, only if the set of values covers less than 80% of the total of the interval, the algorithm considers appropriate to divide it. Once the algorithm decides to divide the interval, the number of divisions gets increased (without exceeding a total of 10 divisions per interval), as explained next. Let's define *range* as the percentage of the total of *interval* that is occupied by the values of the variable under consideration. Let $\overline{d}_{min}$ be the minimum average distance between individuals and let *divisions* be the number of divisions to perform in the interval of the variable:

if $(range < 0.8 * interval)$
  while $(\overline{d}_{min} < 0.2 * interval)$
  $\{\ divisions++;\ interval = 0.2 * interval;\ \}$

5.6 Parameters Required

Our proposed approach requires the following parameters:

1. Crossover rate (p_c) and mutation rate (p_m).
2. Maximum number of generations ($Gmax$).
3. Size of the initial population ($popsize_{init}$) to be used during the first stage and minimum size of the sec-

```
gen = 0
populations = 1
while (gen < Gmax) {
    if(gen = Gmax/4 or Gmax/2 or 3*Gmax/4)
    {
            check_active_populations()
            decision_variables_analysis()
            (compute number of subdivisions)
            construct_new_subpopulations()
            (update populations)
    }
    for (i = 1; i <= populations; i++)
            if (population i contributes
            to the current Pareto front)
                    evolve_and_compete(i)
    elitism()
    reassign_resources()
    gen++ }
```

Figure 3: Pseudocode of our algorithm.

ondary population ($popsize_{sec}$) to be used during the further stages.

6 Comparison of Results

To validate our approach, we used the methodology normally adopted in the evolutionary multiobjective optimization literature [4]. We performed both quantitative comparisons (adopting four metrics) and qualitative comparisons (plotting the Pareto fronts produced) with respect to three MOEAs that are representative of the state-of-the-art in the are: the microGA for multiobjective optimization [3], the Pareto Archived Evolution Strategy (PAES) [10] and the Nondominated Sorting Genetic Algorithm II (NSGA-II) [6]. For our comparative study, we implemented for four following metrics:

1. **Two Set Coverage (SC)**: This metric was proposed in [19], and it can be termed *relative coverage comparison of two sets*. Consider X', X'' as two sets of phenotype decision vectors. SC is defined as the mapping of the order pair (X', X'') to the interval $[0,1]$:

$$SC(X', X'') \triangleq \frac{|\{a'' \varepsilon X''; \exists a' \varepsilon X' : a' \succeq a''\}|}{|X''|} \quad (4)$$

   If all points in X' dominate or are equal to all points in X'', then by definition $SC = 1$. $SC = 0$ implies the opposite. In general, $SC(X', X'')$ and $SC(X'', X')$ both have to be considered due to set intersections not being empty. Of course, this metric can be used for both spaces (objective function or decision variable space), but in this case we applied it in objective function space.

2. **Spacing (SP)**: This metric was proposed by Schott [16] as a way of measuring the range (distance) variance of neighboring vectors in the Pareto front known. This metric is defined as:

$$SP = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{d} - d_i)^2} \quad (5)$$

   where $d_i = min_j(\sum_{k=1}^{m} |f_m^i - f_m^j|)$, $i, j = 1, \ldots, n$, m is the number of objectives, $\bar{d}$ is the mean of all d_i, and n is the number of vectors in the Pareto front found by the algorithm being evaluated. A value of zero for this metric indicates all the nondominated solutions found are equidistantly spaced.

3. **Generational Distance (GD)**: The concept of generational distance was introduced by Van Veldhuizen & Lamont [18] as a way of estimating how far are the elements in the Pareto front produced by our algorithm from those in the true Pareto front of the problem. This metric is defined as:

$$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \quad (6)$$

   where n is the number of nondominated vectors found by the algorithm being analyzed and d_i is the Euclidean distance (measured in objective space) between each of these and the nearest member of the true Pareto front. It should be clear that a value of $GD = 0$ indicates that all the elements generated are in the true Pareto front of the problem. Therefore, any other value will indicate how "far" we are from the global Pareto front of our problem.

