Evolutionary Algorithms Approach to the Solution of Mixed Integer Non-Linear Programming Problems

Lino Costa and **Pedro Oliveira***

Dept. Production and Systems Engineering

University of Minho, 4710 Braga, Portugal

Abstract

The global optimization of mixed integer non-linear problems (MINLP), constitutes a major area of research in many engineering applications. In this work, a comparison is made between an algorithm based on Simulated Annealing (M-SIMPSA) and two Evolutionary Algorithms: Genetic Algorithms (GAs) and Evolution Strategies (ESs). Results concerning the handling of constraints, through penalty functions, with and without penalty parameter setting, are also reported. Evolutionary Algorithms seem a valid approach to the optimization of non-linear problems. Evolution Strategies emerge as the best algorithm in most of the problems studied.

Keywords: Genetic Algorithms, Mixed integer non-linear programming

*E-mail: {lac,pno}@dps.uminho.pt; Fax: +351 253604741

1 Introduction

Real world problems can, in general, be formulated as mixed integer non-linear programming problems (MINLP). These problems, due to their combinatorial nature, are considered difficult problems. Gradient optimization techniques have only been able to tackle special formulations, where continuity or convexity had to be imposed, or by exploiting special mathematical structures. Approaches based on stochastic algorithms have been used. These approaches, also know as adaptive random search, have successfully tackle MINLP, mostly in the area of chemical engineering (Reklaitis et al. 1983, Salcedo 1992, Banga and Sneider 1996). In recent years, a vast amount of work has been published on applications of evolutionary algorithms (Genetic Algorithms, Evolution Strategies, Simulated Annealing, etc.) to the solution of MINLP in many engineering applications. These algorithms are distinct from conventional algorithms since, in general, only the information regarding the objective function is required. Moreover, they start from a pool of points that evolves over time, possibly in the direction of the optimum. It should be stressed that the objective of this on going research is not to find the "best" algorithm for all problem instances, but to compare different algorithms, in order to find out classes of problems which may be more suitable for certain algorithms than others.

The general formulation of the problem is as follows:

$$\min f(\mathbf{x}) \tag{1}$$

subject to

$$\mathbf{h}(\mathbf{x}) = \mathbf{0}$$

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$$

$$x_j \text{ integer }, j \in I$$

$$\mathbf{x} \in X = \left\{\mathbf{x} \mid \mathbf{x} \in R^n, \mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^u\right\}$$

where

$$\mathbf{h} \in R^m, \mathbf{g} \in R^p.$$

In this work, 7 test problems, proposed by independent authors, are studied using Genetic Algorithms (GAs) and Evolution Strategies (ESs). These problems arise from the area of chemical engineering, and represent difficult nonconvex optimization problems, with continuous and discrete variables. Comparisons are made with an M-SIMPSA algorithm (Cardoso et al.1996a, 1996b, 1997, Cardoso 1998) based on the combination of the non-linear simplex method of Nelder and Mead and Simulated Annealing.

2 Description of the Genetic Algorithm

2.1 Introduction

Genetic Algorithms are search algorithms that mimic the process of natural selection (Goldberg 1989). Thus, unlike conventional algorithms, GAs start from a pool of points, usually

referred to as chromosomes. In order to implement a simple GA it is necessary to define the representation of the search space (usually a binary representation) and an evaluation function which permits the comparison between the different chromosomes in terms of their fitness. New points in the search space are generated by the application of genetic operators, thus originating a new generation of points.

Figure 1 presents the GA flow chart. An initial population of bit strings (chromosomes) is generated. Each of these strings is then evaluated. The selection operator assures that strings are copied to the next generation with a probability associated to their fitness values. Therefore, this operator mimics the survival of the fittest in the natural world. Although selection assures that in the next generation the best chromosomes will be present with a higher probability, it does not search the space, because it just copies the previous chromosomes. The search results from the creation of new chromosomes from old ones. The crossover operator, takes two randomly selected chromosomes; one point along their common length is randomly selected, and the characters of the two parent strings are swapped, thus generating two new chromosomes. Crossover by itself does not involve any change in the actual values of the chromosomes. The mutation operator, randomly selects a position in the chromosome and, with a given probability, changes the corresponding value. This operator does assure that new parts of the search space are explored, which selection and crossover could not fully guarantee. The GA proceeds as the following algorithm (figure 1):

Algorithm 1 - *Genetic Algorithm*

1. INITIALIZATION

The initial population, consisting of points in the search space, is randomly created.

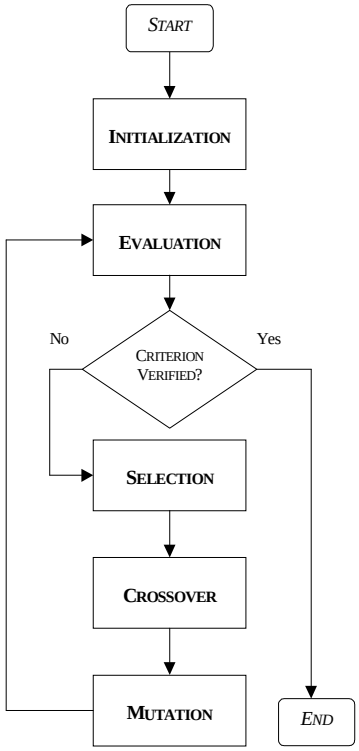

Figure 1: Genetic Algorithm Flow Chart

2. EVALUATION

Each chromosome in the population is evaluated through the objective (fitness) function.

3. GENETIC OPERATORS

The search is performed, by creating a new population from the previous one, through the application of the genetic operators.

4. STOPPING CRITERIUM

These steps (2-3) are repeated till the population converges or the specified number of generations is reached.

2.2 Constraint Handling

Most of GAs implementation on constrained optimization use the penalty function method (Goldberg 1989), in such a way that, the fitness function $F(\mathbf{x})$, is defined as follows,

$$F(\mathbf{x}) = f(\mathbf{x}) + R\left(\sum_{k=1}^{p}[\max\{0, g_k(\mathbf{x})\}]^2 + \sum_{l=1}^{m}[h_l(\mathbf{x})]^2\right) \tag{2}$$

i.e., as the sum of $f(\mathbf{x})$, the objective function and two penalty terms denoting the equality and inequality constraints violations, where R is a penalty coefficient.

