Adding a Diversity Mechanism to a Simple Evolution Strategy to Solve Constrained Optimization Problems

Efrén Mezura Montes & Carlos A. Coello Coello
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
emezura@computacion.cs.cinvestav.mx
ccoello@cs.cinvestav.mx

Abstract- In this paper, we propose the use of a Simple Evolution Strategy (SES) (i.e., a $(1 + \lambda)$-ES with self-adaptation that uses three tournament rules based on feasibility) coupled with a diversity mechanism to solve constrained optimization problems. The proposed mechanism is based on multiobjective optimization concepts taken from an approach called the Niched-Pareto Genetic Algorithm (NPGA). The main advantages of the proposed approach is that it does not require the definition of any extra parameters, other than those required by an evolution strategy. The performance of the proposed approach is shown to be highly competitive with respect to other constraint-handling techniques representative of the state-of-the-art in the area when using a well-known benchmark.

1 Introduction

Evolutionary Algorithms (EAs) are heuristic methods that have been successfully applied to a wide set of application domains [10], both, in global (single-objective) and in multiobjective optimization. Nevertheless, EAs are an unconstrained search technique and lack an explicit mechanism to deal with constrained search spaces. This has motivated the development of a considerable number of approaches to incorporate constraints into the fitness function of an EA [11, 1].

The most common approach adopted to deal with constrained search spaces is the use of penalty functions [12]. 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 [15, 1].

Evolution Strategies (ES) have been found not only efficient in solving a wide variety of optimization problems [4], but also have a strong theoretical background [14]. Motivated by the fact that some of the most recent (and most competitive) approaches to incorporate constraints into an evolutionary algorithm use an ES (see for example [13, 5]), Mezura & Coello [9] proposed a Simple Evolution Strategy (SES) to solve constrained problems. This approach is based on a double mechanism: (1) the original self-adaptation mechanism of

the ES to sample the search space and (2) a selection mechanism that prefers solutions based on feasibility to choose the new starting points for the search. However, our SES presented some drawbacks that are improved in the current work. Our main improvement consists of a diversity mechanism that allows the best of the λ individuals (based on feasibility and objective function value) in the current generation (and not only the child formed by all the parents) to replace the current solution. We also allow the parent with the best objective function value to replace the current solution regardless of its feasibility. This mechanism is controlled by a stochastic parameter called Selection Ration (S_r). This parameter indicates the number of times (as a percentage) that normal selection (i.e., deterministic between the current solution and the child formed by all the parents) will take place. The remaining $1 - S_r$ times, the best parent or the individual with the best value of the objective function will replace the current solution. In the current work, equality constraints are transformed into inequality constraints using a dynamic parameterless tolerance.

This paper is organized as follows: in Section 2 a short survey of constraint-handling techniques similar to our own is presented. Section 3 provides a detailed description of the proposed approach. Section 4, describes the test functions used and presents the results obtained. Section 5 provides a discussion of results. Finally, Section 6 provides our conclusions and some possible paths for future research.

2 Previous Work

The use of tournament selection based on feasibility rules has been explored by other authors. Jiménez and Verdegay [7] 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 [3] used the same tournament rules previously indicated in his approach. However, Deb proposed to use niching as a diversity mechanism, which introduces some extra computational time (niches are an $O(N^2)$ procedure). 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 [2] used tournament selection based on feasibility rules. They also adopted non-dominance checkings using a sample of the population (as the multiobjective optimization approach called NPGA [6]). They adopted a user-defined parameter S_r, 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 [2].

The three approaches discussed before are based on a genetic algorithm. However, several of the most competitive constraint-handling approaches known to date are based on an evolution strategy (e.g., Stochastic Ranking [13] and the Adaptive Segregated Constraint Handling Evolutionary Algorithm (ASCHEA) [5]).

3 Description of the Approach

The motivation of this work was twofold:

1. We hypothesized that the use of an evolution strategy for constrained optimization would be beneficial (with respect to the use of a genetic algorithm).