4. **Error Ratio (ER)**: This metric was proposed by Van Veldhuizen [17] to indicate the percentage of solutions (from the nondominated vectors found so far) that are not members of the true Pareto optimal set:

$$ER = \frac{\sum_{i=1}^{n} e_i}{n}, \quad (7)$$

   where n is the number of vectors in the current set of nondominated vectors available; $e_i = 0$ if vector i is a member of the Pareto optimal set, and $e_i = 1$ otherwise. It should then be clear that $ER = 0$ indicates an ideal behavior, since it would mean that all the vectors generated by our algorithm belong to the true Pareto optimal set of the problem.

For each of the test functions shown below, we perform 30 runs per algorithm. The Pareto fronts that we will show correspond to the median of the 30 runs with respect to the ER metric.

Test Function 1 (6300 evaluations)					
		CO-MOEA	microGA	PAES	NSGA-II
ER	best	0.46	0.42	0.02	0.0
	median	0.61	0.77	0.07	0.02
	worst	0.68	0.98	0.15	0.08
	average	0.60	0.75	0.07	0.03
	std. dev.	0.0610	0.1453	0.0297	0.0211
GD	best	0.0003	0.0008	0.0001	0.0007
	median	0.001	0.0089	0.0006	0.0008
	worst	0.042	0.238	0.0659	0.0009
	average	0.0049	0.0681	0.0066	0.0008
	std. dev.	0.0085	0.0860	0.0163	0.0000
SP	best	0.006	0.017	0.007	0.006
	median	0.012	0.042	0.014	0.008
	worst	0.379	1.539	0.624	0.086
	average	0.039	0.356	0.054	0.010
	std. dev.	0.0727	0.5070	0.1411	0.0143

Test Function 1 - Two Set Coverage Metric *SC*		
X	*SC*(*X*,CO-MOEA)	*SC*(*X*,microGA)
CO-MOEA	0.0	0.54
microGA	0.14	0.0
PAES	0.37	0.63
NSGA-II	0.55	0.57
X	*SC*(*X*,PAES)	*SC*(*X*,NSGA-II)
CO-MOEA	0.01	0.01
microGA	0.0	0.0
PAES	0.0	0.0
NSGA-II	0.07	0.0

Table 1: Comparison of results between our approach (denoted by CO-MOEA), the microGA [3], PAES [10] and the NSGA-II [6] for test function 1.

6.1 Test Function 1

$$\begin{aligned} \text{Minimize } f_1(x_1,x_2) &= x_1 \\ \text{Minimize } f_2(x_1,x_2) &= (1.0+10.0x_2) \\ (1.0-\frac{x_1^2}{1.0+10.0x_2} &- \frac{x_1}{1.0+10.0x_2}\sin(2\pi 4x_1)) \\ 0.0 \leq x_1,x_2 \leq 1.0 & \end{aligned} \tag{8}$$

In this example, our approach used: $popsize_{init} = 100$, $popsize_{rec} = 30$. Table 1 shows the values of the metrics for each of the MOEAs compared.

6.2 Test Function 2

$$\begin{aligned} \text{Minimize } f_1(x_1,x_2) &= \frac{(x_1-2)^2}{2}+\frac{(x_2+1)^2}{13}+3 \\ \text{Minimize } f_2(x_1,x_2) &= \frac{(x_1+x_2-3)^2}{175}+ \\ +\frac{(2x_2-x_1)^2}{17}-13 & \\ \text{Minimize } f_3(x_1,x_2) &= \frac{(3x_1-2x_2+4)^2}{8}+ \\ \frac{(x_1-x_2+1)^2}{27}+15; & \quad -4.0 \leq x_1,x_2 \leq 4.0 \end{aligned}$$

In this case, our approach used: $popsize_{init} = 20$, $popsize_{rec} = 20$. Table 2 shows the values of the metrics for each of the MOEAs compared.

