An empirical study about the usefulness of evolution strategies to solve constrained optimization problems

Efrén Mezura-Montes*†and Carlos A. Coello Coello ‡[*]

*†National Laboratory on Advanced Informatics (LANIA A.C.)
Rébsamen 80 Col. Centro Xalapa, Veracruz, 91000, MÉXICO
`emezura@lania.mx`

‡Evolutionary Computation Group (EVOCINV)
CINVESTAV-IPN. Computer Science Department
Av. IPN No. 2508. Col. San Pedro Zacatenco
México D.F. 07300, MÉXICO
`ccoello@cs.cinvestav.mx`

Abstract

In this paper we explore the capabilities of different types of evolution strategies to solve global optimization problems with constraints. The aim is to highlight the idea that the selection of the search engine is more critical than the selection of the constraint-handling mechanism, which can be very simple indeed. We show how using just three simple comparison criteria based on feasibility, the simple evolution strategy can be led to the feasible region of the search space and find the global optimum solution (or a very good approximation of it). Different evolution strategies including a variation of a $(\mu+1)-ES$ and $(\mu \overset{+}{,} \lambda)-ES$ with or without correlated mutation were implemented. Such approaches were tested using a well-known test suite for constrained optimization. Furthermore, the most competitive version found (among those five)

---

[*] Postal Address: Efren Mezura-Montes c/o Carlos Coello PO Box 60326-394 Houston, Texas 77205-0326, USA. Phone: 011 52 55 5061 3800 ext. 6564 Fax: 011 52 55 5061 3757

was compared against three state-of-the-art approaches and it was also compared against a GA using the same constraint-handling approach. Finally, our evolution strategy was used to solve some engineering design problems.

Keywords: Global optimization, evolutionary algorithms, constraint handling, engineering design.

1 Introduction

Evolution strategies (ES) have been widely used to solve global optimization problems (Greenwood & Liu 1998, Schweitzer, Ebeling, Rosé & Weiss 1998, Arnold 2002). Moreover, there is a theoretical background that supports ES convergence (Schwefel 1995, Bäck 1996, Beyer 2001). However, as other Evolutionary Algorithms, like Evolutionary Programming (Fogel 1999) and Genetic Algorithms (Goldberg 1989), ES, in its original version, lacks an explicit mechanism to deal with constrained search spaces. The recombination and mutation operators cannot distinguish between feasible and infeasible solutions. Therefore, several approaches have been suggested in the literature to allow Evolutionary Algorithms (EAs) to deal with constrained problems (Coello Coello 2002).

The most common approach adopted to deal with constrained search spaces is the use of penalty functions. 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 as to approach efficiently the feasible region (Smith & Coit 1997, Coello Coello 2002).

Several approaches have been proposed to avoid this dependency on the values of the penalty factors. The most known are: Death penalty (Bäck, Hoffmeister & Schwefel 1991), static penalties (Homaifar, Lai & Qi 1994), dynamic penalties (Joines & Houck 1994), annealing penalties (Michalewicz & Attia 1994), adaptive penalties (Rasheed 1998), co-evolutionary penalties (Coello Coello 2000*b*), the segregated genetic algorithm (Riche, Knopf-Lenoir & Haftka 1995) and fuzzy penalties (Wu & Yu 2001). There are alternative approaches, like special encodings, whose aim is to generate only feasible solutions and use special operators to preserve their feasibility during all the evolutionary process (Michalewicz 1996, Schoenauer & Michalewicz 1996, Koziel & Michalewicz 1999). Other alternative approach is the use of repair algorithms, whose goal is to make feasible an infeasible solution (Michalewicz & Nazhiyath 1995, Liepins & Vose 1990). The separation of constraints and objectives is another approach to deal with constrained search spaces, the idea is to avoid the combination of the value of the objective function and the constraints of a problem to assign fitness, like when using a penalty function (Deb 2000, Coello Coello & Mezura-Montes 2002). Finally, there are hybrid approaches whose aim is to combine different techniques (even mathematical programming approaches) into one single approach (Wah & Chen 2001, Jin & Reynolds 1999).

Two of the most recent techniques to handle constraints in EAs found in the literature, the Stochastic Ranking (Runarsson & Yao 2000) and the Adaptive Segregational Constraint Handling Evolutionary Algorithm (ASCHEA) (Hamida & Schoenauer 2002) are both based on an ES. The quality and consistency of the reported results of both approaches are very good and these results are indeed better than those provided by the Homomorphous Maps (Koziel & Michalewicz 1999), which is based on a genetic algorithm.

This suggests that ES's way of sampling the search space might help the approach to deal with constrained search spaces. We think that the emphasis must be on choosing an adequate search engine and the constraint handling technique will not be necessarily complex or difficult to calibrate. The question that arises here is to know what features of an evolution strategy improves its performance the most. Thus, we decided to compare five different types of ES (a variation of a $(\mu + 1)$-ES that we will call V$(\mu + 1)$-ES, a $(\mu + \lambda)$-ES and a (μ, λ)-ES both with correlated and noncorrelated mutation) with only a simple comparison mechanism based on feasibility to handle the constraints of the problem. The aim is to show how the evolution strategy is capable of sampling the search space in a better way than other evolutionary algorithms (like genetic algorithms) and that it does not require a very complicated constraint handling mechanism in order to reach the feasible region of the search space.

We tested these five versions on a well-known benchmark for global nonlinear optimization. The most competitive ES (out of these five) was compared against a similar approach, which was based on a genetic algorithm and it was also compared against three state-of-the-art approaches. Finally, to show its applicability to real-world problems, the approach was used to solve three engineering design problems.

This paper is organized as follows: In Section 2, we describe the problem to be solved. Afterwards, in Section 3 we briefly describe the main concepts of ES. In Section 4, we provide an explanation of the simple constraint handling approach adopted in this work. After that, in Section 5, we describe the experimental design and we present the results obtained in these experiments. Finally, in Section 6 we provide our conclusions and some possible paths of future research.

2 Statement of the problem

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

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

subject to:

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

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

where $\vec{x}$ is the vector of solutions $\vec{x} = [x_1, x_2, \ldots, x_n]^T$, where each $x_i, \;\; i = 1, \ldots, n$ is bounded by lower and upper limits $L_i \leq x_i \leq U_i$; m 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 it 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 $\mathcal{F}$. Most constraint-handling approaches used with EAs tend to deal only with inequality constraints. However, in those cases, equality constraints are transformed into inequality constraints of the form:

$$|h_j(\vec{x})| - \epsilon \leq 0 \tag{4}$$

where ϵ is the tolerance allowed (a very small value).

3 Evolution strategies

ES were proposed by Peter Bienert, Ingo Rechenberg and Hans-Paul Schwefel, who used them to solve hydrodynamical problems (Rechenberg 1965, Schwefel 1968). The first ES version was the $(1+1)$-ES which uses just one individual that is mutated using a normal distributed random number with mean zero, standard deviation of 1 and an identical stepsize value for each decision variable. The expression to generate this mutation for each decision variable i of the problem is presented in equation 5

$$x_i^{'} = x_i + \sigma(t) \cdot N_i(0,1), \forall_i \in \{1, \ldots, n\} \tag{5}$$

where n is the number of decision variables of the problem. The best solution between the parent and the offspring is chosen and the other one is eliminated. Rechenberg derived a convergence rate theory and proposed a rule for changing the stepsize value of mutations, which he called the '$1/5$-success rule' (Rechenberg 1973). This dynamic rule is detailed in equation 6, where p_s is the frequency of successful mutations (when the offspring replaces its parent), measured over intervals of $10 \cdot n$ trials and $0.817 \leq c \leq 1$ (Rechenberg 1973).

$$\sigma(t) = \begin{cases} \sigma(t-n)/c & \text{if } p_s > 1/5 \\ \sigma(t-n) \cdot c & \text{if } p_s < 1/5 \\ \sigma(t-n) & \text{if } p_s = 1/5 \end{cases} \tag{6}$$

The first multimembered ES was the $(\mu + 1)$-ES, which was designed by Rechenberg and is described in detail in (Bäck et al. 1991). In this approach, μ parent solutions recombine to generate one offspring. This solution is also mutated and, if it is better, it will replace the worst parent solution.

The $(\mu + \lambda)$-ES and the (μ, λ)-ES were proposed by Schwefel (1981). In the first one, the best μ individuals out of the union of the μ original parents and their λ offspring will survive for the next generation. On the other hand, in the (μ, λ)-ES the best μ will only be selected from the λ offspring.

The $(\mu + \lambda)$-ES uses an implicit elitist mechanism and solutions can survive more than one generation. Meanwhile, in the (μ, λ)-ES solutions only survive one generation. Instead of the '$1/5$-success rule', each individual includes a stepsize value for each decision variable. Moreover, for each combination of two stepsize (σ) values, a rotation angle is included. These angles are used to perform a correlated mutation. This mutation allows each individual to look for a search direction. The stepsize values and the angles of each individual are called strategy parameters and they are recombined and mutated as well. A $(\mu + \lambda)$-ES or (μ, λ)-ES individual can be seen as follows: $a(i)(\vec{x}, \vec{\sigma}, \vec{\theta})$, where i is the number of individual in the population, $\vec{x} \in \Re^n$ is a vector of n decision variables, $\vec{\sigma}$ is a vector of n stepsize values and $\vec{\theta}$ is a vector of $n(n-1)/2$ rotation angles where $\theta_i \in [-\pi, \pi]$. For a detailed description of the representation of a solution and its differences with a representation in a traditional GA see Figure 1.

[FIGURE 1 MUST BE LOCATED HERE]

Recombination can be sexual (two parents) or panmictic (more than two parents). It is worth reminding that recombination can be applied to the decision variables of the problem as well as to the strategy parameters. There are two main types of recombination: (1) Discrete and (2) Intermediate. Both can be either sexual or panmictic. Also, Schwefel (1995) proposed to generalize intermediate recombination by allowing arbitrary weight factors from the interval $[0, 1]$ to be used anew for each component of the chromosome. For a complete description of the recombination operator we provide the following list:

$$\text{offspring}_i = \begin{cases} \textbf{Operation} & \textbf{Type of Recombination} \\ P1_i \text{ or } P2_i & \text{discrete} \\ P1_i \text{ or } PJ_i & \text{panmictic discrete} \\ P1_i + ((P2_i - P1_i)/2) & \text{intermediate} \\ P1_i + ((PJ_i - P1_i)/2) & \text{panmictic intermediate} \\ P1_i + \chi((P2_i - P1_i)/2) & \text{generalized intermediate} \\ P1_i + \chi_i((PJ_i - P1_i)/2) & \text{panmictic generalized intermediate} \end{cases}$$

where $P1$ and $P2$ are the parents for the sexual recombination, PJ means a different parent for each gene in the chromosome. χ_i is the weight factor created anew for each decision variable and used in the generalized recombination.