This approach has a major drawback. The setting of the penalty parameters usually requires extensive experimentation. In this work, we compare this approach with the constraint handling method proposed by Deb (1998). A penalty term is introduced which does not depend on a penalty parameter. Thus, the fitness function is defined as

$$F(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \text{ and } \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ f_{\max} + \left(\sum_{k=1}^{p}[\max\{0, g_k(\mathbf{x})\}] + \sum_{l=1}^{m}|h_l(\mathbf{x})|\right) & \text{otherwise} \end{cases} \tag{3}$$

where $f_{\max}$, is the maximum function value of all feasible solutions in the population. The fitness value of any infeasible solution will result from the summation of the constraints violations, plus the highest objective function value of all feasible solutions. Thus, any infeasible solution will have a fitness value worst than any of the feasible solutions.

In this approach, two solutions are compared on the base of,

- their objective function values, if feasible;
- the constraint violations, if infeasible.

2.3 Genetic Coding – Chromosome representation

The implementation of a Genetic Algorithm for Mixed Integer Non-Linear Programming problems requires the representation of the potential solutions to the problem (each one being a point in the search space). Thus, each problem solution, a vector consisting of continuous and integer variables is represented by a string, denoted as a chromosome.

The chromosome length depends on the number of variables of the problem considered. The variables, coded using a binary representation, were defined in the following way:

- the continuous variables are represented by a determined number of bits which defines the resolution (in general a 32 bit representation was used);
- the binary variables are naturally represented in a binary code;
- the integer variables (defined in an interval of discrete values) are represented by the least number of bits that allows the representation of all values and, if necessary, restrictions are introduced in order to avoid infeasible values for those variables.

2.4 Genetic Algorithm Parameters

Table 1 presents the data concerning the GA parameters. These values have been set from previous experience in solving MINLP problems. Each problem was solved for 10 times. In any of the implementations, a stochastic uniform selection (Baker 1987), a two point crossover with a probability of 0.7, and an uniform mutation with a probability 0.001 were used (Goldberg 1989). The basic idea of this selection scheme is to allocate to each chromosome a portion of a spinning wheel proportional to its relative fitness. A single spin of the wheel

allows the selection of the desired number of chromosomes. The two point crossover is applied to pairs of chromosomes and consists in a random selection of two crossover points and then, exchange the bit sections between these points. Uniform mutation means that each chromosome position has an equal chance of being mutated. If mutation does occur at a given position then a random value is chosen from $\{0, 1\}$ to that position. All chromosomes were coded as bit strings using a Gray code. The Gray codes have the property that adjacent decoded values differ at exactly one bit position (Wright 1991).

Parameter	Value
Number of experiences	10
Population size	250
Chromosome length	Variable
Crossover probability	0.7
Mutation probability	0.001
Penalization coefficient	Variable
Coding mechanism	Graycode

Table 1: The GA Parameters

In a first phase, all the constraints were treated using a penalty scheme. In a second phase, the inequalities were treated according to the scheme due to Deb (1998). For all the executions, the initial population consisted of 250 chromosomes randomly generated. For a given generation, $f_{\max}$ was estimated by the value of the objective function of the worst chromosome in the population.

2.5 Stopping Criterion

The stopping criterion adopted was to terminate the search process when one of the following conditions was verified:

- the maximum number of generations was reached (assumed 1000 generations);
- a given number of generations was reached without improvement of the fitness function values (assumed 50 generations);
- all the population converged to a single point, i.e., when all the population converged to the same chromosome.

3 Evolution Strategies

3.1 Introduction

Evolution Strategies (Schwefel 1985) are also search procedures that mimic the natural evolution of the species in the natural systems. They work directly with the real representation of the parameter set, searching from an initial population (a set of points). Like GAs, they only require data based on the objective function and constraints, and not derivatives or other auxiliary knowledge. However, the transitions rules are deterministic and the constraints are handled, normally, using an elimination mechanism (the non-feasible points are eliminated). The search of new points is based on one operator, the mutation operator.

We will consider two distinct types of Evolution Strategies differing basically on the selection procedure:

- the $(\mu + \lambda)$ Evolution Strategy and
- the (μ, λ) Evolution Strategy.

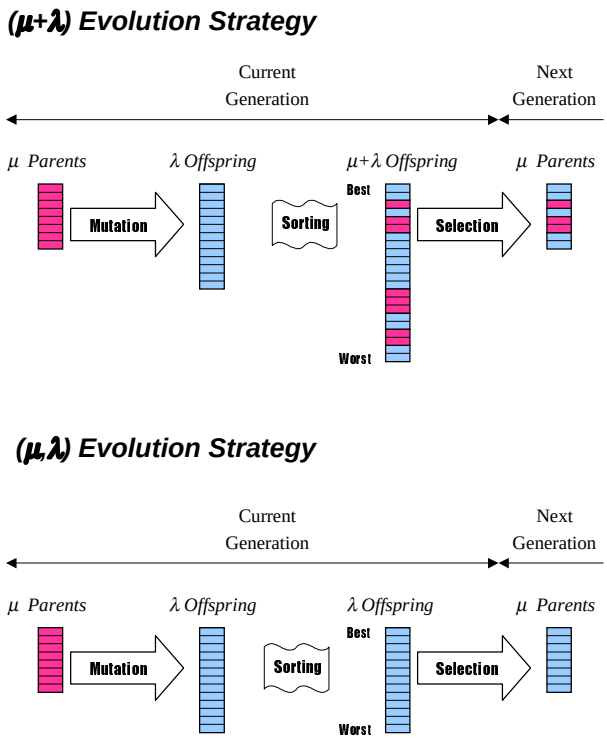

Figure 2: $(\mu + \lambda)$ and (μ, λ) Evolution Strategies

In this nomenclature, μ represents the number of parents of a generation and λ is the number of offspring of a generation. The figure 2 illustrates the difference between these two strategies.