Test Function 2 (1700 evaluations)					
		CO-MOEA	microGA	PAES	NSGA-II
ER	best	0.02	0.04	0.0	0.03
	median	0.08	0.1	0.03	0.06
	worst	0.12	0.16	0.22	0.12
	average	0.07	0.10	0.05	0.06
	std. dev.	0.0253	0.033	0.0566	0.0221
GD	best	0.099	0.0706	0.0134	0.1992
	median	0.147	0.1353	0.0802	0.2503
	worst	0.246	0.2175	0.2952	0.2982
	average	0.150	0.1412	0.1009	0.2501
	std. dev.	0.0306	0.0352	0.0708	0.0291
SP	best	0.163	0.225	0.085	0.159
	median	0.224	0.3	0.240	0.201
	worst	1.3	0.767	1.156	0.313
	average	0.247	0.367	0.323	0.208
	std. dev.	0.2038	0.1576	0.2265	0.0373

Test Function 2 - Two Set Coverage Metric *SC*		
X	*SC*(*X*,CO-MOEA)	*SC*(*X*,microGA)
CO-MOEA	0.0	0.17
microGA	0.02	0.0
PAES	0.01	0.09
NSGA-II	0.09	0.10
X	*SC*(*X*,PAES)	*SC*(*X*,NSGA-II)
CO-MOEA	0.11	0.09
microGA	0.03	0.05
PAES	0.0	0.06
NSGA-II	0.03	0.0

Table 2: Comparison of results between our approach (denoted by CO-MOEA), the microGA [3], PAES [10] and the NSGA-II [6] for test function 2.

6.3 Test Function 3

$$\text{Minimize } f_1(x_1,x_2) = x_1$$
$$\text{Minimize } f_2(x_1,x_2) = \frac{g(x_2)}{x_1}$$

where:

$$g(x_2) = 2.0 - e^{-(\frac{x_2-0.2}{0.004})^2} - 0.8e^{-(\frac{x_2-0.6}{0.4})^2}$$
$$0.1 \leq x_1,x_2 \leq 1.0$$

In this example, our approach used: $popsize_{init} = 100$, $popsize_{rec} = 30$. Table 3 shows the values of the metrics for each of the MOEAs compared.

7 Discussion of Results

In test function 1, we can see that the NSGA-II had the best overall performance (both with respect to all the metrics and with respect to the graphical results shown in Figure 4). It is also clear that the microGA presented the worst performance for this test function. Based on the values of the *ER* and *SC* metrics, we can conclude that our approach had problems to reach the true Pareto front of this problem. Note however, that the values of *GD* and *SP* indicate that our approach converged very closely to the true Pareto front and that it achieved a good distribution of solutions. PAES had a good performance regarding closeness to the true Pareto front, but its performance

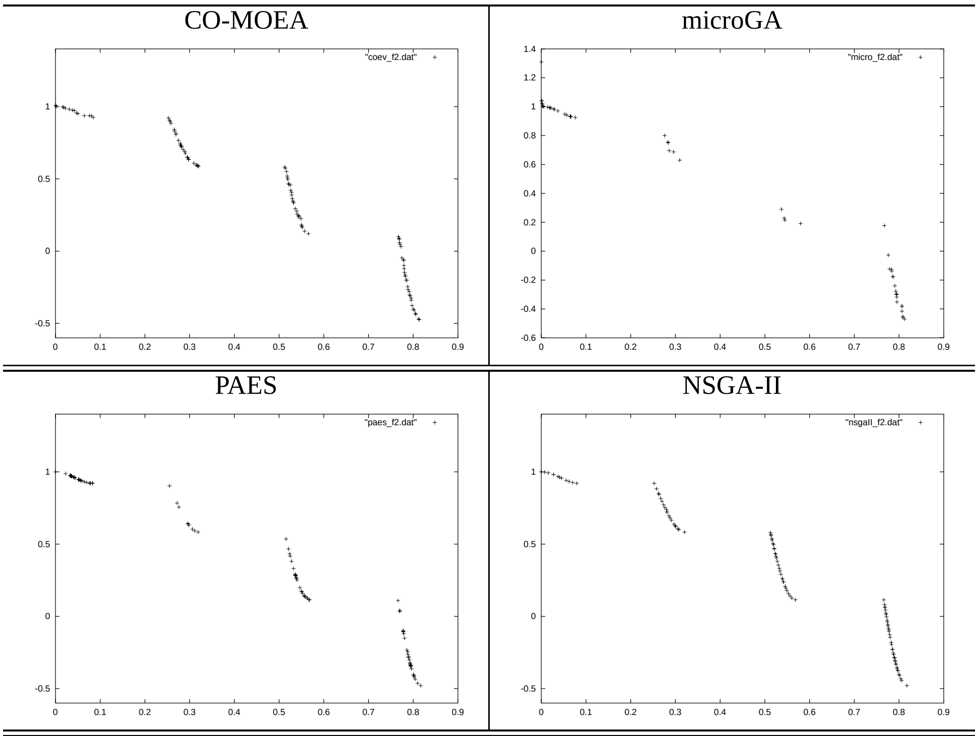

Figure 4: Pareto fronts obtained by our approach (CO-MOEA), the microGA [3], PAES [10] and the NSGA-II [6] for test function 1.