2. We were also aware 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.
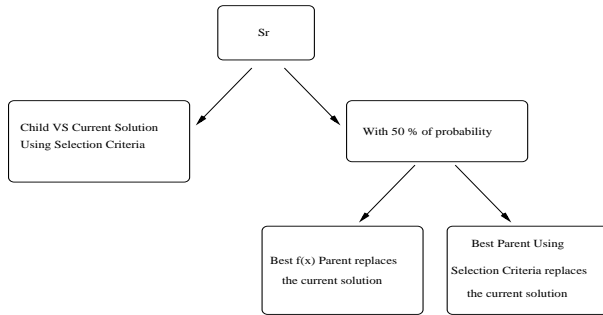

Figure 1: Diagram that illustrates the diversity mechanism implemented for our SES.

In [9], we introduced the so-called Simple Evolution Strategy (SES), which is based on two mechanisms:

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

- The use of tournaments based on feasibility, which are adopted to guide the search in such a way that the global optimum can be approached efficiently.

The three simple selection criteria used in our tournaments are the following (binary tournaments are adopted):

- Between 2 feasible solutions, the one with the highest fitness value wins.

- If one solution is feasible and the other one is infeasible, the feasible solution wins.

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

Our SES uses the $1/5$-success rule for self-adapting the σ value of our ES. By using just one σ value and one fitness function evaluation per generation, the resulting computational cost (per generation) of our approach is very low.

This first version of our approach provided very competitive results [9]. However, it presented premature convergence in some problems due to the high selection pressure caused by the tournaments performed. This motivated the changes introduced in this paper.

The modifications that we have made to the original algorithm are the following:

- The selection process was modified in order to allow either infeasible solutions with a good value of the objective function or the best parent (based on the selection criteria) to replace the current solution (in the last version only the child generated by all the parents could replace the current solution).

- This modified selection process is controlled by a parameter (that is not defined by the user) called Selection Ratio (S_r). This parameter was introduced in [2] and it 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 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 (see Figure 1).

- The S_r parameter is adapted online using the fitness value of the current solution during an interval of time (number of generations). The "mean deviation" (M_d) of the current solution over a certain number of generations is calculated in order to know how different has been the current solution. All the fitnesses are normalized in order to obtain a value between 0 and 1. The expression to adapt the S_r value is the following:

$$S_r(t) = \begin{cases} S_r(t - \text{interval})/1.001 & \text{if } M_d < 0.1 \\ S_r(t - \text{interval}) * 1.001 & \text{if } M_d > 0.2 \\ S_r(t - \text{interval}) & \text{if } 0.1 \leq M_d \leq 0.2 \end{cases}$$

(1)

where *interval* is defined as a percentage of the maximum number of generations. For example if the inter-

Problem	eevu+1					
	Optimal	Best	Mean	Median	Worst	St. Dev.
TF1	-15.000000	-15.000000	-15.000000	-15.000000	-15.000000	0.000000
TF2	0.803619	0.803569	0.769612	0.782466	0.702322	0.027547
TF3	1.000000	1.004329	1.003563	1.003669	1.002604	0.000423
TF4	-30665.539000	-30665.539062	-30665.539062	-30665.539062	-30665.539062	0.000000
TF5	-6961.814000	-6961.813965	-6961.813965	-6961.813965	-6961.813965	0.000000
TF6	24.306000	24.313972	24.418837	24.419113	24.560797	0.071159
TF7	0.095825	0.095826	0.095826	0.095826	0.095473	0.000104
TF8	680.630000	680.669189	680.809829	680.797882	681.199646	0.122624
TF9	7049.330700	7057.044434	10771.416895	10935.448730	16375.266602	2524.075298
TF10	0.750000	0.749018	0.749179	0.749062	0.750647	0.000316
TF11	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000
TF12	0.053950	0.053964	0.264135	0.438692	0.544346	0.206214