In $(\mu + \lambda)$ Evolution Strategy, at a given generation, there are μ parents, and λ offspring are generated by mutation. Then, the $\mu + \lambda$ members are sorted according to their objective function values. Finally, the best μ of all the $\mu + \lambda$ members become the parents of the next generation (i.e., the selection takes place between the $\mu + \lambda$ members).

In (μ, λ) Evolution Strategy, at a given generation, there are μ parents, and λ offspring are generated by mutation (assuming that $\lambda > \mu$). Next, the λ members are sorted according to their objective function values. Then, the μ best of the λ members generated become the

parents of the next generation (i.e., the selection takes place between the λ members).

3.2 $(1+1)$ Evolution Strategy

This two membered scheme is the simplest Evolution Strategy. In this strategy, proposed by Rechenberg (1994), at a given generation, there are only one parent ($\mu = 1$) and one offspring ($\lambda = 1$), and the selection takes place between these two members.

Algorithm 2 - $(1+1)$ *Evolution Strategy*

1. INITIALIZATION

$$x^{(0)} = \begin{bmatrix} x_1^{(0)} \\ \vdots \\ x_n^{(0)} \end{bmatrix} \textit{ such that } g_j(x^{(0)}) \geq 0 \textit{ for all } j = 1, \ldots, m$$

$$k = 0$$

2. MUTATION

$$x_{New}^{(k)} = x^{(k)} + z^{(k)} = \begin{bmatrix} x_1^{(k)} + z_1^{(k)} \\ \vdots \\ x_n^{(k)} + z_n^{(k)} \end{bmatrix}$$

3. SELECTION

$$x^{(k+1)} = \begin{cases} x_{New}^{(k)} \leftarrow f(x_{New}^{(k)}) \leq f(x^{(k)}) \wedge g_j(x_{New}^{(k)}) \geq 0 \textit{ for all } j = 1, \ldots, m \\ x^{(k)} \leftarrow \textit{otherwise} \end{cases}$$

$$k = k + 1$$

IF stopping criterion is not true THEN return to step 2. ELSE end.

Algorithm 2 is the $(1+1)$ Evolution Strategy algorithm. The search starts from an initial point $x^{(0)}$ (an approximation to the optimum). Next, a new point, $x_{New}^{(k)}$, is generated by

mutation by the addition of a random number $z^{(k)}$. Then, the two points are compared and the best one (with lower value of the objective function and satisfying all the constraints) is selected to become parent on next generation. This process is repeated until the stop criterion is verified.

Usually, the random numbers $z^{(k)}$ are generated according to a Gaussian or Normal distribution. Besides, it is convenient that small changes occur frequently, but large ones only rarely. So, two requirements arise together for the generation of the random numbers $z^{(k)}$:

- the expected value of the components $z_i^{(k)}$ of $z^{(k)}$ must be equal to zero, i.e., $E\left[z_i^{(k)}\right] = 0$ for $i = 1, \ldots, n$, and
- the variances σ_i^2 must be small, for $i = 1, \ldots, n$.

In this sense, the random numbers $z_i^{(k)}$ can be generated according to a Normal distribution with mean zero and variance σ_i^2:

$$z_i^{(k)} \sim N(0, \sigma_i^2)$$

The typical initial values for the standard deviations σ_i can be expressed by equation 4, where Δx is a rough measure of the distance from the optimum and n is the dimension of the problem.

$$\sigma_i^{(0)} = \frac{\Delta x}{\sqrt{n}} \tag{4}$$

These standard deviations σ_i are actualized by the 1/5 Success rule. This rule can be formulated in the following manner (Rechenberg 1994):

"From time to time during the optimum search obtain the frequency of successes, i.e., the ratio of the number of successes to the total number of trials (mutations). If the ratio is greater than 1/5, increase the variance, if it is less than 1/5, decrease the variance."

Assuming that this rule is applied periodically, every p generations, it can also be expressed by:

$$\sigma_i^{(k+1)} = \begin{cases} c_{dec}\sigma_i^{(k)} \leftarrow \Pr_{suc}(p) < \frac{1}{5} \\ c_{inc}\sigma_i^{(k)} \leftarrow \Pr_{suc}(p) > \frac{1}{5} \\ \sigma_i^{(k)} \leftarrow \Pr_{suc}(p) = \frac{1}{5} \end{cases}$$

where $\Pr_{suc}(p)$ is the success rate during the last p generations, and $c_{dec} < 1$ and $c_{inc} > 1$ are the decreasing and increasing factors of the standard deviations σ_i, respectively.

3.3 (μ, λ) Evolution Strategy

In this Evolution Strategy, in each generation, there are μ parents and λ offspring, and the selection takes place between the λ offspring.

Algorithm 3 - (μ, λ) *Evolution Strategy*

1. INITIALIZATION

$$x_p^{(0)} = \begin{bmatrix} x_{p,1}^{(0)} \\ \vdots \\ x_{p,n}^{(0)} \end{bmatrix} \text{ such that } g_j(x_p^{(0)}) \geq 0 \text{ for all } j = 1, \ldots, m \text{ and } p = 1, \ldots, \mu$$

$k = 0$

2. MUTATION

$$x_{d,New}^{(k)} = x_u^{(k)} + z^{(k\lambda+d)} = \begin{bmatrix} x_{u,1}^{(k)} + z_1^{(k\lambda+d)} \\ \vdots \\ x_{u,n}^{(k)} + z_n^{(k\lambda+d)} \end{bmatrix} \textit{ such that } g_j(x_{d,New}^{(k)}) \geq 0$$

for all $j = 1, \ldots, m$, $p = 1, \ldots, \mu$ *and* $u \in [1, \mu]$,

$$\textit{e.g., } u = \begin{cases} \mu \leftarrow d = \mu, 2\mu, \ldots, K\mu \textit{ with } K \textit{ integer} \\ d//\mu \leftarrow \textit{otherwise} \end{cases}$$

where // states for the rest of integer division

3. SELECTION

Sort the $x_{d,New}^{(k)}$, *with* $d = 1, \ldots, \lambda$, *so that* $f(x_{a,New}^{(k)}) \leq f(x_{b,New}^{(k)})$

for all $a = 1, \ldots, \mu$ *and* $b = \mu, \ldots, \lambda$

$x_p^{(k+1)} = x_{a,New}^{(k)}$ *with* $p = 1, \ldots, \mu$ *and* $a = 1, \ldots, \mu$

$k = k + 1$

IF stopping criterion is not true THEN return to step 2. ELSE end.

