A Simple Multimembered Evolution Strategy to Solve Constrained Optimization Problems

Efrén Mezura-Montes and Carlos A. Coello Coello, *Senior Member, IEEE*

***Abstract*—This paper presents a simple multimembered evolution strategy to solve global nonlinear optimization problems. The approach does not require the use of a penalty function. 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. A simple feasibility-based comparison mechanism is used to guide the process toward the feasible region of the search space. Also, the initial stepsize of the evolution strategy is reduced in order to perform a finer search and a combined (discrete/intermediate) panmictic 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 cost required by the other techniques compared.**

***Index Terms*—Constrained optimization, multimembered evolution strategy, nonlinear optimization, panmictic recombination technique.**

I. Introduction

EVOLUTIONARY algorithms (EAs) have been widely used to solve several types of optimization problems [1], [8], [10], [11]. Nevertheless, they are unconstrained search techniques and lack an explicit mechanism to bias the search in constrained search spaces. This has motivated the development of a considerable number of approaches to incorporate constraints into the fitness function of an EA [6], [25].

The most common approach adopted to deal with constrained search spaces is the use of penalty functions [28]. 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 [6], [32].

Evolution strategies (ES) have been found not only efficient in solving a wide variety of optimization problems [1], [2], [4], [12], [31], but also have a strong theoretical background [3], [5], [30].

Our approach uses the self-adaptive mutation mechanism of a multimembered evolution strategy to explore constrained search spaces. This is combined with a comparison mechanism which uses three feasibility-based rules to guide the search toward 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 50% probability either from the parents or the offspring population. A hybrid panmictic recombination 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 II, we define the global nonlinear optimization problem that we aim to solve. After that, in Section III a description of previous approaches based on similar ideas is provided. Section IV presents a detailed description of our approach. Then, in Section V, we present the experimental design and show the obtained results which are discussed in Section VI. Section VII provides an experimental study that aims to identify the mechanism that is mainly responsible for the effectiveness of our proposed approach. The rate at which our approach reaches the feasible region (in the test functions adopted) is analyzed in Section VIII. In Section IX, some conclusions are established. Finally, some possible paths for future research are provided in Section X.

II. Statement of the Problem

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

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

Manuscript received August 28, 2003; revised May 6, 2004. The work of E. Mezura-Montes was supported in part by the Mexican Consejo Nacional de Ciencia y Tecnología (CONACyT) through a scholarship to pursue graduate studies in the Electrical Engineering Department, CINVESTAV-IPN. The work of C. A. Coello Coello was supported in part by CONACyT under Project 42 435-Y.

The authors are with the Evolutionary Computation Group (EVOCINV), Computer Science Section, Electrical Engineering Department, CINVESTAV-IPN, México City 07300, México (e-mail: emezura@computacion.cs.cinvestav.mx; ccoello@cs.cinvestav.mx).

Digital Object Identifier 10.1109/TEVC.2004.836819

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 $\mathcal{F}$.

III. Previous Work

The inspiration of our approach was motivated by the idea of exploring the capabilities of multiobjective optimization concepts to solve global optimization problems. We compared four representative approaches using the same test functions adopted in this paper [18], [21]. 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) [23].

Motivated by the fact that the most recent and competitive approaches to solve constrained optimization problems are based on an ES (e.g., stochastic ranking (SR) [29] and the adaptive segregational constraint handling evolutionary algorithm (ASCHEA) [13]), 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 addition of feasibility rules 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) [19]. We also implemented a variation of a $(\mu + 1)$-ES with the "1/5 successful rule" to adapt on-line the sigma value [19]. Constraints were handled using rules based on feasibility (see Section IV for details).

The use of rules based on feasibility has been explored in the past by other authors. Jiménez and Verdegay [15] 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.

Powell and Skolnick [26] proposed to map feasible solutions into the interval $(-\infty, 1)$, and infeasible solutions into the interval $(1, \infty)$. The aim is to consider feasible solutions always superior to infeasible ones.

Individuals are evaluated using [26]

$$\text{fit}(\vec{x}) = \begin{cases} f(\vec{x}), & \text{if feasible} \\ 1 + r\left(\sum_{i=1}^{n} g_i(\vec{x}) + \sum_{j=1}^{p} h_j(\vec{x})\right), & \text{otherwise} \end{cases}. \tag{4}$$

$f(\vec{x})$ is scaled into the interval $(-\infty, 1)$, $g_i(\vec{x})$ and $h_j(\vec{x})$ are scaled into the interval $(1, \infty)$, and r is a constant. Powell and Skolnick [26] used linear ranking selection in order to get a slower convergence.

Deb [9] proposed the selection criteria adopted in our work as tournament rules to select individuals using a genetic algorithm (GA). In his approach, the following expression is used to assign fitness to a solution:

$$\text{fit}_i(\vec{x}) = \begin{cases} f_i(\vec{x}), & \text{if feasible} \\ f_{\text{worst}} + \sum_{j=1}^{n} g_j(\vec{x}), & \text{otherwise} \end{cases} \tag{5}$$

where f_{worst} is the objective function value of the worst feasible solution. However, Deb proposed to use niching as a diversity mechanism, which introduces some extra computational time (niches are an $O(N^2)$ procedure). As in Powell and Skolnick's algorithm [26], 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 and 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 niched Pareto genetic algorithm (NPGA) [14]). 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 the variation of a $(\mu + 1)$-ES [19] in which one child created from μ mutations of the current solution competes against it and the best one is selected as the new current solution. The details of this approach are shown in Fig. 1.

However, the approach presented premature convergence in some test functions [19]. A $(1 + \lambda)$-ES was proposed in [20], 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 and Mezura [7] and mentioned above) is adopted. S_r refers to the percentage of selections that will be performed in a deterministic way (as used in the original version of our ES where the child replaces the current solution using the feasibility-based comparison mechanism). 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 (using again the feasibility-based comparison mechanism) will

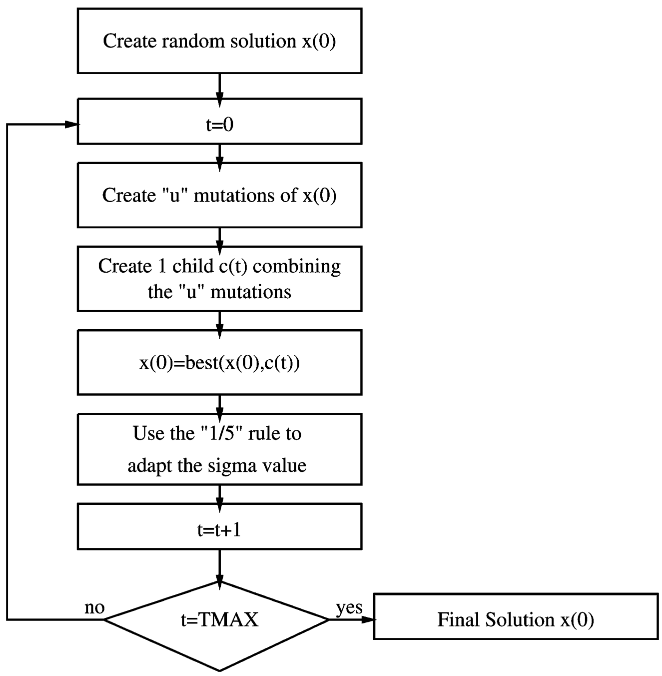

Fig. 1. Algorithm of the variation of a $(\mu + 1)$-ES used in the first version of the approach.

replace the current solution. Both options are given 50% probability each.

The $(1 + \lambda)$-ES approach proposed in [20] 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 single-membered ES lack the explorative power to allow them sample large 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.

IV. Our Approach

Our new approach is based on the same concepts that its predecessors discussed in Section III: (1) the self-adaptation mechanism of an ES and (2) a comparison mechanism based on the following criteria.

1) Between two 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 panmictic recombination.

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 the best infeasible solution) can be chosen either from the parents or the offspring population, with 50% probability. This process of allowing this solution to survive for the next generation happens three 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 boundary of the feasible region will allow the ES to reach global optimum solutions located precisely on 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 three times for every 100 attempts. This works in the following way: When the deterministic replacement is used to form the population for the next generation in an ES, the best individuals from among the parents and offspring are selected using the comparison mechanism previously indicated (in a deterministic way). The 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, three times from every 100 picks, the best infeasible solution (from either the parents or the offspring population with 50% probability each) is copied in the population for the next generation. The pseudocode is listed in Fig. 2.