The mutation operator works on the decision variables and also on the strategy parameters. The mutation is calculated in the following way:

$$\sigma'_i = \sigma_i \cdot exp(\tau' \cdot N(0, 1) + \tau \cdot N_i(0, 1)) \tag{7}$$

$$\theta'_j = \theta_j + \beta \cdot N_j(0, 1) \tag{8}$$

$$\vec{x}' = \vec{x} + \vec{N}(\vec{0}, \mathcal{C}(\vec{\sigma'}, \vec{\theta'})) \tag{9}$$

where τ and τ' are interpreted as 'learning rates' and are defined by Schwefel (Bäck 1996) as: $\tau = (\sqrt{2\sqrt{n}})^{-1}$ and $\tau' = (\sqrt{2n})^{-1}$ and $\beta \approx 0.0873$. $N_i(x, y)$ is a function that returns a real normal-distributed random number with mean x and standard deviation y. The index i indicates that this random number is generated anew for each decision variable (gene of the chromosome).

$\mathcal{C}(\vec{\sigma'}, \vec{\theta'})$ is the covariance matrix represented by the set of n stepsizes and the $n(n-1)/2$ rotation angles. The mutation on Equation 9 is implemented as follows: To calculate this $\vec{N}(\vec{0}, \mathcal{C}(\vec{\sigma'}, \vec{\theta'}))$, which represents the vector of stepsizes but now updated using correlated mutation (we call this vector $\vec{\sigma''}$) we perform the following: For each angle θ'_k, we calculate its corresponding two stepsize values in its corresponding axes σ'_i and σ'_j and we calculate the following: $\sigma_i^{''} = \sigma'_i \cdot \cos\theta_k - \sigma'_j \cdot \sin\theta_k$ and $\sigma_j^{''} = \sigma'_i \cdot \sin\theta_k + \sigma'_j \cdot \cos\theta_k$ (Schwefel 1995). In this way, the $\vec{\sigma''}$ values are now mutated in a correlated way and can be used to mutate the $\vec{x}$ vector of decision variables.

Some authors use correlated mutation, but it implies an extra computational effort to process the value of each angle and also to rotate the individual. Moreover, some extra memory space is needed to store all the different angles per individual (the angles are formed by the combination of all the axis based on the number of decision variables of the problem). If non-correlated mutation is preferred, the computational cost and the storage space for each individual get lower.

If a non-correlated mutation is used, the mutation expressions are:

$$\sigma_i' = \sigma_i \cdot exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)) \tag{10}$$

$$x_i' = x_i + \sigma_i' \cdot N_i(0,1) \tag{11}$$

The general ES algorithm is detailed in Figure 2.

[FIGURE 2 MUST BE LOCATED HERE]

It is important to note that the selection process in an evolution strategy takes place after all offspring have been generated. Some authors prefer to call it 'deterministic replacement', because only the best solutions will remain in the population. The worst ones have zero probabilities of surviving. Furthermore, the selection of parents to reproduce is performed randomly with a uniform probability distribution (all solutions have the same chance of being selected regardless of their fitness). In contrast, in other approaches like genetic algorithms, the selection process based on feasibility is performed when the parents are selected for reproduction.

In this work, we use a variation of the $(\mu + 1)$-ES, called by us as V$(\mu + 1)$-ES. Its pseudocode is presented in Figure 3.

[FIGURE 3 MUST BE LOCATED HERE].

The aim in our V$(\mu + 1)$-ES is to extend the use of a typical $(1 + 1)$-ES by increasing the capabilities of this current parent to generate better offspring. It works in the following way. Instead of using a population of μ solutions, just one solution (called parent) is considered. This only solution will generate μ mutations by using the traditional mutation operator (Gaussian Noise). After that, these mutations are combined into one single solution, which we call 'child', by using panmictic-discrete recombination. This child will be evaluated and also compared against the parent and the best between them will survive as the parent for the next generation.

The goal of the mutations is to explore more in-depth the neighborhood of the parent when generating its child. Furthermore, each mutation is not evaluated. This is to maintain the feature of evaluating just one new solution per generation (as in the original $(1+1)$-ES). It is also worth reminding that only one sigma value is used for all decision variables of the problem and for all solutions generated. Therefore, we use the '1/5' rule to dynamically update this only sigma value. To generate the offspring from the μ mutations we used a panmictic-discrete like recombination in the following way: For each variable of the child, we generate a uniform-distributed integer random number to select one of the μ mutations. The selected mutation will give its corresponding value to the child. We allow a parent to be selected more than once in the process. In Figure 4 there is a graphical explanation of the operator.

[FIGURE 4 MUST BE LOCATED HERE].

4 Constraint-handling approach

As it was pointed out in Section 1, we argue that the sampling mechanism of evolution strategies is useful to bias the evolutionary search through a constrained space. Hence, for our experiments, we will use neither any complex constraint handling mechanism nor a penalty function approach. In this way, just a simple comparison mechanism of three criteria based on feasibility and proposed by Deb (2000), is used to select the best individuals from one generation:

- Between 2 feasible solutions, the one with the higher fitness value is preferred.
- If one solution is feasible and the other one is infeasible, the feasible one is preferred.
- If both solutions are infeasible, the one with the lowest sum of constraint violation is preferred. This sum is calculated as: $\sum_{i=1}^{n} \max(0, g(\vec{x})) + \sum_{j=1}^{p} max(0, |h_j(\vec{x})| - \epsilon)$.

5 Experiments and results

We divided our experiments in four phases. Each part has an specific aim. The first phase consists on testing different version of evolution strategies (testing different types of mutation and selection operators) on a set of 10 benchmark problems. We did not test the recombination operator at all, because it is considered secondary (mutation is the main operator) in an ES. However, some comments are presented about it. The aim of this part was to know which ES provided the most competitive performance when solving a set of benchmark problems.

The second part of the experiments includes the comparison of the most competitive approach from the previous experiments with respect to three state-of-the-art approaches used to solve constrained problems adopting evolutionary algorithms. The goal of this part is to verify how good is our approach compared with a set of very competitive algorithms.

The third phase involved the comparison of our most competitive ES against an approach with the same simple constraint handling mechanism but using a genetic algorithm as a search engine. The objective of this experiment is to show the influence (positive in this case) of using an ES as a search engine in place of a genetic algorithm.

The final part of the experiments comprised the use of our ES now to solve real-world problems with constraints (engineering design problems in this case) and compare the provided results against state-of-the-art approaches adopted in engineering design. This final experiment will give us some insight about the applicability of our approach. In the first three parts of the experiments we decided to use a set of benchmark problems proposed in (Michalewicz & Schoenauer 1996). The detailed description of each test problem is provided in an appendix at the end of this paper.

To get an estimate of how difficult is to generate feasible solutions, a ρ measure (as suggested by Michalewicz & Schoenauer (1996)) was computed using the following expression:

$$\rho = |F|/|S| \tag{12}$$

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. This measure gives some insight about the ratio between the feasible region and the whole search space

.

[TABLE I MUST BE HERE].

The different 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 and NE is the number of nonlinear equalities. It can be clearly seen that in problems 1, 3, 5, 6 and 9 it is very difficult to generate feasible solutions and therefore the size of the feasible region seems to be very small with respect to the whole search space.

As we are not using a penalty function approach, we will use the terms objective function and fitness function interchangeably, because in our approach they are the same.

The number of evaluations of the objective function will be considered as a computational cost measure because it is commonly used in the specialized literature on evolutionary algorithms (Jin 2005) and also because its importance is indeed a research topic nowadays (Runarsson 2004, Won & Ray 2004). One of its advantages is that it is 'hardware-independent' (i.e. it does not depend of the computer's features where the algorithm is tested), which facilitates the comparison among different approaches, and it also stresses a point that is critical when using EAs: the high number of evaluations usually required by these types of heuristic approaches in order to achieve competitive results.

5.1 Experimental phase 1

We implemented five different types of ES:

- The variation of a $(\mu + 1)$-ES (V$(\mu + 1)$-ES)
- $(\mu + \lambda)$-ES without correlated mutation.
- $(\mu + \lambda)$-ES with correlated mutation.
- (μ, λ)-ES without correlated mutation.
- (μ, λ)-ES with correlated mutation.

The number of fitness function evaluations was fixed to 350000 for all 5 different ES. We performed 30 independent runs for each problem and for each type of ES. For test problems which have equality constraints, we used equation 4 with a tolerance value of $\epsilon = 0.0001$.

For the $(\mu + 1)$-ES, the initial values are:

- $\sigma = 4.0$.
- $c = 0.99$.
- $\mu = 5$.
- Number of generations = 350000.

These values, and also all values used for the approaches in the remaining experiments, were found empirically, always looking for the most competitive performance and also considering that each approach must perform the same number of evaluations of the objective function.

Nevertheless, it is important to note that the apparent high value of the initial σ and the value of $c = 0.99$ are set in order to allow a slow decrease in the value of sigma. In this way, the ES will be able to explore more the search space and the probability of being trapped in local optima is decreased.

The pseudocode of the algorithm used for the four remaining ES is presented in Figure 2. We used traditional panmictic discrete recombination for both, strategy parameters and decision variables. The learning rates values were calculated as shown in Section 3. The initial values for the stepsize (σ values) were 3.0 for all the decision variables for all solutions.

The initial values for the remaining ES are:

- $\mu = 100$.
- $\lambda = 300$.
- Number of generations = 1166.

The statistical results obtained (best, mean and worst solution found and also the corresponding standard deviation) for the five ES are summarized in Table II.

[TABLE II MUST BE LOCATED HERE].

The discussion of results, in this experiment and also in the remaining tests, will be based on quality and robustness of results. We measure the quality (accuracy) with the best value obtained by an approach in a set of independent runs. This is the best solution found by each approach (first row for each problem in Tables II, V, VI, VIII, IX and X. The robustness (precision) of each approach is measured by the mean and standard deviation values presented in the second and fourth rows per problem in Tables II, V, VI, VIII, IX and X.