Algorithm 3 is the algorithm of a (μ, λ) Evolution Strategy. The search starts generating μ points from an initial point (an approximation to the optimum). Then, λ points are generated by mutation, as a result of the addition of random numbers $z^{(k)}$. Next, all the λ points generated are sorted according to the objective function and the constraints. The μ best points are selected to become parents on the next generation. This process is repeated until the stopping criterion is satisfied.

The same requirements regarding the normal distribution of the random quantities $z_i^{(k)}$ are observed, i.e., $z_i^{(k)} \sim N(0, \sigma_i^2)$ for $i = 1, \ldots, n$. Like in the $(1+1)$ Evolution Strategy, the

typical initial values for the standard deviations σ_i can be expressed by the same equation 4. However, the standard deviations σ_i are now actualized by equation 5 where $z^{(k)}$ are determined according to a Normal distribution with mean zero and variance $\Delta\sigma^2$, where $\Delta\sigma$ is a parameter of the algorithm.

$$\sigma_i^{(k+1)} = \sigma_i^{(k)} e^{z_i^{(k)}} \text{ with } z_i^{(k)} \sim N(0, \Delta\sigma^2) \tag{5}$$

3.4 $(\mu + \lambda)$ Evolution Strategy

In this Evolution Strategy, like in the previous one, in each generation, there are μ parents and λ offspring; however, the selection takes place between all the $\mu+\lambda$ offspring. The $(\mu+\lambda)$ Evolution Strategy algorithm is also similar to the previous one (algorithm 3), except at the selection procedure which chooses the best μ of the $\mu + \lambda$ members, for parents on the next generation.

All requirements regarding the normal distribution of the random quantities $z_i^{(k)}$, the typical initial values for the standard deviations σ_i (equation 4) and the actualization of the standard deviations σ_i (equation 5) are identical to the (μ, λ) Evolution Strategy.

3.5 Constraint Handling

All the constraints were treated using the elimination mechanism (the non-feasible points are eliminated). Depending on the problem, an initial approximation for the global optimum was considered. This initial approximation allowed the random generation of the initial population. A simple mechanism was used in order to guarantee that all points in this initial

population were feasible.

3.6 Evolution Strategies Parameters

Table 2 presents the data concerning the Evolution Strategies parameters. As previously, each problem was solved for 10 times.

Parameter	Value
Number of experiences	10
Initial approximation	Variable
Initial value of σ	Variable
Lower bound of σ (absolute)	10^{-5}
Lower bound of σ (relative)	10^{-5}
Actualization factor of σ	0.85
Penalization coefficient	10^3

Table 2: The ES Parameters

3.7 Stopping Criterion

The stopping criterion adopted for $(1 + 1)$ Evolution Strategy was to terminate the search process when one of the following conditions was verified:

- the maximum number of generations was reached (assumed 2000 generations);
- $\left|f^{(k+\Delta k)} - f^{(k)}\right| < 10^{-5}$ with $\Delta k = 20$;
- $\frac{\left|f^{(k+\Delta k)} - f^{(k)}\right|}{\left|f^{(k+\Delta k)}\right|} < 10^{-5}$ with $\Delta k = 20$.

The stopping criterion adopted for $(\mu+\lambda)$ Evolution Strategy and (μ, λ) Evolution Strategy was to terminate the search process when one of the following conditions was verified:

- the maximum number of generations was reached (assumed 1000 generations);

Problem	n	r	i	b	p	m
#1	2	1	-	1	2	-
#2	3	2	-	1	1	1
#2'	2	1	-	1	1	-
#3	3	2	-	1	3	-
#4	9	7	-	2	4	6
#4'	5	4	-	1	6	-
#5	2	3	-	4	9	-
#6	5	3	2	-	3	-
#7	10	7	3	-	18	-

Table 3: Mixed Test Problems

- $\left| f_{\max}^{(k)} - f_{\min}^{(k)} \right| < 10^{-5}$;
- $\frac{\left| f_{\max}^{(k)} - f_{\min}^{(k)} \right|}{\left| f_{mean}^{(k)} \right|} < 10^{-5}$.

4 Case Studies

Seven problems chosen from the chemical engineering area are presented in Table 3. In this table, n is the total number of variables in the problem; r, i and b, are, respectively, the number of continuous, integer and binary variables ($n = r + i + b$); m and p are, respectively, the number of equality and inequality constraints in the problem. Problems #2 and #4, with equality constraints, were reformulated in order to change these constraints into inequality ones (problems #2´and #4´).

Problem #1

This problem has a nonlinear restriction and has been proposed by Kocis and Grossmann (1988); it has been solved also by Floudas et al. (1989), Ryoo and

Sahinidis (1995) and Cardoso et al (1997).

$$\min f(x, y) = 2x + y$$

subject to

$$1.25 - x^2 - y \leq 0$$

$$x + y \leq 1.6$$

$$0 \leq x \leq 1.6$$

$$y \in \{0, 1\}$$

The global optimum is $(x, y; f) = (0.5, 1; 2)$.

Problem #2

This problem, with a nonlinear constraint, has been proposed by Kocis and Grossmann (1987) and also studied by Salcedo (1992) and Cardoso et al (1997).

$$\min f(x_1, x_2, y) = -y + 2x_1 + x_2$$

subject to

$$x_1 - 2\exp(-x_2) = 0$$

$$-x_1 + x_2 + y \leq 0$$

$$0.5 \leq x_1 \leq 1.4$$

$$y \in \{0, 1\}$$

The global optimum is $(x_1, x_2, y; f) = (1.375, 0.375, 1; 2.124)$.