  Based on the empirical evidence observed in the previous version of the approach [20], where we used a population of three offspring, we decided to use a small number of copies of the best infeasible solutions for the next generation of our approach. For values larger than three, the quality and robustness of our approach tend to decrease.
- **Combined Recombination**: We use panmictic recombination, but with a combination of the discrete and intermediate recombination operators. Each gene in the chromosome can be processed with any of these two recombination operators with 50% probability. This operator is applied to both, strategy parameters (sigma values) and decision variables of the problem. The pseudocode is shown in Fig. 3. Note that we use intermediate recombination by just computing the average between the values of the variables of each parent (as originally proposed by Schwefel [30]).
- **Reduction of the Initial Stepsize of the ES**: The previous versions of our algorithm are based on a variation of a $(\mu+1)$-ES [19] and a $(1+\lambda)$-ES [20]. Thus, they do not use a population of solutions but employ the most simple scheme of an ES, where only one sigma value is used for all the decision variables. We observed that when this sigma value was close to zero, the previous approaches

function population_for_next_generation()

For i=1 to μ **Do**

If flip(0.97)

Select the best individual based on the comparison mechanism 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

Fig. 2. Pseudocode of the generation of the population for the next generation with the diversity mechanism incorporated. $flip(P)$ is a function that returns TRUE with probability P.

function combined_recombination()

Select randomly mate_1 from the parents population

For i=1 to NUMBER_OF_VARIABLES **Do**

Select randomly 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

Fig. 3. Pseudocode of the panmictic combined (discrete-intermediate) recombination operator used by our approach. $flip(P)$ is a function that returns TRUE with probability P.

were capable of reaching the global optimum, or at least improve the value of the final solution. Therefore, in our new approach based on a multimembered ES, we decided to favor finer movements in the search space. We experimented with just a percentage of the quantity obtained by the formula proposed by Schwefel [30]. We initialize the sigma values (we use one for each decision variable) for each individual in the initial population with only a 40%

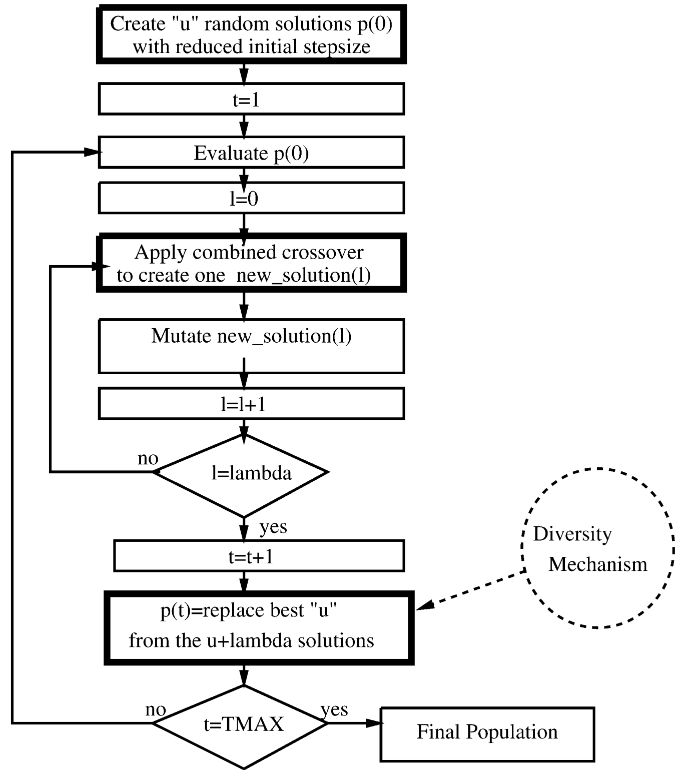

Fig. 4. Algorithm of our $(\mu + \lambda)$-ES. The thick boxes indicate the three modifications made to the original ES.

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

$$\sigma_i(0) = 0.4 \times \left(\frac{\Delta x_i}{\sqrt{n}}\right) \tag{6}$$

where Δx_i is approximated with the expression (suggested in [29]), $\Delta x_i \approx x_i^u - x_i^l$, where $x_i^u - x_i^l$ are the upper and lower bounds of the decision variable i.

Summarizing, our approach works over a simple multimembered evolution strategy (SMES): $(\mu+\lambda)$-ES. The only modifications introduced are the reduction of the initial stepsize of the sigma values, the panmictic combined (discrete-intermediate) recombination, and the changes to the original deterministic replacement of the ES (made by sorting the solutions using the comparison mechanism based on feasibility discussed at the beginning of this section), allowing the best infeasible solution, from either the parents or the offspring population, to remain in the next generation. The details of our approach are presented in Fig. 4.

Unlike Deb's [9] technique, our approach does not use niches in order to maintain diversity in the population. This is because inside the replacement process used to produce the population for the next generation, we incorporate a mechanism that allows slightly infeasible solutions with a good objective function value to be considered better than feasible ones. This ES-like replacement makes also a difference with respect to Powell and Skolnick's approach [26], which uses proportional selection (with linear ranking) on a GA-based approach.

TABLE I
VALUES OF ρ FOR THE 13 TEST PROBLEMS CHOSEN

Problem	n	Type of function	ρ	LI	NI	LE	NE
g01	13	quadratic	0.0003%	9	0	0	0
g02	20	nonlinear	99.9973%	1	1	0	0
g03	10	nonlinear	0.0026%	0	0	0	1
g04	5	quadratic	27.0079%	0	6	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%	3	3	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

V. EXPERIMENTS AND RESULTS

To evaluate the performance of the proposed approach, we used the 13 test functions described in [29]. 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 in the Appendix, at the end of the paper.

To get an estimate of how difficult is to generate feasible points through a purely random process, we computed the ρ metric (as suggested by Michalewicz and Schoenauer [25]) using the following expression:

$$\rho = \frac{|F|}{|S|} \tag{7}$$

where $|S|$ is the number of random solutions generated ($S = 1,000,000$ in our case), and $|F|$ is the number of feasible solutions found (out of the total $|S|$ solutions randomly generated).

The values of ρ for each of the functions chosen are shown in Table I, 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 [30] (where n is the number of decision variables of the problem)

$$\tau = \left(\sqrt{2\sqrt{n}}\right)^{-1} \qquad \tau' = \left(\sqrt{2n}\right)^{-1}. \tag{8}$$

The initial values for the standard deviations were calculated using (6).

For the experiments, we used the following parameters:

- $\mu = 100$;
- $\lambda = 300$;
- number of generations $= 800$;
- number of objective function evaluations $= 240000$.

The combined recombination operator explained in detail in Section IV 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 [22].

To deal with equality constraints, a dynamic mechanism originally proposed in ASCHEA [13] and used in [20] is adopted. The tolerance value ϵ is decreased with respect to the current generation using the following expression:

$$\epsilon_j(t+1) = \frac{\epsilon_j(t)}{1.001\,95}. \tag{9}$$

The initial ϵ_0 was set to 0.001. Note that the use of the value 1.001 95 in (9) causes the allowable tolerance for the equality constraints to go from 0.001 (initial value) to 0.0004 (final value) given the number of iterations adopted by our approach (if more iterations are performed, this value will tend to zero).

For problem g13, ϵ_0 was set to a much larger value (3.0), because in this case it is very difficult to generate feasible solutions during the initial generations of our approach. Thus, by using a large tolerance value, more individuals will be able to satisfy the equality constraints and will serve as reference solutions that the algorithm will improve over time. Given that this larger value is adopted, we also changed the constant decreasing value. So, instead of using 1.00195, we adopt, in this case, a value of 1.0145. Such a value causes the allowable equality constraint violation to go from 3.0 (initial value) to 0.00003 (final value) given the number of iterations adopted by our approach. Note that the final allowable tolerance is smaller in this case, despite the initial larger value. As a matter of fact, we recommend to use this second setup for the tolerance of the equality constraints in problems in which no feasible solutions can be found by our algorithm when using a small initial ϵ_0.

Additionally, for problems $g03$ and $g13$, the initial stepsize required a more dramatic decrease. They were defined as 0.01 (just a 5% instead of the 40% used for the other test functions) for $g03$ and 0.05 (2.5%) 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 these types of problems, finer movements provide a better sampling of the search space using an evolution strategy.

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

We compared our approach against three state-of-the-art approaches: the homomorphous maps (HM) [17], SR [29], and the ASCHEA [13]. The best results obtained by each approach are shown in Table III. The mean values provided are compared in Table IV and the worst results are presented in Table V. The results provided by these approaches were taken from the original references for each method.

HM performs a homomorphous mapping between an n-dimensional cube and a feasible search space (either convex or nonconvex). The main idea of this approach is to transform the original problem into another (topologically equivalent) function that is easier to optimize by an EA. HM handles two

TABLE II
STATISTICAL RESULTS OBTAINED BY OUR SMES FOR THE 13 TEST FUNCTIONS OVER 30 INDEPENDENT RUNS. A RESULT IN **BOLDFACE** INDICATES THAT THE GLOBAL OPTIMUM (OR BEST KNOWN SOLUTION) WAS REACHED

	Statistical Results of the Simple Multimembered Evolution Strategy (SMES)					
Problem	**Optimal**	**Best**	**Mean**	**Median**	**Worst**	**St. Dev.**
g01	−15.000	**−15.000**	**−15.000**	**−15.000**	**−15.000**	0
g02	0.803619	0.803601	0.785238	0.792549	0.751322	1.67E-2
g03	1.000	**1.000**	**1.000**	**1.000**	**1.000**	2.09E-4
g04	−30665.539	**−30665.539**	**−30665.539**	**−30665.539**	**−30665.539**	0
g05	5126.498	5126.599	5174.492	5160.198	5304.167	50.06E+0
g06	−6961.814	**−6961.814**	−6961.284	**−6961.814**	−6952.482	1.85E+0
g07	24.306	24.327	24.475	24.426	24.843	1.32E-1
g08	0.095825	**0.095825**	**0.095825**	**0.095825**	**0.095825**	0
g09	680.63	680.632	680.643	680.642	680.719	1.55E-2
g10	7049.25	7051.903	7253.047	7253.603	7638.366	136.02E+0
g11	0.75	**0.75**	**0.75**	**0.75**	**0.75**	1.52E-4
g12	1.000	**1.000**	**1.000**	**1.000**	**1.000**	0
g13	0.053950	0.053986	0.166385	0.061873	0.468294	1.77E-1

cases: convex feasible space and nonconvex feasible space. HM uses a binary-coded GA with gray codes, proportional selection without elitism and traditional crossover and mutation operators.

