Eng. Opt., 2002, Vol. 34, pp. 307–322

A COMPARISON OF EVOLUTIONARY ALGORITHMS FOR MECHANICAL DESIGN COMPONENTS

LAURENCE GIRAUD-MOREAU* and PASCAL LAFON†

Laboratoire des Systèmes Mécaniques et d'Ingénierie Simultanée (LASMIS), Université de Technologie de Troyes, 12 rue Marie Curie - BP 2060, 10010 Troyes Cedex, France

(Received 19 February 2001; In final form 28 October 2001)

Two evolutionary algorithms – the genetic algorithm and the evolution strategy – are compared in respect of mechanical design problems. Mechanical design problems are real world problems, characterized by a number of inequality constraints, nonlinear equations, mixed discrete-continuous variables and the presence of interdependent discrete parameters whose values are taken from standardized tables. The selection, recombination and mutation operators, and the chosen constraint-handling method are presented for both the genetic algorithm and the evolution strategy. In order to find the best combination of operators for each algorithm which will solve mechanical design problems, a number of selection and recombination operators are compared in respect of these problems. A comparison of these two algorithms with regard to three mechanical design problems extends the results of comparisons presented in the literature for unimodal and multimodal test functions with continuous variables only, and without constraints.

Keywords: Genetic algorithms; Evolution strategy; Mixed variables; Inequality constraints

1 INTRODUCTION

In mechanical design, sizing a mechanical system implies solving a problem of optimization; this is called the optimal design problem. The characteristics of these real world problems are mixed discrete–continuous variables, inequality constraints and nonlinear equations. Optimal design problems generally involve interdependent discrete parameters whose values are taken from standardized tables, *i.e.* lists of commercially available prefabricated sizes. These discrete parameters directly depend on the choice of one of the discrete variables. These optimal design problems constitute a particular class of problems, with which this article is concerned. Because of the presence of interdependent discrete parameters, the gradients cannot be calculated in general. Thus, classical gradient-based methods are not suitable for this class of problems.

This article proposes using two evolutionary algorithms – genetic algorithms (GAs) and evolution strategies (ESs) – to solve these problems of optimal design. By choosing zeroth order heuristic methods, standardized tables of discrete parameters can be used directly. The purpose is to compare GAs and ESs for the class of optimal design problem. Several GA and ES

* Corresponding author. E-mail: laurence.moreau@utt.fr
† E-mail: pascal.lafon@utt.fr

ISSN 0305-215X print; ISSN 1029-0273 online
DOI: 10.1080/03052150290030406

selection and recombination operators will be compared with respect to these problems in order to find the best combination of operators with each algorithm to solve this type of problem.

First, optimal design problems are expounded after which GAs and ESs are briefly reviewed. Details of the different operators chosen for study with GAs and ESs are then described. Finally, numerical experiments on mechanical design problems are presented.

2 MATHEMATICAL FORMULATION OF THE OPTIMAL DESIGN PROBLEMS

The optimal design problems (P) can be expressed in the following form:

$$\text{Minimize} \quad f(x_C, x_D, P(x_D)) \tag{1}$$

$$\text{Subject to} \quad g_j(x_C, x_D, P(x_D)) \leq 0 \qquad j = 1, \ldots, m \tag{2}$$

$$x_{Ll} \leq x_{Cl} \leq x_{Ul} \qquad l = 1, \ldots, n_C \tag{3}$$

$$x_{Di} \in D_i, D_i = (d_{i1}, d_{i2}, \ldots, d_{iq_i}) \qquad i = 1, \ldots, n_d \tag{4}$$

The optimal design problem (P) is a mixed discrete-continuous nonlinear optimization problem, where f and g_j are objective and constraint functions respectively. The components of the mixed variable vector x are divided into n_C continuous variables expressed as $x_C \in \mathbb{R}^{nc}$, where x_{Ll} and x_{Ul} are lower and upper limits, and n_D discrete variables, expressed as x_D. D_i is the set of discrete values for the ith variable, while q_i is the number of discrete values for the ith discrete variable. $P(x_D)$ is a vector of n_p discrete parameters, whose values are taken from standardized tables of standard sizes, and which directly depend on the choice of one of the discrete variables x_{Dk} ($k \in [1, n_D]$). The derivates $\partial P_i/\partial x_{Dk}$ and also $\partial f/\partial x_{Dk}$, $\partial g_j/\partial x_{Dk}$ ($j = 1, \ldots, m$) cannot be computed.

3 EVOLUTIONARY ALGORITHMS

Evolutionary algorithms are based on the principle of evolution, *i.e.* survival of the fittest. Unlike classical methods, they do not use a single search point but a population of points called individuals. Each individual represents a potential solution to the problem. In these algorithms, the population evolves toward increasingly better regions of the search space by undergoing statistical transformations called recombination, mutation and selection. The general structure is the following:

```
t: = 0
initialization of the population
evaluation
while not terminated do
begin
    selection (GAs)
    recombination
    mutation
    evaluation
    selection (ESs)
    t = t + 1
end
```

A good review of GAs and ESs can be found in Refs. [1, 4–6, 12, 15].

It should be noted that the order of reproduction and selection procedures is different for GAs and ESs. With GAs, an intermediate population is selected first and then recombination and mutation are applied to the individuals selected. With ESs, each offspring is the result of the recombination of two or more individuals and mutation. When λ offspring individuals are created, the selection operator reduces this intermediate population of λ offspring to μ individuals.

In the next section, the individuals' representation and details of the operators which will be tested with each algorithm are described.

4 IMPLEMENTATION

4.1 Mixed Continuous-Discrete Representation

GAs The canonical GA uses a binary representation of individuals as fixed-length strings over the alphabet $\{1, 0\}$, which are similar to chromosomes in biological systems [8]. In this paper, a binary representation using the Gray coding [12] has been chosen. The advantage is that the Hamming distance is always equal to 1. Continuous variables are treated as discrete variables with small increments [1, 16, 17]. The segment of the chromosome, corresponding to the continuous variable x_{Ci} is decoded to yield the corresponding integer value, and the integer value is mapped to the interval $[x_{Li}, x_{Ui}]$.

ESs The standard representation of ESs is a real-valued vector which can only be applied to continuous variables. Different extensions of standard ESs have been proposed to solve mixed-discrete problems [3, 7, 14]. In this article, the representation proposed by Bäck and Schütz [3] will be used. Each individual is represented by four vectors: $a = (x_C, x_D, \sigma, p)$ where $\sigma \in \mathbb{R}_+^{nc}$ and $p \in [0, 1]^{nd}$ are strategy parameters which control the application of mutation to the continuous and discrete parameters.