Problem #2'

Problem #2 can also be formulated without the nonlinear equality constraint with the same global optimum.

$$\min f(x_1, y) = -y + 2x_1 - \ln(x_1/2)$$

subject to

$$-x_1 - \ln(x_1/2) + y \leq 0$$

$$0.5 \leq x_1 \leq 1.4$$

$$y \in \{0, 1\}$$

Problem #3

This problem was firstly studied by Floudas (1995) and presents a nonlinear

constraint. It has also been solved by Cardoso et al (1997).

$$\min f(x_{1,}x_{2,}y) = -0.7y + 5(x_1 - 0.5)^2 + 0.8$$

subject to

$$-\exp(x_1 - 0.2) - x_2 \leq 0$$

$$x_2 + 1.1y \leq -1$$

$$0.2 \leq x_1 \leq 1$$

$$-2.22554 \leq x_2 \leq -1$$

$$y \in \{0, 1\}$$

The global optimum is $(x_{1,}x_2, y; f) = (0.94194, -2.1, 1; 1.07654)$.

Problem #4

In this problem, taken from Kocis and Grossmann (1989), the objective is to select one between two candidate reactors in order to minimize the production cost. It has also been solved by Diwekar et al. (1992), Diwekar and Rubin (1993) and Cardoso et al (1997).

$$\min f(x, y_1, y_2, v_1, v_2) = 7.5y_1 + 5.5y_2 + 7v_1 + 6v_2 + 5x$$

subject to

$$y_1 + y_2 = 1$$

$$z_1 = 0.9\left[1 - \exp(-0.5v_1)\right] x_1$$

$$z_2 = 0.8\left[1 - \exp(-0.4v_2)\right] x_2$$

$$z_1 + z_2 = 10$$

$$x_1 + x_2 = x$$

$$z_1 y_1 + z_2 y_2 = 10$$

$$v_1 \leq 10y_1$$

$$v_2 \leq 10y_2$$

$$x_1 \leq 20y_1$$

$$x_2 \leq 20y_2$$

$$x_1, x_2, z_1, z_2, v_1, v_2 \geq 0$$

$$y_1, y_2 \in \{0, 1\}$$

The global optimum is $(x, y_1, y_2, v_1, v_2; f) = (13.362272, 1, 0, 3.514237, 0; 99.245209)$.

Problem #4'

The previous problem, without equality constraints, can be formulated as follows:

$$\min f(y_1, v_1, v_2) = 7.5y_1 + 5.5(1 - y_1) + 7v_1 + 6v_2 + 50\frac{1 - y_1}{0.8[1 - \exp(-0.4v_2)]} + 50\frac{y_1}{0.9[1 - \exp(-0.5v_1)]}$$

subject to

$$0.9[1 - \exp(-0.5v_1)] - 2y_1 \leq 0$$

$$0.8[1 - \exp(-0.4v_2)] - 2(1 - y_1) \leq 0$$

$$v_1 \leq 10y_1$$

$$v_2 \leq 10(1 - y_1)$$

$$v_1, v_2 \geq 0$$

$$y_1 \in \{0, 1\}$$

Problem #5

This problem was studied by Floudas et al. (1989), Yuan et al. (1989), Salcedo (1992), Ryoo and Sahinidis (1995) and Cardoso et al (1997) and presents several nonlinear constraints.

$$\min f(x_1, x_2, x_3, y_1, y_2, y_3, y_4) = (y_1 - 1)^2 + (y_2 - 1)^2$$
$$+ (y_3 - 1)^2 - \ln(y_4 + 1) + (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2$$

subject to

$$y_1 + y_2 + y_3 + x_1 + x_2 + x_3 \leq 5$$

$$y_3^2 + x_1^2 + x_2^2 + x_3^2 \leq 5.5$$

$$y_1 + x_1 \leq 1.2$$

$$y_2 + x_2 \leq 1.8$$

$$y_3 + x_3 \leq 2.5$$

$$y_4 + x_1 \leq 1.2$$

$$y_2^2 + x_2^2 \leq 1.64$$

$$y_3^2 + x_3^2 \leq 4.25$$

$$y_2^2 + x_3^2 \leq 4.64$$

$$x_1, x_2, x_3 \geq 0$$

$$y_1, y_2, y_{3,} y_4 \in \{0, 1\}$$

The global optimum is

$(x_1, x_2, x_3, y_1, y_2, y_3, y_4; f) = (0.2, 1.280624, 1.954483, 1, 0, 0, 1; 3.557463)$.

Problem #6

This is maximization problem taken from Wong (1990) and also studied by Car-

doso et al (1997).

$$\max f(x_1, x_2, x_3, y_1, y_2) = -5.357854x_1^2 - 0.835689y_1x_3 - 37.29329y_1 + 40792.141$$

subject to

$$a_1 + a_2y_2x_3 + a_3y_1x_2 - a_4x_1x_3 \leq 92$$

$$a_5 + a_6y_2x_3 + a_7y_1y_2 + a_8x_1^2 - 90 \leq 20$$

$$a_9 + a_{10}x_1x_3 + a_{11}y_1x_1 + a_{12}x_1x_2 - 20 \leq 5$$

$$27 \leq x_1, x_2, x_3 \leq 45$$

$$y_1 \in \{78, ..., 102\}, \text{integer}$$

$$y_2 \in \{33, ..., 45\}, \text{integer}$$

where a_1 to a_{12} are given by table 4.

a_1	85.334407	a_5	80.51249	a_9	9.300961
a_2	0.0056858	a_6	0.0071317	a_{10}	0.0047026
a_3	0.0006262	a_7	0.0029955	a_{11}	0.0012547
a_4	0.0022053	a_8	0.0021813	a_{12}	0.0019085

Table 4: Problem #6 Data

The global optimum is, for any combination of x_2, y_2, $(x_1, x_3, y_1; f) = (27, 27, 78; 32217.4)$.

