A Simple Multimembered Evolution Strategy to Solve Constrained Optimization Problems

Efrén Mezura-Montes and Carlos A. Coello Coello*
Evolutionary Computation Group (EVOCINV)
CINVESTAV-IPN
Computer Science Section
Electrical Engineering Department
México, D.F., 07300, MEXICO
emezura@computacion.cs.cinvestav.mx
ccoello@cs.cinvestav.mx

August 30, 2003

Abstract

This paper presents a simple multimembered evolution strategy (SMES) to solve global nonlinear optimization problems. The approach does not require the use of a penalty function and it does not require any extra parameters (besides those used with an evolution strategy). Instead, it uses a simple diversity mechanism based on allowing infeasible solutions to remain in the population This technique helps the algorithm to find the global optimum despite reaching reasonably fast the feasible region of the search space. Some simple selection criteria are used to guide the process to the feasible region of the search space. Also, the initial step size of the evolution strategy is reduced in order to perform a finer search and a combined (discrete/intermediate) recombination technique improves its exploitation capabilities. The approach was tested with a well-known benchmark. The results obtained are very competitive, when comparing the proposed approach against other state-of-the art techniques and its computational cost (measured by the number of fitness function evaluations) is lower than the required cost of the other techniques compared.

1 Introduction

Evolutionary algorithms (EAs) have been widely used to solve several types of optimization problems [10, 3, 8]. Nevertheless, they are unconstrained search techniques and lack an explicit mechanism to bias the search in constrained search spaces. This

*Technical Report EVOCINV-04-2003, Evolutionary Computation Group at CINVESTAV, Sección de Computación, Departamento de Ingeniería Eléctrica, CINVESTAV-IPN, México, August 2003.

has motivated the development of a considerable number of approaches to incorporate constraints into the fitness function of an EA [22, 6].

The most common approach adopted to deal with constrained search spaces is the use of penalty functions [24]. When using a penalty function, the amount of constraint violation is used to punish or "penalize" an infeasible solution so that feasible solutions are favored by the selection process. Despite the popularity of penalty functions, they have several drawbacks from which the main one is that they require a careful fine tuning of the penalty factors that accurately estimates the degree of penalization to be applied so that we can approach efficiently the feasible region [28, 6].

Evolution Strategies (ES) have been found not only efficient in solving a wide variety of optimization problems [27, 11, 4, 2, 1], but also have a strong theoretical background [26, 3, 5].

Our approach uses the self-adaptive mutation mechanism of a multimembered evolution strategy to explore constrained search spaces. This mechanism is combined with three simple selection criteria to guide the search towards the global optima of constrained optimization problems. To avoid a high selection pressure and maintain infeasible solutions in the population, a simple diversity mechanism is added. The idea is to allow the individual with the lowest amount of constraint violation and the best value of the objective function to be selected for the next population. This solution can be chosen with a 50% of probability either from the parents or the offspring population. A hybrid crossover operator that combines discrete and intermediate recombination is used to improve the exploitation mechanism of our algorithm.

With these combined elements, the algorithm first focuses on reaching the feasible region of the search space. After that, it is capable of moving over the feasible region as to reach the global optimum. The infeasible solutions that remain in the population are used to sample points in the boundaries between the feasible and the infeasible regions. Thus, the main focus of this paper is to show how a multimembered evolution strategy coupled with very simple mechanisms is able to produce results that are highly competitive with respect to other constraint-handling approaches that are representative of the state-of-the-art in evolutionary optimization.

This paper is organized as follows: In Section 2 we define the global nonlinear optimization problem that we aim to solve. After that, in Section 3 a description of previous approaches based on similar ideas is provided. Section 4 presents a detailed description of our approach. Then, in Section 5, we present the experimental design and show the obtained results which are discussed in Section 6. In Section 7 some conclusions are established. Finally, some possible paths for future research are provided in Section 8.

2 Statement of the Problem

We are interested in the general nonlinear programming problem in which we want to:

$$\text{Find } \vec{x} \text{ which optimizes } f(\vec{x}) \tag{1}$$

subject to:

$$g_i(\vec{x}) \leq 0, \quad i = 1, \ldots, n \tag{2}$$

$$h_j(\vec{x}) = 0, \quad j = 1, \ldots, p \tag{3}$$

where $\vec{x}$ is the vector of solutions $\vec{x} = [x_1, x_2, \ldots, x_r]^T$, n is the number of inequality constraints and p is the number of equality constraints (in both cases, constraints could be linear or nonlinear).

If we denote with $\mathcal{F}$ to the feasible region and with $\mathcal{S}$ to the whole search space, then it should be clear that $\mathcal{F} \subseteq \mathcal{S}$.

For an inequality constraint that satisfies $g_i(\vec{x}) = 0$, then we will say that is active at $\vec{x}$. All equality constraints h_j (regardless of the value of $\vec{x}$ used) are considered active at all points of $\mathcal{F}$.

3 Previous Work

The inspiration of our approach was motivated by the idea of exploring the capabilities of multiobjecitve optimization concepts to solve global optimization problems. We compared four representative approaches using the same test functions adopted in this paper [17, 19]. One of the conclusions of this work was the importance of a mechanism to maintain diversity in the population (i.e., to allow feasible and infeasible solutions to remain in the population during all the evolutionary process).

Motivated by the fact that the most recent and competitive approaches to solve constrained optimization problems are based on an Evolution Strategy (e.g. Stochastic Ranking [25] and ASCHEA [12]) we hypothesized the following:

1. The self-adaptation mechanism of an ES helps to sample the search space well enough as to reach the feasible region reasonably fast.

2. The simple additional of tournaments based on feasibility to an ES should be enough to guide the search in such a way that the global optimum can be approached efficiently.

Thus, based on these ideas, we implemented a generic ES-based approach to solve constrained optimization problems. Then, we performed an empirical study in which we varied the type of selection ("+" or ",") and the type of mutation (noncorrelated or correlated) [21]. We also implemented a simple $(\mu + 1)$ with the "1/5 successful rule" to self-adapt the sigma value [21]. Constraints were handled using tournaments based on feasibility (see Section 4 for details).

The use of tournaments based on feasibility has been explored in the past by other authors. Jiménez and Verdegay [14] proposed an approach similar to a min-max formulation used in multiobjective optimization combined with tournament selection. The rules used by them are similar to those adopted in this work. However, Jiménez and Verdegay's approach lacks an explicit mechanism to avoid the premature convergence produced by the random sampling of the feasible region because their approach is guided by the first feasible solution found.