With these two measures, based on statistical results, we may know which approach provides the best approximation to the best known solution (or global optimum) and also how often an approach is able to find solutions close to the optimum (or best known) solution.

From a pragmatic point of view, when using ES, as well as other evolutionary algorithms, it is important to know, based on a set of independent runs, their quality (accuracy) and robustness (precision). Sometimes, for an interested user, it is useful to have an approach which is able to find a very good solution, at least once in several runs, because the approach can be executed several times and the quality is the high-priority. On the other hand, the user should be interested in obtaining good (not necessarily very good) solutions consistently (robustness), because the evaluation of the objective function is either very expensive or time consuming, or maybe because several solutions are required.

Based on the previous comments, in this experiment, we seek for the ES with the best quality as a high priority and, as a second criterion, we look for a good robustness.

Problem P10 was excluded from discussion because all 5 ES reached the global optimum consistently in this problem. P10 is a problem with a low dimensionality (3 variables) whose feasible region is disjoint.

As a summary of Table II, in Table III we present the technique that provided the best approximation to the global optimum and also the technique which provided the most robust results

.

[TABLE III MUST BE LOCATED HERE].

In order to have more statistical support, we calculated the confidence intervals for the mean statistic for each of the five ES tested.

To verify if the distributions provided by the samples per test problem were close to a normal, we performed a one-sample Kolmogorov-Smirnov test for each sample for each function. In all cases the results showed that the distributions were not close to a normal one. After that, we performed a bootstrapping test with 1000 re-samples. Briefly, the aim of bootstrapping is to create several new samples by sampling with replacement (allowing a data to be repeated in the same resample) from the original sample. Each sample is of the same size of the original sample. Then the desired statistic is calculated for each resample. The distribution of these resample statistics is called a bootstrap distribution, which gives information about the shape, center and spread of the sampling distribution of the statistic. We used the Data-plot software. The obtained bootstrapping distributions were close to a normal. The summary of results with the confidence intervals for the mean statistic, with 95% confidence is presented in Table IV. Problem P10 was excluded because all five ES reached the global optimum in every single run.

[TABLE IV MUST BE LOCATED HERE].

5.1.1 Discussion of results

In order to allow a more reasonable discussion of results, we performed the following binary comparisons:

- Overall results obtained by each technique.
- '+' selection against ',' selection.
- Non-correlated against correlated mutation.
- Some findings about recombination.

Overall results obtained by each technique.

Based on the results in column 3 on Table II, the V($\mu+1$)-ES reached the global optimum in six problems (P1, P3, P4, P5, P7, P9) and it provided very good approximations to the global optimum in the remaining three (P2, P6 and P8). Besides, the V($\mu+1$)-ES provided the most robust values in three problems (P3, P5 and P7). The results obtained by the non-correlated $(\mu+\lambda)$-ES in column 4 on Table II show that this approach only reached the global optimum in one problem (P5), but it provided the best approximations to the best solution in three problems (P2 and P6 and P8). In addition, the non-correlated $(\mu+\lambda)$-ES was more robust in two problems (P2 and P6). The correlated $(\mu+\lambda)$-ES gave the following results (column 5 Table II: It reached the global optimum in two problems (P7 and P9). Also, it provided the most robust results in five problems (P1, P4, P7, P8 and P9). The results of the non-correlated (μ,λ)-ES in column 6 from Table II indicate that the approach only reached the global optimum in one problem (P7) and it failed to provide robust results in any other problem. Finally, the correlated (μ,λ)-ES only reached the global optimum in one problem (P7) and it also failed to provide robust results in any of the problems.

From these results (summarized in Table III we can state that the best approximations to the global optimum were provided by the V($\mu+1$)-ES followed by the non-correlated $(\mu+\lambda)$-ES. On the other hand, the most robust results were provided by the correlated $(\mu+\lambda)$-ES followed by the V($\mu+1$)-ES.

These results are supported by the confidence intervals for the mean statistic presented in Table IV, where the best intervals were obtained for the correlated $(\mu + \lambda)$-ES in five problems (P1, P4, P7, P8 and P9), followed by the V$(\mu + 1)$-ES, with best intervals in three problems (P3, P5 and P7).

From this comparison, and based in our criterion that quality is our main goal and robustness is a secondary goal, we obtained that the most competitive approach is the V$(\mu + 1)$-ES (it provided the best approximations to the global optimum and was the second more robust approach). However, it is interesting that in test functions where the V$(\mu + 1)$-ES could not find better results (P2, P6 and P8), the number of decision variables is higher than in problems where good results were found by this approach. The exception is P1, but the difference here is that P1 has only linear constraints. Those results may suggest that the V$(\mu + 1)$-ES has difficulties with high dimensionality problems coupled with nonlinear constraints.

'+' selection against ',' selection.

The overall results suggest that there is a clear superiority of the '+' selection over the ',' selection, because none of the two ES with ',' selection provided a good performance in any problem. From the results in Table II and the summary in Table III, we can observe that the non-correlated (μ, λ)-ES and the correlated (μ, λ)-ES only reached the best known solution in two problems (P7 and P10), and it is worth remarking that P10 was the easiest problem to solve by all five ES tested. Furthermore, the confidence intervals shown in Table IV show that none of the ',' selection ES were able to provide robust results in any given problem.

These results suggest that the implicit elitism that the '+' selection has, is important to help the ES to avoid losing the feasible solutions found. Despite the fact that it is well known that the ',' selection is less sensitive to get trapped in local optima (Schwefel 1995, Bäck 1996), in this experiment we can argue that elitism plays an important role in constrained optimization.

Non-correlated against correlated mutation

The results from Table II (columns 6 and 7) show no evidence about an improvement on neither the quality of approximation to the global optimum nor the robustness of the approach (confidence intervals in Table IV when using correlated mutation with ',' selection. In fact, for some problems the results are poorer than those obtained when using correlated mutation (P1, P2, P6, and P7).

For the case of '+' selection (implicit elitism), there is a slightly positive difference when using correlated mutation in terms of consistency (most robust results in problems P1, P4, P7, P8 and P9 in Table II and also better confidence intervals for the mean statistic for these problems in Table IV. However, it is not the same case when looking for quality results which are almost the same in all test problems.

We argue that these results suggest that the correlated mutation does improve the robustness of the evolutionary search in constrained spaces. However, the quality of results does not seem to get better. This issue is important (computationally speaking), because there is an extra computational cost and storage associated with the implementation of this type of mutation.

There is also evidence indicating that the comparison criteria explained in Section 4 added to the ‘,’ selection causes the search to be consistently trapped in local optimal solutions.

Recombination operator

The results obtained suggest that, for panmictic discrete recombination, the version used in the V$(\mu + 1)$-ES provided more quality results than those obtained by the recombination used in the multimembered ES. The main difference between these two operators is that the first (used in the V$(\mu + 1)$-ES) controls the number of parents which participate in the process by a user defined parameters (‘μ’). On the other hand, in the recombination used in the four multimembered ES, the number of parents depends of the dimensionality of the problem (‘$n + 1$’ parents are used , where ‘n’ is the number of decision variables of the problem).

This finding deserves more experimentation (i.e., to isolate the recombination operator in order to analyze carefully its behaviour) and it is suggested as a path for future work.

5.1.2 Some remarks

From the comparison of the five types of ES we can conclude the following:

- The most competitive approach, based on quality of results is the V$(\mu + 1)$-ES. It seems that the use of a dynamic approach to adapt the stepsize of the mutation is enough such as to provide quality results. In consequence, a large number of strategy parameters seems to cause difficulties to converge in constrained search spaces.
- The correlated $(\mu + 1)$-ES provided the most robust results. However, the closest approximations to the best known solutions were not as good as those provided by the V$(\mu + 1)$-ES.
- The elitism that the ‘+’ selection provides is more adequate to solve this set of constrained problems.
- The correlated mutation provides no significant improvements on the performance of an ES in constrained search spaces.

5.2 Experimental phase 2

In this phase, we compare the results of our V($\mu + 1$)-ES against three state-of-the art techniques based on evolutionary algorithms: Stochastic Ranking (SR) (Runarsson & Yao 2000), the Adaptive Segregational Constraint Handling Evolutionary Algorithm (ASCHEA) (Hamida & Schoenauer 2002) and the Self-Adaptive Fitness Formulation (SAFF) (Farmani & Wright 2003).

The aim of Stochastic Ranking (Runarsson & Yao 2000) 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. Instead, the selection process is based on a ranking process and a user-defined parameter called P_f that sets the probability of using only the objective function to compare two solutions when sorting them. The remaining comparisons will be performed using only the penalty function that consists, in this case, of the sum of constraint violation.

ASCHEA (Hamida & Schoenauer 2002) is based on three components: (1) an adaptive penalty function, (2) a constraint-driven recombination which: combines an infeasible solution with a feasible one and applies it when there is a low number of feasible solutions with respect to a pre-defined rate and (3) a segregational selection based on feasibility which allows to choose a defined ratio of feasible solutions based on their fitness to be part of the population for the next generation.

The Self-Adaptive Fitness Formulation (Farmani & Wright 2003) consists on applying a two-step adaptive penalty function. The aim of the approach is to assign a competitive fitness value to those slightly infeasible solutions with a good value of the objective function. The penalty function is calculated based on the objective function and the sum of constraint violation of the best solution in the population, the worst of the infeasible solutions and the solution with the worst value of the objective function.

SR and ASCHEA use an evolution strategy as a search engine. In contrast, SAFF uses a GA. The statistical results of SR are from a set of 30 independent runs, ASCHEA's are from a set of 31 independent runs and SAFF's are from a set of 20 independent runs. All results were taken from their corresponding publications.

The results of each approach compared with our V($\mu + 1$)-ES are summarized in Table V.

[TABLE V MUST BE LOCATED HERE].

5.2.1 Discussion of results

Now, we discuss the results of our approach against each of the three techniques used for comparison. As in the previous experiment, we eliminate problem P10 from discussion because all approaches found the global optimum consistently and the results from one of the approaches was not available.