was not so good regarding uniform distribution of solutions (this is corroborated graphically as well). In the second test function, all the algorithms compared converged to the true Pareto front, but their distribution of results was not as uniform as desirable (see the values of the *SP* metric from Table 2 and Figure 5). Nevertheless, our approach and the microGA produced the best Pareto fronts (both in terms of closeness to the true Pareto front and in terms of distribution of solutions) for this problem (see Figure 5).

Based on the values of the *ER* and *SC* metrics, the microGA and the NSGA-II had the best performance in test function 3. However, when we analyze the graphical results obtained for this problem (see Figure 6), the Pareto front obtained by our approach looks quite similar to the fronts obtained by both the microGA and the NSGA-II. Also note the poor distribution of solutions obtained by PAES. An interesting aspect of this test function is that it has a local attractor. In the experiments performed, all the approaches converged at least once to this false attractor (this explains that the worst value of the *ER* metric is 1.0 for all of them). This had an obvious impact on the standard deviation of the *ER* metric for all the algorithms compared. It is also worth noticing that the microGA had the lowest standard deviation for the *ER* metric because it only converged once to the false attractor of this test function.

8 Conclusions and Future Work

We have proposed a coevolutionary multi-objective evolutionary algorithm whose main idea is to divide the search space into different subregions, as to detect the most “promising” of such regions, focusing the search on them. The proposed algorithm performs a relatively simple analysis to detect what decision variables are the most important and, based on such analysis, it divides the search space. The proposed approach was validated using several test functions taken from the specialized literature. Our comparative study showed that the proposed approach is competitive with respect three other algorithms that are representative of the state-of-the-art in evolutionary multiobjective optimization. Currently, the main drawback of our proposed approach is the number of populations that it could potentially need to handle. Once the first phase has finished, the number of populations that it could need to handle is given by: $div_1 \times div_2 \times \cdots \times div_{\#var}$. Thus, as part of our future work, we are considering a redesign of the algorithm in which such multiple populations are no longer needed. It is also important to decrease the high selection pressure introduced by our elitist scheme, since in our current version of the algorithm this may cause premature convergence when a false attractor exists. Additionally, we are considering the use of a clustering algorithm to determine the most critical decision variables of the problem.