Deb [9] used the same tournament rules adopted in our approach. However, Deb proposed to use niching as a diversity mechanism, which introduces some extra computational time (niches are an $O(N^2)$ procedure). Also, in Deb's approach, feasible solutions are always considered better than infeasible ones. This contradicts the idea of allowing infeasible individuals to remain in the population. Therefore, this approach will have difficulties in problems in which the global optimum lies on the boundary between the feasible and the infeasible regions.

Coello & Mezura [7] used tournament selection based on feasibility rules (this is one of four different multiobjective-based techniques compared). They also adopted nondominance checking using a sample of the population (as the multiobjective optimization approach called NPGA [13]). In this approach, a user-defined parameter S_r is used to control the diversity in the population. This approach provided good results in some well-known engineering problems and in some benchmark problems, but presented problems when facing high dimensionality [7].

From our ES's comparative study, the best results were provided by a $(\mu + 1)$-ES [21] in which one child created from μ mutations of the current solution competes against it and the better one is selected as the new current solution. However, the approach presented premature convergence in some test functions [21]. A $(1 + \lambda)$-ES was proposed in [18], which improved the robustness and quality of the previous ES proposed by the same authors. In this case, a self-adaptive parameter called Selection Ratio (S_r) (similar to that proposed by Coello & Mezura and mentioned before [7]) refers to the percentage of selections that will be performed in a deterministic way (as used in the original version of our SES where the child replaces the current solution based on the three selection criteria). In the remaining $1 - S_r$ selections, there are two choices: (1) either the parent (out of the λ) with the best value of the objective function will replace the current solution (regardless of its feasibility) or (2) the best parent (based on the three selection criteria) will replace the current solution. Both options are given a 50% probability each.

The $(1+\lambda)$-ES approach proposed in [18] made evident that having a good mechanism to maintain diversity is one of the keys to produce a constraint-handling approach that is competitive with the techniques representative of the state-of-the-art in the area.

However, these two approaches, based on a non-population ES lack the explorative power that allows them to sample larger search spaces. Thus, we decided to re-evaluate the use of a $(\mu + \lambda)$-ES to solve this limitation, but in this case, adding the diversity mechanism implemented in our previous approaches.

4 Our approach

Our new approach is based on the same concepts that its predecessors discussed in Section 3: (1) the self-adaptation mechanism of an ES and (2) three simple selection criteria:

1. Between 2 feasible solutions, the one with the highest fitness value wins.
2. If one solution is feasible and the other one is infeasible, the feasible solution wins.

3. If both solutions are infeasible, the one with the lowest sum of constraint violation is preferred.

Also, it has a simple diversity mechanism similar to that used in the $(1+\lambda)$-ES and a combination of discrete and intermediate crossover.

The detailed features of our algorithm are the following:

- **Diversity Mechanism**: With an idea similar to that used in the $(1+\lambda)$-ES version, we allow infeasible solutions to remain in the population. However, unlike this previous approach, where the best parent based only on the objective function (regardless of its feasibility) can survive, in this new approach we allow the infeasible individual with the best value of the objective function and with the lowest amount of constraint violation to survive for the next generation. This solution (called by us as the best infeasible solution) can be chosen either from the parents or the offspring population, with a 50% of probability. This process of allowing the best infeasible solution to survive for the next generation happens 3 times every 100 during the same generation. However, it is a desired behavior because a few copies of this solution will allow its recombination with several solutions in the population, specially with feasible ones. Recombining feasible solutions with infeasible solutions in promising areas (based on the good value of the objective function) and close to the the boundary of the feasible region will allow the ES to reach global optimum solutions located in the boundary of the feasible region of the search space (which are known as the most difficult solutions to be reached). Following the idea of allowing just a few infeasible solutions (one in case of the $(1+\lambda)$-ES approach) we allow the best infeasible solution to be copied into the population for the next generation just 3 times for every 100 attempts. It works in the following way: When the selection process occurs, the best individuals among the parents and offspring are selected based on the three selection criteria previously indicated. The selection process will pick feasible solutions with a better value of the objective function first, followed by infeasible solutions with a lower value of constraint violation. However, 3 times from every 100 picks, the best infeasible solution (from either the parents or the offspring population with a 50% of probability each) the best infeasible solution is copied in the population for the next generation. The pseudocode is listed in Figure 1.

- **Combined crossover**: We use global crossover, but with a combination of the discrete and intermediate recombination operators. Each gene in the chromosome can be processed with any of these two crossover operators with a 50% of probability. This operator is applied to both, strategy parameters (sigma values) and decision variables of the problem. The pseudocode is shown in Figure 2.

- **Reduction of the initial stepsize of the ES**: Based on the results obtained in the previous versions of our algorithm, which only use a sigma value to define the stepsize of the mutation operator, we decided to experiment with just a percentage of the quantity obtained by the formula proposed by Schwefel [26]. We initialize the sigma values for all the individuals in the initial population with

```
function selection()
    For i=1 to μ Do
        If flip(0.97)
            Select the best individual based on the selection criteria
            from the union of the parents and offspring population,
            add it to the population for the next generation and delete
            it from this union.
        Else
            If flip(0.5)
                Select the best infeasible individual from the parents
                population and add it to the population for the next
                generation.
            Else
                Select the best infeasible individual from the offspring
                population and add it to the population for the next
                generation.
            End If
        End If
    End For
End
```

Figure 1: Pseudocode of the selection procedure with the diversity mechanism incorporated. $flip(P)$ is a function that returns TRUE with probability P

only a 40% of the value obtained by the following formula (where n is the number of decision variables):

$$\sigma_i(0) = 0.4 \times \left(\frac{\Delta x_i}{\sqrt{n}} \right) \tag{4}$$

where Δx_i is approximated with the expression (suggested in [25]), $\Delta x_i \approx x_i^u - x_i^l$, where $x_i^u - x_i^l$ are the upper and lower limits of the decision variable i.

The idea of this reduction is to favor finer movements in the search space, because the previous versions were capable of reaching optimum solutions when the sigma value was close to 0.0.

Summarizing, our approach works over a simple multimembered evolution strategy: $(\mu + \lambda)$-ES. The only modifications introduced are the reduction of the initial stepsize of the sigma values, the global combined (discrete-intermediate) crossover and the changes to the original deterministic selection of the ES (made by sorting the solutions based on the three selection criteria discussed in Section 3), allowing the best infeasible solution, from either the parents or the offspring population, to remain in the next generation.

5 Experiments and Results

To evaluate the performance of the proposed approach we used the 13 test functions described in [25]. The test functions chosen contain characteristics that are representative

```
function crossover()
    Select mate_1 from the parents population
    For i=1 to NUMBER_OF_VARIABLES Do
        Select mate_2 from the parents population
        If flip(0.5)
            If flip(0.5)
                child_i = mate_1_i
            Else
                child_i = mate_2_i
            End If
        Else
                child_i = mate_1_i + ((mate_2_i - mate_1_i/2, 0)
        End If
    End For
End
```

Figure 2: Pseudocode of the global combined (discrete-intermediate) crossover operator used by our approach. $flip(P)$ is a function that returns TRUE with probability P

of what can be considered "difficult" global optimization problems for an evolutionary algorithm. Their expressions are provided next.