Problem #7

This is a multi-product batch plant problem with M serial processing stages, where fixed amounts Q_i from N products must be produced. The objective is to find out for each stage j, the number of parallel units N_j, together with the respective sizes V_j and, for each product i, the corresponding batch sizes B_i and cycle times T_{Li}. The data are the horizon time H, the size factors S_{ij}, the processing times t_{ij} of product i in stage j, the required productions Q_i and the cost coefficients α_j and β_j. This problem, studied by Grossmann and Sargent (1979), Kocis and Grossmann (1988), Salcedo (1992) and Cardoso et al (1997) has the following mathematical formulation:

$$\min f = \sum_{j=1}^{M} \alpha_j N_j V_j^{\beta_j}$$

subject to

$$\sum_{i=1}^{N} \frac{Q_i T_{Li}}{B_i} \leq H$$

$$V_j \geq S_{ij} B_i$$

$$N_j T_{Li} \geq t_{ij}$$

$$1 \leq N_j \leq N_j^u$$

$$V_j^l \leq V_j \leq V_j^u$$

$$T_{Li}^l \leq T_{Li} \leq T_{Li}^u$$

$$B_j^l \leq B_j \leq B_j^u$$

$$N_j, \text{integer}$$

where, for the specific problem considered, $M = 3$, $N = 2$, $H = 6000$, $\alpha_j = 250$, $\beta_j = 0.6$, $N_j^u = 3$, $V_j^l = 250$ and $V_j^u = 2500$. The values of T_{Li}^l, T_{Li}^u, B_j^l and B_j^u are given by:

$$\begin{aligned} T_{Li}^l &= \max \frac{t_{ij}}{N_j^u} \\ T_{Li}^u &= \max_j t_{ij} \\ B_i^l &= \frac{Q_i}{H} T_{Li} \\ B_j^u &= \min(Q_i, \min_j \frac{V_j^u}{S_{ij}}) \end{aligned}$$

The table 5 gives the values of S_{ij} and t_{ij}.

S_{ij}			t_{ij}		
2	3	4	8	20	8
4	6	3	16	4	4

Table 5: Problem #7 Data

The global optimum is

$(N_1, N_2, N_3, V_1, V_2, V_3, B_1, B_2, T_{L1}, T_{L2}; f) = (1, 1, 1, 480, 720, 960, 240, 120, 20, 16; 38499.8)$.

5 Results

Figures 3 and 4 summarize the results obtained when GAs were used. In these experiments, different values of the penalty coefficient of the constraints in equation 2 (R equals to 10, 10^3 and 10^5), and the Deb Scheme as described in equation 3 were considered. Figure 3 presents

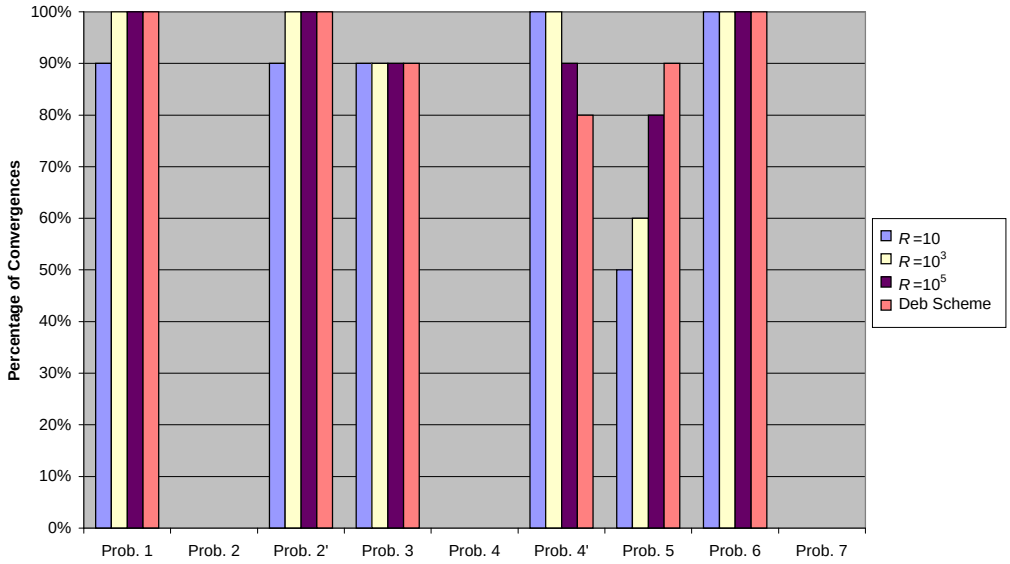

Figure 3: GA convergence

the percentage of convergencies to the global optimum and figure 4 shows the mean number of objective function evaluations needed to reach convergence.

Figures 5 and 6 summarize the results obtained when ESs were considered. In these experiments, three ESs were applied, the $(1+1)$ -ES, the $(\mu+\lambda)$ - ES and (μ,λ) - ES. The number of parents (μ) and number of offspring (λ) were, respectively, 10 and 100. As before, figure 5 presents the percentage of convergencies to the global optimum and figure 6 shows the mean number of objective function evaluations needed to reach convergence.

Table 6 presents the comparison, in terms of the number of objective function evaluations and proportion of convergencies to the optimum, between the GA (with the two penalty schemes), the $(\mu+\lambda)$-ES and the M-SIMPSA algorithm. In this table, $\overline{\#F}$ and $\#C$ represent, respectively, the mean number of objective function evaluations over all the 10 executions, and the percentage of convergencies to the global optimum. The results presented are the