The aim of SR is to balance the influence of the objective function and the penalty function when assigning fitness to a solution. SR does not require the definition of a penalty factor. The selection process is based on a ranking process. Instead, a user-defined parameter called P_f sets the probability of using only the objective function to compare two solutions to sort them. Then, when the solutions are sorted using a bubble-sort like algorithm, sometimes, depending of the P_f value, the comparison between two adjacent solutions will be performed using only the objective function. The remaining comparisons will be performed using only the penalty function that consists, in this case, of the sum of constraint violation. SR uses a (30,200)-ES with global intermediate recombination applied only to the strategy parameters (not to the decision variables of the problem).

ASCHEA is based on three components: 1) an adaptive penalty function; 2) a constraint-driven recombination; and 3) a segregational selection based on feasibility. In ASCHEA's most recent version [13], the authors propose to use a penalty factor for each constraint of the problem. Also, the authors added a niching mechanism to improve the performance of the algorithm in multimodal functions. Finally, the authors added a dynamic and an adaptive scheme to decrease the tolerance value used in the transformation of equality constraints into two inequality constraints. The approach uses a (100 + 300)-ES with standard arithmetical recombination.

VI. DISCUSSION OF RESULTS

As described in Table II, 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 VI, we show the number of runs in which the global optimum (or best known solution) was reached. In addition, we show the lowest and the average generation number in which such global optimum was found. The results obtained suggest that for the problems in which the global optimum was 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 (see Tables III–V).

A. 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 HM 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.

B. 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 better "mean" and "worst" results in functions g05, g07, and g13.

TABLE III
COMPARISON OF THE BEST SOLUTIONS FOUND BY OUR SMES AGAINST THE HOMOMORPHOUS MAPS (HM), STOCHASTIC RANKING (SR), ASCHEA, OUR GA VERSION AND TWO OTHER VERSIONS OF OUR SMES: ONE THAT USES ONLY RECOMBINATION AND ANOTHER ONE THAT USES BOTH RECOMBINATION AND STEPSIZE REDUCTION. A RESULT IN **BOLDFACE** INDICATES A BETTER RESULT OR THAT THE GLOBAL OPTIMUM (OR BEST KNOWN SOLUTION) WAS REACHED. "-" MEANS THAT NO FEASIBLE SOLUTIONS WERE FOUND. NA = NOT AVAILABLE

	Comparison of the best solution obtained.							
P	**Optimal**	**HM**	**SR**	**ASCHEA**	**SMES**	**GA**	**Recomb.**	**Recomb.& Step.Reduc.**
g01	−15.000	−14.7886	**−15.000**	**−15.0**	**−15.000**	−14.440	**−15.000**	**−15.000**
g02	0.803619	0.79953	0.803515	0.785	**0.803601**	0.796231	0.803589	0.803592
g03	1.000	0.9997	**1.000**	**1.0**	**1.000**	0.990	0.800	**1.000**
g04	−30665.539	−30664.5	**−30665.539**	−30665.5	**−30665.539**	−30626.053	−30665.445	−30665.422
g05	5126.498	−	**5126.497**	5126.5	5126.599	−	5133.935	5126.988
g06	−6961.814	−6952.1	**−6961.814**	−6961.81	**−6961.814**	−6952.472	**−6961.814**	**−6961.814**
g07	24.306	24.620	**24.307**	24.3323	24.327	31.097	24.360	24.343
g08	0.095825	**0.0958250**	**0.095825**	**0.095825**	**0.095825**	**0.095825**	**0.095825**	**0.095825**
g09	680.63	680.91	**680.630**	**680.630**	680.632	685.994	680.632	680.631
g10	7049.25	7147.9	7054.316	7061.13	**7051.903**	9079.770	7231.497	7062.754
g11	0.75	**0.75**	**0.750**	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**
g12	1.000	0.999999857	**1.000000**	*NA*	**1.000**	**1.000**	**1.000**	**1.000**
g13	0.053950	*NA*	**0.053957**	*NA*	0.053986	0.134057	0.171855	0.058037

TABLE IV
COMPARISON OF THE MEAN SOLUTIONS FOUND BY OUR SMES AGAINST THE HOMOMORPHOUS MAPS (HM), STOCHASTIC RANKING (SR), ASCHEA, OUR GA VERSION AND TWO OTHER VERSIONS OF OUR SMES: ONE THAT USES ONLY RECOMBINATION AND ANOTHER ONE THAT USES BOTH RECOMBINATION AND STEPSIZE REDUCTION. A RESULT IN **BOLDFACE** INDICATES A BETTER RESULT OR THAT THE GLOBAL OPTIMUM (OR BEST KNOWN SOLUTION) WAS REACHED. "-" MEANS THAT NO FEASIBLE SOLUTIONS WERE FOUND. NA = NOT AVAILABLE

	Comparison of the mean solution obtained.							
P	**Optimal**	**HM**	**SR**	**ASCHEA**	**SMES**	**GA**	**Recomb.**	**Recomb.& Step.Reduc.**
g01	−15.000	−14.7082	**−15.000**	−14.84	**−15.000**	−14.236	**−15.000**	**−15.000**
g02	0.803619	0.79671	0.781975	0.59	0.785238	0.788588	**0.802376**	0.798786
g03	1.000	0.9989	**1.000**	0.99989	**1.000**	0.976	0.529	**1.000**
g04	−30665.539	−30655.3	**−30665.539**	−30665.5	**−30665.539**	−30590.455	−30665.445	−30661.106
g05	5126.498	−	**5128.881**	5141.65	5174.492	−	5133.935	5158.739
g06	−6961.814	−6342.6	−6875.940	**−6961.81**	−6961.284	−6872.204	**−6961.814**	**−6961.814**
g07	24.306	24.826	**24.374**	24.66	24.475	34.980	24.472	24.474
g08	0.095825	0.0891568	**0.095825**	**0.095825**	**0.095825**	0.095799	**0.095825**	**0.095825**
g09	680.63	681.16	680.656	680.641	680.643	692.064	**680.637**	**680.637**
g10	7049.25	8163.6	7559.192	**7193.11**	7253.047	10003.225	7355.564	7193.887
g11	0.75	**0.75**	**0.750**	**0.75**	**0.75**	**0.75**	0.752	0.752
g12	1.000	0.999134613	**1.000000**	*NA*	**1.000**	**1.000**	**1.000**	**1.000**
g13	0.053950	*NA*	**0.057006**	*NA*	0.166385	−	0.787648	0.247404

C. 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, and 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. Also, 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.

TABLE V
COMPARISON OF THE WORST SOLUTIONS FOUND BY OUR SMES AGAINST THE HOMOMORPHOUS MAPS (HM), STOCHASTIC RANKING (SR), ASCHEA, OUR GA VERSION AND TWO OTHER VERSIONS OF OUR SMES: ONE THAT USES ONLY RECOMBINATION AND ANOTHER ONE THAT USES BOTH RECOMBINATION AND STEPSIZE REDUCTION. A RESULT IN **BOLDFACE** INDICATES A BETTER RESULT OR THAT THE GLOBAL OPTIMUM (OR BEST KNOWN SOLUTION) WAS REACHED. "-" MEANS THAT NO FEASIBLE SOLUTIONS WERE FOUND. NA = NOT AVAILABLE

	Comparison of the worst solution obtained.							
P	**Optimal**	**HM**	**SR**	**ASCHEA**	**SMES**	**GA**	**Recomb.**	**Recomb.& Step.Reduc.**
g01	−15.000	−14.6154	**−15.000**	NA	**−15.000**	−14.015	**−15.000**	**−15.000**
g02	0.803619	**0.79119**	0.726288	NA	0.751322	0.779140	0.787626	0.785255
g03	1.000	0.9978	**1.000**	NA	**1.000**	0.956	0.294	0.999
g04	−30665.539	−30645.9	**−30665.539**	NA	**−30665.539**	−30567.105	−30649.424	−30647.484
g05	5126.498	−	**5142.472**	NA	5304.167	−	5246.968	5201.935
g06	−6961.814	−5473.9	−6350.262	NA	−6952.482	−6784.255	−5218.657	**−6961.814**
g07	24.306	25.069	**24.642**	NA	24.843	38.686	24.658	24.789
g08	0.095825	0.0291438	**0.095825**	NA	**0.095825**	0.095723	**0.095825**	**0.095825**
g09	680.63	683.18	680.763	NA	680.719	698.297	**680.649**	680.664
g10	7049.25	9659.3	8835.655	NA	7638.366	11003.533	7548.530	**7368.333**
g11	0.75	**0.75**	**0.750**	NA	**0.75**	0.752	0.785	0.767
g12	1.000	0.991950498	**1.000000**	NA	**1.000**	0.999	**1.000**	**1.000**
g13	0.053950	NA	**0.216915**	NA	0.468294	−	−	0.466266

TABLE VI
NUMBER OF RUNS (OUT OF 30) WHERE THE GLOBAL OPTIMUM (OR BEST KNOWN SOLUTION) WAS FOUND. WE ALSO SHOW THE BEST AND AVERAGE GENERATION NUMBER AT WHICH THE GLOBAL OPTIMUM (OR BEST KNOWN SOLUTION) WAS FOUND

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

D. Advantages of the Approach

Our approach can deal with moderately constrained problems (g04), highly constrained problems, problems with low (g06 and g08), moderated (g09) and high (g01, g02, g03,and 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, and g09). This behavior suggests that the mechanism of maintaining the best infeasible solution helps the search to sample the boundaries between the feasible and infeasible regions.