1. **g01**:
   Minimize: $f(\vec{x}) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i$ subject to:

$$
\begin{aligned}
g_1(\vec{x}) &= 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0 \\
g_2(\vec{x}) &= 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0 \\
g_3(\vec{x}) &= 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0 \\
g_4(\vec{x}) &= -8x_1 + x_{10} \leq 0 \\
g_5(\vec{x}) &= -8x_2 + x_{11} \leq 0 \\
g_6(\vec{x}) &= -8x_3 + x_{12} \leq 0 \\
g_7(\vec{x}) &= -2x_4 - x_5 + x_{10} \leq 0 \\
g_8(\vec{x}) &= -2x_6 - x_7 + x_{11} \leq 0 \\
g_9(\vec{x}) &= -2x_8 - x_9 + x_{12} \leq 0
\end{aligned}
$$

   where the bounds are $0 \leq x_i \leq 1$ ($i = 1, \ldots, 9$), $0 \leq x_i \leq 100$ ($i = 10, 11, 12$) and $0 \leq x_{13} \leq 1$. The global optimum is at $x^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1)$ where $f(x^*) = -15$. Constraints g_1, g_2, g_3, g_4, g_5 and g_6 are active.

2. **g02**:
Maximize: $f(\vec{x}) = \left| \frac{\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right|$ subject to:

$$\begin{aligned} g_1(\vec{x}) &= 0.75 - \prod_{i=1}^{n} x_i \leq 0 \\ g_2(\vec{x}) &= \sum_{i=1}^{n} x_i - 7.5n \leq 0 \end{aligned} \tag{5}$$

where $n = 20$ and $0 \leq x_i \leq 10$ $(i = 1, \ldots, n)$. The global maximum is unknown; the best reported solution is [25] $f(x^*) = 0.803619$. Constraint g_1 is close to being active ($g_1 = -10^{-8}$).

3. **g03**:
Maximize: $f(\vec{x}) = (\sqrt{n})^n \prod_{i=1}^{n} x_i$

subject to:

$h(\vec{x}) = \sum_{i=1}^{n} x_i^2 - 1 = 0$

where $n = 10$ and $0 \leq x_i \leq 1$ $(i = 1, \ldots, n)$. The global maximum is at $x_i^* = 1/\sqrt{n}$ $(i = 1, \ldots, n)$ where $f(x^*) = 1$.

4. **g04**:
Minimize: $f(\vec{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$
subject to:
$g_1(\vec{x}) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \leq 0$
$g_2(\vec{x}) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0$
$g_3(\vec{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leq 0$
$g_4(\vec{x}) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0$
$g_5(\vec{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0$
$g_6(\vec{x}) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0$

where: $78 \leq x_1 \leq 102$, $33 \leq x_2 \leq 45$, $27 \leq x_i \leq 45$ $(i = 3, 4, 5)$. The optimum solution is $x^* = (78, 33, 29.995256025682, 45, 36.775812905788)$ where $f(x^*) = -30665.539$. Constraints g_1 y g_6 are active.

5. **g05**
Minimize:$f(\vec{x}) = 3x_1 + 0.000001x_1^3 + 2x_2 + (0.000002/3)x_2^3$

subject to:
$g_1(\vec{x}) = -x_4 + x_3 - 0.55 \leq 0$

$g_2(\vec{x}) = -x_3 + x_4 - 0.55 \le 0$
$h_3(\vec{x}) = 1000\sin(-x_3 - 0.25) + 1000\sin(-x_4 - 0.25) + 894.8 - x_1 = 0$
$h_4(\vec{x}) = 1000\sin(x_3 - 0.25) + 1000\sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0$
$h_5(\vec{x}) = 1000\sin(x_4 - 0.25) + 1000\sin(x_4 - x_3 - 0.25) + 1294.8 = 0$

where $0 \le x_1 \le 1200$, $0 \le x_2 \le 1200$, $-0.55 \le x_3 \le 0.55$, and $-0.55 \le x_4 \le 0.55$. The best known solution is $x^* = (679.9453, 1026.067, 0.1188764, -0.3962336)$ where $f(x^*) = 5126.4981$.

6. **g06**
Minimize: $f(\vec{x}) = (x_1 - 10)^3 + (x_2 - 20)^3$
subject to:
$g_1(\vec{x}) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \le 0$
$g_2(\vec{x}) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \le 0$

where $13 \le x_1 \le 100$ and $0 \le x_2 \le 100$. The optimum solution is $x^* = (14.095, 0.84296)$ where $f(x^*) = -6961.81388$. Both constraints are active.

7. **g07**
Minimize: $f(\vec{x}) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$
subject to:
$g_1(\vec{x}) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 0$
$g_2(\vec{x}) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0$
$g_3(\vec{x}) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \le 0$
$g_4(\vec{x}) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \le 0$
$g_5(\vec{x}) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \le 0$
$g_6(\vec{x}) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \le 0$
$g_7(\vec{x}) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \le 0$
$g_8(\vec{x}) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \le 0$

where $-10 \le x_i \le 10$ $(i = 1, \ldots, 10)$. The global optimum is $x^* = (2.171996, 2.363683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927)$ where $f(x^*) = 24.3062091$. Constraints g_1, g_2, g_3, g_4, g_5 and g_6 are active.

8. **g08**
Maximize: $f(\vec{x}) = \frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1 + x_2)}$
subject to:
$g_1(\vec{x}) = x_1^2 - x_2 + 1 \le 0$
$g_2(\vec{x}) = 1 - x_1 + (x_2 - 4)^2 \le 0$

where $0 \le x_1 \le 10$ and $0 \le x_2 \le 10$. The optimum solution is located at $x^* = (1.2279713, 4.2453733)$ where $f(x^*) = 0.095825$.

9. **g09**
Minimize: $f(\vec{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 +$

$7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$
subject to:

$$g_1(\vec{x}) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 0$$
$$g_2(\vec{x}) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0$$
$$g_3(\vec{x}) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0$$
$$g_4(\vec{x}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0$$

where $-10 \leq x_i \leq 10$ $(i = 1, \ldots, 7)$. The global optimum is $x^* = (2.330499, 1.951372, -0.4775414, 4.365726, -0.6244870, 1.038131, 1.594227)$ where $f(x^*) = 680.6300573$. Two constraints are active (g_1 and g_4).