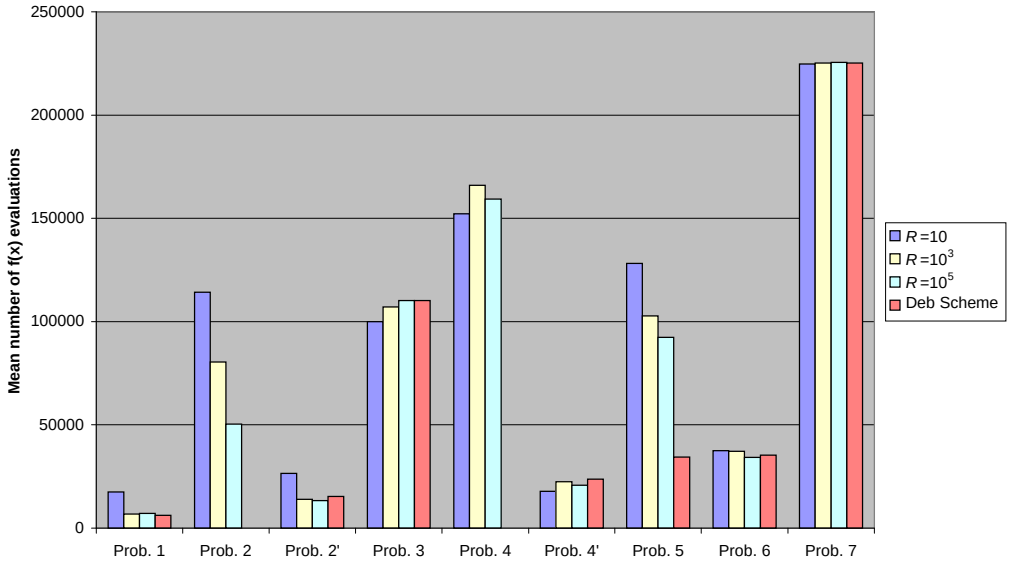

Figure 4: GA number of function evaluations

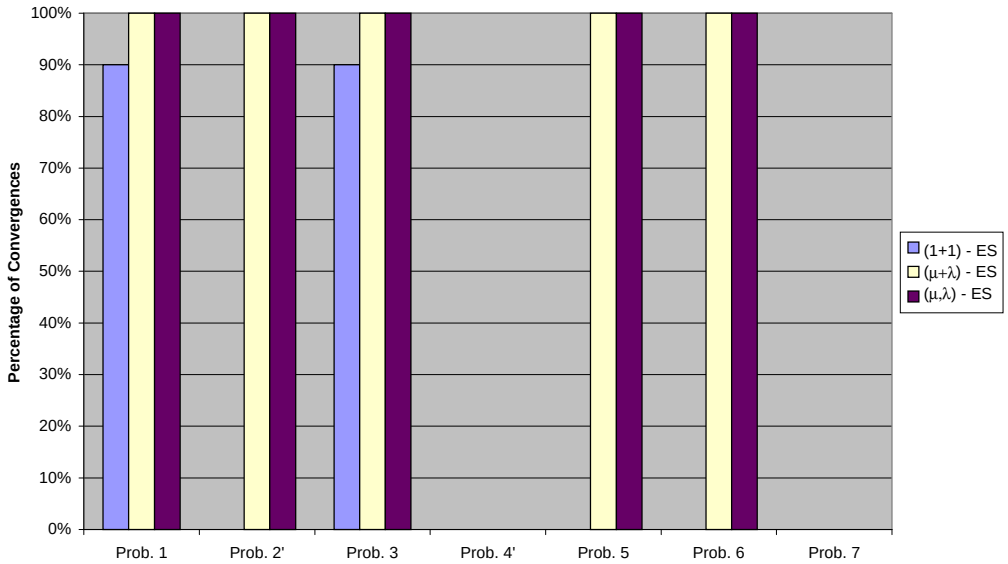

Figure 5: ES convergence

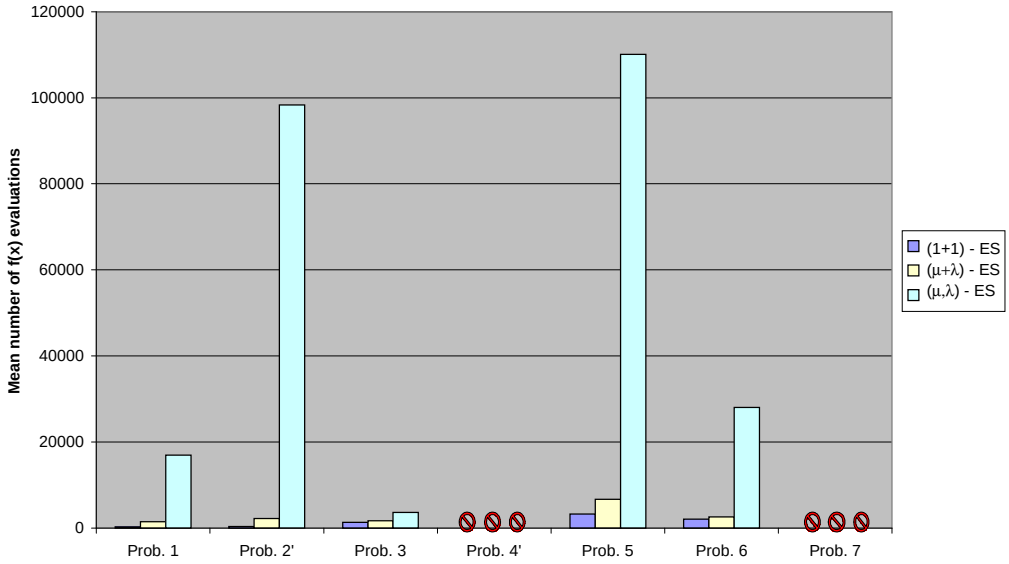

Figure 6: ES number of function evaluations

best obtained for each problem. For problem #5 a better solution was found by GAs and ESs than the best obtained by M-SIMPSA.

	GA-$R = 10^3$		GA-Deb sch.		$(\mu + \lambda)$-ES		M-SIMPSA		M-SIMPSA-pen.	
Prob. #	$\overline{\#F}$	$C\%$	$\overline{\#F}$	$C\%$	$\overline{\#F}$	$C\%$	$\overline{\#F}$	$C\%$	$\overline{\#F}$	$C\%$
1	6787	**100**	6191	**100**	**1518**	**100**	607	99	16282	**100**
2'	13939	**100**	15298	**100**	**2255**	**100**	10582	83	14440	**100**
3	107046	90	110233	90	**1749**	**100**	-	0	38042	**100**
4'	22489	**100**	23730	80	-	0	**14738**	**100**	42295	**100**
5	102778	60	34410	90	**6710**	**100**	22309	60*	63751	97*
6	37167	**100**	35255	**100**	**2536**	**100**	27410	87	33956	95
7	225176	0	225173	0	-	0	-	0	**257536**	**97**

* converged to a non-optimal solution - execution halted

Table 6: GA versus M-SIMPSA

6 Conclusions

It is well known that one of the main disadvantages of Evolutionary Algorithms lies in the required computational time. On the other hand, requiring only the objective function values to execute the search, based on a pool of points, allows Evolutionary Algorithms to deal with nonconvex problems with reduced chances of being trapped on a local optimum. This work compared Genetic Algorithms, Evolution Strategies with previous published results of M-SIMPSA (Cardoso et al 1997), a Simulated Annealing based algorithm.