SR against $V(\mu+1)$-ES

With respect to SR, the $V(\mu+1)$-ES obtained a 'similar' best result in seven problems (P1, P3, P4, P5, P7, and P9). It also provided a better mean result in problem P5 and a 'similar' mean results in two problems (P3, P7). In problem P5, the worst result found by our approach is better than that provided by SR. Except for problems P3, P5 and P7, the standard deviations provided by SR were smaller than those provided by our ES. We can observe that the $V(\mu+1)$-ES is able to provide similar 'best' results to those provided by SR, but it lacks the consistency shown by SR.

ASCHEA against $V(\mu+1)$-ES

With respect to ASCHEA, our $V(\mu+1)$-ES provided 'better' best results in three problems (P2, P4 and P5) and 'similar' best results in other four test functions (P1, P3, P8 and P9). Besides, our approach found 'better' mean results in four problems (P2, P3, P4, P5) and 'similar' mean results in two problems (P1 and P7). There is no comparison of worst results and standard deviation values because they were not available for ASCHEA. However, it is clear to see that our approach presented a very competitive performance (based on the best and mean results found) compared with that provided by ASCHEA.

SAFF against $V(\mu+1)$-ES

Compared with the SAFF, the $V(\mu+1)$-ES provided a 'better' best result in four problems (P4, P5, P6 and P8) and a 'similar' best result in other four test functions (P1, P3, P7, P9). Moreover, our approach found 'better' mean result in five problems (P3, P4, P5, P6 and P8) and a 'similar' mean result in problem P7. Finally, the $V(\mu+1)$-ES provided a 'better' worst result in four problems (P3, P4, P5 and P6) and 'similar' worst result in problem P7. From these results, we can see also a competitive performance by the $V(\mu+1)$-ES, and, sometimes a better performance by our technique.

5.2.2 Remarks

Our approach can deal with moderately constrained problems (P4), highly constrained problems, problems with low (P5, P7), moderated (P8) and high (P1, P2, P3, P6) dimensionality, with different types of combined constraints (linear, nonlinear, equality and inequality) and with very large (P2), very small (P1, P5 and P6) or even disjoint (P10) feasible regions. Also, the algorithm is able to deal with large search spaces, based on the intervals of the decision variables, (see Appendix A for details) and with a very small feasible region (P5 and P6). Furthermore, the approach can find the global optimum in problems where such optimum lies on the boundaries of the feasible region (P1, P2, P4, P5, P6 and P9). See the description of each test function where problems with active constraints are indicated.

It is important to mention that the V$(\mu + 1)$-ES presented a lack of consistency (based on the mean, worst and standard deviation values) in some test functions. We argue that this is due to the high selection pressure of the comparison mechanism used to deal with constraints. Infeasible solutions have no probability of surviving when compared with a feasible one. In this way, we can have a situation of an infeasible solution close to the boundaries of the feasible region and located near the global optimum, but this infeasible solution will be discarded when it is compared against a feasible one located far from the global optimum. Furthermore, the V$(\mu + 1)$-ES works only with one solution as starting point, which could make it sensitive to the region where this point is generated. This problem will be addressed in our future work.

Besides still being a very simple approach, it is worth reminding that the $V(\mu + 1)$-ES does not add any extra parameter due to the constraint handling mechanism adopted. In contrast, the SAFF (Farmani & Wright 2003) requires a parameter for the second part of its penalty function which the authors mention that it influences the performance of the approach. Stochastic ranking requires the definition of a parameter called P_f, whose value has an important impact on the performance of the approach (Runarsson & Yao 2000). 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 (Hamida & Schoenauer 2002).

The computational cost measured in terms of the number of fitness function evaluations (FFE) performed by any approach is at least equal for the V$(\mu + 1)$-ES with respect to the others to which it was compared. This is an additional (and important) advantage, mainly if we wish to use this approach for solving real-world problems. The V$(\mu + 1)$-ES performed $350,000$ FFE, the Stochastic Ranking performed also $350,000$ FFE, the SAFF performed $1,400,000$ FFE, and ASCHEA required $1,500,000$ FFE.

5.3 Experimental phase 3

In this experiment we want to show the positive influence of using an ES when solving constrained problems. Hence, we implemented a GA which uses exactly the same constraint handling technique of our $V(\mu+1)$-ES (and discussed in Section 4).

We chose a real-coded GA because of its encoding similarities with the ES. We also selected simulated binary crossover and parameter-based mutation because they are two of the most competitive operators for real-coded GAs when solving global optimization problems (Deb 2000). The selection process is by binary tournament selection using the comparison mechanism based on feasibility (see Section 4). The parameters were empirically chosen and are the following:

- Population size: 200
- Number of generations: 1750
- Crossover rate. 0.8
- Mutation rate. 0.6
- Number of total evaluations of the objective function: 350,000 (the same used by the $V(\mu+1)$-ES).

We tuned the GA parameters as to obtain the best performance so far. Also, we took care of promoting a fair comparison based on fixing the same number of fitness function evaluations for both approaches (ES and GA) and using the same constraint-handling mechanism applied in the selection process for each approach. The aim is to analyze, under similar conditions, the capabilities of each search engine to generate better solutions.

We performed 30 independent runs and the statistical results are summarized and compared against those provided by the $V(\mu+1)$-ES in Table VI.

It is very clear to see the better results obtained by the $V(\mu+1)$-ES over the GA: 'Better' best, mean and worst results in eight problems (P1, P2, P3, P4, P5, P6, P8 and P9). The GA only reaches a similar performance in two functions: P7 and P10 (which are the easiest to solve, based on the previous experiments). This experiment confirms the idea that the ES way of sampling constrained search spaces helps an EA to provide better results than using other types of search engine (a GA in our case), when using the constraint-handling mechanism explained in Section 4.

[TABLE VI MUST BE LOCATED HERE].

5.4 Experimental phase 4

To show the performance of the $V(\mu+1)$-ES when solving real-world problems, we tested it on three engineering design problems (Coello Coello 2000*a*).The details of the problems can be found in Appendix A at the end of this paper. The main features of each problem are detailed in Table VII.

[TABLE VII MUST BE LOCATED HERE].

We used the same set of parameters adopted in the previous experiments, except for the number of generations and the total number of evaluations of the fitness function. In this case we used 25000 generations and 25000 evaluations as well (at each generation, the $V(\mu+1)$-ES performs only one evaluation of the objective function).

This reduction in the number of generations was decided because of the fact that, after a trial-and-error process, we realized that, for these types of engineering design problems, the number of evaluations required for our approach to provide competitive results is lower than the one used in previous experiments. Furthermore, this number of evaluations makes the $V(\mu+1)$-ES competitive against the approaches of the state-of-the-art, as shown below.

The summary of the statistical results of 30 independent runs and a comparison against different state-of-the art techniques are provided in Table VIII for the welded-beam problem, in Table IX for the pressure vessel problem and in Table X for the tension-compression spring problem. We used for comparison four techniques taken from the literature. The Socio-Behavioral model (SB) (Ray & Liew 2003) and by Akhtar, Tai & Ray (2002). SB is a particle swarm optimization approach whose constraint handling mechanism is based on ranking the population using Pareto Dominance (Coello Coello, Van Veldhuizen & Lamont 2002). Deb's (2000) approach uses a GA-based approach, a similar selection mechanism to the one used in this work and a niching mechanism to help the approach to maintain diversity. The EMO approach by Coello Coello & Mezura-Montes (2002) uses Pareto dominance (Coello Coello et al. 2002) in a tournament selection to guide the search to the feasible region of the search space. Finally, we used a penalty approach proposed by Coello Coello (2000*b*) whose main feature is that penalty factors are self-adapted using an embedded GA inside the main GA which optimizes the solutions of the problem. We present the details of the best solution found for each engineering design problem in Tables XI and XII.

[TABLE VIII MUST BE LOCATED HERE].

[TABLE IX MUST BE LOCATED HERE].

[TABLE X MUST BE LOCATED HERE].

5.4.1 Discussion of results

As it can be seen in Table VIII, the $V(\mu+1)$-ES provided the most competitive results for the welded beam design using the lowest number of evaluations of the objective function. For the results of the pressure vessel design shown in Table IX, the best solution was provided by the $V(\mu+1)$-ES. However, the 'best' mean result was provided by the EMO approach and the 'best' worst result was provided by the Socio-Behavioral approach (whose number of evaluations was also the lowest). Finally, our $V(\mu+1)$-ES provided very competitive results against the Socio-Behavioral approach and the self-adaptive penalty approach for the spring design problem (see Table X). Furthermore, the number of evaluations required by our approach was the lowest. As a final conclusion for this experiment we can observe a competitive performance of $V(\mu+1)$-ES against evolutionary-based state-of-the-art approaches to solve engineering design problems. However, as mentioned before, the $V(\mu+1)$-ES presented some premature convergence to local optima also in two of these three real-world problems. In our discussion in Section 5.2.2, we argue that this undesired behaviour may be caused by the combination of two factors: (1) The high selection pressure of the constraint-handling mechanism and (2) the fact that the $V(\mu+1)$-ES is a single-membered ES and its exploration capabilities, for certain types of problems, may depend of the initial point which is generated at random.

[TABLE XI MUST BE LOCATED HERE].

[TABLE XII MUST BE LOCATED HERE].

6 Conclusions and future work

We have presented an empirical study to analyze the usefulness of using evolution strategies to solve constrained optimization problems. As a first experiment, we implemented and tested different types of ES in order to compare two types of selection mechanisms and also two types of mutation operators. The second part of our study consisted on comparing the most competitive ES of the first experiment, the $V(\mu + 1)$-ES, against three state-of-the-art approaches. The comparison showed a competitive performance of our approach despite a lack of robustness due to the inability of the approach to keep slightly infeasible solutions located in promising areas of the search space. However, it is worth reminding that the $V(\mu + 1)$-ES is very easy to implement (see Figure 3 for details) and it does not add any extra parameter to the ES and the computational cost required (measured by the number of evaluations of the objective function) was equal or lower than those required by the approaches used for comparison. Furthermore, the feasible region was reached in any single run for all test problems. In order to emphasize the positive influence of using an ES as a search engine, we compared our $V(\mu + 1)$-ES against a GA whose constraint handling approach was the same used in our ES. The results confirmed our idea. Finally, we tested the $V(\mu+1)$-ES on three engineering design problems and we compared the results against state-of-the-art approaches in the area. The results were very competitive at a very low computational cost (measured by the number of evaluations of the objective function). Our future paths of research consists of:

- Adding a diversity mechanism to the selection process which allows the $V(\mu + 1)$-ES to maintain slightly infeasible solutions located in promising areas in order to avoid convergence to local optimum solutions.
- Performing a comparison among other evolutionary algorithms (differential evolution (Price 1999), particle swarm optimization (Kennedy & Eberhart 2001)) in order to verify which one is the most competitive when dealing with constrained search spaces.
- Analyzing in more detail the effect of the recombination operator used in the ES implemented in this work.
- Solving problems in presence of a higher number of equality constraints. In this paper we solved problems with only one equality constraints (P3 and P9).