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 HMs performed 1 400 000 FFE, and ASCHEA required 1 500 000 FFE.

VII. FINDING THE STRENGTH OF THE APPROACH

Once we corroborated the effectiveness of our approach, it became particularly relevant to identify the key component (or combination of them) that was mainly responsible for the good performance of our algorithm. For that sake, we designed two experiments.

The aim of the first experiment was to know which of the three modifications to the $(\mu+\lambda)$-ES was mandatory, or if only the combined effect of all three made the algorithm work.

The goal of the second experiment was to reinforce our hypothesis regarding the effectiveness of the self-adaptation mechanism of an ES to sample constrained search spaces.

The experiments consisted of the following.

- **Cross Validation of Our ES' Mechanisms**: We tested our SMES using each of its mechanisms separately and combining them in pairs, in order to recognize which of them was mandatory. It is important to note that removing the diversity mechanism implies disallowing the best infeasible solution to remain in the population for the next generation of the algorithm. The comparison mechanism (the three rules based on feasibility) remains in all cases in order to guide the search to the feasible region of the search space.
- **ES Against GA**: Our second experiment consisted on implementing a real-coded GA with the same combined recombination and the same diversity mechanism used in

TABLE VII
BEST SOLUTIONS FOUND BY OUR SMES WITH ITS THREE MECHANISMS ANALYZED SEPARATELY. "∗" MEANS INFEASIBLE. A RESULT IN **BOLDFACE** INDICATES A BETTER RESULT OR THAT THE GLOBAL OPTIMUM (OR BEST KNOWN SOLUTION) WAS FOUND

	Best solutions obtained by our SMES with its three mechanisms analyzed separately			
Problem	**Optimal**	**Only Combined Recombination**	**Only Diversity Mech.**	**Only Stepsize Reduction**
g01	−15.000	**−15.000**	**−15.000**	**−15.000**
g02	0.803619	**0.803589**	0.763226	0.744524
g03	1.000	0.800	**0.995**	0.482
g04	−30665.539	**−30665.445**	−30663.625	−30664.609
g05	5126.498	5133.935	5127.187	**5126.938**
g06	−6961.814	**−6961.814**	**−6961.814**	**−6961.814**
g07	24.306	**24.360**	24.576	24.429
g08	0.095825	**0.095825**	**0.095825**	**0.095825**
g09	680.63	**680.632**	680.654	680.654
g10	7049.25	7231.497	7078.823	**7059.549**
g11	0.75	**0.75**	**0.75**	**0.75**
g12	1.000	**1.000**	**1.000**	**1.000**
g13	0.053950	**0.171855**	0.025667∗	0.013617∗

TABLE VIII
MEAN SOLUTIONS FOUND BY OUR SMES WITH ITS THREE MECHANISMS ANALYZED SEPARATELY. "∗" MEANS INFEASIBLE. A RESULT IN **BOLDFACE** INDICATES A BETTER RESULT OR THAT THE GLOBAL OPTIMUM (OR BEST KNOWN SOLUTION) WAS FOUND

	Mean solutions obtained by our SMES with its three mechanisms analyzed separately			
Problem	**Optimal**	**Only Combined Recombination**	**Only Diversity Mech.**	**Only Stepsize Reduction**
g01	−15.000	**−15.000**	−14.055	−14.493
g02	0.803619	**0.802376**	0.674	0.627237
g03	1.000	0.529	**0.692**	0.212
g04	−30665.539	**−30665.445**	−30630.231	−30633.003
g05	5126.498	**5133.935**	5373.424	5271.296
g06	−6961.814	**−6961.814**	−6950.373	−6961.439
g07	24.306	**24.472**	26.883	26.694
g08	0.095825	**0.095825**	**0.095825**	**0.095825**
g09	680.63	**680.637**	681.098	681.299
g10	7049.25	**7355.564**	7783.965	7527.588
g11	0.75	**0.752**	0.755	**0.752**
g12	1.000	**1.000**	**1.000**	**1.000**
g13	0.053950	**0.787648**	0.052238∗	0.201332∗

our SMES. Here, we wanted to see if the use of a GA instead of an ES would make any significant difference in terms of performance.

We will discuss next the results obtained in each of these two experiments.

A. Cross Validation of Our ES' Mechanisms

We tested six different versions of our SMES.

- Only combined recombination.
- Only diversity mechanism.
- Only stepsize reduction.
- Combined recombination and diversity mechanism.
- Combined recombination and stepsize reduction.
- Stepsize reduction and diversity mechanism.

The parameters used in these six versions are exactly the same used in the experiments described in Section V. Thus, the number of evaluations of the objective function is also the same (240 000).

The best results obtained for the three first versions (with only one feature) are presented in Table VII. Mean results are shown in Table VIII and worst results are shown in Table IX. The best, mean, and worst results obtained for the last three versions (combination of two features) are shown in Tables X–XII.

TABLE IX
WORST SOLUTIONS FOUND BY OUR SMES WITH ITS THREE MECHANISMS ANALYZED SEPARATELY. "*" MEANS INFEASIBLE. A RESULT IN **BOLDFACE** INDICATES A BETTER RESULT OR THAT THE GLOBAL OPTIMUM (OR BEST KNOWN SOLUTION) WAS FOUND

	Worst solutions obtained by our SMES with its three mechanisms analyzed separately			
Problem	**Optimal**	**Only Combined Recombination**	**Only Diversity Mech.**	**Only Stepsize Reduction**
g01	−15.000	**−15.000**	−10.875	−12.585
g02	0.803619	**0.787626**	0.586408	0.499773
g03	1.000	0.294	**0.441**	0.021
g04	−30665.539	**−30649.424**	−30447.381	−30582.023
g05	5126.498	**5246.968**	6018.426	6090.623
g06	−6961.814	−5218.657	−6618.615	**−6952.750**
g07	24.306	**24.658**	38.710	31.982
g08	0.095825	**0.095825**	**0.095825**	**0.095825**
g09	680.63	**680.649**	681.752	683.611
g10	7049.25	**7548.530**	9089.470	8585.027
g11	0.75	0.785	0.824	**0.767**
g12	1.000	**1.000**	0.999	**1.000**
g13	0.053950	1.0004*	**0.06212**	1.965371*

TABLE X
BEST SOLUTIONS FOUND BY OUR SMES WITH ALL POSSIBLE COMBINATIONS OF TWO OF ITS (THREE) MECHANISMS. "*" MEANS INFEASIBLE. A RESULT IN **BOLDFACE** INDICATES A BETTER RESULT OR THAT THE GLOBAL OPTIMUM (OR BEST KNOWN SOLUTION) WAS FOUND

	Best solutions obtained by our SMES with two of its mechanisms combined.			
Problem	**Optimal**	**Combined Recombination & Diversity Mechanism**	**Combined Recombination & Stepsize Reduction**	**Stepsize Reduction & Diversity Mechanism**
g01	−15.000	**−15.000**	**−15.000**	**−15.000**
g02	0.803619	0.803549	**0.803592**	0.741027
g03	1.000	0.998	**1.000**	0.725
g04	−30665.539	**−30665.539**	−30665.422	−30665.318
g05	5126.498	5105.347*	5126.988	**5126.534**
g06	−6961.814	**−6961.814**	**−6961.814**	**−6961.814**
g07	24.306	24.353	**24.343**	24.478
g08	0.095825	**0.095825**	**0.095825**	**0.095825**
g09	680.63	680.633	**680.631**	680.671
g10	7049.25	7092.887	**7062.754**	7095.610
g11	0.75	**0.75**	**0.75**	**0.75**
g12	1.000	**1.000**	**1.000**	**1.000**
g13	0.053950	**0.055491**	0.058037	0.033529*

From the results shown in Tables VII–IX, it is clear that the version with only the combined recombination provided the better "best" results, as well as the best "mean" and "worst" results for most of the functions. The version with only the diversity mechanism obtained better "best," "mean," and "worst" results only for function g03, and was unable to reach the feasible region in g13. The version with only the stepsize reduction obtained better "best" results for functions g05 and g10, and it also obtained a better "worst" result for function g06. However, this version was also unable to reach the feasible region in g13.