Table 1: Statistical results obtained of the 30 runs performed using the new SES with the 12 test problems adopted.

val is defined as 0.05 and the number of generations is 100, the update process will take place at every 5 generations. As can be seen, S_r will be decreased if the current solution has not significantly changed during the given interval (i.e., $M_d < 0.1$) allowing a parent (which may be infeasible) with a good fitness value to replace the current solution. This is meant to increase diversity. On the other hand, S_r will increase if the solution has been significantly different (i.e., $M_d > 0.2$) during the interval, thus favoring deterministic selection to impel convergence. S_r will keep its current value if the variation of the current solution in the interval has been moderated (i.e., $0.1 \leq M_d \leq 02$).

- In order to always keep the best solution found during the process a superelitist mechanism is included. Its only goal is to keep the best feasible solution found. This is required because the diversity mechanism adopted makes the current solution to be replaced by another solution which is not necessarily better and may be infeasible. Its implementation does not add any significant extra computational or storage cost to the algorithm.

- To deal with equality constraints, a parameterless dynamic mechanism similar to that used in ASCHEA [5] is adopted. The tolerance value ϵ is decreased with respect to the current generation using the following expression:

$$\epsilon_j(t + 1) = \epsilon_j(t)/1.000001 \qquad (2)$$

It is worth emphasizing that allowing the parents to replace the current solution in a $(1 + \lambda)$-ES (used by our SES) enhances the search power (global and local) of the ES because it explores more deeply the regions that surround the current solution. As the σ value is high early in the process, it will generate solutions (parents) on far regions of the search space (global search). This will allow to sample better the search space and to select the most promising point. The same behavior will occur later in the process, but in this case the points close to the current solution will be the more deeply explored (local search). A graphical explanation of this behavior is provided in Figure 2.

Another aspect to note is that in order to maintain the low cost of our SES, the parents are only evaluated if some of them are to replace the current solution. In the same way,
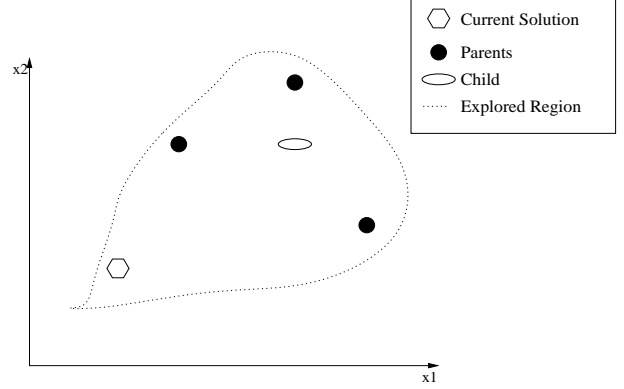

Figure 2: Diagram that illustrates the explored region of the search space in the new version of our SES. In the old version only the points in white (child and current solution) could be selected.

the child will be evaluated only if it is to compete against the current solution. The pseudo-code of this approach is shown in Figure 3.

4 Experiments and Result

To evaluate the performance of the new approach we used 12 of the test functions described in [13]. The test functions chosen contain characteristics that are representative of what can be considered "difficult" global optimization problems for an evolutionary algorithm. Their expressions are provided next.

- **TF1**:
  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:

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

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.

- **TF2**:

  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:

  $$g_1(\vec{x}) = 0.75 - \prod_{i=1}^{n} x_i \leq 0$$

  $$g_2(\vec{x}) = \sum_{i=1}^{n} x_i - 7.5n \leq 0 \qquad (3)$$

  where $n = 20$ and $0 \leq x_i \leq 10$ $(i = 1, \ldots, n)$. The global maximum is unknown; the best reported solution is [13] $f(x^*) = 0.803619$. Constraint g_1 is close to being active $(g_1 = -10^{-8})$.

- **TF3**:

  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 $f(x^*) = 1$.