10. **g10**
Minimize: $f(\vec{x}) = x_1 + x_2 + x_3$
subject to: $g_1(\vec{x}) = -1 + 0.0025(x_4 + x_6) \leq 0$
$g_2(\vec{x}) = -1 + 0.0025(x_5 + x_7 - x_4) \leq 0$
$g_3(\vec{x}) = -1 + 0.01(x_8 - x_5) \leq 0$
$g_4(\vec{x}) = -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \leq 0$
$g_5(\vec{x}) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0$
$g_6(\vec{x}) = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \leq 0$

where $100 \leq x_1 \leq 10000$, $1000 \leq x_i \leq 10000$, $(i = 2, 3)$, $10 \leq x_i \leq 1000$, $(i = 4, \ldots, 8)$. The global optimum is: $x^* = (579.3167, 1359.943, 5110.071, 182.0174, 295.5985, 217.9799, 286.4162, 395.5979)$, where $f(x^*) = 7049.3307$. g_1, g_2 and g_3 are active.

11. **g11**
Minimize: $f(\vec{x}) = x_1^2 + (x_2 - 1)^2$
subject to:
$h(\vec{x}) = x_2 - x_1^2 = 0$

where: $-1 \leq x_1 \leq 1$, $-1 \leq x_2 \leq 1$. The optimum solution is $x^* = (\pm 1/\sqrt{2}, 1/2)$ where $f(x^*) = 0.75$.

12. **g12**
Maximize: $f(\vec{x}) = \frac{100 - (x_1-5)^2 - (x_2-5)^2 - (x_3-5)^2}{100}$
subject to:
$g_1(\vec{x}) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \leq 0$

where $0 \leq x_i \leq 10$ $(i = 1, 2, 3)$ and $p, q, r = 1, 2, \ldots, 9$. The feasible region of the search space consists of 9^3 disjointed spheres. A point (x_1, x_2, x_3) is feasible if and only if there exist p, q, r such the above inequality (12) holds. The global optimum is located at $x^* = (5, 5, 5)$ where $f(x^*) = 1$.

13. **g13**
Minimize: $f(\vec{x}) = e^{x_1x_2x_3x_4x_5}$
subject to:

Problem	n	Type of function	ρ	LI	NI	LE	NE
g01	13	quadratic	0.0003%	9	0	0	0
g02	20	nonlinear	99.9973%	2	0	0	0
g03	10	nonlinear	0.0026%	0	0	0	1
g04	5	quadratic	27.0079%	4	2	0	0
g05	4	nonlinear	0.0000%	2	0	0	3
g06	2	nonlinear	0.0057%	0	2	0	0
g07	10	quadratic	0.0000%	3	5	0	0
g08	2	nonlinear	0.8581%	0	2	0	0
g09	7	nonlinear	0.5199%	0	4	0	0
g10	8	linear	0.0020%	6	0	0	0
g11	2	quadratic	0.0973%	0	0	0	1
g12	3	quadratic	4.7697%	0	9^3	0	0
g13	5	nonlinear	0.0000%	0	0	1	2

Table 1: Values of ρ for the 13 test problems chosen.

$g_1(\vec{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$
$g_2(\vec{x}) = x_2 x_3 - 5 x_4 x_5 = 0$
$g_3(\vec{x}) = x_1^3 + x_2^3 + 1 = 0$
where $-2.3 \leq x_i \leq 2.3$ $(i = 1, 2)$ and $-3.2 \leq x_i \leq 3.2$ $(i = 3, 4, 5)$. The optimum solution is $x^* = (-1.717143, 1.595709, 1.827247, -0.7636413, -0.763645)$ where $f(x^*) = 0.0539498$.

To get a measure of the difficulty of solving each of these problems, a ρ metric (as suggested by Koziel and Michalewicz [16]) was computed using the following expression:

$$\rho = |F|/|S| \tag{6}$$

where $|F|$ is the number of feasible solutions and $|S|$ is the total number of solutions randomly generated. In this work, $S = 1,000,000$ random solutions.
The different values of ρ for each of the functions chosen are shown in Table 1, where n is the number of decision variables, LI is the number of linear inequalities, NI the number of nonlinear inequalities, LE is the number of linear equalities and NE is the number of nonlinear equalities.

We performed 30 independent runs for each test function. The learning rates values were calculated using the formulas proposed by Schwefel [26] (where n is the number of decision variables of the problem):

$$\tau = \left(\sqrt{2\sqrt{n}}\right)^{-1} \tau' = \left(\sqrt{2n}\right)^{-1} \tag{7}$$

The initial values for the standard deviations were calculated using the reduced value proposed in Section 4.

For the experiments we used the following parameters:

	Statistical Results of Simple Multimembered Evolution Strategy (SMES)					
Problem	**Optimal**	**Best**	**Mean**	**Median**	**Worst**	**St. Dev.**
g01	-15.000000	-15.000000	-15.000000	-15.000000	-15.000000	0.000000
g02	0.803619	0.803601	0.785238	0.792549	0.751322	0.016757
g03	1.000000	1.001038	1.000989	1.001017	1.000579	0.000209
g04	-30665.539000	-30665.539062	-30665.539062	-30665.539062	-30665.539062	0.000000
g05	5126.498000	5126.599609	5174.492301	5160.197754	5304.166992	50.057854
g06	-6961.814000	-6961.813965	-6961.283984	-6961.813965	-6952.481934	1.851141
g07	24.306000	24.326715	24.474926	24.426246	24.842829	0.132385
g08	0.095825	0.095826	0.095826	0.095826	0.095826	0.000000
g09	680.630000	680.631592	680.643410	680.641571	680.719299	0.015528
g10	7049.330700	7051.902832	7253.047005	7253.603027	7638.366211	136.023716
g11	0.750000	0.749090	0.749358	0.749357	0.749830	0.000152
g12	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000
g13	0.053950	0.053986	0.166385	0.061873	0.468294	0.176855

Table 2: Statistical results obtained by the SMES for the 13 test functions with 30 independent runs.

- $\mu = 100$.
- $\lambda = 300$.
- Number of generations = 800.
- Number of objective function evaluations = $240,000$.

The combined crossover operator explained in detail in Section 4 was used both for the decision variables of the problem and for the strategy parameters (sigma values). Note that we do not use correlated mutation [20].