Acknowledgments

The first author acknowledges support from CONACyT through project number 52048-Y. The second author acknowledges support from CONACyT through project number 42435-Y.

Appendix A

The details of the thirteen test functions used in this work are the following:

1. **Problem 1**:

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

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

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

2. **Problem 2**:

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

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

where $n = 20$ and $0 \leq x_i \leq 10$ $(i = 1, \ldots, n)$. The global maximum is unknown; the best reported solution is (Runarsson & Yao 2000) $f(x^*) = 0.803619$. Constraint g_1 is close to being active ($g_1 = -10^{-8}$).

3. **Problem 3**:

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

   subject to:

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

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

4. **Problem 4**:

   Minimize: $f(\vec{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$

   subject to:

   $g_1(\vec{x}) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \leq 0$

   $g_2(\vec{x}) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0$

   $g_3(\vec{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leq 0$

   $g_4(\vec{x}) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0$

   $g_5(\vec{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0$

   $g_6(\vec{x}) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0$

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

5. **Problem 5**

   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.

6. **Problem 6**

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

   subject to:

   $g_1(\vec{x}) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0$

   $g_2(\vec{x}) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0$

   $g_3(\vec{x}) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0$

   $g_4(\vec{x}) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0$

   $g_5(\vec{x}) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0$

   $g_6(\vec{x}) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0$

   $g_7(\vec{x}) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \leq 0$

   $g_8(\vec{x}) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0$

   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.

7. **Problem 7**

   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 $f(x^*) = 0.095825$.

8. **Problem 8**

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

   subject to:

   $g_1(\vec{x}) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 0$

   $g_2(\vec{x}) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0$

   $g_3(\vec{x}) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0$

   $g_4(\vec{x}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0$

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

9. **Problem 9**

   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 $f(x^*) = 0.75$.

10. **Problem 10**

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

11. **Design of a Welded Beam**

    [FIGURE 5 MUST BE LOCATED HERE].

A welded beam is designed for minimum cost subject to constraints on shear stress (τ), bending stress in the beam (σ), buckling load on the bar (P_c), end deflection of the beam (δ), and side constraints (Rao 1996). There are four design variables as shown in Figure 5 (Rao 1996): h (x_1), l (x_2), t (x_3) and b (x_4).

The problem can be stated as follows:

Minimize:

$$f(\vec{x}) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2)$$

Subject to:

$$\begin{aligned}
g_1(\vec{x}) &= \tau(\vec{x}) - \tau_{max} \leq 0 \\
g_2(\vec{x}) &= \sigma(\vec{x}) - \sigma_{max} \leq 0 \\
g_3(\vec{x}) &= x_1 - x_4 \leq 0 \\
g_4(\vec{x}) &= 0.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \leq 0 \\
g_5(\vec{x}) &= 0.125 - x_1 \leq 0 \\
g_6(\vec{x}) &= \delta(\vec{x}) - \delta_{max} \leq 0 \\
g_7(\vec{x}) &= P - P_c(\vec{x}) \leq 0
\end{aligned}$$

where

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$$

$$\tau' = \frac{P}{\sqrt{2}x_1 x_2}, \tau'' = \frac{MR}{J}, M = P\left(L + \frac{x_2}{2}\right)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$

$$J = 2\left\{\sqrt{2}x_1 x_2 \left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$$\sigma(\vec{x}) = \frac{6PL}{x_4 x_3^2}, \delta(\mathbf{X}) = \frac{4PL^3}{E x_3^3 x_4}$$

$$P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$P = 6000\,lb,\;\; L = 14\,in,\;\; E = 30 \times 10^6\,psi,\;\; G = 12 \times 10^6\,psi$$

$$\tau_{max} = 13,600\, psi, \;\; \sigma_{max} = 30,000\, psi, \;\; \delta_{max} = 0.25\, in$$

where $0.1 \leq x_1 \leq 2.0$, $0.1 \leq x_2 \leq 10.0$, $0.1 \leq x_3 \leq 10.0$ y $0.1 \leq x_4 \leq 2.0$.

12. **Design of a Pressure Vessel**

    [FIGURE 6 MUST BE LOCATED HERE].

    A cylindrical vessel is capped at both ends by hemispherical heads as shown in Figure 6. The objective is to minimize the total cost, including the cost of the material, forming and welding. There are four design variables: T_s (thickness of the shell), T_h (thickness of the head), R (inner radius) and L (length of the cylindrical section of the vessel, not including the head). T_s and T_h are integer multiples of 0.0625 inch, which are the available thicknesses of rolled steel plates, and R and L are continuous. Using the same notation given by Kannan & Kramer (1994), the problem can be stated as follows:

    Minimize :

    $$f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

    Subject to :

    $$\begin{aligned} g_1(\vec{x}) &= -x_1 + 0.0193x_3 \leq 0 \\ g_2(\vec{x}) &= -x_2 + 0.00954x_3 \leq 0 \\ g_3(\vec{x}) &= -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \leq 0 \\ g_4(\vec{x}) &= x_4 - 240 \leq 0 \end{aligned}$$

    where $1 \leq x_1 \leq 99$, $1 \leq x_2 \leq 99$, $10 \leq x_3 \leq 200$ y $10 \leq x_4 \leq 200$.

13. **Minimization of the Weight of a Tension/Compression String**

    [FIGURE 7 MUST BE LOCATED HERE].

    This problem was described by Arora (1989) and Belegundu (1982), and it consists of minimizing the weight of a tension/compression spring (see Figure 7) subject to constraints on minimum deflection, shear stress, surge frequency, limits on outside diameter and on design variables. The design variables are the mean coil diameter D (x_2), the wire diameter d (x_1) and the number of active coils N (x_3).

Formally, the problem can be expressed as:

Minimize:

$$(N+2)Dd^2$$

Subject to:

$$\begin{aligned} g_1(\vec{x}) &= 1 - \frac{D^3N}{71785d^4} \leq 0 \\ g_2(\vec{x}) &= \frac{4D^2 - dD}{12566(Dd^3 - d^4)} + \frac{1}{5108d^2} - 1 \leq 0 \\ g_3(\vec{x}) &= 1 - \frac{140.45d}{D^2N} \leq 0 \\ g_4(\vec{x}) &= \frac{D+d}{1.5} - 1 \leq 0 \end{aligned}$$

where $0.05 \leq x_1 \leq 2$, $0.25 \leq x_2 \leq 1.3$ y $2 \leq x_3 \leq 15$.

References

Akhtar, S., Tai, K. & Ray, T. (2002), 'A Socio-behavioural simulation model for engineering design optimization', *Engineering Optimization* **34**(4), 341–354.

Arnold, D. V. (2002), *Noisy optimization with evolution strategies*, Kluwer Academic Publishers, New York. ISBN 1-4020-7105-1.

Arora, J. S. (1989), *Introduction to Optimum Design*, McGraw-Hill, New York.

Bäck, T. (1996), *Evolutionary algorithms in theory and practice*, Oxford University Press, New York.

Bäck, T., Hoffmeister, F. & Schwefel, H.-P. (1991), A survey of evolution strategies, *in* R. K. Belew & L. B. Booker, eds, 'Proceedings of the Fourth International Conference on Genetic Algorithms', Morgan Kaufmann Publishers, San Mateo, California, pp. 2–9.

Belegundu, A. D. (1982), A study of mathematical programming methods for structural optimization, Department of civil and environmental engineering, University of Iowa, Iowa.

Beyer, H.-G. (2001), *The theory of evolution strategies*, Springer, Berlin.

Coello Coello, C. A. (2000*a*), 'Constraint-handling using an evolutionary multiobjective optimization technique', *Civil Engineering and Environmental Systems* **17**, 319–346.

Coello Coello, C. A. (2000*b*), 'Use of a self-adaptive penalty approach for engineering optimization problems', *Computers in Industry* **41**(2), 113–127.

Coello Coello, C. A. (2002), '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.

Coello Coello, C. A. & Mezura-Montes, E. (2002), 'Constraint-handling in genetic algorithms through the use of dominance-based tournament selection', *Advanced Engineering Informatics* **16**(3), 193–203.

Coello Coello, C. A., Van Veldhuizen, D. A. & Lamont, G. B. (2002), *Evolutionary algorithms for solving multi-objective problems*, Kluwer Academic Publishers, New York. ISBN 0-3064-6762-3.

Deb, K. (2000), 'An efficient constraint handling method for genetic algorithms', *Computer Methods in Applied Mechanics and Engineering* **186**(2/4), 311–338.

Farmani, R. & Wright, J. A. (2003), 'Self-adaptive fitness formulation for constrained optimization', *IEEE Transactions on Evolutionary Computation* **7**(5), 445–455.

Fogel, L. J. (1999), *Intelligence through simulated evolution. Forty years of evolutionary programming*, John Wiley & Sons, New York.

Goldberg, D. E. (1989), *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley Publishing Co., Reading, Massachusetts.

Greenwood, G. W. & Liu, Y.-P. (1998), Finding low energy conformations of atomic clusters using evolution strategies, *in* V. W. Porto, N. Saravanan, D. Waagen & A. E. Eiben, eds, 'Evolutionary Programming VII', Springer, Berlin, pp. 493–502. Lecture Notes in Computer Science 1447.

Hamida, S. B. & Schoenauer, M. (2002), ASCHEA: new results using adaptive segregational constraint handling, *in* 'Proceedings of the Congress on Evolutionary Computation 2002 (CEC'2002)', Vol. 1, IEEE Service Center, Piscataway, New Jersey, pp. 884–889.

Homaifar, A., Lai, S. H. Y. & Qi, X. (1994), 'Constrained optimization via genetic algorithms', *Simulation* **62**(4), 242–254.