- **TF4**:

  Minimize: $f(\vec{x}) = 5.3578547 x_3^2 + 0.8356891 x_1 x_5 + 37.293239 x_1 - 40792.141$
  subject to:
  $g_1(\vec{x}) = 85.334407 + 0.0056858 x_2 x_5 + 0.0006262 x_1 x_4 - 0.0022053 x_3 x_5 - 92 \leq 0$
  $g_2(\vec{x}) = -85.334407 - 0.0056858 x_2 x_5 - 0.0006262 x_1 x_4 + 0.0022053 x_3 x_5 \leq 0$
  $g_3(\vec{x}) = 80.51249 + 0.0071317 x_2 x_5 + 0.0029955 x_1 x_2 + 0.0021813 x_3^2 - 110 \leq 0$
  $g_4(\vec{x}) = -80.51249 - 0.0071317 x_2 x_5 - 0.0029955 x_1 x_2 - 0.0021813 x_3^2 + 90 \leq 0$
  $g_5(\vec{x}) = 9.300961 + 0.0047026 x_3 x_5 + 0.0012547 x_1 x_3 + 0.0019085 x_3 x_4 - 25 \leq 0$
  $g_6(\vec{x}) = -9.300961 - 0.0047026 x_3 x_5 - 0.0012547 x_1 x_3 - 0.0019085 x_3 x_4 + 20 \leq 0$

  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.

- **TF5**

  Minimize: $f(\vec{x}) = (x_1 - 10)^3 + (x_2 - 20)^3$
  subject to:

$g_1(\vec{x}) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0$
$g_2(\vec{x}) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0$

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

- **TF6**

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

  where $-10 \leq x_i \leq 10$ $(i = 1, \ldots, 10)$. 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.

- **TF7**

  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:
  $g_1(\vec{x}) = x_1^2 - x_2 + 1 \leq 0$
  $g_2(\vec{x}) = 1 - x_1 + (x_2 - 4)^2 \leq 0$

  where $0 \leq x_1 \leq 10$ and $0 \leq x_2 \leq 10$. The optimum solution is located at $x^* = (1.2279713, 4.2453733)$ where $f(x^*) = 0.095825$. The solutions is located within the feasible region.

- **TF8**

  Minimize: $f(\vec{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10 x_5^6 + 7 x_6^2 + x_7^4 - 4 x_6 x_7 - 10 x_6 - 8 x_7$
  subject to:

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

  where $-10 \leq x_i \leq 10$ $(i = 1, \ldots, 7)$. The global optimum is $x^* = (2.330499, 1.951372, -0.4775414, 4.365726, -0.6244870, 1.038131, 1.594227, 9.828726, 8.280092, 8.375927)$ where $f(x^*) = 680.6300573$. Two constraints are active (g_1 and g_4).

- **TF9**

  Minimize: $f(\vec{x}) = x_1 + x_2 + x_3$

subject to: $g_1(\vec{x}) = -1 + 0.0025(x_4 + x_6) \leq 0$

$g_2(\vec{x}) = -1 + 0.0025(x_5 + x_7 - x_4) \leq 0$

$g_3(\vec{x}) = -1 + 0.01(x_8 - x_5) \leq 0$

$g_4(\vec{x}) = -x_1 x_6 + 833.33252 x_4 + 100 x_1 - 83333.333 \leq 0$

$g_5(\vec{x}) = -x_2 x_7 + 1250 x_5 + x_2 x_4 - 1250 x_4 \leq 0$

$g_6(\vec{x}) = -x_3 x_8 + 1250000 + x_3 x_5 - 2500 x_5 \leq 0$

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.

- **TF10**

  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 $f(x^*) = 0.75$.

- **TF11**

  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 (4) holds. The global optimum is located at $x^* = (5, 5, 5)$ where $f(x^*) = 1$.