To deal with equality constraints, a parameterless dynamic mechanism originally proposed in ASCHEA [12] and used in [21] and in [18] is adopted. The tolerance value ϵ is decreased with respect to the current generation using the following expression:

$$\epsilon_j(t+1) = \epsilon_j(t)/1.00195 \tag{8}$$

The initial ϵ_0 was set to 0.001. For problem g13, ϵ_0 was set to 3.0 and, in consequence, the factor to decrease the tolerance value was modified to $\epsilon_j(t+1) = \epsilon_j(t)/1.0145$. Also, for problems $g03$ and $g13$ the initial stepsize required a more dramatic decrease of the stepsize. They were defined as 0.01 (just a 5% instead of the 40%) for $g03$ and 0.05 (a 2.5% instead of the 40%) for g13. These two test functions seem to provide better results with very smooth movements. It is important to note that these two problems share the following features: moderately high dimensionality (five or more decision variables), nonlinear objective function, one or more equality constraints, and moderate size of the search space (based on the range of the decision variables). These common features suggest that for this type of problem, finer movements provide a better sampling of the search space using an evolution strategy.

The statistical results of our SMES are summarized in Table 2.

In order to know how useful is the diversity mechanism we obtained the percentage of feasible solutions per generation in the population. The results are shown in Figure 3.

We compare our approach against three state-of-the-art approaches: the Homomorphous Maps (HM) [16] in Table 3, Stochastic Ranking (SR) [25] in Table 4 and the

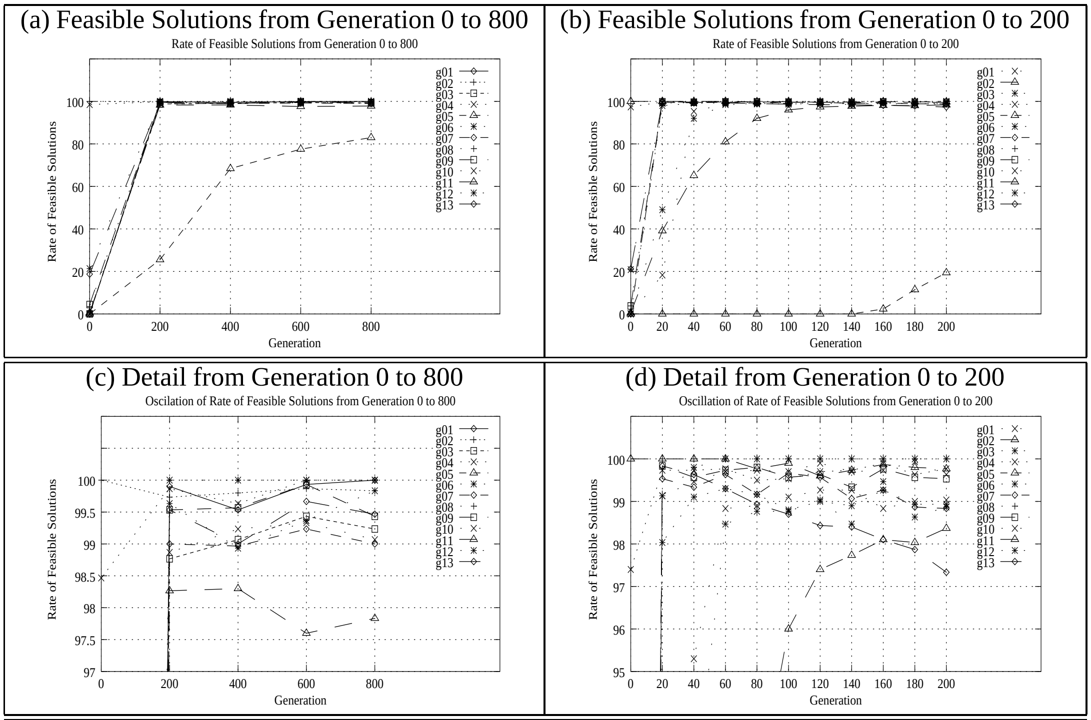

Figure 3: Rate of Feasible Solutions (a) every 200 generations (from 0 to 800), (b) every 20 generations (from 0 to 200), (c) a detailed oscillation of feasible and infeasible solutions (from 0 to 800) and (d) a detailed oscillation of feasible and infeasible solutions (from 0 to 200)

Adaptive Segregational Constraint Handling Evolutionary Algorithm (ASCHEA) [12] in Table 5.

6 Discussion of Results

As described in Table 2, our approach was able to find the global optimum in seven test functions (g01, g03, g04, g06, g08, g11 and g12) and it found solutions very close to the global optimum in the remaining six (g02, g05, g07, g09, g10, g13). In Table 6 we show in how many runs the optimum was reached. In addition, we show the lowest and the average generation in which the optimum was found. The results obtained suggest that for these problems where the global optimum is reached, the algorithm is capable of finding it using no more than 250 generations (about $75,000$ evaluations of the objective function), except for function g01 where the number of generations is 671.

When compared with respect to the three state-of-the-art techniques previously indicated, we found the following:

		Best Result		Mean Result		Worst Result	
Problem	Optimal	SMES	HM	SMES	HM	SMES	HM
g01	-15.000000	-15.000000	-14.7886	-15.000000	-14.7082	-15.000000	-14.6154
g02	0.803619	0.803601	0.79953	0.785238	0.79671	0.751322	0.79119
g03	1.000000	1.001038	0.9997	1.000989	0.9989	1.000579	0.9978
g04	-30665.539000	-30665.539062	-30664.5	-30665.539062	-30655.3	-30665.539062	-30645.9
g05	5126.49800	5126.599609	$-$	5174.492301	$-$	5304.166992	$-$
g06	-6961.814000	-6961.813965	-6952.1	-6961.283984	-6342.6	-6952.481934	-5473.9
g07	24.306000	24.326715	24.620	24.474926	24.826	24.842829	25.069
g08	0.095825	0.095826	0.0958250	0.095826	0.0891568	0.095826	0.0291438
g09	680.630000	680.631592	680.91	680.643410	681.16	680.719299	683.18
g10	7049.330700	7051.902832	7147.9	7253.047005	8163.6	7638.366211	9659.3
g11	0.750000	0.749090	0.75	0.749358	0.75	0.749830	0.75
g12	1.000000	1.000000	0.999999857	1.000000	0.999134613	1.000000	0.991950498
g13	0.053950	0.053986	NA	0.166385	NA	0.468294	NA

Table 3: Comparison of our approach (SMES) with respect to the Homomorphous Maps (HM).