4.2 Fitness Function and Constraint-Handling

When the problem of minimization does not contain any constraint, the fitness function is generally the objective function. For problems which involve constraints, many methods have been developed, especially with GAs (see Refs. [11–13]).

In order to compare GAs and ESs, the same constraint-handling method for GAs and ESs will be used. A dynamic penalty method in which the penalty coefficient increases during the evolution process has been chosen. Penalty methods generally give good results in a reasonable computation time. The aim of increasing this coefficient during the search is to initially keep a slow exploration of the design space and then force a greater exploitation of promising regions.

4.3 Selection

GAs and ESs have very different selection processes – probabilistic in the case of GAs, and deterministic in the case of ESs.

GAs In the case of GAs, the chance of being selected is proportional to the individual's fitness. The rule is that the best individuals have the most copies, the average individual stays even and the worst dies off. In the case of classical selection (proportional selection) [8], a linear scaling method can be used to determine the selection probability of the individual a_i: $p(a_i) = (F\max - F(a_i))/(\mu F\max - \sum_{i=1}^{\mu} F(a_i))$, where $F(a_i)$ is the fitness of the individual a_i and Fmax is the worst fitness value F in the current population. μ individuals are then selected according to that probability distribution (roulette wheel selection). Rank-based selection methods are now often used rather than this classical selection [12]. The probability assigned to each individual depends only on its position in the individual's rank. In the application, the classical selection and a rank-based selection method used by Le Riche [10] will be tested.

ESs With ESs, the selection operator is completely deterministic. Two selection methods exist *i.e.* $(\mu,\ \lambda)$ selection and $(\mu+\lambda)$ selection [1]. $(\mu,\ \lambda)$ selection is recommended by Schwefel [15], but the experimental studies of Gehlhaar and Fogel [5] show that the $(\mu+\lambda)$ strategy performs better than the $(\mu,\ \lambda)$ strategy. As a result, the two selection strategies will be tested with respect to the present optimal design problems.

4.4 Recombination

The recombination operator strongly depends on the way individuals are represented. As GAs and ESs do not use the same representation of individuals (binary and floating-point vectors respectively), the operators are different.

GAs Many crossover operators have been proposed for GAs (see Ref. [12]). In this article three different crossover operators will be tested:

- the traditional one-point crossover
- the two-point crossover
- the uniform crossover, in which each bit is chosen randomly from the corresponding parental bits.

The crossover is applied with a given probability, generally between 0.6 and 0.9 [10]. A probability p_c equal to 0.8 has been chosen.

ESs Not only object variables but also strategy parameters are subject to recombination and this operator may be different for continuous variables, discrete variables, standard deviations corresponding to continuous variables, and mutation probabilities corresponding to discrete variables.

As with GAs, many recombination operators exist [1]. They can all be used in local form (noted with small letters), where two randomly selected parent individuals produce an offspring, or global form (indicated in capitals), where one randomly chosen parent is held fixed and the second parent is randomly chosen anew for each single variable. Table I describes 8 different recombination operators. x_i^{P1} and x_i^{P2} are the ith variables of the two parents $P1$ and $P2$ chosen randomly from the population. x_i^{Pj} is the ith variable of a new parent j chosen randomly for each component in the case of the global form. x_i is the ith variable of the offspring created by recombination. α_i indicates that α is resampled for each variable i.

As no theoretical basis exists for choosing recombination operators, different combinations of recombination operators for the object variables and the strategy parameters will be tested with respect to different optimal design problems. The chosen recombination operators are:

- The discrete recombination in its local and global form for the discrete variables: $r_{xd} \in \{d, D\}$
- The local discrete, global discrete, global intermediate and global extended generalized intermediate recombination for the mutation probability: $r_p \in \{d, D, I, A1.5\}$
- The next 6 recombination pairs for the continuous variables x_C and the standard deviations σ: $(r_{xc}, r_\sigma) \in \{(d, I), (I, d), (I, I), (a1.5, a1.5), (A1, A1), (A1.5, A1.5)\}$

TABLE I Recombination Operators for ESs.

Recombination	*Computation:* $x_i =$	*Interval for* α	*Form*	*Notation*
Discrete	x_i^{p1} or x_i^{P2}		local	d
Global discrete	x_i^{p1} or x_i^{Pj}		global	D
Intermediate	$\alpha x_i^{P1} + (1-\alpha)x_i^{P2}$	$\alpha = 0.5$	local	i
Global intermediate	$\alpha x_i^{P1} + (1-\alpha)x_i^{Pj}$	$\alpha = 0.5$	global	I
Generalized intermediate	$\alpha_i x_i^{P1} + (1-\alpha_i)x_i^{P2}$	$\alpha_i \in [0, 1]$	local	$a1$
Global generalized intermediate	$\alpha_i x_i^{P1} + (1-\alpha_i)x_i^{Pj}$	$\alpha_i \in [0, 1]$	global	$A1$
Extended generalized intermediate	$\alpha_i x_i^{P1} + (1-\alpha_i)x_i^{P2}$	$\alpha_i \in [-0.5, 1.5]$	local	$a1.5$
Global extended generalized intermediate	$\alpha_i x_i^{P1} + (1-\alpha_i)x_i^{Pj}$	$\alpha_i \in [-0.5, 1.5]$	global	$A1.5$

4.5 Mutation

GAs In GAs, the mutation changes single bits of chromosomes according to a small predetermined probability p_m. This operator can be applied on a bit by bit basis with a probability inversely proportional to the number of bits of the chromosome representing each individual. A predetermined low probability can also be chosen by the user (generally $p_m \in [0.001, 0.02]$). In this application, a bit by bit mutation with $p_m = 0.02$ has been chosen.

ESs In the evolution strategy, the strategy parameters (standard deviation and probability of mutation) are modified during the optimization process by a mechanism of self adaptation. The strategy parameters undergo the evolution process (crossover and mutation).

Self adaptation in ESs is very different from what happens in GAs. In GAs, the parameters p_c and p_m (probability of crossover and mutation) are constant during the evolution process and they are chosen by the user, whereas in ESs, the strategy parameters σ and p undergo crossover and mutation, and are modified during the evolution process.

For continuous variables, the mutation operator operates by first mutating the standard deviations with a multiplicative, logarithmic, normally-distributed process, and then by mutating the continuous variables with a normally-distributed random vector [2]. For discrete variables, the mechanism of mutation used in this article is similar to that of continuous variables. The mutation probability is mutated first and then the discrete object variables are modified. This mechanism is described by Bäck and Schütz [3].