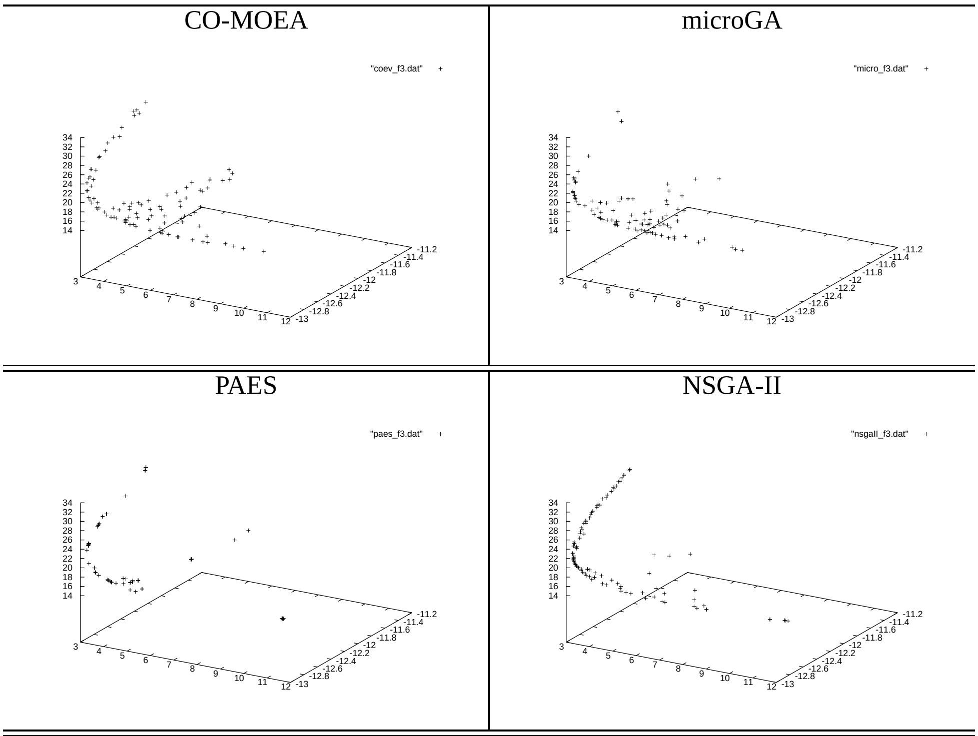

Figure 5: Pareto fronts obtained by our approach (CO-MOEA), the microGA [3], PAES [10] and the NSGA-II [6] for test function 2.

Acknowledgments

The first author acknowledges support from CONACyT through project number 34201-A. The second author acknowledges support from CONACyT through a scholarship to pursue graduate studies at CINVESTAV-IPN's Electrical Engineering Department.

Bibliography

[1] H. J. Barbosa and A. M. Barreto. An interactive genetic algorithm with co-evolution of weights for multiobjective problems. In L. Spector, E. D. Goodman, A. Wu, W. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 203–210, San Francisco, California, 2001. Morgan Kaufmann Publishers.

[2] H. J. C. Barbosa. A coevolutionary genetic algorithm for a game approach to structural optimization. In T. Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 545–552, San Mateo, California, July 1997. Michigan State University, Morgan Kaufmann Publishers.

[3] C. A. Coello Coello and G. Toscano Pulido. Multiobjective Optimization using a Micro-Genetic Algorithm. In L. S. et al., editor, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 274–282, San Francisco, California, 2001. Morgan Kaufmann Publishers.

[4] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, May 2002. ISBN 0-3064-6762-3.

[5] C. Darwin. *The Origin of Species by Means of Natural Selection or the Preservation of Favored Races in the Struggle for Life*. The Book League of America, 1929. (Originally published in 1859).

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.

[7] P. Ehrlich and P. Raven. Butterflies and Plants: A Study in Coevolution. *Evolution*, 18:586–608, 1964.

[8] C. M. Fonseca and P. J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California, 1993. Morgan Kauffman Publishers.

[9] N. Keerativuttitumrong, N. Chaiyaratana, and V. Varavithya. Multi-objective co-operative co-evolutionary genetic algorithm. In J. M. G. et al., editor, *Proceedings of the Parallel Problem Solving from Nature VII Conference (PPSN VII)*, pages 288–297, Granada, Spain, September 2002. Springer-Verlag.

[10] J. D. Knowles and D. W. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000.

[11] J. Paredis. Coevolutionary algorithms. In T. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *The Handbook of Evolutionary Computation, 1st supplement*, pages 225–238. Institute of Physics Publishing and Oxford University Press, 1998.

[12] I. C. Parmee and A. H. Watson. Preliminary Airframe Design Using Co-Evolutionary Multiobjective Genetic Algorithms. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Gar-