		Best Result		Mean Result		Worst Result	
Problem	Optimal	SMES	SR	SMES	SR	SMES	SR
g01	-15.000000	-15.000000	-15.000	-15.000000	-15.000	-15.000000	-15.000
g02	0.803619	0.803601	0.803515	0.785238	0.781975	0.751322	0.726288
g03	1.000000	1.001038	1.000	1.000989	1.000	1.000579	1.000
g04	-30665.539000	-30665.539062	-30665.539	-30665.539062	-30665.539	-30665.539062	-30665.539
g05	5126.49800	5126.599609	5126.497	5174.492301	5128.881	5304.166992	5142.472
g06	-6961.814000	-6961.813965	-6961.814	-6961.283984	-6875.940	-6952.481934	-6350.262
g07	24.306000	24.326715	24.307	24.474926	24.374	24.842829	24.642
g08	0.095825	0.095826	0.095825	0.095826	0.095825	0.095826	0.095825
g09	680.630000	680.631592	680.630	680.643410	680.656	680.719299	680.763
g10	7049.330700	7051.902832	7054.316	7253.047005	7559.192	7638.366211	8835.655
g11	0.750000	0.749090	0.750	0.749358	0.750	0.749830	0.750
g12	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
g13	0.053950	0.053986	0.053957	0.166385	0.057006	0.468294	0.216915

Table 4: Comparison of our approach (SMES) with respect to Stochastic Ranking (SR).

		Best Result		Mean Result		Worst Result	
Problem	Optimal	SMES	ASCHEA	SMES	ASCHEA	SMES	ASCHEA
g01	-15.000000	-15.000000	-15.0	-15.000000	-14.84	-15.000000	NA
g02	0.803619	0.803601	0.785	0.785238	0.59	0.751322	NA
g03	1.000000	1.001038	1.0	1.000989	0.99989	1.000579	NA
g04	-30665.539000	-30665.539062	30665.5	-30665.539062	30665.5	-30665.539062	NA
g05	5126.49800	5126.599609	5126.5	5174.492301	5141.65	5304.166992	NA
g06	-6961.814000	-6961.813965	-6961.81	-6961.283984	-6961.81	-6952.481934	NA
g07	24.306000	24.326715	24.3323	24.474926	24.66	24.842829	NA
g08	0.095825	0.095826	0.095825	0.095826	0.095825	0.095826	NA
g09	680.630000	680.631592	680.630	680.643410	680.641	680.719299	NA
g10	7049.330700	7051.902832	7061.13	7253.047005	7193.11	7638.366211	NA
g11	0.750000	0.749090	0.75	0.749358	0.75	0.749830	NA
g12	1.000000	1.000000	NA	1.000000	NA	1.000000	NA
g13	0.053950	0.053986	NA	0.166385	NA	0.468294	NA

Table 5: Comparison of our approach (SMES) with respect to the Adaptive Segregational Constraint Handling Evolutionary Algorithm (ASCHEA). NA = Not Available.

Problem	Runs that find the optimum	Lowest generation	Average
g01	30	634	671
g03	30	41	184
g04	30	113	129
g06	15	47	249
g08	30	11	18
g11	30	28	88
g12	30	63	77

Table 6: Number of runs (out of 30) where the optimum is found. We also show the best and average generation number at which the optimum is found.

6.1 Compared with the Homomorphous Maps (HM)

Our approach found a better "best" solution in ten problems (g01, g02, g03, g04, g05, g06, g07, g09, g10 and g12) and a similar "best" result in other two (g08 and g11). Also, our technique reached better "mean" and "worst" results in ten problems (g01, g03, g04, g05, g06, g07, g08, g09, g10 and g12). A "similar" mean and worst result was found in problem g11. The Homomorphous maps found a "better" mean and worst result in function g02. No comparisons were made with function g13 because such results were not available for HM.

6.2 Compared with Stochastic Ranking (SR)

With respect to SR, our approach was able to find a better "best" result in functions g02 and g10. In addition, it found a "similar" best solution in seven problems (g01, g03, g04, g06, g08, g11 and g12). Slightly better "best" results were found by SR in the remaining functions (g05, g07, g09 and g13). Our approach found better "mean" and "worst" results in four test functions (g02, g06, g09 and g10). It also provided similar "mean" and "worst" results in six functions (g01, g03, g04, g08, g11 and g12). Finally, SR found again just slightly better "mean" and "worst" results in function g05, g07 and g13.

6.3 Compared with the Adaptive Segregational Constraint Handling Evolutionary Algorithm (ASCHEA)

Compared against ASCHEA, our algorithm found "better" best solutions in three problems (g02, g07 and g10) and it found "similar" best results in six functions (g01, g03, g04, g06, g08, g11). ASCHEA found slightly "better" best results in function g05 and g09. Additionally, our approach found "better" mean results in four problems (g01, g02, g03 and g07) and it found "similar" mean results in three functions (g04, g08 and g11). ASCHEA surpassed our mean results in four functions (g05, g06, g09 and g10). We did not compare the worst results because they were not available for ASCHEA. We did not perform comparisons with respect to ASCHEA using functions g12 and g13 for the same reason.

As we can see, our approach showed a very competitive performance with respect to these three state-of-the-art approaches.

6.4 Advantages of the Approach

Our approach can deal with moderately constrained problems (g04), highly constrained problems, problems with low (g06, g08), moderated (g09) and high (g01, g02, g03, g07) dimensionality, with different types of combined constraints (linear, nonlinear, equality and inequality) and with very large (g02), very small (g05 and g13) or even disjoint (g12) feasible regions. Also, the algorithm is able to deal with large search spaces (based on the intervals of the decision variables) with a very small feasible region (g10). Furthermore, the approach can find the global optimum in problems where such optimum lies on the boundaries of the feasible region (g01, g02, g04, g06, g07, g09). This behavior suggests that the mechanism of maintaining the best infeasible solution helps the search to sample the boundaries.

Besides still being a very simple approach, it is worth reminding that our algorithm does not require any extra parameters (other than those used with an evolution strategy). In contrast, the Homomorphous maps require an additional parameter (called v) which has to be found empirically [16]. Stochastic ranking requires the definition of a parameter called P_f, whose value has an important impact on the performance of the approach [25]. ASCHEA also requires the definition of several extra parameters, and in its latest version, it uses niching, which is a process that also has at least one additional parameter [12].

Regarding computational cost, we can say that the number of fitness function evaluations (FFE) performed by our approach is lower than the other techniques with respect to which it was compared. Our approach performed $240,000$ FFE. Stochastic ranking performed $350,000$ FFE, the Homomorphous maps performed $1,400,000$ FFE, and ASCHEA required $1,500,000$ FFE.