The results clearly show the difficulty in dealing with equality constraints (Problems #2 and #4); however, when reformulated without these constraints, the algorithms exhibited a high rate of convergence.

The method proposed by Deb (1998) to deal with the constraints is clearly superior to the usual penalty scheme, since it does not require the setting of a penalty parameter. The proposal of Cardoso et al. (1997) for the M-SIMPSA algorithm can generate infeasible points with fitness values superior to feasible points. Evolutionary strategies exclude all non feasible points, which corresponds to an infinite penalty. All algorithms presented have great difficulties with problem #7 which is not surprising since the problem is highly constrained; the global optimum corresponds to a point where a very small variation in any of the continuous variables produces infeasibility. Thus, it seems of utmost importance the development of appropriate methods to deal with constraints in the context of Evolutionary Algorithms.

Evolutionary Strategies exhibit difficulties in highly constrained problems but, in general, they are the most efficient in terms of function evaluations. In summary, the solution of

MINLP problems with Evolutionary Algorithms is a valid approach in non convex problems where computational time is not of primary importance. Future work will address the comparison of evolutionary algorithms with adaptive random search algorithms, as well as the treatment of equality constraints.

References

Baker, J. (1987), Reducing Bias and Inefficiency in the Selection Algorithm, in Proceedings of the Second International Conference on Genetic Algorithms and their Applications, Lawrence Erlbaum Associates, Hillsdale, New Jersey, USA, 14-21.

Banga, J. R., Seider, W. D, (1996), Global optimization of chemical processes using stochastic algorithms, State of the Art in Global Optimization: Computational Methods and Applications, C.A. Floudas and P. M. Pardalos (Eds.), Kluwer Academic Publishers, Dordrecht, The Netherlands, 563-583.

Cardoso, M. M. F. C.(1998), À Procura do Óptimo Global, PhD Thesis, Engineering Faculty, University of OPorto, Portugal.

Cardoso, M. F., Salcedo, R. L., Feyo de Azevedo, S. (1996a), Nonequilibrium Simulated Annealing: A Faster Approach to Combinatorial Minimization, Ind. Eng. Chem. Res, 33, 1908-1918.

Cardoso, M. F., Salcedo, R. L., Feyo de Azevedo, S. (1996b), The Simplex-Simulated Annealing Approach to Continuous Non-Linear Optimization, Computers chem. Engng., Vol. 20, 9, 1065-1080.

Cardoso, M. F., Salcedo, R. L., Feyo de Azevedo, S., Barbosa, D. (1997), A Simulated Annealing Approach to the Solution of MINLP Problems, Computers chem. Engng., Vol.

21, 12, 1349-1364.

Deb, K. (1998), An Efficient Constraint Handling Method for Genetic Algorithms, Comp. Meth. Appl. Mech. Eng. (to appear).

Diwekar, U. M., Grossmann, I. E., Rubin, E. S. (1992), An MINLP Process Synthesizer for a Sequential Modular Simulator, Ind. Eng. Chem. Res, 31, 313-322.

Diwekar, U. M., Rubin, E. S. (1993), Efficient Handling of the Implicit Constraints Problem for the ASPEN MINLP Synthesizer, Ind. Eng. Chem. Res, 32, 2006-2011.

Floudas, C. A. (1995), Nonlinear and mixed-integer optimization, Oxford University Press, New York.

Floudas, C. A., Aggarwal, A., Ciric, A. R. (1989), Global Optimum Search for Nonconvex NLP and MINLP Problems, Computers chem. Engng., Vol. 13, 10, 1117-1132.

Goldberg, D. E. (1989), Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, Massachusetts.

Grossmann, I. E., Sargent, R. W. H., (1979) Optimal design of multipurpose chemical plants, Ind. Eng. Chem. Process Des. Dev., Vol. 18, 343.

Kocis, G. R., Grossmann, I. E. (1987), Relaxation Strategy for the Structural Optimization of Process Flow Sheets, Ind. Eng. Chem. Res, 26, 1869-1880.

Kocis, G. R., Grossmann, I. E. (1988), Global Optimization of Nonconvex Mixed-Integer Nonlinear Programming (MINLP) Problems in Process Synthesis, Ind. Eng. Chem. Res, 27, 1407-1421.

Kocis, G. R., Grossmann, I. E. (1989), A Modelling and Decomposition Strategy for the MINLP Optimization of Process Flowsheets, Computers chem. Engng., Vol. 13, 7, 797-819.

Rechenberg, I. (1994), Evolutionsstrategie '94, Frommann-Holzboog, Stuttgart.

Reklaitis, G. V., Ravindran, A., Ragsdell, K. M. (1983), Engineering Optimization: Methods and Applications, Wiley, New York, 277-287.

Ryoo, H. S., Sahinidis, B. P. (1995), Global Optimization of nonconvex NLPs and MINLPs with applications in process design, Comp. chem. Engng., Vol. 19(5), 551.

Salcedo, R. L. (1992), Solving nonconvex nonlinear programming and mixed-integer nonlinear programming problems with adaptive random search, Ind. Eng. Chem. Res, 31, 262.

Schwefel, H.-P.(1985), Evolution and Optimal Seeking, John Wiley & Sons

Winston, W. L.(1994), Operations Research Applications and Algorithms, 3 ed., Duxburg

Wright, A. H. (1991), Genetic Algorithms for Real Parameter Optimization, in G.J.E. Rawlings (Ed.), Foundations of Genetic Algorithms, Morgam Kaufmann Publishers, San Mateo, CA, 205-218

Wong, J. (1990), Computational experience with a general nonlinear programming algorithm, COED J., Vol. 10, 19.

Yuan, X., Zhang, S., Pibouleau, L., Domenech, S.(1989), Une Methode d'optimization nonlineaire en variables mixtes pour la conception de procedes, in RAIRO-Oper. Res.