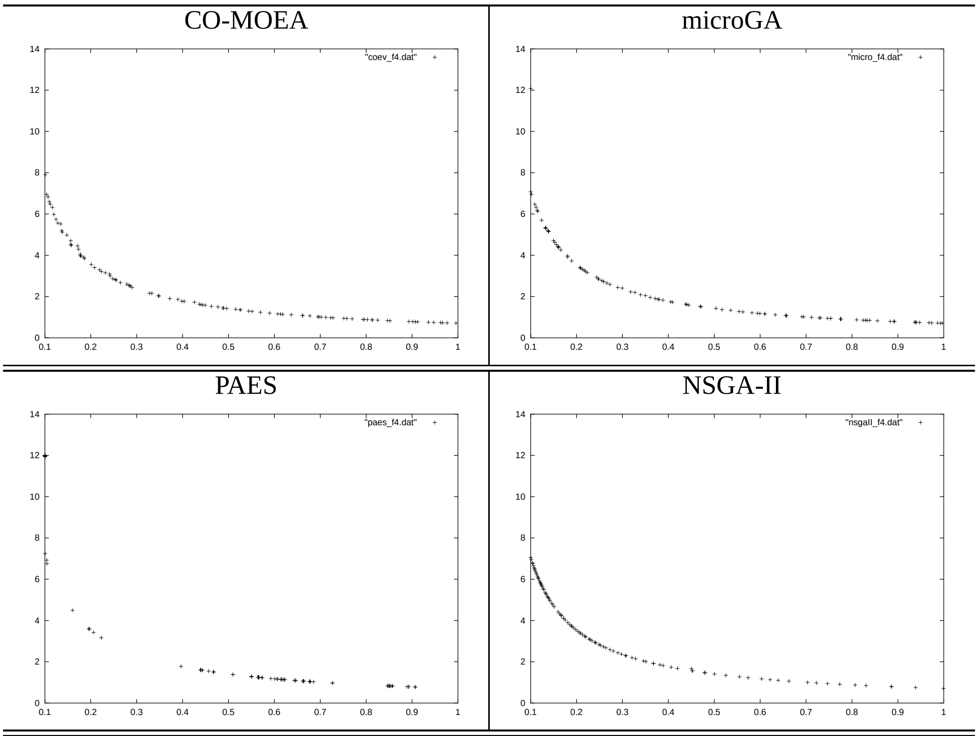

Figure 6: Pareto fronts obtained by our approach (CO-MOEA), the microGA [3], PAES [10] and the NSGA-II [6] for test function 3.

zon, V. Honavar, M. Jakiela, and R. E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99)*, volume 2, pages 1657–1665, San Francisco, California, July 1999. Morgan Kaufmann.

[13] C. Pimpawat and N. Chaiyaratana. Using a co-operative co-evolutionary genetic algorithm to solve a three-dimensional container loading problem. In *2001 Congress on Evolutionary Computation (CEC'2001)*, volume 2, pages 1197–1204, Seoul, Korea, 2001. IEEE Press.

[14] M. Potter and K. D. Jong. A cooperative coevolutionary approach to function optimization. In *Proceedings from the Fifth Parallel Problem Solving from Nature*, pages 530–539, Jerusalem, Israel, 1994. Springer-Verlag.

[15] Rosin and R. Belew. New methods for competitive coevolution. *Evolutionary Computation*, 5(1):1–29, 1996.

[16] J. R. Schott. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1995.

[17] D. A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, May 1999.

[18] D. A. Van Veldhuizen and G. B. Lamont. On Measuring Multiobjective Evolutionary Algorithm Performance. In *2000 Congress on Evolutionary Computation*, volume 1, pages 204–211, Piscataway, New Jersey, July 2000. IEEE Service Center.

[19] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.

Test Function 3 (9600 evaluations)					
		CO-MOEA	microGA	PAES	NSGA-II
ER	best	0.16	0.05	0.0	0.0
	median	0.28	0.15	0.24	0.01
	worst	1.0	1.0	1.0	1.0
	average	0.40	0.19	0.44	0.243
	std. dev.	0.277	0.180	0.4271	0.4251
GD	best	0.01	0.0045	0.0027	0.0085
	median	0.075	0.0349	0.0517	0.0091
	worst	0.18	0.154	0.2257	0.2028
	average	0.080	0.042	0.0771	0.0314
	std. dev.	0.0361	0.0415	0.0717	0.0562
SP	best	0.041	0.033	0.029	0.026
	median	0.193	0.071	0.087	0.033
	worst	1.5	0.906	0.504	0.056
	average	0.274	0.220	0.110	0.034
	std. dev.	0.2186	0.2328	0.1065	0.0066

Test Function 3 - Two Set Coverage Metric *SC*		
X	$SC(X,$CO-MOEA$)$	$SC(X,$microGA$)$
CO-MOEA	0.0	0.09
microGA	0.24	0.0
PAES	0.06	0.04
NSGA-II	0.35	0.12
X	$SC(X,$PAES$)$	$SC(X,$NSGA-II$)$
CO-MOEA	0.01	0.01
microGA	0.16	0.02
PAES	0.0	0.01
NSGA-II	0.24	0.0

Table 3: Comparison of results between our approach (denoted by CO-MOEA), the microGA [3], PAES [10] and the NSGA-II [6] for test function 3.