Jin, X. & Reynolds, R. G. (1999), Using knowledge-based evolutionary computation to solve nonlinear constraint optimization problems: a cultural algorithm approach, *in* ‘1999 Congress on Evolutionary Computation’, IEEE Service Center, Washington, D.C., pp. 1672–1678.

Jin, Y. (2005), ‘A comprehensive survey of fitness approximation in evolutionary computation’, *Soft Computing - A Fusion of Foundations, Methodologies and Applications* **9**(1), 3–12.

Joines, J. & Houck, C. (1994), On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GAs, *in* D. Fogel, ed., ‘Proceedings of the first IEEE Conference on Evolutionary Computation’, IEEE Press, Orlando, Florida, pp. 579–584.

Kannan, B. K. & Kramer, S. N. (1994), ‘An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design’, *Journal of Mechanical Design. Transactions of the ASME* **116**, 318–320.

Kennedy, J. & Eberhart, R. C. (2001), *Swarm intelligence*, Morgan Kaufmann, UK.

Koziel, S. & Michalewicz, Z. (1999), ‘Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization’, *Evolutionary Computation* **7**(1), 19–44.

Liepins, G. E. & Vose, M. D. (1990), ‘Representational issues in genetic optimization’, *Journal of Experimental and Theoretical Computer Science* **2**(2), 4–30.

Michalewicz, Z. (1996), *Genetic algorithms + data structures = evolution programs*, third edn, Springer-Verlag.

Michalewicz, Z. & Attia, N. F. (1994), Evolutionary optimization of constrained problems, *in* ‘Proceedings of the 3rd Annual Conference on Evolutionary Programming’, World Scientific, pp. 98–108.

Michalewicz, Z. & Nazhiyath, G. (1995), Genocop III: A co-evolutionary algorithm for numerical optimization with nonlinear constraints, *in* D. B. Fogel, ed., ‘Proceedings of the Second IEEE International Conference on Evolutionary Computation’, IEEE Press, Piscataway, New Jersey, pp. 647–651.

Michalewicz, Z. & Schoenauer, M. (1996), ‘Evolutionary algorithms for constrained parameter optimization problems’, *Evolutionary Computation* **4**(1), 1–32.

Price, K. V. (1999), An introduction to differential evolution, *in* D. Corne, M. Dorigo & F. Glover, eds, ‘New Ideas in Optimization’, Mc Graw-Hill, UK, pp. 79–108.

Rao, S. S. (1996), *Engineering optimization*, third edn, John Wiley and Sons.

Rasheed, K. (1998), An adaptive penalty approach for constrained genetic-algorithm optimization, *in* J. R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M. H. Garzon, D. E. Goldberg, H. Iba & R. L. Riolo, eds, 'Proceedings of the Third Annual Genetic Programming Conference', Morgan Kaufmann Publishers, San Francisco, California, pp. 584–590.

Ray, T. & Liew, K. (2003), 'Society and civilization: An optimization algorithm based on the simulation of social behavior', *IEEE Transactions on Evolutionary Computation* **7**(4), 386–396.

Rechenberg, I. (1965), 'Cybernetic solution path of an experimental problem', *Royal Aircraft Establishment* . Library Translation No. 1122, Farnborough, Hants, UK.

Rechenberg, I. (1973), *Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution*, Frommann-Holzboog, Stuttgart.

Riche, R. G. L., Knopf-Lenoir, C. & Haftka, R. T. (1995), A segregated genetic algorithm for constrained structural optimization, *in* L. J. Eshelman, ed., 'Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA-95)', University of Pittsburgh, Morgan Kaufmann Publishers, San Mateo, California, pp. 558–565.

Runarsson, T. P. (2004), Constrained evolutionary optimization by approximate ranking and surrogate models, *in* X. Yao, E. Burke, J. A. Lozano, J. Smith, , J. J. Merelo-Guervós, J. A. Bullinaria, J. Rowe, P. Tiňo, A. Kabán & H.-P. Schwefel, eds, 'Proceedings of 8th Parallel Problem Solving From Nature (PPSN VIII)', Birmingham, UK, Springer-Verlag, Heidelberg, Germany, pp. 401–410. Lecture Notes in Computer Science Vol. 3242.

Runarsson, T. P. & Yao, X. (2000), 'Stochastic ranking for constrained evolutionary optimization', *IEEE Transactions on Evolutionary Computation* **4**(3), 284–294.

Schoenauer, M. & Michalewicz, Z. (1996), Evolutionary computation at the edge of feasibility, *in* H.-M. Voigt, W. Ebeling, I. Rechenberg & H.-P. Schwefel, eds, 'Proceedings of the Fourth Conference on Parallel Problem Solving from Nature (PPSN IV)', Berlin, Germany, Springer-Verlag, Heidelberg, Germany, pp. 245–254.

Schwefel, H. (1968), Projekt MHD-staustrahlrohr: experimentelle optimierung einer zweiphasendüse, teil I, Technical Report Technischer Bericht 11.034/68, 35, AEG Forschungsinstitut, Berlin.

Schwefel, H.-P. (1981), *Numerical optimization of computer models*, John Wiley & Sons, Great Britain.

Schwefel, H.-P., ed. (1995), *Evolution and optimization seeking*, John Wiley & Sons, New York.

Schweitzer, F., Ebeling, W., Rosé, H. & Weiss, O. (1998), 'Optimization of road networks using evolutionary strategies', *Evolutionary Computation* **5**(4), 419–438.

Smith, A. E. & Coit, D. W. (1997), Constraint handling techniques—penalty functions, *in* T. Bäck, D. B. Fogel & Z. Michalewicz, eds, 'Handbook of Evolutionary Computation', Oxford University Press and Institute of Physics Publishing, chapter C 5.2.

Wah, B. W. & Chen, Y. (2001), Hybrid constrained simulated annealing and genetic algorithms for nonlinear constrained optimization, *in* 'Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001)', Vol. 2, IEEE Service Center, Piscataway, New Jersey, pp. 925–932.

Won, K.-S. & Ray, T. (2004), Performance of kriging and cokriging based surrogate models within the unified framework for surrogate assisted optimization, *in* 'Proceedings of the Congress on Evolutionary Computation 2004 (CEC'2004)', Vol. 2, Portland, Oregon, USA, IEEE Service Center, Piscataway, New Jersey, pp. 1577–1585.

Wu, B. & Yu, X. (2001), Fuzzy penalty function approach for constrained function optimization with evolutionary algorithms, *in* 'Proceedings of the 8th International Conference on Neural Information Processing', Fudan University Press, Shanghai, China, pp. 299–304.

Table captions

Table I: Main features of the ten problems used in experiments 1 to 3.

Table II: Statistical results obtained by the 5 types of ES. A result in **boldface** means a better (or best) solution obtained. ‘-’ means no feasible solutions were found.

Table III: Most competitive techniques by problem (best approximation to the best known solution (or global optimum) and more robust approach (based on statistical results).

Table IV: 95%-confidence intervals obtained for the 5 types of ES. A result in **boldface** means a better interval obtained. ‘-’ means no feasible solutions were found in the original sample. ‘BKS’ means Best Known Solution per problem.

Table V: Comparison of results of our V$(\mu + 1)$-ES against state-of-the-art approaches. A result in **boldface** means a better (or best) solution obtained. NA means not available.

Table VI: Comparison of results of our V$(\mu+1)$-ES against a GA with the same constraint handling technique. A result in **boldface** means a better (or best) solution obtained.

Table VII: Main features of the 3 engineering design problems.

Table VIII: Comparison of results for the welded beam design problem. A result in **boldface** means that a better solution was obtained.

Table IX: Comparison of results for the pressure vessel design problem. A result in **boldface** means that a better solution was obtained.

Table X: Comparison of results for the spring design problem. A result in **boldface** means that a better solution was obtained.

Table XI: Summary of best results found by each approach compared for the first two engineering design problems in experimental phase 4. All solutions are feasible.

Table XII: Summary of best results found by each approach compared for the last engineering design problem in experimental phase 4. All solutions are feasible.

Tables on individuals pages

Problem	**n**	**Type of function**	ρ	**LI**	**NI**	**NE**
1	13	quadratic	0.0003%	9	0	0
2	20	nonlinear	99.9973%	2	0	0
3	10	nonlinear	0.0026%	0	0	1
4	5	quadratic	27.0079%	4	2	0
5	2	nonlinear	0.0057%	0	2	0
6	10	quadratic	0.0000%	3	5	0
7	2	nonlinear	0.8581%	0	2	0
8	7	nonlinear	0.5199%	0	4	0
9	2	quadratic	0.0973%	0	0	1
10	3	quadratic	4.7697%	0	9^3	0