With respect to the comparison among versions which use two (out of three) combined mechanisms, the results indicate that the combination of the recombination with the stepsize reduction provided the best and more robust results (see Tables X–XII). This version obtained better "best" results for problems g02, g03, g07, g09, and g10. Also, it found similar "best" results for problems g01, g03, g06, g08, g11, and g12. The combined recombination coupled with the stepsize reduction obtained better "mean" results for problems g02, g03, g05, g07, g09, g10, g11, and g13, and it obtained similar "mean" results for problems g01, g06, g08, and g12. Finally,

TABLE XI
MEAN SOLUTIONS FOUND BY OUR SMES WITH ALL POSSIBLE COMBINATIONS OF TWO OF ITS (THREE) MECHANISMS. A RESULT IN **BOLDFACE** INDICATES A BETTER RESULT OR THAT THE GLOBAL OPTIMUM (OR BEST KNOWN SOLUTION) WAS FOUND

Problem	Mean solutions obtained by our SMES with two of its mechanisms combined.			
	Optimal	**Combined Recombination & Diversity Mechanism**	**Combined Recombination & Stepsize Reduction**	**Stepsize Reduction & Diversity Mechanism**
g01	−15.000	**−15.000**	**−15.000**	−14.125
g02	0.803619	0.775841	**0.798786**	0.609223
g03	1.000	0.808	**1.000**	0.315
g04	−30665.539	**−30665.539**	−30661.106	−30637.253
g05	5126.498	5249.087*	**5158.739**	5303.175
g06	−6961.814	−6900.247	**−6961.814**	**−6961.814**
g07	24.306	24.559	**24.474**	26.327
g08	0.095825	**0.095825**	**0.095825**	**0.095825**
g09	680.63	680.643	**680.637**	681.040
g10	7049.25	7605.077	**7193.887**	7823.012
g11	0.75	0.754	**0.752**	0.757
g12	1.000	**1.000**	**1.000**	**1.000**
g13	0.053950	0.372581	**0.247404**	0.108290*

TABLE XII
WORST SOLUTIONS FOUND BY OUR SMES WITH ALL POSSIBLE COMBINATIONS OF TWO OF ITS (THREE) MECHANISMS. A RESULT IN **BOLDFACE** INDICATES A BETTER RESULT OR THAT THE GLOBAL OPTIMUM (OR BEST KNOWN SOLUTION) WAS FOUND

Problem	Worst solutions obtained by our SMES with two of its mechanisms combined.			
	Optimal	**Combined Recombination & Diversity Mechanism**	**Combined Recombination & Stepsize Reduction**	**Stepsize Reduction & Diversity Mechanism**
g01	−15.000	**−15.000**	**−15.000**	−11.694
g02	0.803619	0.647445	**0.785255**	0.446562
g03	1.000	0.243	**0.999**	0.088
g04	−30665.539	**−30665.539**	−30647.484	−30523.984
g05	5126.498	*5877.772	**5201.935**	6005.305
g06	−6961.814	−6173.165	**−6961.814**	−6961.808
g07	24.306	25.136	**24.789**	30.682
g08	0.095825	**0.095825**	**0.095825**	**0.095825**
g09	680.63	**680.664**	**680.664**	681.724
g10	7049.25	13883.840	**7368.333**	9099.229
g11	0.75	0.854	**0.767**	0.909
g12	1.000	**1.000**	**1.000**	**1.000**
g13	0.053950	1.229679*	**0.466266**	0.469699*

this version provided better "worst" results for problems g02, g03, g05, g06, g07, g10, g11, and g13, and similar "worst" results for problems g01, g08, and g12.

Based on the results obtained, we decided to compare the results provided by the two most competitive versions of our SMES (the version with only the combined recombination and the version with the combined recombination coupled with the stepsize reduction). The comparison of results is shown in the last three columns from Tables III–V. The results indicated that the version with both the recombination and the stepsize reduction provided better "best" results in seven problems (g02, g03, g05, g07, g09, g10, and g13) and similar "best" results in other five (g01, g06, g08, g11, and g12). This version with two mechanisms reached better "mean" results in three problems (g03, g10, and g13), and similar "mean" results in six functions (g01, g06, g08, g09, g11, and g12). Finally, this version provided better "worst" results in six problems (g03, g05, g06, g10, g11. and g13), and it provided similar "worst" results in three more (g01, g08. and g12). All these results suggest that the stepsize reduction, which provides finer mutation movements in the search space, help the combined recombination to sample the feasible region as to find competitive results.

The main question that arose at this point was: what is the role of the diversity mechanism in the success of our approach? In order to answer this question, we compared the results of the version with combined recombination and stepsize reduction against the version with the three mechanisms. The results can be seen in columns 9 and 6, respectively, from Tables III–V. The complete version provided better "best" results in six functions (g02, g04, g05, g07, g10, and g13), and similar "best" results in other six (g01, g03, g06, g08, g11, and 12). Moreover, the complete version provided better "mean" results for three problems (g04, g11 and g13), and similar "mean" results in other four (g01, g03, g08 and g12). Finally, the complete version obtained better "worst" results in three problems (g03, g04, and g11), and it reached similar "worst" solutions for other three (g01, g08, and g12).

Thus, our approach provides results of a better quality when using the diversity mechanism. However, the price paid for this higher quality of results is a slight decrease in robustness. Also, the overall results (providing competitive results in all 13 test functions) are better when the diversity mechanism is incorporated into our SMES. It is also worth reminding that the goal of the diversity mechanism is to allow the search to generate solutions in the boundaries of the feasible region (which is something critical when dealing with constraints that are active in the global optimum). Hence, the use of such diversity mechanism seems a logical choice for dealing with active constraints.

To conclude, the combined recombination seems to be the dominant mechanism, which is assisted by the fine mutation movements provided by the reduction of the initial stepsize. Finally, the diversity mechanism helps to sample solutions located on the boundaries between the feasible and infeasible regions.

B. *ES Against GA*

For the comparison of performance between a GA and an evolution strategy, we used a real-coded GA with nonuniform mutation [24]. Such a GA used the same comparison mechanism (with the diversity mechanism) adopted by our SMES. It is important to note that we tested different mutation operators for real-coded GAs and nonuniform mutation provided the best results. Furthermore, we intended that the GA used the same features of the ES (except for the self-adaptive mutation which we hypothesized was the main strength of our ES-based approach). Finally, the same dynamic mechanism to handle the tolerance for equality constraints was employed.

The parameters used by our real-coded GA were the following:

- population size: 200;
- maximum number of generations: 1200;
- crossover rate: 0.8;
- mutation rate: 0.6;
- number of objective function evaluations: 240 000 (the same performed by our SMES).

We performed 30 runs for each test problem. The results obtained by the GA are presented in Tables III–V in column 7, and they are compared against those provided by the SMES in column 6. As can be seen, both the quality and robustness of the results provided by the GA are significantly poorer than those obtained with the evolution strategy in all the test functions adopted. The exceptions are g08, g11, and g12, in which the GA was able to find competitive results. These results highlight the strong influence (positive in this case) of using a more adequate search engine, in our case an ES over a GA. Therefore, the results seem to confirm our initial hypothesis about the usefulness of an ES to sample constrained search spaces in a more appropriate way.

VIII. Reaching the Feasible Region

After discussing the quality, robustness, and competitiveness of our approach, and after studying the effect of its three main mechanisms, we wanted to verify the rate (measured in terms of generations) at which the algorithm was able to reach the feasible region. This is an important issue, because in many real-world problems it is normally desirable to find feasible (even if not optimal) solutions with the lowest possible number of FFEs. This fact is due to different reasons typically found in industry: time-expensive evaluations of the objective function, restricted time to provide results in highly constrained problems, etc. Note, however, that a too fast arrival to the feasible region is not always desirable, since it may bias the search and keep us from reaching the global optimum.

To study this issue, we monitored the percentage of feasible solutions in the population of our SMES at every 200 generations (let us keep in mind that the total number of generations was fixed to 800). The results are presented in Fig. 5(a). As can be seen, for all the test problems our approach reaches the feasible region by generation 200. For problem g05, more than 20% of the population is feasible by generation 200 and for the remaining functions almost all the population is feasible by then. Based on the results found, we were interested in answering two questions:

1) What is the behavior before generation 200 (i.e., how fast does the population become almost feasible—where "almost feasible" refers to having a population in which at least 50% of the individuals are feasible—)?
2) How well is the diversity maintained at late stages of the evolutionary process?

The results obtained for these two questions are shown in Fig. 5(b) and (d) for question 1 and in Fig. 5(c) for question 2. Fig. 5(b) shows that the feasible region is reached at generation 20 in most cases. This means that (except for g05 and g13) the approach only requires 6100 FFE to find feasible solutions. In Table XIII, we show the statistical results obtained at this stage of the search. Note that although the results are still far from the optimum, with the exception of problems g05 and g13, most of the solutions are feasible. In Fig. 5(d), we observe in a close-up of Fig. 5(b) that the algorithm has the capability of maintaining some infeasible solutions despite the almost-feasible population (which follows the main motivation for using the diversity mechanism adopted). In addition, we show the statistical results obtained at generation 200 in Table XIV. A substantial improvement of the quality and robustness of the results is shown at generation 200, where only 20 000 FFE have been