5 COMPUTATIONAL PROCEDURE

The purpose of this paper is to find the best strategy to solve mechanical optimal design problems. As no theoretical basis exists for the choice of genetic operators with GAs and ESs to solve this type of problem, the different selection and recombination operators presented in the previous chapter will be tested with regard to these problems. All in all, 6 tests with GAs and 96 tests with ESs will be carried out in order to find the best combination of ES and GA operators for the class of design problems.

To obtain statistically significant data, 100 runs for each combination of GA and ES operators will be performed. A population of 200 individuals is chosen. With GAs, 200 parents create 200 offspring. With ESs, the $(30+200)$-ES and the $(30, 200)$-ES are studied.

6 NUMERICAL EXAMPLES

6.1 Formulation of the Problems

GAs and ESs are applied to three mechanical design problems.

Problem 1 (Fig. 1) The first problem is the design of a pressure vessel [16]. The dimensions T_s (the shell thickness), T_h (the spherical head thickness), R (the radius of the cylindrical shell), and L (the length of the shell) are the design variables. The aim of the design is to minimize the total manufacturing cost for the pressure vessel. The optimization problem contains 2 discrete variables (T_s, T_h), 2 continuous variables (R, L) and 7 inequality constraints. Its formulation is as follows:

Minimize the objective function

$$f(T_s, T_h, R, L) = 0.6224\ T_s R L + 1.7781\ T_h R^2 + 3.1611\ T_s^2 L + 19.84\ T_s^2 R \tag{5}$$

Subject to the constraints:

$$\begin{array}{ll} g_1(T_s, T_h, R, L) = 0.0193\ R - T_s \le 0 & g_5(T_s, T_h, R, L) = 0.6 - T_h \le 0 \\ g_2(T_s, T_h, R, L) = 0.00954\ R - T_h \le 0 & g_6(T_s, T_h, R, L) = R - 70 \le 0 \\ g_3(T_s, T_h, R, L) = 1.1 - T_s \le 0 & g_7(T_s, T_h, R, L) = L - 50 \le 0 \\ g_4(T_s, T_h, R, L) = 752 \times 1728 - \pi R^2 L^2 - 4\pi R^2/3 \le 0 & \end{array} \tag{6}$$

Problem 2 (Fig. 2) The second problem is a coupling with a bolted rim [9]. A torque is transmitted by adhesion using N bolts of diameter d placed at radius R_B. The problem is to find the coupling with the smallest radius, the smallest number of bolts and the lowest torque. This objective function is a multi criteria function with weighting coefficients β_1, β_2, β_3. In this study it is assumed that the restrained linkage between the shaft and the coupling is chosen. The formulated optimization problem contains one discrete variable (d), one integer variable (N), two continuous variables (R_B, M), 11 inequality constraints and 5 discrete bolt

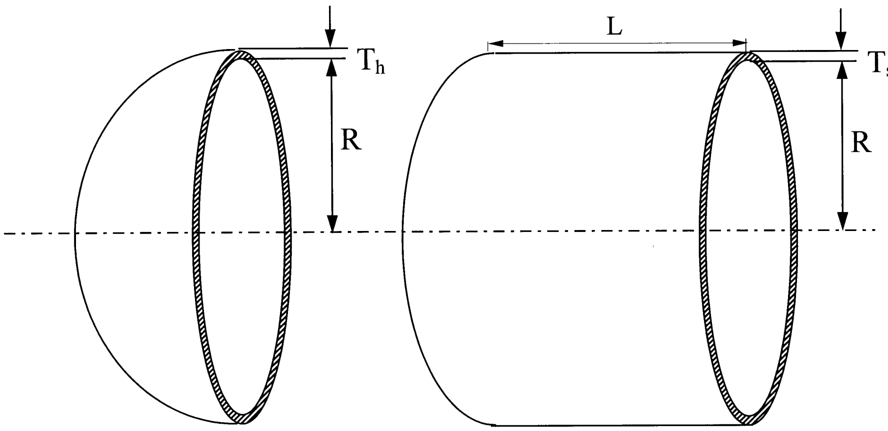

FIGURE 1 Pressure vessel.

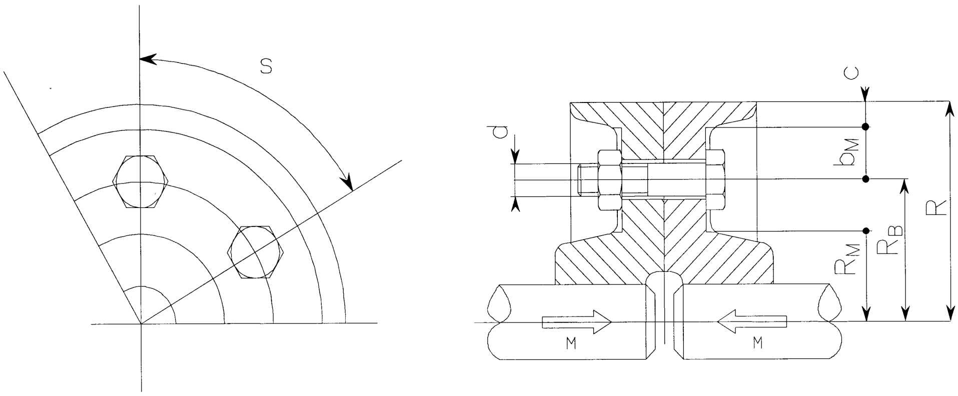

FIGURE 2 A coupling with bolted rim.

parameters ($\phi_i(d)$, $i = 1, \ldots, 5$). Details of the equations of the problem and values of the data can be found in Ref. [9]. The formulation of the problem is as follows:

Minimize the objective function:

$$f(d, N, R_B, M) = \beta_1\left(\frac{N}{N_m}\right) + \beta_2\left(\frac{(R_B + \Phi_4(d) + c)}{R_M}\right) + \beta_3\left(\frac{M}{M_T}\right) \tag{7}$$

Subject to the constraints:

$$\begin{aligned}
&g_1(d, N, R_B, M) = \frac{\alpha M}{(N.R_B.K(d))} - 1 \leq 0 && g_6(d, N, R_B, M) = N_m - N \leq 0\\
&g_2(d, N, R_B, M) = 1 - \frac{2\pi R_B}{\Phi_5(d).N} \leq 0 && g_7(d, N, R_B, M) = R_M - R_B \leq 0\\
&g_3(d, N, R_B, M) = 1 - \frac{R_B}{\Phi_4(d)} + R_M \leq 0 && g_8(d, N, R_B, M) = M - M_{MAXI} \leq 0\\
&g_4(d, N, R_B, M) = N - N_{MAXI} \leq 0 && g_9(d, N, R_B, M) = M_T - M \leq 0\\
&g_5(d, N, R_B, M) = R_B - R_{MAXI} \leq 0 && g_{10}(d, N, R_B, M) = d - 24 \leq 0\\
&g_{11}(d, N, R_B, M) = 6 - d \leq 0 &&
\end{aligned} \tag{8}$$