Problem &		**Different ES tested**				
Best Known Sol.	**Stats**	$V(\mu+1)$**-ES**	$(\mu+\lambda)$**-ES**	$(\mu+\lambda)$**-ES Corr**	(μ,λ)**-ES**	(μ,λ)**-ES Corr**
	best	**−15.000**	−14.986	−14.999	−14.995	−14.931
P1	mean	−14.840	−14.974	**−14.998**	−14.971	−14.915
−15.000	worst	−12.999	−14.954	**−14.973**	−14.931	−14.889
	St. Dev	4.1E-1	7.79E-3	4.62E-3	1.56E-2	**9.78E-4**
	best	0.793083	**0.803607**	0.803594	0.792393	0.797201
P2	mean	0.698932	**0.800743**	0.796618	0.779795	0.777913
0.803619	worst	0.576079	**0.792375**	0.785246	0.753796	0.748130
	St. Dev	4.1E-1	**4.64E-3**	5.86E-3	1.20E-2	1.25E-2
	best	**1.000**	0.474	0.472	0.465	0.445
P3	mean	**1.000**	0.238	0.202	0.165	0.108
1.000	worst	**1.000**	0.027	0.086	0.007	0.000
	St. Dev	**1.4E-5**	1.14E-1	1.00E-1	1.34E-1	1.40E-1
	best	**−30665.539**	−30664.838	−30665.529	−30432.131	−30664.217
P4	mean	−30665.442	−30651.001	**−30665.520**	−30309.273	−30662.855
−30665.539	worst	−30663.496	−30619.619	**−30665.508**	−30204.131	−30661.170
	St. Dev	3.9E-1	13.16E+0	**5.17E-3**	52.56E+0	7.72E-1
	best	**−6961.814**	**−6961.814**	−6961.761	−6916.590	−6802.235
P5	mean	**−6961.814**	−6938.453	−6960.628	−6711.116	−6538.026
−6961.814	worst	**−6961.814**	−6567.754	−6957.259	−6068.743	−6277.651
	St. Dev	**0**	83.16E+0	1.15E+0	206.01E+0	127.24E+0
	best	24.368	**24.329**	24.330	24.484	24.651
P6	mean	24.703	**24.391**	24.422	24.929	24.887
24.306	worst	25.517	**24.478**	24.563	25.485	25.238
	St. Dev	2.4E-1	**4.67E-2**	6.52E-2	2.71E-1	1.42E-1
	best	**0.095825**	**0.095825**	**0.095825**	**0.095825**	**0.095825**
P7	mean	**0.095825**	0.095823	**0.095825**	**0.095825**	0.095822
0.095825	worst	**0.095825**	0.095771	**0.095825**	0.095821	0.095811
	St. Dev	**0**	1.0E-5	**0**	1.0E-6	4.0E-6
	best	680.632	**680.631**	680.633	680.809	680.775
P8	mean	680.674	680.640	**680.638**	681.351	681.139
680.63	worst	680.915	680.666	**680.645**	682.871	681.498
	St. Dev	5.2E-2	1.04E-2	**2.70E-3**	4.85E-1	1.43E-1
	best	**0.75**	0.751	**0.75**	–	0.88
P9	mean	0.78	0.88	**0.752**	–	0.95
0.75	worst	0.88	0.99	**0.81**	–	0.99
	St. Dev	3.73E-2	8.53E-2	**1.13E-2**	–	2.80E-2
	best	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
P10	mean	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
1.000	worst	**1.000**	0.999	**1.000**	**1.000**	**1.000**
	St. Dev	**0**	1.0E-6	**0**	**0**	**0**

Problem	Best approximation	Most robust
P1	V$(\mu+1)$-ES	$(\mu+\lambda)$-ES Corr
P2	$(\mu+\lambda)$-ES Non-corr	$(\mu+\lambda)$-ES Non-corr
P3	V$(\mu+1)$-ES	V$(\mu+1)$-ES
P4	V$(\mu+1)$-ES	$(\mu+\lambda)$-ES Corr
P5	V$(\mu+1)$-ES	V$(\mu+1)$-ES
P5	$(\mu+\lambda)$-ES Non-corr	
P6	$(\mu+\lambda)$-ES Non-corr	$(\mu+\lambda)$-ES Non-corr
P7	V$(\mu+1)$-ES $(\mu+\lambda)$-ES Corr (μ,λ)-ES Non-Corr (μ,λ)-ES Corr	V$(\mu+1)$-ES $(\mu+\lambda)$-ES Corr
P8	$(\mu+\lambda)$-ES Non-corr	$(\mu+\lambda)$-ES Corr
P9	V$(\mu+1)$-ES $(\mu+\lambda)$-ES Corr	$(\mu+\lambda)$-ES Corr
P10	all approaches	all approaches

P. & BKS	**Different ES tested**				
	$V(\mu+1)$**-ES**	$(\mu+\lambda)$**-ES**	$(\mu+\lambda)$**-ES Corr**	(μ,λ)**-ES**	(μ,λ)**-ES Corr**
P1 -15.000	[-14.984,-14.755]	[-14.971, -14.965]	**[-14998,-14.996]**	[-14.982,-14.974]	[-14.918,-14.910]
P2 0.803619	[0.645508,0.723947]	**[0.799717,0.801954]**	[0.794875,0.800625]	[0.772763,0.780614]	[0.770939,0.780263]
P3 1.000	**[1.000,1.000]**	[0.812,0.885]	[0.174660,0.283622]	[0.172,0.311]	[0.061,0.173]
P4 -30665.539	[-30665.539,-30665.480]	[-30652.140,-30640.620]	**[-30665.520,-30665.520]**	[-30318.030,30269.180]	[-30663.340,-30662.830]
P5 -6961.814	**[-6961.814,-6961.814]**	[-6948.833,-6842.289]	[-6960.879,-6959.812]	[6777.012,6669.917]	[-6540.594,-6458.280]
P6 24.306	[24.641,24.904]	**[24.374,24.417]**	[24.405,24.466]	[24.738,24.969]	[24.824,24.921]
P7 0.095825	**[0.095825,0.095825]**	[0.095820,0.095825]	**[0.095825,0.095825]**	[0.095822,0.095825]	[0.095820,0.095823]
P8 680.63	[680.676,680.741]	[680.650,680.696]	**[680.638,680.642]**	[681.141,681.453]	[681.136,681.247]
P9 0.75	[0.76,0.79]	[0.79,0.84]	**[0.75,0.76]**	-	[0.93,0.96]

Problem &		**State-of-the-art approaches compared**			
Best Known Sol.	**Stats**	**SR**	**ASCHEA**	**SAFF**	$V(\mu+1)$**-ES**
	best	**−15.000**	**−15.000**	**−15.000**	**−15.000**
P1	mean	**−15.000**	−14.840	**−15.000**	−14.840
−15.000	worst	**−15.000**	NA	**−15.000**	−12.999
	St. Dev	**0**	NA	**0**	4.1E-1
	best	**0.803515**	0.785000	0.802970	0.793083
P2	mean	0.781975	0.590000	**0.790100**	0.698932
0.803619	worst	0.726288	NA	**0.760430**	0.576079
	St. Dev	2.0E-2	NA	**1.2E-2**	4.1E-1
	best	**1.000**	**1.000**	**1.000**	**1.000**
P3	mean	**1.000**	0.99989	0.9999	**1.000**
1.000	worst	**1.000**	NA	0.9997	**1.000**
	St. Dev	1.9E-4	NA	7.5E-5	**1.4E-5**
	best	**−30665.539**	−30665.500	−30665.500	**−30665.539**
P4	mean	**−30665.539**	−30665.500	−30663.200	−30665.442
−30665.539	worst	**−30665.539**	NA	−30663.300	−30663.496
	St. Dev	**2.0E-5**	NA	4.85E-1	3.9E-1
	best	**−6961.814**	−6961.810	−6961.800	**−6961.814**
P5	mean	−6875.940	−6961.810	−6961.800	**−6961.814**
−6961.814	worst	−6350.262	NA	−6961.800	**−6961.814**
	St. Dev	1.6E+2	NA	**0**	**0**
	best	**24.307**	24.332	24.480	24.368
P6	mean	**24.374**	24.660	26.580	24.703
24.306	worst	**24.642**	NA	28.400	25.517
	St. Dev	**6.6E-2**	NA	1.14E+0	2.4E-1
	best	**0.095825**	**0.095825**	**0.095825**	**0.095825**
P7	mean	**0.095825**	**0.095825**	**0.095825**	**0.095825**
0.095825	worst	**0.095825**	NA	**0.095825**	**0.095825**
	St. Dev	2.6E-17	NA	**0**	**0**
	best	**680.630**	**680.630**	680.640	680.632
P8	mean	680.656	**680.641**	680.720	680.674
680.63	worst	**680.763**	NA	680.870	680.915
	St. Dev	**3.4E-2**	NA	5.92E-2	5.2E-2
	best	**0.75**	**0.75**	**0.75**	**0.75**
P9	mean	**0.75**	**0.75**	**0.75**	0.78
0.75	worst	**0.75**	NA	**0.75**	0.88
	St. Dev	8.0E-5	NA	**0**	3.73E-2
	best	**1.000**	NA	**1.000**	**1.000**
P10	mean	**1.000**	NA	**1.000**	**1.000**
1.000	worst	**1.000**	NA	**1.000**	**1.000**
	St. Dev	**0**	NA	**0**	**0**

Problem &		**GA VS ES**	
Best Known Sol.	**Stats**	**GA**	$V(\mu+1)$**-ES**
	best	−5.727	**−15.000**
P1	mean	−4.600	**−14.840**
−15.000	worst	−4.090	**−12.999**
	St. Dev	**3.25E-1**	4.1E-1
	best	0.630084	**0.793083**
P2	mean	0.505746	**0.698932**
0.803619	worst	0.439669	**0.576079**
	St. Dev	**6.1E-2**	4.1E-1
	best	0.967	**1.000**
P3	mean	0.853	**1.000**
1.000	worst	0.711	**1.000**
	St. Dev	6.6E-2	**1.4E-5**
	best	−30365.748	**−30665.539**
P4	mean	−30004.441	**−30665.442**
−30665.539	worst	−29721.688	**−30663.496**
	St. Dev	2.01E+2	**3.9E-1**
	best	−6961.057	**−6961.814**
P5	mean	−6953.089	**−6961.814**
−6961.814	worst	−6939.063	**−6961.814**
	St. Dev	6.1E+0	**0**
	best	25.321	**24.368**
P6	mean	27.988	**24.703**
24.306	worst	35.559	**25.517**
	St. Dev	2.3E+0	**2.4E-1**
	best	**0.095825**	**0.095825**
P7	mean	**0.095825**	**0.095825**
0.095825	worst	**0.095825**	**0.095825**
	St. Dev	**0**	**0**
	best	680.853	**680.632**
P8	mean	681.199	**680.674**
680.63	worst	681.767	**680.915**
	St. Dev	2.4E-1	**5.2E-2**
	best	0.753	**0.75**
P9	mean	0.90	**0.78**
0.75	worst	0.99	**0.88**
	St. Dev	6.2E-2	**3.73E-2**
	best	**1.000**	**1.000**
P10	mean	**1.000**	**1.000**
1.000	worst	**1.000**	**1.000**
	St. Dev	**0**	**0**

Problem	**n**	**Type of function**	ρ	**LI**	**NI**	**LE**	**NE**
Welded beam	4	quadratic	39.6762%	3	1	0	0
Pressure vessel	4	quadratic	2.6859%	6	1	0	0
Spring	3	quadratic	0.7537%	1	3	0	0

Problem &		**Engineering design approaches compared**		
Best Known Sol.	**Stats**	**SB**	**Deb**	$V(\mu+1)$**-ES**
Welded beam	best	2.385435	2.38119	$\mathbf{1.737300}$
	mean	3.255137	2.39289	$\mathbf{1.813290}$
	worst	6.399679	2.64583	$\mathbf{1.994651}$
	St. Dev	9.59E-1	NA	$\mathbf{7.05}$E-$\mathbf{2}$
	evaluations	33095	40080	$\mathbf{25000}$