6.5 Reaching the Feasible Region

After discussing the quality, robustness and competitiveness of our approach, we wanted to verify how fast the algorithm reaches the feasible region, because in real-world problems it is important for an optimization algorithm to provide results (in some cases, at least feasible solutions) with a moderate number of objective function evaluations. Therefore, we performed an analysis of the rate of feasible solutions at every 200 generations (let's keep in mind that the total number of generations was fixed to 800). The results are presented in Figure 3(a). As it can be seen, for all the test problems our approach finds the feasible region between the initial generation and generation 200. For problem g05, more than 20% of the population is feasible and for the remaining functions almost all the population is feasible. Then, we were interested in analyzing two issues:

1. What is the behavior before generation 200 (how fast the population becomes almost feasible).

2. How well the diversity is maintained late in the evolutionary process.

The results obtained for these two questions are on Figure 3(b) and (d) for question 1 and in Figure 3(c) for question 2. Figure 3(b) shows that the feasible region was reached at generation 20. This means that the approach only required $6,100$ FFE to find feasible solutions. In Table 7, we show the statistical results obtained at this stage of the search. Note that although the results are still far from the optimum, except for problem g05, most of the solutions are feasible. In Figure 3(d) we observe in a close-up of figure 3(b) that the algorithm has the capacity to maintain some infeasible solutions despite the almost-feasible population (as originally proposed in the diversity mechanism). In addition, we show the statistical results obtained in generation 200 in Table 8. A marginal improvement of the quality and robustness of the results is shown in generation 200 where only $20,000$ FFE have been performed. Indeed, the results are close to the optimum in most of the problems (for problems g08 and g12 the algorithm has reached the global optimum). This means that the approach is about to converge. This highlights the importance of the diversity mechanism in order to avoid that the algorithm gets trapped in local optima.

On the other hand, Figure 3(c) shows a zooming of Figure 3(a), where it is possible to see again in detail the smooth oscillation on the rate of feasible solutions during the evolutionary process after generation 200. This behavior suggests that the diversity mechanism still works well, maintaining close-to-be-feasible solutions with a good value of the objective function in the population (between 1 and 3 infeasible solutions are enough based on the previous results of the $(1+\lambda)$-ES approach [21], which is able to avoid local optima with only a few copies of the best infeasible solution).

The final results (on generation 800) provided in Table 2, compared with those on generation 200 (Table 8), suggest that our diversity mechanism does its job of avoiding premature convergence and, when coupled with the combination of discrete/intermediate recombination and the self-adaptation mechanism of the ES leads the evolutionary search towards the global optimum of a problem.

It is important to remark that the process of finding the global optimum takes almost $3/4$ of the evolutionary search and only $1/4$ (or less) is necessary to find the feasible region of the search space. We argue that this behavior depends mostly of the landscape of the function, but such idea is not explored any further in this work.

The point here is that our approach is fast on finding the feasible region and, depending of the difficult of the landscape of the function, it can avoid local optima and reach the global one.

7 Conclusions

A new simple approach to handle constraints in evolutionary optimization was proposed. The proposed approach does not require the use of a penalty function and it does not require the definition by the user of any extra parameter (other than those required by an evolution strategy). The proposed approach uses the self-adaptation mechanism of a multimembered ES to sample enough the search space in order to reach the feasible region and it uses three simple selection criteria based on feasibility

Problem	SMES					
	Optimal	Best	Mean	Median	Worst	St. Dev.
g01	−15.000000	−7.290860	−5.685555	−5.685416	−3.819835	0.905351
g02	0.803619	0.442641	0.372686	0.370697	0.317450	0.026550
g03	1.000000	0.948809	0.785052	0.828657	0.526123	0.113725
g04	−30665.539000	−30563.615234	−30473.318750	−30462.027344	−30401.755859	40.262563
g05	5126.498000	∗5067.896973	5211.510775	5199.974121	5643.923340∗	103.261984
g06	−6961.814000	−6890.164062	−6235.588916	−6188.202881	−5552.386230	371.894270
g07	24.306000	62.136288	135.968836	121.868412	682.451904	107.721741
g08	0.095825	0.095826	0.095825	0.095826	0.095817	0.000002
g09	680.630000	686.591919	704.350751	704.377106	719.150879	8.907023
g10	7049.330700	12777.324219	17407.559310	17284.081055	25774.398438	2368.597366
g11	0.750000	0.750130	0.783047	0.763978	0.896942∗	0.040775
g12	1.000000	0.999998	0.999894	0.999903	0.999710	0.000070
g13	0.053950	0.001348	0.009035	0.004388	0.026345	0.008456

Table 7: Statistical Results of our approach after 20 generations. ("*" means infeasible)

Problem	SMES					
	Optimal	Best	Mean	Median	Worst	St. Dev.
g01	−15.000000	−14.999743	−14.960266	−14.999444	−13.827706	0.210311
g02	0.803619	0.801158	0.777458	0.787125	0.678203	0.024360
g03	1.000000	1.002948	0.998650	0.999797	0.986612	0.003744
g04	−30665.539000	−30665.539062	−30665.531055	−30665.536133	−30665.472656	0.013582
g05	5126.498000	5126.988281	5179.163021	5162.322998	5379.227051	63.500478
g06	−6961.814000	−6961.807617	−6959.910075	−6961.624268	−6938.690430	4.392025
g07	24.306000	24.472904	24.733662	24.711020	25.400835	0.215600
g08	0.095825	0.095826	0.095826	0.095826	0.095826	0.000000
g09	680.630000	680.642273	680.679742	680.680237	680.735535	0.024312
g10	7049.330700	7076.724609	7330.398486	7319.404785	7816.830078	153.716092
g11	0.750000	0.748746	0.749941	0.749342	0.766443	0.003075
g12	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000
g13	0.053950	0.041436	0.145069	0.045766	0.387152	0.153471

Table 8: Statistical Results of our approach after 200 generations.

to guide the search towards the global optimum. Furthermore, the proposed technique adopts a diversity mechanism which consists of allowing infeasible solutions close to the boundaries of the feasible region to remain in the next population. Additionally, a combination of discrete and intermediate crossover is used to improve the exploitation effort. Finally, in order to favor finer movements in the search space, the initial values of the stepsize (sigma values) are decreased 60%. This approach is very easy to implement and the computational cost (based on the number of fitness function evaluations) is considerably lower than the cost reported by other three constraint-handling techniques which are representative of the state-of-the-art in evolutionary optimization.

8 Future Work

Our future paths of research consist of applying our approach in the solution of real-world problems in engineering design problems. Additionally we will implement our constraint handling mechanism using other heuristics such as Differential Evolution [23] and Particle Swarm Optimization [15]. This aims to explore the possibility of decreasing its computational cost (measured in terms of the number of fitness function evaluations), after reaching the feasible region, since in the current approach almost 3/4 part of the search process is spent on this task.

Acknowledgements

The first author acknowledges support from the Mexican Consejo Nacional de Ciencia y Tecnología (CONACyT) through a scholarship to pursue graduate studies at CINVESTAV-IPN's Electrical Engineering Department. The second author acknowledges support from CONACyT through project number 32999-A.