Data:

M_T, M_{MAXI}: lower and upper bound of the torque to be transmitted.
f_m, f_1: friction coefficient between the two rims and between the nuts and the screw.
α: torque wrench precision.
R_e: yield stress of bolts.
N_m, N_{MAXI} : lower and upper bound on the number of bolts.
R_M, R_{MAXI} : lower and upper bound on the radius of bolts
c: minimum thickness of the rim

$$\text{with } K(d) = \frac{0.9 f_m R_e \pi(\Phi_1(d))^2}{4\sqrt{1 + 3(0.16\Phi_3(d) + 0.583\Phi_2(d)f_1)/\Phi_1(d))^2}}$$

Problem 3 (Fig. 3) The last problem is a ball bearing pivot link. The aim is to find the lengths x_1, x_2 and the two ball bearings R_1 and R_2 in order to minimize the weight of the assembly composed of a shaft and two ball bearings. The ball bearings were chosen from a standardized table of prefabricated sizes. The formulated optimization problem contains 4 variables, 2 continuous (x_1, x_2), 2 integer (R_1, R_2) variables, 12 discrete parameters and 10 inequality constraints. R_1 and R_2 represent the choice of the two ball bearings. In order to solve this problem with GAs and ESs, we numbered the ball bearings, having a diameter of 30 to 45, from 1 to 28 in the same order as the standardized table. The parameters of the 2 ball bearings are (C_1, d_1, D_1, b_1, d_3, m_1) and (C_2, d_2, D_2, b_2, d_4, m_2) respectively, depending on the choice of the ball bearings. This gives the following formulation, with $X = \{R_1, R_2, x_1, x_2\}$:

Minimize the objective function:

$$f(X) = m_1 + m_2 + \rho\frac{\pi}{4}[0.5(d_1^2(b_1 - b_0) - (d_2 + 2t_2)^2(b_1 + b_2))] + \rho\frac{\pi}{4}[(d_1 + 2t_1)^2 b_3 + d_2^2 l_2 + x_1 d_1^2 + d_2^2(x_2 - b_3)] \tag{9}$$

Subject to the constraints:

$$\begin{array}{ll} g_1(X) = 0.5\,b_1 - x_1 + (0.5\,b_0 + e_1) \leq 0 & g_2(X) = D_2 - D_1 \leq 0 \\ g_3(X) = 29216\left(1 + \dfrac{x_1}{x_2}\right) - C_1 \leq 0 & g_4(X) = d_0 - d_1 \leq 0 \\ g_5(X) = 29216\left(\dfrac{x_1}{x_2}\right) - C_2 \leq 0 & g_6(X) = d_5 - d_2 \leq 0 \\ g_7(X) = (615.51x_1 + 3930)^{1/3} - d_1 \leq 0 & g_8(X) = D_1 - D_M \leq 0 \\ g_9(X) = 0.5\,b_1 + 0.5\,b_2 - x_2 + (e_4 + b_3) \leq 0 & g_{10}(X) = x_2 + x_1 + 0.5\,b_2 - 177 \leq 0 \\ \text{Data: } \{b_0, b_3, e_1, e_4, e_2, D_M, L_M, b_5, d_0, \rho\} & \end{array} \tag{10}$$

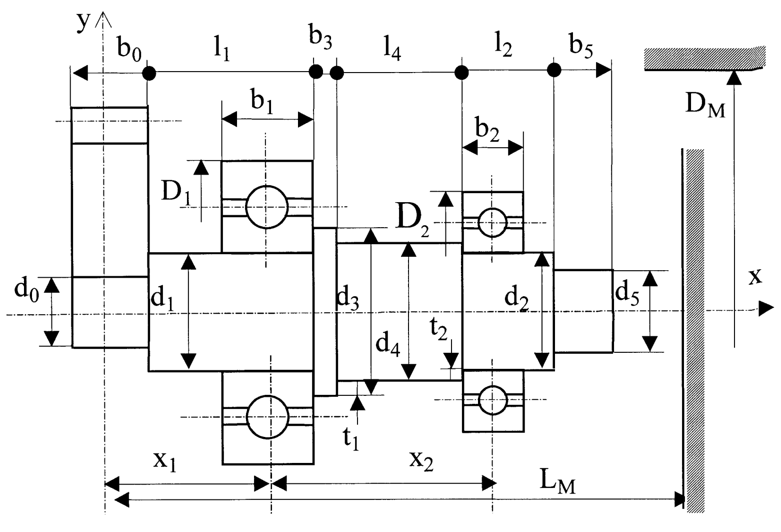

FIGURE 3 Ball bearing pivot link.

These 3 problems are real world mechanical engineering problems. For the three mechanical design problems studied in this paper, the analytical solution is known, so the results can be compared with the theoretical solution. These problems allow the best combination of operators to be found for the class of problems, of which they are part. Knowing these results, it will be possible to apply the best strategy to problems of the same class, but for which the theoretical solution is unknown.

6.2 The Best Combination of Operators for GAs and ESs

From the 6 tests made with GAs and the 96 tests made with ESs with respect to the design problems described above, the following conclusions can be drawn:

6.2.1 Selection

Based on the results of the different tests, the conclusions about the selection operators are the following:

- For GAs the rank-based method gives better results than the proportional one for the three problems (see Fig. 4 for example).
- For ESs the best results are obtained with the $(\mu + \lambda)$ selection from the viewpoint of convergence reliability and velocity (see Table II).