- **TF12**

  Minimize: $f(\vec{x}) = e^{x_1 x_2 x_3 x_4 x_5}$

  subject to:

  $g_1(\vec{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$

  $g_2(\vec{x}) = x_2 x_3 - 5 x_4 x_5 = 0$

  $g_3(\vec{x}) = x_1^3 + x_2^3 + 1 = 0$

  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 [8]) was computed using the following expression:

$$\rho = |F|/|S| \qquad (4)$$

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 3, 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

Problem	n	Type of function	ρ	LI	NI	LE	NE
TF1	13	quadratic	0.0003%	9	0	0	0
TF2	20	nonlinear	99.9973%	2	0	0	0
TF3	10	nonlinear	0.0026%	0	0	0	1
TF4	5	quadratic	27.0079%	4	2	0	0
TF5	2	nonlinear	0.0057%	0	2	0	0
TF6	10	quadratic	0.0000%	3	5	0	0
TF7	2	nonlinear	0.8581%	0	2	0	0
TF8	7	nonlinear	0.5199%	0	4	0	0
TF9	8	linear	0.0020%	6	0	0	0
TF10	2	quadratic	0.0973%	0	0	0	1
TF11	3	quadratic	4.7697%	0	9^3	0	0
TF12	5	nonlinear	0.0000%	0	0	1	2

Table 3: Values of ρ for the 12 test functions chosen.

and NE is the number of nonlinear equalities.

The parameters used in the experiments are the following (30 runs were performed for each problem): the total number of fitness function evaluations was set to 330,000. Equality constraints were transformed into inequalities using an initial tolerance value of 0.001. The initial values for the $(1 + \lambda)$-ES parameters were: $\sigma = 4.0$, $C = 0.99$, $\lambda = 3$, and maximum number of generations = 275,000. The interval of the S_r updates was almost negligible (0.9). This means that the update will take place until generation 247,500. We anticipated that our approach would not be too sensitive to the S_r parameter and our experiments confirmed this hypothesis. The statistical results are presented in Table 1.

5 Discussion of Results

In Tables 2, 4, 5 and 6, we compare the new SES against the last (old) SES, the homomorphous maps [8], stochastic ranking [13] and ASCHEA [5], respectively. Such approaches were selected for comparison because they are representative of the state-of-the-art in the area. There are several issues derived from this comparison that deserve some discussion:

- The new SES was able to converge to the global optimum in 7 of the test 12 functions used (**TF1**, **TF3**, **TF4**, **TF5**, **TF7**, **TF10** and **TF11**), and it was able to converge very close to the optimum in **TF2**, **TF6**, **TF8**, **TF9** and **TF12**.

- With respect to the last version of the SES (Table 2), this New SES improved the robustness of the results in problems **TF1**, **TF2**, **TF4**, **TF6** and **TF10**. Also, the quality of the results was improved in problems **TF9** and **TF12**.

- With respect to the homomorphous maps (see Table 4), we can see that our SES converged to a better "best" solution in 9 problems (**TF1**, **TF2**, **TF3**, **TF4**, **TF5**, **TF6**, **TF8**, **TF9** and **TF11**). Also, it found a better average and a better "worst" solution in 8 problems (**TF1**, **TF3**, **TF4**, **TF5**, **TF6**, **TF7**, **TF8**, and **TF11**). Thus, it should be clear that our SES had a highly competitive performance, even improving the results of the homomorphous maps in several test functions. No comparison was made with problem **TF12** because such results were not available.

- With respect to stochastic ranking (see Table 5), the

Problem	Optimal	Best Result		Mean Result		Worst Result	
		NEW	OLD	NEW	OLD	NEW	OLD
TF1	-15.000000	-15.000000	-15.000000	-15.000000	-14.848614	-15.000000	-12.999997
TF2	0.803619	0.803569	0.793083	0.769612	0.698932	0.702322	0.576079
TF3	1.000000	1.004329	1.000497	1.003563	1.000491	1.002604	1.000424
TF4	-30665.539000	-30665.539062	-30665.539062	-30665.539062	-30665.539062	-30665.539062	-30663.496094
TF5	-6961.814000	-6961.813965	-6961.813965	-6961.813965	-6961.813965	-6961.813965	-6961.813965
TF6	24.306000	24.313972	24.368050	24.418837	24.702525	24.560797	25.516653
TF7	0.095825	0.095826	0.095826	0.095784	0.095826	0.095473	0.095826
TF8	680.630000	680.669189	680.631653	680.809829	680.673645	681.199646	680.915100
TF9	7049.330700	7057.044434	-	10771.416895	-	16375.266602	-
TF10	0.750000	0.749018	0.749900	0.749179	0.784395	0.750647	0.879522
TF11	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
TF12	0.053950	0.053964	-	0.264135	-	0.544346	-