References

[1] Dirk V. Arnold. *Noisy Optimization with Evolution Strategies*. Kluwer Academic Publishers, New York, June 2002. ISBN 1-4020-7105-1.

[2] Torsten Asselmeyer, Werner Ebeling, and Helge Rosé. Evolutionary strategies of optimization. *Phys. Rev. E*, 56(1):1171–1180, July 1997.

[3] Thomas Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York, 1996.

[4] Thomas Bäck and Martin Schütz. Evolution strategies for mixed-integer optimization of optical multilayer systems. In J. R. McDonnell, R. G. Reynolds, and D. B. Fogel, editors, *Evolutionary Programming IV: Proc. of the Fourth Annual Conf. on Evolutionary Computation*, pages 33–51, Cambridge, MA, 1995. MIT Press.

[5] H.-G. Beyer and D.V. Arnold. Theory of evolution strategies—a tutorial. In L. Kallel, N. Baudts, and A. Rogers, editors, *Theoretical Aspects of Evolutionary Computing*, pages 109–133. Springer, Berlin, 2001.

[6] Carlos A. Coello Coello. Theoretical and Numerical Constraint Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, January 2002.

[7] Carlos A. Coello Coello and Efrén Mezura-Montes. Handling Constraints in Genetic Algorithms Using Dominance-Based Tournaments. In I.C. Parmee, editor, *Proceedings of the Fifth International Conference on Adaptive Computing Design and Manufacture (ACDM 2002)*, volume 5, pages 273–284, University of Exeter, Devon, UK, April 2002. Springer-Verlag.

[8] Carlos A. Coello Coello, David A. Van Veldhuizen, and Gary B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, June 2002. ISBN 0-3064-6762-3.

[9] Kalyanmoy Deb. An Efficient Constraint Handling Method for Genetic Algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2/4):311–338, 2000.

[10] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Co., Reading, Massachusetts, 1989.

[11] Garrison W. Greenwood and Yi-Ping Liu. Finding low energy conformations of atomic clusters using evolution strategies. In V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, editors, *Evolutionary Programming VII*, pages 493–502, Berlin, 1998. Springer. Lecture Notes in Computer Science 1447.

[12] Sana Ben Hamida and Marc Schoenauer. ASCHEA: New Results Using Adaptive Segregational Constraint Handling. In *Proceedings of the Congress on Evolutionary Computation 2002 (CEC'2002)*, volume 1, pages 884–889, Piscataway, New Jersey, May 2002. IEEE Service Center.

[13] Jeffrey Horn, Nicholas Nafpliotis, and David E. Goldberg. A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, volume 1, pages 82–87, Piscataway, New Jersey, June 1994. IEEE Service Center.

[14] Fernando Jiménez and José L. Verdegay. Evolutionary techniques for constrained optimization problems. In Hans-Jürgen Zimmermann, editor, *7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99)*, Aachen, Germany, 1999. Verlag Mainz. ISBN 3-89653-808-X.

[15] James Kennedy and Russell C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann, San Francisco, California, 2001.

[16] Slawomir Koziel and Zbigniew Michalewicz. Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. *Evolutionary Computation*, 7(1):19–44, 1999.

[17] Efrén Mezura-Montes and Carlos A. Coello Coello. A Numerical Comparison of some Multiobjective-Based Techniques to Handle Constraints in Genetic Algorithms. Technical Report EVOCINV-03-2002, Evolutionary Computation Group at CINVESTAV, Sección de Computación, Departamento de Ingeniería Eléctrica, CINVESTAV-IPN, México D.F., México, 2002. Available in the List of References on Constraint-Handling Techniques used with Evolutionary Algorithms at http://www.cs.cinvestav.mx/~constraint.

[18] Efrén Mezura-Montes and Carlos A. Coello Coello. Adding a Diversity Mechanism to a Simple Evolution Strategy to Solve Constrained Optimization Problems. In *Proceedings of IEEE International Congress on Evolutionary Computation (CEC'2003)*. Camberra, Australia, IEEE Neural Networks Society, December 2003. (Accepted for Publication).

[19] Efrén Mezura-Montes and Carlos A. Coello Coello. Multiobjective-Based Concepts to Handle Constraints in Evolutionary Algorithms. In *Proceedings of the Fourth Mexican International Conference on Computer Science (ENC' 03)*. Apizaco, Tlaxcala, México, IEEE Computer Society, July 2003. Accepted for Publication.

[20] Efrén Mezura-Montes and Carlos A. Coello Coello. On the Usefulness of the Evolution Strategies Self-Adaptation Mechanism to Handle Constraints in Global Optimization. Technical Report EVOCINV-01-2003, Evolutionary Computation Group at CINVESTAV, Sección de Computación, Departamento de Ingeniería Eléctrica, CINVESTAV-IPN, México D.F., México, 2003. Available in the Constraint Handling Techniques in Evolutionary Algorithms Repository at http://www.cs.cinvestav.mx/˜constraint/.

[21] Efrén Mezura-Montes and Carlos A. Coello Coello. A Simple Evolution Strategy to Solve Constrained Optimization problems. In *Proceedings of Genetic and Evolutionary Computation Conference(GECCO'2003)*, pages 641–642, Heidelberg, Germany, July 2003. Chicago, Illinois, USA, Springer-Verlag. Lecture Notes in Computer Science No. 2723.

[22] Zbigniew Michalewicz and Marc Schoenauer. Evolutionary Algorithms for Constrained Parameter Optimization Problems. *Evolutionary Computation*, 4(1):1–32, 1996.

[23] Kenneth V. Price. An Introduction to Differential Evolution. In David Corne, Marco Dorigo, and Fred Glover, editors, *New Ideas in Optimization*, pages 79–108. Mc Graw-Hill, UK, 1999.

[24] Jon T. Richardson, Mark R. Palmer, Gunar Liepins, and Mike Hilliard. Some Guidelines for Genetic Algorithms with Penalty Functions. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms (ICGA-89)*, pages 191–197, San Mateo, California, June 1989. George Mason University, Morgan Kaufmann Publishers.

[25] Thomas P. Runarsson and Xin Yao. Stochastic Ranking for Constrained Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 4(3):284–294, September 2000.

[26] Hans Paul Schwefel. *Evolution and Optimal Seeking*. John Wiley & Sons Inc., New York, 1995.

[27] Frank Schweitzer, Werner Ebeling, Helge Rosé, and Olaf Weiss. Optimization of road networks using evolutionary strategies. *Evolutionary Computation*, 5(4):419–438, 1998.

[28] Alice E. Smith and David W. Coit. Constraint Handling Techniques—Penalty Functions. In Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors, *Handbook of Evolutionary Computation*, chapter C 5.2. Oxford University Press and Institute of Physics Publishing, 1997.