6.2.2 Recombination

GAs For the recombination methods with GAs, the two point crossover clearly increases the convergence velocity and the convergence reliability (see Fig. 5). The reason is that the

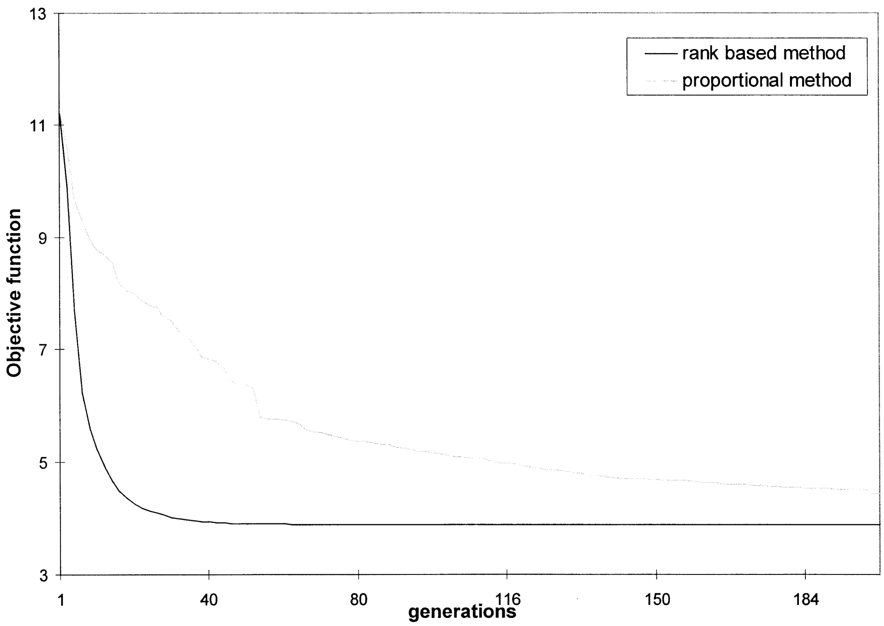

FIGURE 4 The two methods of selection with GAs (problem 2).

TABLE II Comparison of the Two Selection Methods of ES (Realized with $r_{xd}=D$, $r_p=I$, $r_{xc}=d$, $r_\sigma=I$).

	Problem 1		*Problem 2*		*Problem 3*	
	(μ, λ)	$(\mu+\lambda)$	(μ, λ)	$(\mu+\lambda)$	(μ, λ)	$(\mu+\lambda)$
Average error (%)	2.38	1.49	0.57	0.46	1.95	1.18

one-point crossover method has a very high probability of separating bits located at the extreme ends of the chromosome. Some recent studies [17] have shown that crossover operators with a higher number of crossover points may sometimes be more effective but they do not always give the best results for the present class of design problems. The results depend on the problem treated. The two-point crossover appears to be the best strategy.

ESs With regard to recombination for discrete variables and probability of mutation, for problems 2 and 3 it can be noted that:

- For the discrete variables, the results are a little more precise with the global form $((r_{xd}=D))$. For example for problem 2, the average error is equal to 0.46% with the global form and 0.56% with the local one.
- For the probability of mutation, the average reliability is better with the global intermediate method ($r_p=I$). For example, for problem 2, the average error equal to 0.46% with $r_p=I$ is respectively equal to 0.48%, 0.56%, 0.72% for $r_p=d$, D, $A1.5$. So $((r_{xd}=D)$ and $(r_p=I))$ will be kept for the remainder of this article.

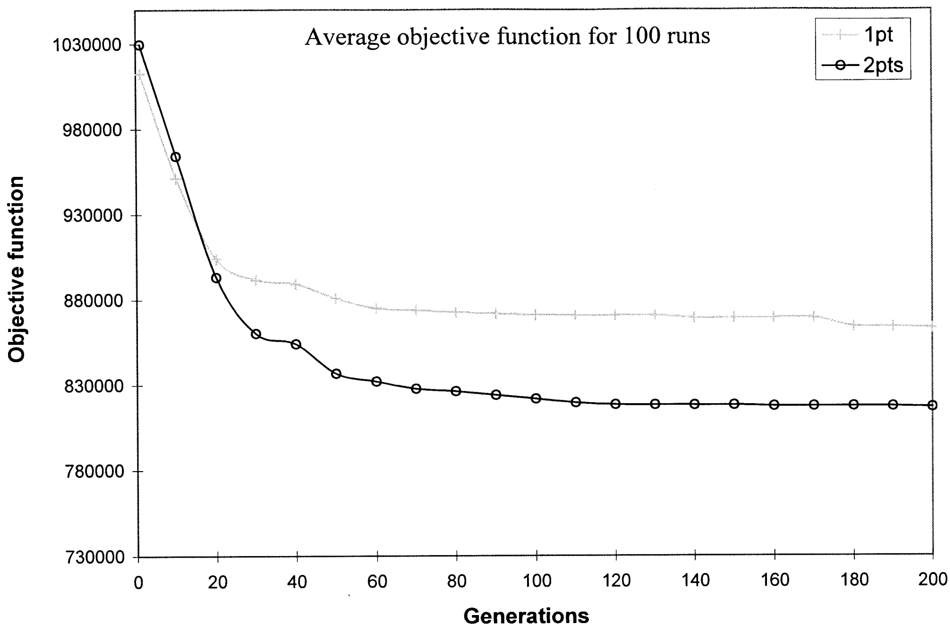

FIGURE 5 1-point crossover and 2-point crossover (problem 3).

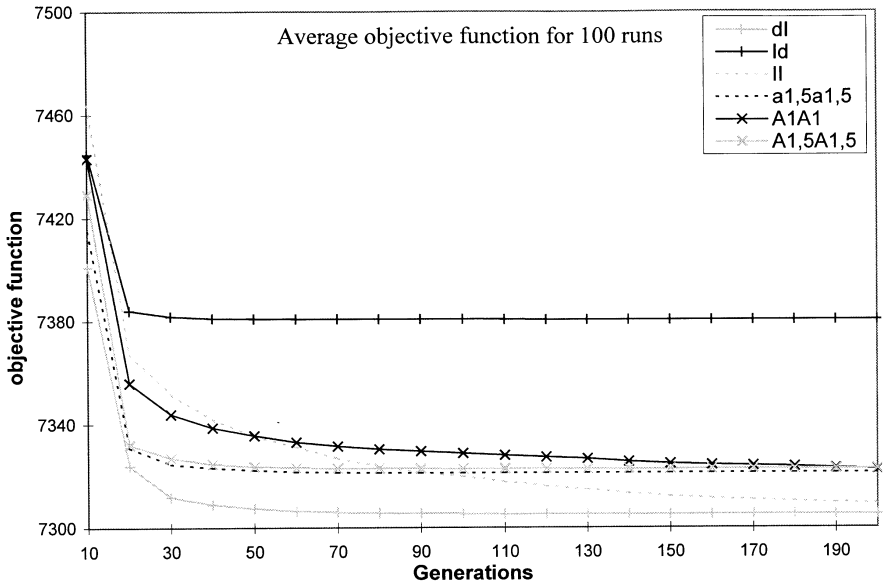

FIGURE 6 Comparison of the recombination operators of ES (problem 1) with $r_{xd}=D$, $r_p=I$.