Problem & Best Known Sol.		**Engineering design approaches compared**		
	Stats	**SB**	**EMO approach**	$V(\mu+1)$-**ES**
Pressure vessel	best	6171.000000	6059.946341	**6059.745605**
	mean	6335.050000	**6177.253268**	6850.004948
	worst	**6453.650000**	6469.322010	7332.879883
	St. Dev	NA	**13.09**E+**1**	4.26E+2
	evaluations	**12630**	80000	25000

Problem & Best Known Sol.	**Stats**	**Engineering design approaches compared**		
		SB	**Coello**	$V(\mu+1)$**-ES**
Spring	best	$\mathbf{0.012669}$	0.012705	0.012698
	mean	0.012923	$\mathbf{0.012769}$	0.013461
	worst	0.016717	$\mathbf{0.012822}$	0.016485
	St. Dev	5.92E-4	NA	9.66E-4
	evaluations	25167	900000	$\mathbf{25000}$

	Welded beam			**Pressure vessel**		
	SB	**Deb**	$V(\mu+1)$**-ES**	**SB**	**EMO**	$V(\mu+1)$**-ES**
x_1	0.244438	NA	0.199742	0.8125	0.8125	0.8125
x_2	6.237967	NA	3.612060	0.4375	0.4375	0.4375
x_3	8.288576	NA	9.037500	41.9768	42.097398	42.098087
x_4	0.244566	NA	0.206082	182.2845	176.654047	176.640518
$f(x)$	2.385435	2.38119	**1.737300**	6171.000	6059.946341	**6059.745605**

	Tension/Compression Spring		
	SB	**Coello**	$V(\mu+1)$**-ES**
x_1	0.368159	0.351661	0.355360
x_2	0.052160	0.051480	0.051643
x_3	10.648442	11.632201	11.397926
$f(x)$	**0.012669**	0.012705	0.012698

Figure captions

Figure 1: Representation of individuals of a genetic algorithm and an evolution strategy.

Figure 2: ES general algorithm.

Figure 3: Algorithm of the V$(\mu + 1)$-ES. Function best(x,y) selects the best solution between x and y using the comparison mechanism based on feasibility discussed in Section 4.

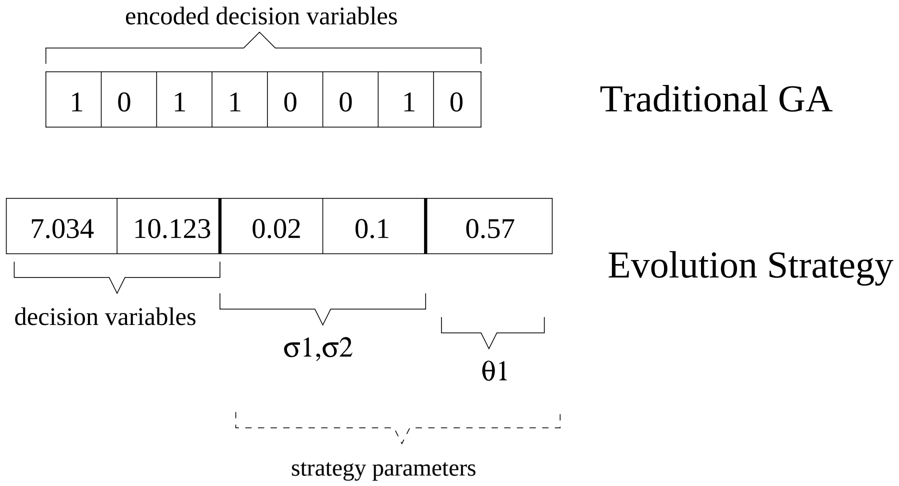
Figure 4: Recombination operator used to generate one child from the μ mutations in our V$(\mu + 1)$-ES.

Figure 5: The welded beam used for problem 11.

Figure 6: Center and end section of the pressure vessel used for problem 12.

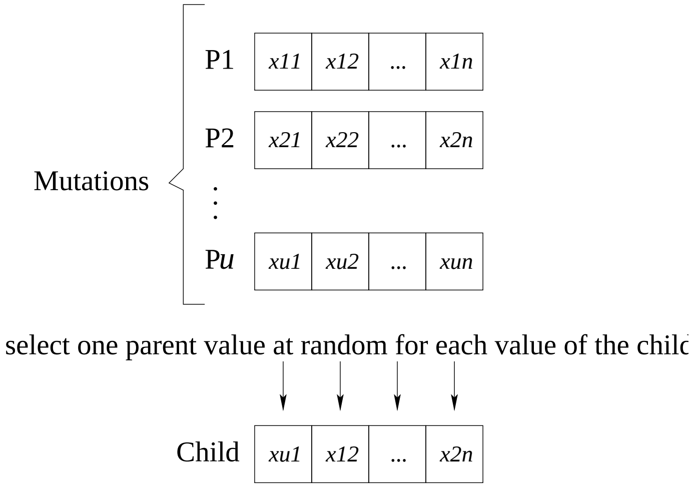
Figure 7: Tension/compression string used for problem 13.

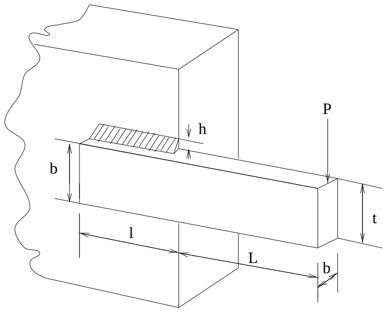
Figures on individual pages

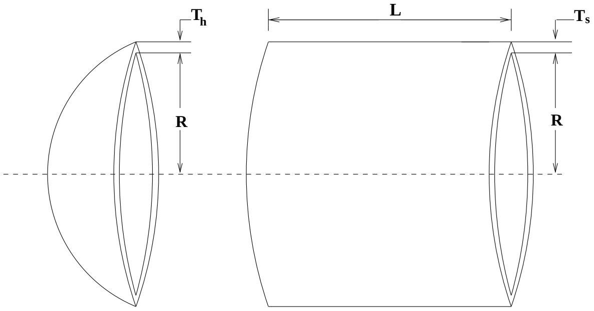

encoded decision variables
1 0 1 1 0 0 1 0
Traditional GA
7.034
10.123
0.02
0.1
0.57
Evolution Strategy
decision variables
σ1,σ2
θ1
strategy parameters

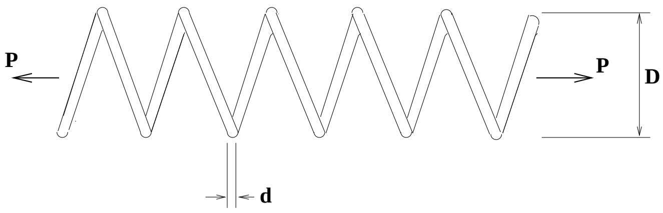
```
Begin
    t=0
    Create $\mu$ random solutions for the initial population.
    Evaluate all $\mu$ individuals
    Assign a fitness value to all $\mu$ individuals
    For t=1 to MAX_GENERATIONS Do
        Produce $\lambda$ offspring by recombination of the $\mu$ parents
        Mutate each child (with or without correlated mutation)
        Evaluate all $\lambda$ offspring
        Assign a fitness value to all $\lambda$ individuals
        If Selection = '+' Then
            Select the best $\mu$ individuals from the $\mu + \lambda$ individuals
        Else
            Select the best $\mu$ individuals from the $\lambda$ individuals
        End If
    End For
End
```

```
Begin
    t=0
    Create a random solution $x(t)$.
    Evaluate $x(t)$
    For t=1 to MAX_GENERATIONS Do
        Produce $\mu$ solutions of $x(t)$ by mutation
        Create one offspring $x'$ from the $\mu$ solutions using
        panmictic-discrete like recombination
        Evaluate $x'$
        $x(t+1) = best(x(t), x')$
        Use the '1/5' rule to adapt the sigma value
    End For
End
```


Mutations
P1
x11
x12
...
x1n
P2
x21
x22
...
x2n
Pu
xu1
xu2
...
xun
select one parent value at random for each value of the chilc
Child
xu1
x12
...
x2n


P
h
b
t
l
L
b


T_h
L
T_s
R
R


P
P
D
d

Authors' Biographies

Efrén Mezura-Montes received the B.Sc. in computer systems engineering from the Universidad de las Américas Puebla, in Puebla, México in 1997, a M.Sc. degree in artificial intelligence from the Universidad Veracruzana, in Xalapa, Veracruz, México in 2002 and a PhD degree in computer science from the Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV-IPN), in México City in 2004.

He is currently a full-time researcher at the Laboratorio Nacional de Informática Avanzada (LANIA A.C.) in Xalapa, Veracruz, México. Dr. Mezura-Montes has authored and co-authored over 30 technical papers and some book chapters. He has also served in the program committees of about 8 international conferences and has been technical reviewer for over 10 international journals (IEEE Transactions on Evolutionary Computation, International Journal for Numerical Methods in Engineering, Structural and Multidisciplinary Optimization, among others). He is member of the IEEE and the Mexican National System of Researchers (SNI) Level 1.

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

Carlos A. Coello Coello received the B.Sc. degree in civil engineering from the Universidad Autónoma de Chiapas, México, and the M.Sc. and the PhD degrees in computer science from Tulane University, USA, in 1991, 1993, and 1996, respectively.

He is currently a full professor (CINVESTAV-3D Researcher) and chair of the computer science department of CINVESTAV-IPN, in Mexico City, México. Dr. Coello has authored and co-authored over 150 technical papers and several book chapters. He has also co-authored the book *Evolutionary Algorithms for Solving Multi-Objective Problems* (Kluwer Academic Publishers, 2002). Additionally, Dr. Coello has served in the program committees of over 40 international conferences and has been technical reviewer for over 40 international journals. He is associate editor of the journals *IEEE Transactions on Evolutionary Computation*, *Evolutionary Computation*, and the *Journal of Heuristics*, and is an Editorial Board member of the journals *Soft Computing*, *Engineering Optimization* and *Computational Optimization and Applications*. He is a senior member of the IEEE, and a regular member of the Mexican Academy of Sciences.

His major research interests are: evolutionary multi-objective optimization, constraint-handling techniques for evolutionary algorithms, and evolvable hardware.