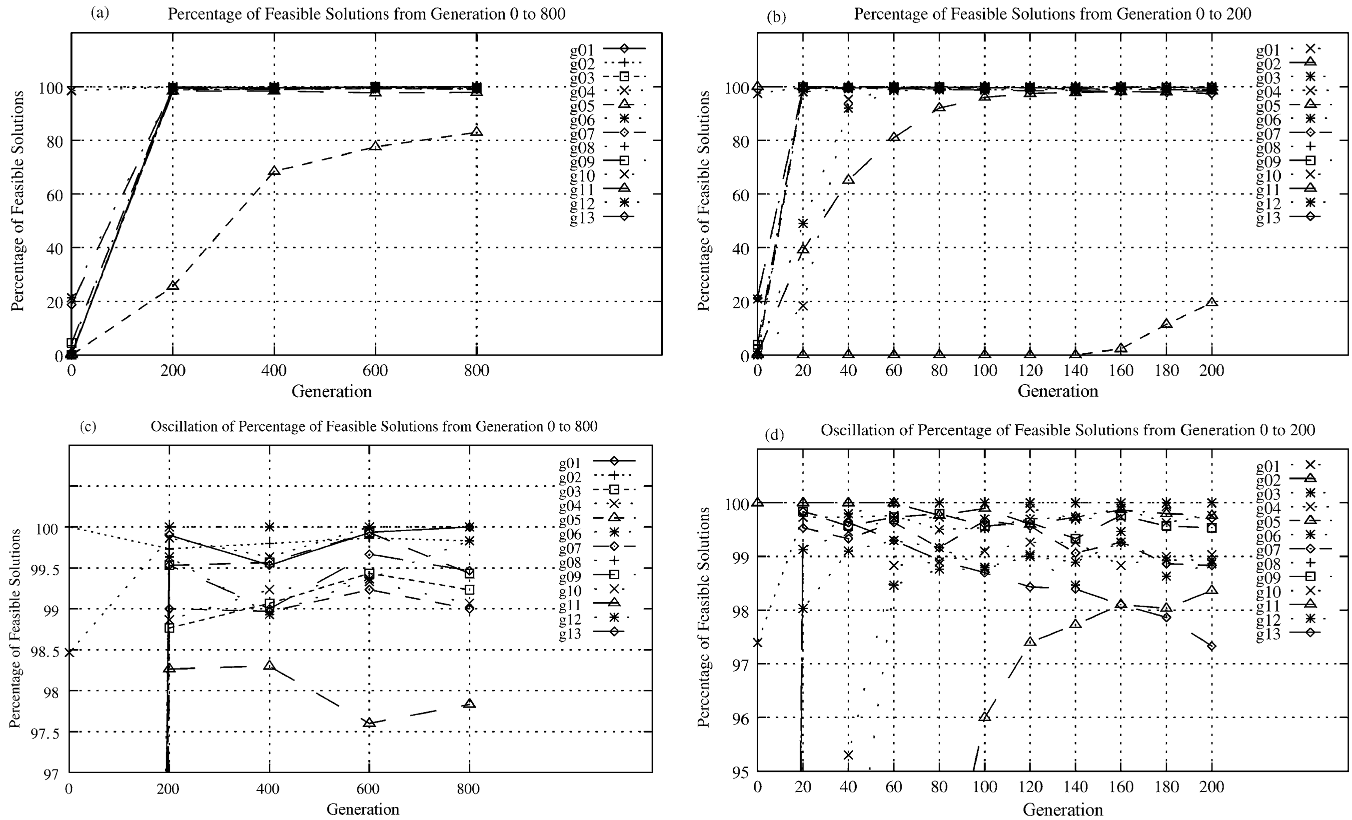

Fig. 5. Percentage of feasible solutions. (a) Every 200 generations (from 0 to 800). (b) Every 20 generations (from 0 to 200). (c) Detailed oscillation of feasible and infeasible solutions (from 0 to 800). (d) Detailed oscillation of feasible and infeasible solutions (from 0 to 200).

TABLE XIII
STATISTICAL RESULTS OF OUR APPROACH AFTER 20 GENERATIONS. ("$*$" MEANS INFEASIBLE). A RESULT IN **BOLDFACE** INDICATES THAT THE GLOBAL OPTIMUM (OR BEST KNOWN SOLUTION) WAS REACHED

	SMES after 20 generations.					
Problem	**Optimal**	**Best**	**Mean**	**Median**	**Worst**	**St. Dev.**
g01	−15.000	−7.291	−5.686	−5.685	−3.820	9.05E-1
g02	0.803619	0.442641	0.372686	0.370697	0.317450	2.65E-2
g03	1.000	0.949	0.785	0.827	0.526	1.13E-1
g04	−30665.539	−30563.615	−30473.319	−30462.027	−30401.756	40.26E+0
g05	5126.498	*5067.897	5211.511	5199.974	*5643.923	103.26E+0
g06	−6961.814	−6890.164	−6235.589	−6188.203	−5552.386	371.89E+0
g07	24.306	62.136	135.969	121.868	682.452	107.72E+0
g08	0.095825	**0.095825**	**0.095825**	**0.095825**	0.095817	2.0E-6
g09	680.63	686.592	704.351	704.377	719.151	8.91E+0
g10	7049.25	12777.324	17407.559	17284.081	25774.398	2368.59E+0
g11	0.75	**0.750**	0.783	0.764	*0.897	4.07E-2
g12	1.000	0.999	0.999	0.999	0.999	7.0E-5
g13	0.053950	*0.001348	*0.009035	*0.004388	*0.026345	8.45E-3

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 in most cases. This highlights the importance of the diversity mechanism in order to avoid that the algorithm gets trapped in local optima and it can reach a better solution (even the global optimum).

On the other hand, Fig. 5(c) shows a zooming of Fig. 5(a), where it is possible to see again in detail the smooth oscillation on the percentage of feasible solutions during the evolutionary

TABLE XIV
STATISTICAL RESULTS OF OUR APPROACH AFTER 200 GENERATIONS. ("$*$"MEANS INFEASIBLE). A RESULT IN **BOLDFACE** INDICATES THAT THE GLOBAL OPTIMUM (OR BEST KNOWN SOLUTION) WAS REACHED

	SMES after 200 generations.					
Problem	**Optimal**	**Best**	**Mean**	**Median**	**Worst**	**St. Dev.**
g01	−15.000	−14.999	−14.960	−14.999	−13.828	2.10E-1
g02	0.803619	0.801158	0.777458	0.787125	0.678203	2.43E-2
g03	1.000	**1.000**	0.999	0.999	0.987	3.74E-3
g04	−30665.539	**−30665.539**	−30665.531	−30665.536	−30665.473	1.35E-2
g05	5126.498	5126.988	5179.163	5162.323	5379.227	63.5E+0
g06	−6961.814	−6961.808	−6959.910	−6961.624	−6938.690	4.39E+0
g07	24.306	24.473	24.734	24.711	25.401	2.15E-1
g08	0.095825	**0.095825**	**0.095825**	**0.095825**	**0.095825**	0
g09	680.63	680.643	680.680	680.680	680.736	2.43E-2
g10	7049.25	7076.725	7330.398	7319.405	7816.830	153.72E+0
g11	0.75	**0.75**	**0.75**	**0.75**	0.76	3.07E-3
g12	1.000	**1.000**	**1.000**	**1.000**	**1.000**	0
g13	0.053950	$*$0.041436	0.145069	$*$0.045766	0.387152	1.53E-1

process after generation 200. This behavior suggests that the diversity mechanism still works well, maintaining near-feasible solutions with a good value of the objective function in the population (between one and three infeasible solutions are enough based on the previous results of the $(1+\lambda)$-ES approach [20], 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 II, compared with those on generation 200 (Table XIV), suggest that our diversity mechanism does its job of avoiding premature convergence and, when coupled with the combination of discrete and intermediate recombination and the self-adaptation mechanism of the ES leads the evolutionary search toward 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 on the landscape of the function, but such an idea is not explored any further in this work.

The point we want to make here is that our approach is fast at reaching the feasible region, while managing to avoid local attractors as to converge to the global optimum or its close vicinity.

IX. CONCLUSION

A new approach to handle constraints in evolutionary optimization was proposed in this paper. The proposed approach does not require the use of a penalty function, it uses the original self-adaptation mechanism of a multimembered ES to sample the search space in order to reach the feasible region and it adopts a simple comparison mechanism based on feasibility to guide the search toward the global optimum. Furthermore, the proposed technique adopts a combination of discrete and intermediate panmictic recombination in order to improve the exploitation effort. Additionally, to favor finer movements in the search space, the initial values of the stepsize (sigma values) are decreased in 60% with respect to the values normally adopted with a multimembered ES. Finally, the approach uses a diversity mechanism which consists of allowing infeasible solutions close to the boundaries of the feasible region to remain in the next population.

This approach is very easy to implement and its computational cost (based on the number of FFEs) is considerably lower than the cost reported by three other constraint-handling techniques which are representative of the state of the art in evolutionary optimization.

X. FUTURE WORK

One issue that deserves some further study is the sensitivity of our approach to the initial stepsize, since it is important to have a better understanding of the influence of this parameter on the performance of our algorithm. In addition, due to the relevance shown by the combined recombination operator, we are interested in testing other types of recombination operators like generalized panmictic intermediate recombination.

Another path of future research consists of applying our approach to the solution of real-world (engineering) optimization problems. Additionally, we are also interested in implementing our constraint handling mechanism using other heuristics such as differential evolution [27] and particle swarm optimization [16]. This aims to explore the possibility of decreasing its computational cost (measured in terms of the number of FFEs), after reaching the feasible region, since in the current approach almost 75% of the search process is spent on this task.

APPENDIX
TEST FUNCTIONS

1) **g01**

$$\text{Minimize } f(\vec{x}) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i$$

subject to

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

where the bounds are $0 \le x_i \le 1 (i = 1, \ldots, 9)$, $0 \le x_i \le 100 (i = 10, 11, 12)$, and $0 \le x_{13} \le 1$. 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**

$$\text{Maximize } f(\vec{x}) = \left| \frac{\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right|$$

subject to

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

$$g_2(\vec{x}) = \sum_{i=1}^{n} x_i - 7.5n \le 0 \qquad (10)$$

where $n = 20$ and $0 \le x_i \le 10 (i = 1, \ldots, n)$. The global maximum is unknown; the best reported solution is [29] $f(x^*) = 0.803\,619$. Constraint g_1 is close to being active ($g_1 = -10^{-8}$).