With regard to recombination operators for continuous variables and standard deviations, it is very difficult to tell from the tests carried out for the three mechanical problems which operators are best (see Figs. 6–8). The two pairs of recombination operators $(r_{xc}, r_\sigma)=(a1.5, a1.5)$ and $(A1.5, A1.5)$ have similar behavior for all three problems. They both give

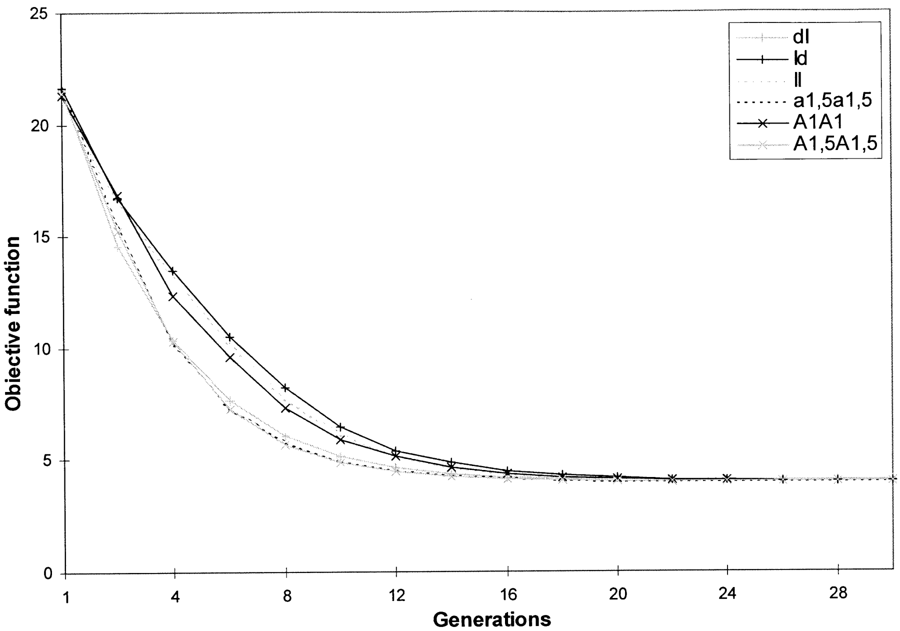

FIGURE 7 Comparison of the recombination operators of ES (problem 2) with $r_{xd}=D$, $r_p=I$.

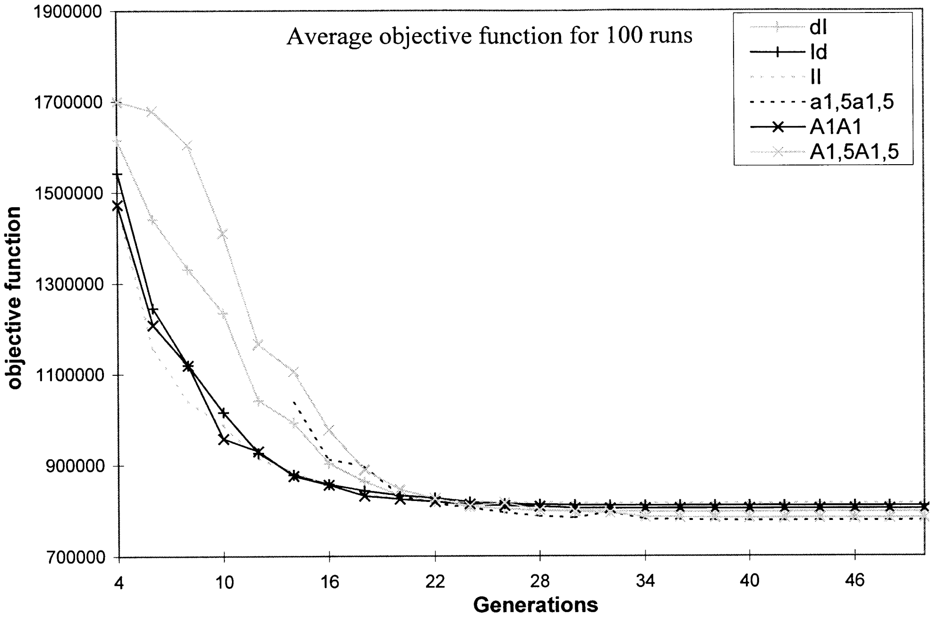

FIGURE 8 Comparison of the recombination operators of ES (problem 3) with $r_{xd} = D$, $r_p = I$.

the best convergence reliability for problems 2 and 3. For problem 1, the best convergence reliability and velocity is obtained with the pair of operators (d, I).

Table III presents the conclusions drawn from Figures 6–8 concerning the two pairs of recombination operators (d, I) and ($A1.5$, $A1.5$), the average of the objective function from 100 runs ($\bar{f}$) and its standard deviation (V). Other pairs of operators are not presented because their performance was not as good.

Finally, for problems 2 and 3, the operators ($A1.5$, $A1.5$) are kept for (r_{xc}, r_σ) because they give the best reliability and a good performance velocity (in the case of problem 2). These operators are different from the standard operators, advised by Bäck [1] ((r_{xc}, r_σ) = (d, I)) or from the operators used by Bäck and Schütz [3] ((r_{xc}, r_σ) = (I, I)). These results show that the recombination operator ($A1.5$) is better adapted, when difficult real-world mechanical design problems are solved, because it favors the exploration.

For problem 1, operators (r_{xc}, r_σ) = (d, I) which gave the best performance for this problem are kept. Problem 1 is the easiest problem. Exploration of the search space does not seem as

TABLE III Conclusions Concerning the Continuous Recombination Operators for ESs.

	Problem 1	*Problem 2*	*Problem 3*
(d, I)	reliability ↑ initial velocity ↑ $\bar{f}$=7307.32 V = 60.05	reliability → fast initial velocity ↑ $\bar{f}$=4.043 V = 0.239	reliability → initial velocity ↓ $\bar{f}$=796693.6 V = 91536.4
($A1.5$, $A1.5$)	reliability → f initial velocity ↑ $\bar{f}$=7323.54 V = 61.86	reliability ↑ initial velocity ↑ $\bar{f}$=3.986 V = 0.163	reliability ↑ initial velocity ↓ $\bar{f}$=783205.3 V = 45080.2

TABLE IV Operators Chosen.

	GAs	*ESs*
Selection	rank-based	$(\mu+\lambda)$
Recombination	two-point	$r_{xd}=D$
		$r_p=I$
		$(r_{xc}, r_\sigma)=(d, I)$ (prob 1)
		$(r_{xc}, r_\sigma)=(A1.5, A1.5)$ (prob 2, 3)

extensive as it is for the other problems. The best combinations of operators with GAs and ESs for the mechanical design problems are summarized in Table IV.