Table 2: Comparison of results between the new SES and the old SES proposed in [9]. "-" means no feasible solutions were found.

Problem	Optimal	Best Result		Mean Result		Worst Result	
		SES	HM	SES	HM	SES	HM
TF1	-15.000000	-15.000000	-14.7886	-15.000000	-14.7082	-15.000000	-14.6154
TF2	0.803619	0.803569	0.79953	0.769612	0.79671	0.702322	0.79119
TF3	1.000000	1.004329	0.9997	1.003563	0.9989	1.002604	0.9978
TF4	-30665.539000	-30665.539062	-30664.5	-30665.539062	-30655.3	-30665.539062	-30645.9
TF5	-6961.814000	-6961.813965	-6952.1	-6961.813965	-6342.6	-6961.813965	-5473.9
TF6	24.306000	24.313972	24.620	24.418837	24.826	24.560797	25.069
TF7	0.095825	0.095826	0.0958250	0.095784	0.0891568	0.095473	0.0291438
TF8	680.630000	680.669189	680.91	680.809829	681.16	681.199646	683.18
TF9	7049.330700	7057.044434	7147.9	10771.416895	8163.6	16375.266602	9659.3
TF10	0.750000	0.749018	0.75	0.749179	0.75	0.750647	0.75
TF11	1.000000	1.000000	0.999999857	1.000000	0.999134613	1.000000	0.991950498
TF12	0.053950	0.053964	NA	0.264135	NA	0.544346	NA

Table 4: Comparison of results between our approach (SES) and the Homomorphous Maps (HM) [8] NA = Not Available.

new SES was able to converge to similar "best" solutions in 10 problems (**TF1**, **TF3**, **TF4**, **TF5**, **TF6**,**TF7**, **TF8**, **TF10**, **TF11** and **TF12**). SES found a slightly better result in problem **TF2**. Moreover, SES found a similar average solution in 8 problems (**TF1**, **TF3**, **TF4**, **TF6**, **TF7**,**TF8**, **TF10**, and **TF11**). SES found a better average solution in problem **TF5**. Finally, SES found similar "worst" solutions in 5 problems (**TF1**, **TF3**, **TF4**, **TF6** and **TF11**) and reached a better "worst" individual in problem **TF5**. Though the Stochastic Ranking performed with a little more consistency, SES performed almost at a similar level than this highly competitive approach.

- Finally, with respect to ASCHEA (see Table 6), the new SES converged to similar "best" results in 7 problems (**TF1**, **TF3**, **TF4**, **TF5**, **TF7**,**TF8** and **TF10**) and it found better "best"results in 3 problems (**TF2**, **TF6** and **TF9**). A better average result was found by the SES in 4 problems (**TF1**, **TF2**, **TF3** and **TF6**), and a similar average solution was reached in 4 problems (**TF4**, **TF5**, **TF8** and **TF10**). Note that we didn't present any results for ASCHEA in problems **TF11** and **TF12** because such results were not available.

From the previous comparison, we can see that the new SES produced very competitive results with respect to three techniques representative of the state-of-the-art in constrained optimization. The new SES can deal with highly constrained problems, problems with low (TF5 and TF7) and high (TF1, TF2, TF3, TF6) dimensionality, with different types of combined constraints (linear, nonlinear, equality and inequality) and with very large (TF2) or very small (TF12) or even disjoint (TF11) feasible regions. However, our approach presented some robustness problems in TF9. This function has a very large search space (because of the intervals of the decision variables). Therefore, our SES required to explore several regions of the search space in order to find promising areas consistently. This can be obtained more easily by using a population-based ES (e.g., a $(\mu + \lambda)$-ES).