3) **g03**

$$\text{Maximize } f(\vec{x}) = \left(\sqrt{n}\right)^n \prod_{i=1}^{n} x_i$$

subject to

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

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

4) **g04**

$$\begin{aligned}
\text{Minimize } f(\vec{x}) = {}& 5.357\,854\,7x_3^2 + 0.835\,689\,1x_1x_5\\
&+ 37.293\,239x_1 - 40\,792.141
\end{aligned}$$

subject to

$$\begin{aligned}
g_1(\vec{x}) = {}& 85.334\,407 + 0.005\,685\,8x_2x_5\\
&+ 0.000\,626\,2x_1x_4 - 0.002\,205\,3x_3x_5\\
&- 92 \le 0\\
g_2(\vec{x}) = {}& -85.334\,407 - 0.005\,685\,8x_2x_5\\
&- 0.000\,626\,2x_1x_4 + 0.002\,205\,3x_3x_5 \le 0\\
g_3(\vec{x}) = {}& 80.512\,49 + 0.007\,131\,7x_2x_5\\
&+ 0.002\,995\,5x_1x_2 + 0.002\,181\,3x_3^2\\
&- 110 \le 0\\
g_4(\vec{x}) = {}& -80.512\,49 - 0.007\,131\,7x_2x_5\\
&- 0.002\,995\,5x_1x_2 - 0.002\,181\,3x_3^2\\
&+ 90 \le 0\\
g_5(\vec{x}) = {}& 9.300\,961 + 0.004\,702\,6x_3x_5\\
&+ 0.001\,254\,7x_1x_3 + 0.001\,908\,5x_3x_4\\
&- 25 \le 0\\
g_6(\vec{x}) = {}& -9.300\,961 - 0.004\,702\,6x_3x_5\\
&- 0.001\,254\,7x_1x_3 - 0.001\,908\,5x_3x_4\\
&+ 20 \le 0
\end{aligned}$$

where $78 \le x_1 \le 102$, $33 \le x_2 \le 45$, $27 \le x_i \le 45 (i = 3,4,5)$. The optimum solution is $x^* = (78, 33, 29.995\,256\,025\,682, 45, 36.775\,812\,905\,788)$, where $f(x^*) = -30\,665.539$. Constraints $g_1 y$ g_6 are active.

5) **g05**

$$\begin{aligned}
\text{Minimize } f(\vec{x}) = {}& 3x_1 + 0.000\,001x_1^3 + 2x_2\\
&+ \left(\frac{0.000\,002}{3}\right) x_2^3
\end{aligned}$$

subject to

$$\begin{aligned}
g_1(\vec{x}) &= -x_4 + x_3 - 0.55 \le 0\\
g_2(\vec{x}) &= -x_3 + x_4 - 0.55 \le 0\\
h_3(\vec{x}) &= 1000\sin(-x_3 - 0.25)\\
&\quad + 1000\sin(-x_4 - 0.25) + 894.8\\
&\quad - x_1 = 0\\
h_4(\vec{x}) &= 1000\sin(x_3 - 0.25)\\
&\quad + 1000\sin(x_3 - x_4 - 0.25) + 894.8\\
&\quad - x_2 = 0\\
h_5(\vec{x}) &= 1000\sin(x_4 - 0.25)\\
&\quad + 1000\sin(x_4 - x_3 - 0.25)\\
&\quad + 1294.8 = 0
\end{aligned}$$

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 $x^* = (679.9453, 1026.067, 0.118\,876\,4, -0.396\,233\,6)$, where $f(x^*) = 5126.4981$.

6) **g06**

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

subject to

$$\begin{aligned}
g_1(\vec{x}) &= -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \le 0\\
g_2(\vec{x}) &= (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \le 0
\end{aligned}$$

where $13 \le x_1 \le 100$ and $0 \le x_2 \le 100$. The optimum solution is $x^* = (14.095, 0.842\,96)$, where $f(x^*) = -6961.813\,88$. Both constraints are active.

7) **g07**

$$\begin{aligned}\text{Minimize } f(\vec{x}) = {} & x_1^2 + x_2^2 + x_1x_2 - 14x_1 \\ & - 16x_2 + (x_3-10)^2 + 4(x_4-5)^2 \\ & + (x_5-3)^2 + 2(x_6-1)^2 + 5x_7^2 \\ & + 7(x_8-11)^2 + 2(x_9-10)^2 \\ & + (x_{10}-7)^2 + 45\end{aligned}$$

subject to

$$\begin{aligned}g_1(\vec{x}) &= -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 0 \\ g_2(\vec{x}) &= 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0 \\ g_3(\vec{x}) &= -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \le 0 \\ g_4(\vec{x}) &= 3(x_1-2)^2 + 4(x_2-3)^2 + 2x_3^2 - 7x_4 \\ &\quad - 120 \le 0 \\ g_5(\vec{x}) &= 5x_1^2 + 8x_2 + (x_3-6)^2 - 2x_4 - 40 \le 0 \\ g_6(\vec{x}) &= x_1^2 + 2(x_2-2)^2 - 2x_1x_2 + 14x_5 \\ &\quad - 6x_6 \le 0 \\ g_7(\vec{x}) &= 0.5(x_1-8)^2 + 2(x_2-4)^2 + 3x_5^2 - x_6 \\ &\quad - 30 \le 0 \\ g_8(\vec{x}) &= -3x_1 + 6x_2 + 12(x_9-8)^2 - 7x_{10} \le 0\end{aligned}$$

where $-10 \le x_i \le 10 (i = 1, \ldots, 10)$. The global optimum is $x^* = (2.171\,996\,2.363\,683\,8.773\,926\,5.095\,984\,0.990\,654\,8, 1.430\,574\,1.321\,644\,9.828\,726\,8.280\,092\,8.375\,927)$, where $f(x^*) = 24.306\,209\,1$. Constraints g_1, g_2, g_3, g_4, g_5, and g_6 are active.

8) **g08**

$$\text{Maximize } f(\vec{x}) = \frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1+x_2)}$$

subject to

$$\begin{aligned}g_1(\vec{x}) &= x_1^2 - x_2 + 1 \le 0 \\ g_2(\vec{x}) &= 1 - x_1 + (x_2-4)^2 \le 0\end{aligned}$$

where $0 \le x_1 \le 10$ and $0 \le x_2 \le 10$. The optimum solution is located at $x^* = (1.227\,971\,3, 4.245\,373\,3)$, where $f(x^*) = 0.095\,825$.

9) **g09**

$$\begin{aligned}\text{Minimize } f(\vec{x}) = {} & (x_1-10)^2 + 5(x_2-12)^2 \\ & + x_3^4 + 3(x_4-11)^2 + 10x_5^6 \\ & + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7\end{aligned}$$

subject to

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

where $-10 \le x_i \le 10 (i = 1, \ldots, 7)$. The global optimum is $x^* = (2.330\,499\,1.951\,372, -0.477\,541\,4, 4.365\,726, -0.624\,487\,0, 1.038\,131\,1.594\,227)$, where $f(x^*) = 680.630\,057\,3$. Two constraints are active (g_1 and g_4).

10) **g10**

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

subject to

$$\begin{aligned}g_1(\vec{x}) &= -1 + 0.0025(x_4 + x_6) \le 0 \\ g_2(\vec{x}) &= -1 + 0.0025(x_5 + x_7 - x_4) \le 0 \\ g_3(\vec{x}) &= -1 + 0.01(x_8 - x_5) \le 0 \\ g_4(\vec{x}) &= -x_1x_6 + 833.332\,52x_4 + 100x_1 \\ &\quad - 83\,333.333 \le 0 \\ g_5(\vec{x}) &= -x_2x_7 + 1250x_5 + x_2x_4 \\ &\quad - 1250x_4 \le 0 \\ g_6(\vec{x}) &= -x_3x_8 + 1\,250\,000 + x_3x_5 \\ &\quad - 2500x_5 \le 0\end{aligned}$$

where $100 \le x_1 \le 10\,000$, $1000 \le x_i \le 10\,000$, $(i = 2, 3)$, $10 \le x_i \le 1000$, $(i = 4, \ldots, 8)$. The global optimum is: $x^* = (579.19, 1360.13, 5109.92, 182.0174, 295.5985, 217.9799, 286.40, 395.5979)$, where $f(x^*) = 7049.25$. g_1, g_2 and g_3 are active.

11) **g11**

$$\text{Minimize } f(\vec{x}) = x_1^2 + (x_2-1)^2$$

subject to

$$h(\vec{x}) = x_2 - x_1^2 = 0$$

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

12) **g12**

$$\text{Maximize } f(\vec{x}) = \frac{100 - (x_1-5)^2 - (x_2-5)^2 - (x_3-5)^2}{100}$$

subject to

$$\begin{aligned}g_1(\vec{x}) &= (x_i-p)^2 + (x_2-q)^2 + (x_3-r)^2 \\ &\quad - 0.0625 \le 0\end{aligned}$$

where $0 \le x_i \le 10 (i = 1, 2, 3)$ 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**

$$\text{Minimize } f(\vec{x}) = e^{x_1x_2x_3x_4x_5}$$

subject to

$$\begin{aligned}h_1(\vec{x}) &= x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0 \\ h_2(\vec{x}) &= x_2x_3 - 5x_4x_5 = 0 \\ h_3(\vec{x}) &= x_1^3 + x_2^3 + 1 = 0\end{aligned}$$

where $-2.3 \le x_i \le 2.3 (i = 1, 2)$ and $-3.2 \le x_i \le 3.2 (i = 3, 4, 5)$. The optimum solution is $x^* = (-1.717\,143\,1.595\,709\,1.827\,247, -0.763\,641\,3, -0.763\,645)$, where $f(x^*) = 0.053\,949\,8$.