6.3 Comparison of GAs and ESs

The results of this comparison are summarized on Figures 9–11 and in Tables V and VI. Table V presents, for each problem, the error for the final best objective function value with regard to the theoretical solution (best error %), and the number of evaluations which were necessary to obtain the theoretical objective function with an error of 10% (Nb eval Ftheo 10%). Concerning the convergence reliability of the two algorithms, Table V shows that the theoretical global optimum was always identified by ESs, after 50 generations, whereas this is not so with GAs. GAs give good approximations to the global optimum for the three problems but never the exact solution. So a better convergence reliability is obtained with ESs. This first result can easily be explained by the fact that GAs work with a binary representation of individuals whereas ESs work with real vectors. The reliability given with GAs is always limited by the increment of discretization.

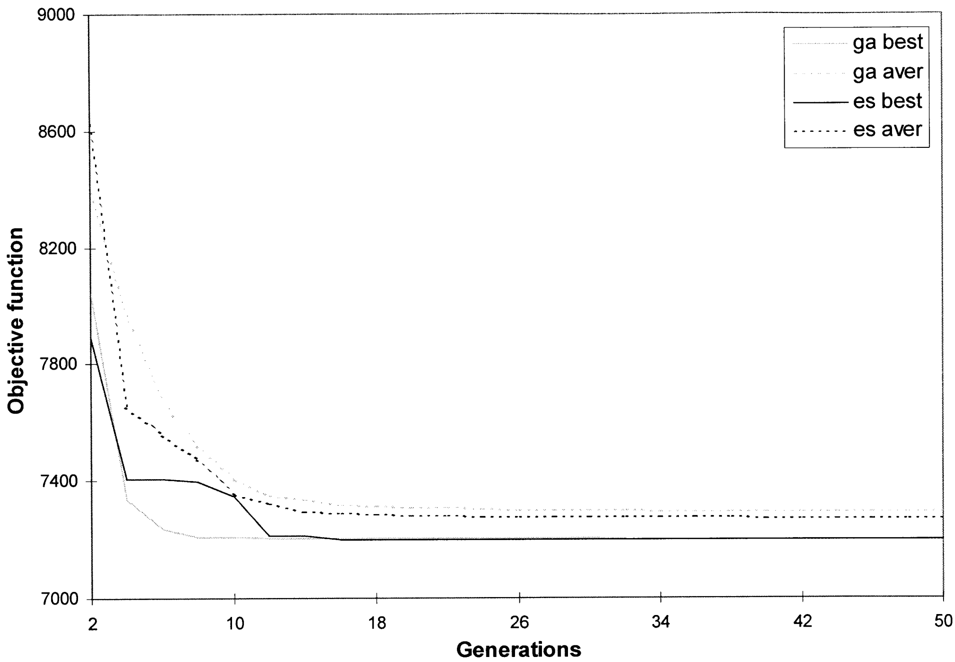

FIGURE 9 Comparison (problem 1).

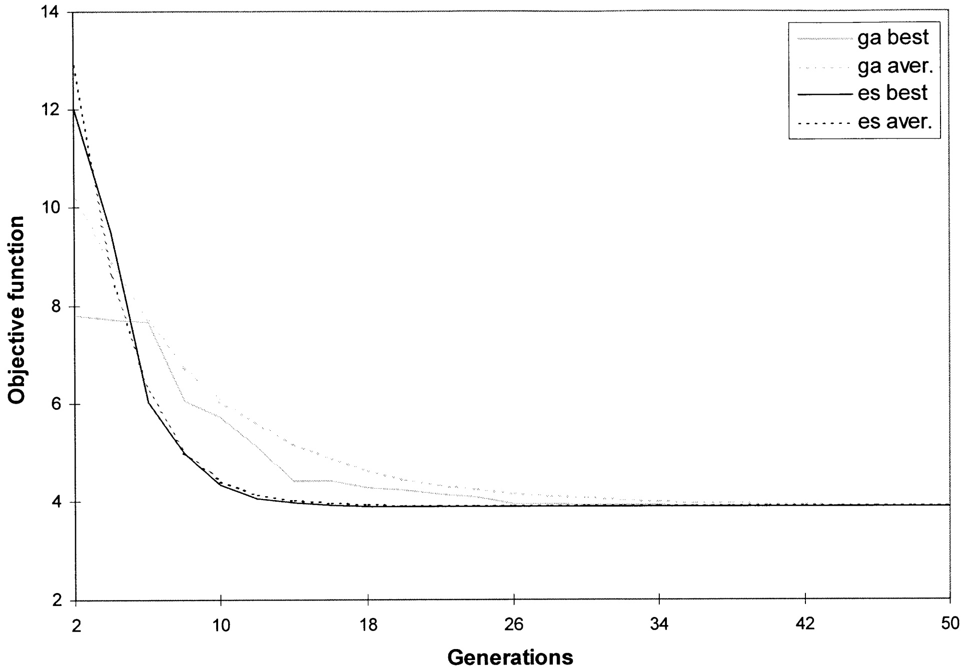

FIGURE 10 Comparison (problem 2).

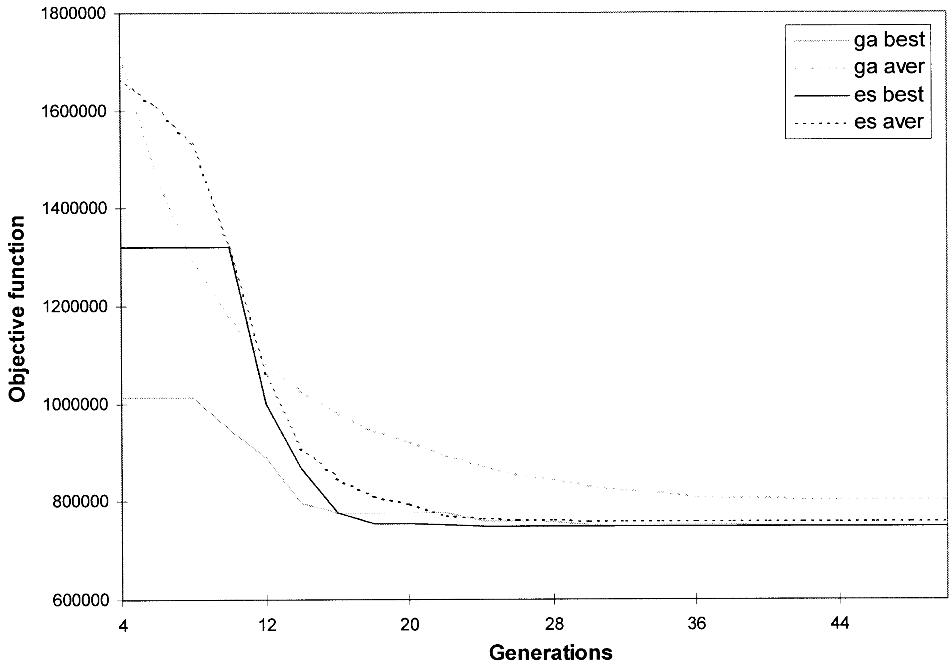

FIGURE 11 Comparison (problem 3).