Besides still being a very simple approach, it is worth reminding that SES does not require any extra parameters (besides those used with an evolution strategy) because the S_r is adapted online. In contrast, the homomorphous maps require an additional parameter (called v) which has to be found empirically [8]. Stochastic ranking requires the definition of a parameter called P_f, whose value has an important impact on the performance of the approach [13]. 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 [5].

Measuring the computational cost, 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 330,000 FFE. Stochastic ranking performed 350,000 FFE, the homomorphous maps performed 1,400,000 FFE, and ASCHEA performed 1,500,000 FFE.

It is interesting to note that with a very small population size (only 3 individuals), the search power is improved when we allow to the parents to replace (in a deterministic way) the current solution regardless of their feasibility.

6 Conclusions and Future Work

We have introduced the addition of a diversity maintenance mechanism to a simple evolution strategy previously proposed to solve constrained optimization problems [9]. Such a

```
Begin
    t=0, S_r=0.9, interval=0.9
    Create a random initial solution x^0 and store it as the superelitist solution x^S
    Evaluate f(x^0)
    For t=1 to MAX_GENERATIONS Do
        Produce λ mutations of x^(t-1) using:
            x_i^j = x_i^{t-1} + σ[t] · N_i(0,1) ∀i ∈ n, j = 1,2,...,λ
        Generate one child x^c by the combination of the λ mutations using
            m=randint(1,λ)
            x_i^c = x_i^m, ∀i ∈ n
        If (flip(S_r)) Then
            Evaluate f(x^c)
            Apply comparison criteria to select the best individual x^t between x^(t-1) and x^c
        else
            Evaluate all the λ mutations obtained
            If (flip(0.5)) Then
                x^t = parent with best objective function value regardless of feasibility
            else
                x^t = best parent based on the comparison criteria
            endif
        endif
        If (x^t better than x^S) (using selection criteria) Then
            x^S = x^t
        endif
        t = t + 1
        If (t mod n = 0) Then
                     ⎧ σ[t-n]/c    if p_s > 1/5
            σ[t] =   ⎨ σ[t-n]·c    if p_s < 1/5
                     ⎩ σ[t-n]      if p_s = 1/5
        End If
        If (t mod interval = 0) Then
            Calculate the mean deviation (M_d) of the current solutions in the interval
            If (M_d < 0.1) Then
                S_r(t) = S_r(t - interval)/1.001
            else
                If (M_d > 0.2) Then
                    S_r(t) = S_r(t - interval)/1.001
                End If
            End If
        End If
        ε_i(t) = ε_i(t-1)/1.000001 (only for each i equality constraint)
    End For
End
```

Figure 3: Our algorithm of SES (n is the number of decision variables of the problem, $flip(P)$ is a function that returns TRUE with probability P).

mechanism was found to improve the quality and robustness of our approach.

We also introduced the use of an online adaptive mechanism that automatically updates the value of the S_r parameter (responsible for controlling diversity in the population). As its predecessor, the new SES does not require a penalty function or any extra parameters (other than the original parameters of an evolution strategy) to bias the search towards the feasible region of a problem. Additionally, this improved approach has a low computational cost and it is easy to implement.

Our main path for future research is to explore in more detail the impact of the S_r parameter on the performance of the approach. We also intend to use our SES to solve some engineering (real-world) optimization problems.

Bibliography

[1] C. 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.