Acknowledgment

The authors gratefully acknowledge the valuable comments from Dr. M. Schoenauer and the anonymous reviewers which greatly helped them to improve the contents of this paper.

REFERENCES

[1] D. V. Arnold, *Noisy Optimization With Evolution Strategies*. Norwell, MA: Kluwer, 2002.

[2] T. Asselmeyer, W. Ebeling, and H. Rosé, "Evolutionary strategies of optimization," *Phys. Rev. E*, vol. 56, no. 1, pp. 1171–1180, July 1997.

[3] T. Bäck, *Evolutionary Algorithms in Theory and Practice*. New York: Oxford Univ. Press, 1996.

[4] T. Bäck and M. Schütz, "Evolution strategies for mixed-integer optimization of optical multilayer systems," in *Proc. 4th Annu. Conf. Evolutionary Computation, Evolutionary Programming IV*, J. R. McDonnell, R. G. Reynolds, and D. B. Fogel, Eds., Cambridge, MA, 1995, pp. 33–51.

[5] H.-G Beyer and D. V. Arnold, "Theory of evolution strategies—A tutorial," in *Theoretical Aspects of Evolutionary Computing*, L. Kallel, N. Baudts, and A. Rogers, Eds. Berlin, Germany: Springer, 2001, pp. 109–133.

[6] C. A. C. Coello, "Theoretical and numerical constraint handling techniques used with evolutionary algorithms: A survey of the state of the art," *Comput. Methods Appl. Mech. Eng.*, vol. 191, no. 11–12, pp. 1245–1287, January 2002.

[7] C. A. C. Coello and E. Mezura-Montes, "Handling constraints in genetic algorithms using dominance-based tournaments," in *Proc. 5th Int. Conf. Adaptive Computing Design Manufacture (ACDM 2002)*, vol. 5, I. C. Parmee, Ed., Devon, U.K., Apr. 2002, pp. 273–284.

[8] C. A. C. Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Norwell, MA: Kluwer, 2002.

[9] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, no. 2/4, pp. 311–338, 2000.

[10] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Berlin, Germany: Springer-Verlag, 2003.

[11] D. Goldberg, *The Design of Innovation*. Norwell, MA: Kluwer, 2002.

[12] G. W. Greenwood and Y.-P. Liu, "Finding low energy conformations of atomic clusters using evolution strategies," in *Lecture Notes in Computer Science*, V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, Eds. Berlin, Germany: Springer-Verlag, 1998, vol. 1447, Proc. Evolutionary Programming VII, pp. 493–502.

[13] S. B. Hamida and M. Schoenauer, "ASCHEA: New results using adaptive segregational constraint handling," in *Proc. Congr. Evolutionary Computation*, vol. 1, May 2002, pp. 884–889.

[14] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," in *Proc. 1st IEEE Conf. Evolutionary Computation, IEEE World Congr. Computational Intelligence*, vol. 1, June 1994, pp. 82–87.

[15] F. Jiménez and J. L. Verdegay, "Evolutionary techniques for constrained optimization problems," in *Proc. 7th Eur. Congr. Intelligent Techniques Soft Computing (EUFIT'99)*, H.-J. Zimmermann, Ed., Aachen, Germany, 1999, ISBN 3-89 653-808-X.

[16] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*. San Mateo, CA: Morgan Kaufmann, 2001.

[17] S. Koziel and Z. Michalewicz, "Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization," *Evol. Comput.*, vol. 7, no. 1, pp. 19–44, 1999.

[18] E. Mezura-Montes and C. A. C. Coello, "Multiobjective-based concepts to handle constraints in evolutionary algorithms," in *Proc. 4th Mexican Int. Conf. Computer Science (ENC'2003)*, E. Chávez, J. Favela, M. Mejía, and A. Oliart, Eds., Los Alamitos, CA, Sept. 2003, pp. 192–199.

[19] ——, "A simple evolution strategy to solve constrained optimization problems," in *Proc. Genetic and Evolutionary Computation Conf. (GECCO'2003)*, E. Cantú-Paz, J. A. Foster, K. Deb, L. D. Davis, R. Roy, U. -M. O' Reilly, H. -G. Beyer, R. Standish, G. Kendall, S. Wilson, M. Harman, J. Wegener, D. Dasgupta, M. A. Potter, A. C. Schultz, K. A. Dowsland, N. Jonoska, and J. Miller, Eds., Heidelberg, Germany, July 2003, pp. 640–641.

[20] ——, "Adding a diversity mechanism to a simple evolution strategy to solve constrained optimization problems," in *Proc. Congr. Evolutionary Computation 2003 (CEC'2003)*, vol. 1, Dec. 2003, pp. 6–13.

[21] ——, (2002) A numerical comparison of some multiobjective-based techniques to handle constraints in genetic algorithms, Tech. Rep. EVOCINV-03-2002. Evol. Comput. Group, CINVESTAV, Sección de Computación, CINVESTAV-IPN, Dept. de Ingeniería Eléctrica, México D.F., México. [Online]. Available: http://www.cs.cinvestav.mx/constraint/

[22] ——, (2003) On the Usefulness of the evolution strategies self-adaptation mechanism to handle constraints in global optimization. Tech. Rep. EVOCINV-01-2003. Evol. Comput. Group, CINVESTAV, Sección de Computación, Dept. de Ingeniería Eléctrica, CINVESTAV-IPN, México D.F., México. [Online]. Available: http://www.cs.cinvestav.mx/constraint/

[23] Z. Michalewicz and M. Michalewicz, "Pro-life versus pro-choice strategies in evolutionary computation techniques," in *Computational Intelligence: A Dynamic System Perspective*, M. Palaniswami, Y. Attikiouzel, R. J. Marks, D. Fogel, and T. Fukuda, Eds. Piscataway, NJ: IEEE Press, 1995, pp. 137–151.

[24] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, third ed. Berlin, Germany: Springer-Verlag, 1996.

[25] Z. Michalewicz and M. Schoenauer, "Evolutionary algorithms for constrained parameter optimization problems," *Evol. Comput.*, vol. 4, no. 1, pp. 1–32, 1996.

[26] D. Powell and M. M. Skolnick, "Using genetic algorithms in engineering design optimization with nonlinear constraints," in *Proc. 5th Int. Conf. Genetic Algorithms (ICGA-93)*, S. Forrest, Ed., San Mateo, CA, July 1993, pp. 424–431.

[27] K. V. Price, "An introduction to differential evolution," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. New York: McGraw-Hill, 1999, pp. 79–108.

[28] J. T. Richardson, M. R. Palmer, G. Liepins, and M. Hilliard, "Some guidelines for genetic algorithms with penalty functions," in *Proc. 3rd Int. Conf. Genetic Algorithms (ICGA-89)*, J. D. Schaffer, Ed., San Mateo, CA, June 1989, pp. 191–197.

[29] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 4, pp. 284–294, Sept. 2000.

[30] H. P. Schwefel, *Evolution and Optimal Seeking*. New York: Wiley, 1995.

[31] F. Schweitzer, W. Ebeling, H. Rosé, and O. Weiss, "Optimization of road networks using evolutionary strategies," *Evol. Comput.*, vol. 5, no. 4, pp. 419–438, 1998.

[32] A. E. Smith and D. W. Coit, "Constraint handling techniques—Penalty functions," in *Handbook of Evolutionary Computation*, T. Bäck, D. B. Fogel, and Z. Michalewicz, Eds. New York: Oxford Univ. Press and Inst. Physics, 1997, ch. C 5.2.

Efrén Mezura-Montes received the B.Sc. degree in computer systems engineering from the Universidad de las Américas Puebla, Puebla, México, in 1997, and the M.Sc. degree in artificial intelligence from the Universidad Veracruzana, Veracruz, México, in 2002. He is currently working towards the Ph.D. degree in electrical engineering at the Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV-IPN), México City, México.

His current research interests include evolutionary computation, global optimization, constraint handling techniques for evolutionary algorithms, and multiobjective optimization.

Carlos A. Coello Coello (M'99–SM'04) received the B.Sc. degree in civil engineering from the Universidad Autónoma de Chiapas, México, in 1991, and the M.Sc. and the Ph.D. degrees in computer science from Tulane University, New Orleans, LA, in 1993 and 1996, respectively.

He is currently an Associate Professor (CINVESTAV-3B Researcher) with the Electrical Engineering Department, CINVESTAV-IPN, México City, México. He has authored or coauthored over 100 technical papers and several book chapters. He has also coauthored the book *Evolutionary Algorithms for Solving Multi-Objective Problems* (Norwell, MA: Kluwer, 2002). His major research interests are: evolutionary multiobjective optimization, constraint-handling techniques for evolutionary algorithms, and evolvable hardware.

Dr. Coello has served on the program committees of over 30 international conferences and has been technical reviewer for over 30 international journals including the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION for which he also serves as an Associate Editor. He also Chairs the Task Force on Multiobjective Evolutionary Algorithms of the IEEE Neural Networks Society. He is a member of the Association for Computing Machinery (ACM), Sigma Xi, and the Mexican Academy of Sciences.