TABLE V Error for the Final Best Objective Function Value at the 50th Generation.

	Problem 1		*Problem 2*		*Problem 3*	
	Best error (%)	*Nb eval. Ftheo 10%*	*Best error (%)*	*Nb eval. Ftheo 10%*	*Best error (%)*	*Nb eval. Ftheo 10%*
GA	0.0182	944	0.0272	5320	0.1994	6298
ES	0	584	0	2120	0	3125

TABLE VI Standard Deviations for 100 Runs after 50 Generations.

	GAs	*ESs*	*Optimum*
Problem 1	44.34	62.04	7197,73
Problem 2	0.0183	0.0675	3,8797
Problem 3	51683.02	28744.49	748315,6

Figures 9–11 show that convergence is faster with ESs than GAs for all three problems. Moreover, the more difficult the mechanical design problem, the greater is the difference in performance between GAs and ESs. GAs give poorer convergence reliability and velocity when the problem becomes difficult (problem 3). Figure 11 shows a rapid initial convergence rate with GAs for problem 3, which is better than with ESs. However, this convergence rate rapidly decreases over time (Figs. 10, 11). This comes from the fact that the probability of mutation with GAs is not adjusted during the optimization process, contrary to ESs which benefit from self adaptation. For the three problems, Table V shows that the number of evaluations needed to obtain a given reliability (the theoretical optimum with a 10% error) is about twice as much with GAs as it is with ESs. This confirms once again that the convergence velocity is better with ESs than it is with GAs.

Finally, for the three problems, ESs show a better convergence reliability and convergence velocity than GAs. These results were also found by Bäck and Schwefel [4] for three functions *i.e.* two unimodal functions and a multimodal function without constraints and with only continuous variables. The results of this paper, which were obtained for three highly constrained, mixed discrete-continuous nonlinear functions, show that ESs still give the best performance with this type of complex problem.

CONCLUSION

GAs and ESs have enabled solutions to be found of mechanical design problems expressed as highly constrained, mixed discrete–continuous nonlinear optimization problems. One of the principal advantages of these evolutionary algorithms is that they do not require any derivative information. This means that complex mechanical design problems can be solved for which the gradient functions cannot be computed and which therefore cannot be solved using classical methods.

GAs and ESs are both evolutionary algorithms based on the same concepts. The tests which were carried out with different selection and recombination operators enabled the best combination of operators to be found for each algorithm for the mechanical design problems studied. These operators are often different from the standard operators. These results show that the standard operators do not always give the best results, when difficult real-world problems of optimal design are solved.

Finally the comparison of GAs and ESs for three design problem indicates a better convergence reliability and convergence velocity with ESs. The problem with GAs is that the GA parameters (probability of crossover and mutation) are not adjusted during the optimization process, contrary to ESs.

The tests realized here on three design problems will now allow mechanical design problems of the same class to be solved by using the best combination of operators found in this article. Our future research will be on the coupling of an evolutionary algorithm with a deterministic algorithm in order to reduce the computational cost.

References

[1] Bäck, T. (1996). *Evolutionary algorithms in Theory and Practice.* Oxford University Press, New York.

[2] Bäck, T., Hammel, U. and Schwefel, H-P. (1997). Evolutionary computation: comments on the history and current state. *IEEE Transactions on evolutionary computation*, **1**, 1–15.

[3] Bäck, T. and Schütz, M. (1995). Evolution strategies for mixed-integer optimization of optical multilayer systems. In: McDonnell, J. R., Reynolds, R. G. and Fogel, D. B. (Eds.), *Proceedings of the Fourth Annual Conference on Evolutionary Programming.* MIT Press, Cambridge, MA, pp. 33–51.

[4] Bäck, T. and Schwefel, H. P. (1993). An overview of evolutionary algorithms for parameter optimization. *Evolutionary computation*, **1**, 1–23.

[5] Gehlhaar, D. K. and Fogel, D. B. (1996). Tuning evolutionary programming for conformationally flexible molecular docking. In: Fogel, L. J., Angeline, P. J. and Bäck, T. (Eds.), *Proceedings of the Fifth Annual Conference on Evolutionary programming*, MIT Press, Cambridge, MA, pp. 419–429.

[6] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison Wesley Publishing Company.

[7] Herdy, M. (1991). Application of the evolution strategy to discrete optimization problems. *Proceedings of the First International Conference on Parallel Problem Solving from Nature (PPSN)*, Lecture note in Computer Science, Vol. 496, Springer-Verlag.

[8] Holland, J-H. (1975). *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, Ann. Arbor, MI.

[9] Lafon, P. (1994). *Conception optimale de systèmes mécaniques: Optimisation en variables mixtes.* Thèse 3ème cycle, n° d'ordre 273, Institut National des Sciences Appliquées de Toulouse.

[10] Le Riche, R. (1994). *Optimisation de structures composites par algorithmes génétiques.* Thèse 3ème cycle, Université de Technologie de Compiègne.

[11] Michalewicz, Z. (1995). Genetic Algorithms, numerical optimization, and constraints. *ICGA-95, Proceeding of the Sixth International Conference on Genetic Algorithms*, Pittsburg.

[12] Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs.* Springer, Berlin.

[13] Michalewicz, Z. and Schoenauer, M. (1996). Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation,* **4**, 1–32.

[14] Schütz, M. and Sprave, J. (1995). Application of parallel mixed-integer evolution strategies with mutation rate pooling. In: Fogel, L. J., Angeline, P. J. and Bäck, T. (Eds.), *Proceedings of the Fifth Annual Conference on Evolutionary Programming (EP96)*, San Diego CA, pp. 345–354.

[15] Schwefel, H. P. (1981). *Numerical Optimization of Computer Models.* Wiley, Chichester.

[16] Wu, S-J. and Chow, P. T. (1994). Genetic algorithms for solving mixed-discrete optimization problems. *The Franklin Institute*, **331B**(4), 381–401.

[17] Wu, S-J. and Chow, P. T. (1995). Steady-state genetic algorithms for discrete optimization of trusses. *Computers and Structures*, **56**(6), 979–991.