Problem	Optimal	Best Result		Mean Result		Worst Result	
		SES	SR	SES	SR	SES	SR
TF1	−15.000000	−15.000000	−15.000	−15.000000	−15.000	−15.000000	−15.000
TF2	0.803619	0.803569	0.803515	0.769612	0.781975	0.702322	0.726288
TF3	1.000000	1.004329	1.000	1.003563	1.000	1.002604	1.000
TF4	−30665.539000	−30665.539062	−30665.539	−30665.539062	−30665.539	−30665.539062	−30665.539
TF5	−6961.814000	−6961.813965	−6961.814	−6961.813965	−6875.940	−6961.813965	−6350.262
TF6	24.306000	24.313972	24.307	24.418837	24.374	24.560797	24.642
TF7	0.095825	0.095826	0.095825	0.095784	0.095825	0.095473	0.095825
TF8	680.630000	680.669189	680.630	680.809829	680.656	681.199646	680.763
TF9	7049.330700	7057.044434	7054.316	10771.416895	7559.192	16375.266602	8835.655
TF10	0.750000	0.749018	0.750	0.749179	0.750	0.750647	0.750
TF11	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
TF12	0.053950	0.053964	0.053957	0.264135	0.057006	0.544346	0.216915

Table 5: Comparison of results between our approach (SES) and Stochastic Ranking (SR) [13].

Problem	Optimal	Best Result		Mean Result		Worst Result	
		SES	ASCHEA	SES	ASCHEA	SES	ASCHEA
TF1	−15.000000	−15.000000	−15.0	−15.000000	−14.84	−15.000000	NA
TF2	0.803619	0.803569	0.785	0.769612	0.59	0.702322	NA
TF3	1.000000	1.004329	1.0	1.003563	0.99989	1.002604	NA
TF4	−30665.539000	−30665.539062	30665.5	−30665.539062	30665.5	−30665.539062	NA
TF5	−6961.814000	−6961.813965	−6961.81	−6961.813965	−6961.81	−6961.813965	NA
TF6	24.306000	24.313972	24.3323	24.418837	24.66	24.560797	NA
TF7	0.095825	0.095826	0.095825	0.095784	0.095825	0.095473	NA
TF8	680.630000	680.669189	680.630	680.809829	680.641	681.199646	NA
TF9	7049.330700	7057.044434	7061.13	10771.416895	7193.11	16375.266602	NA
TF10	0.750000	0.749018	0.75	0.749179	0.75	0.750647	NA
TF11	1.000000	1.000000	NA	1.000000	NA	1.000000	NA
TF12	0.053950	0.053964	NA	0.264135	NA	0.544346	NA

Table 6: Comparison of results between our approach (SES) and ASCHEA [5]. NA = Not Available.

[2] C. A. Coello Coello and E. Mezura-Montes. Handling Constraints in Genetic Algorithms Using Dominance-Based Tournaments. In I. 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.

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

[4] M. Gorges-Schleuter, I. Sieber, and W. Jakob. Local interaction evolution strategies for design optimization. In *1999 Congress on Evolutionary Computation*, pages 2167–2174, Piscataway, NJ, 1999. IEEE Service Center.

[5] S. B. Hamida and M. 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.

[6] J. Horn, N. Nafpliotis, and D. 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.

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

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

[9] E. Mezura-Montes and C. A. Coello Coello. A Simple Evolution Strategy to Solve Constrained Optimization problems. In *Proceedings of Genetic and Evolutionary Computation Conference(GECCO'2003)*, Heidelberg, Germany, July 2003. Chicago, Illinois, USA, Springer-Verlag. (Accepted for publication).

[10] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, third edition, 1996.

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

[12] J. T. Richardson, M. R. Palmer, G. Liepins, and M. Hilliard. Some Guidelines for Genetic Algorithms with Penalty Functions. In J. D. 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.

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

[14] H. P. Schwefel. *Evolution and Optimal Seeking*. John Wiley & Sons Inc., New York, 1995.

[15] A. E. Smith and D. W. Coit. Constraint Handling Techniques—Penalty Functions. In T. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, chapter C 5.2. Oxford University Press and Institute of Physics Publishing, 1997.