Advanced solution methods in structural optimization based on evolution strategies

Manolis Papadrakakis and Nikolaos D. Lagaros
Institute of Structural Analysis and Seismic Research, National Technical University of Athens, Athens, Greece and
Georg Thierauf and Jianbo Cai
Department of Civil Engineering, University of Essen, Essen, Germany

Introduction
During the last three decades there has been a growing interest in problem solving systems based on algorithms which rely on analogies to natural processes. The best known algorithms in this class include evolutionary programming (Fogel, 1992; Fogel et al., 1966), genetic algorithms (GAs) (Goldberg, 1989; Holland, 1975), evolution strategies (ESs) (Rechenberg, 1973; Schwefel, 1981), simulated annealing (Kirkpatrick et al., 1983), classifier systems and neural networks (Berke and Hajela, 1990; Papadrakakis et al., 1996). Evolution-based systems maintain a population of potential solutions. These systems have some selection process based on fitness of individuals and some recombination operators.

Evolution strategies were proposed for parameter optimization problems in the 1970s by (Rechenberg, 1973) and (Schwefel, 1981). Similar to GAs the ESs imitate biological evolution in nature and have two characteristics that differ from other conventional optimization algorithms:

1. in place of the usual deterministic operators, they use randomized operators: mutation, selection and recombination;
2. instead of a single design point, they work simultaneously with population of design points in the space of variables.

The second characteristic allows for natural implementation of GAs and ESs in a parallel computer environment. The ESs, however, achieve a higher rate of convergence than GAs owing to their self-adaptation search mechanism and are considered more efficient for solving real world problems (Hoffmeister and Back, 1991). The ESs developed by Rechenberg (1973) and Schwefel (1981) were commonly applied for continuous optimization problems. For solving discrete optimization problems a modified implementation has been introduced by Thierauf and Cai, (Cai, 1995; Cai and Thierauf, 1993; Thierauf and Cai, 1994;
In the following, the original and the modified evolution strategies for continuous and discrete optimization problems are described.

Optimization of large-scale structures such as multi-storey 3D frames subjected to actual constraints imposed by design codes is a computationally intensive task. In conventional optimization approaches, where sensitivity analysis plays a very important role 60 to 90 per cent of the computations are for the solution of the finite element equilibrium equations (Papadrakakis et al., 1996). When evolution strategies based optimization are adopted the computations involved in the solution of the finite element equations become greater than 95 per cent. In the present study two innovative finite element solution methods are implemented, based on the preconditioned conjugate gradient (PCG) algorithm, which are properly adjusted to the special characteristics of the evolution strategy process. The first method is a PCG algorithm with a preconditioner resulting from a complete Cholesky factorization of the stiffness matrix and the second is a PCG algorithm in which a truncated Neumann series expansion is used as a preconditioner. The numerical tests presented demonstrate the computational advantages of the proposed methods which become more pronounced in large-scale or computationally intensive optimization problems as well as in a parallel computing environment.

Evolution strategy for structural optimization
For convenience of further discussion the original and modified evolution strategies will be described briefly. The ESs can be divided into two-membered evolution strategy (2-ES) and multi-membered evolution strategy (M-ES).

The two-member ES
The earliest evolution strategies were based on a population consisting of one individual only. The two membered scheme is the minimal concept for an imitation of organic evolution. The two principles of mutation and selection, which Darwin in 1859 recognised to be most important, are taken as rules for variation of the parameters and for recursion of the iteration sequence respectively.

The following optimization problem is considered:

\[
\begin{align*}
\min & \quad f(X), \\
\text{subject to} & \quad g_i(X) \geq 0, \quad i = 1,2,\ldots,m.
\end{align*}
\]  

(1)

The two-membered ESs for the solution of the optimization problem (1) works in two steps:

Step 1 (mutation). The parent \(X_p^{(g)}\) of the generation \(g\) produces an offspring \(X_0^{(g)}\), whose genotype is slightly different from that of the parent:

\[
X_0^{(g)} = X_p^{(g)} + Z^{(g)}
\]  

(2)
where $Z^{(g)} = [Z_1^{(g)}, Z_2^{(g)}, ..., Z_n^{(g)}]^T$ is a random vector.

Step 2 (selection). The selection chooses the best individual between the parent and the offspring to survive:

$$X_p^{(g+1)} = \begin{cases} 
X_o^{(g)} & \text{if } g_1(X_o^{(g)}) \geq 0 \quad i=1,2,...,m \text{ and } f(X_o^{(g)}) \leq f(X_p^{(g)}) \\
X_p^{(g)} & \text{otherwise}
\end{cases} \quad (3)$$

The question is how to choose the random vector $Z^{(g)}$ in Step 1. This choice is the role of mutation. Mutation is understood nowadays to be random, purposeless events, which, furthermore, occur very rarely. If one interprets them, as is done here, as a sum of many individual events, it is a natural choice to use a probability distribution according to which small changes occur frequently, but large ones only rarely. Two requirements arise together by analogy with natural evolution: the expectation value $\xi_i$ for a component $z_i$ has the value zero; and the variance $\sigma_i^2$, the average squared deviation from the mean, is small.

The probability density function for normally distributed random events is given by

$$p(z_i) = \frac{1}{\sqrt{(2\pi)\sigma_i}} \exp\left(-\frac{(z_i - \xi_i)^2}{2\sigma_i^2}\right) \quad (4)$$

If $\xi_i = 0$ the so-called $(0, \sigma_i)$ normal distribution is obtained. By analogy with other deterministic search strategies, $\sigma_i$ can be called step length, in the sense that it represents average values of the length of the random steps.

If the step length is too small the search takes an unnecessarily large number of iterations, if it is too large, on the other hand, the optimum can only be crudely approached, and the search can even get stuck far from the global optimum. Thus, as in all optimization strategies, the step length control is the most important part of the algorithm after the recursion formula, and it is, furthermore, closely linked to the convergence behaviour.

The standard deviation $\sigma_i$ which is considered as the step length can be adjusted during the search as follows (Rechenberg's $\frac{1}{5}$ success rule (Rechenberg, 1973)): “The ratio of successful mutations to all mutations should be $\frac{1}{5}$. If it is greater, increase; if it is less, decrease the standard deviations $\sigma_i^2$. According to (Schwefel, 1981), the check should take place every $n$ mutations over the preceding $10n$ mutations, while the increase and decrease factors of the step length should be $(1/0.85)$ and 0.85 respectively.
Multi-membered ES

The multi-membered evolution strategies differ from the previous two-membered strategies in the size of the population. In this case a population of \( \mu \) parents will produce \( \lambda \) offspring. Thus the two steps are defined as follows.

Step 1 (recombination and mutation). The population of \( \mu \) parents at \( g \)th generation produces \( \lambda \) offspring. The genotype of any descendant differs only slightly from that of its parents.

Step 2 (selection). There are two different types of the multi-membered ES:

- \(( \mu + \lambda )\)-ES: The best \( \mu \) individuals are selected from a temporary population of \(( \mu + \lambda )\) individuals to form the parents of the next generation.
- \(( \mu, \lambda )\)-ES: The \( \mu \) individuals produce \( \lambda \) offsprings \(( \mu < \lambda )\) and the selection process defines a new population of \( \mu \) individuals from the set of \( \lambda \) offsprings only.

In the second type, the life of each individual is limited to one generation. This allows the \(( \mu, \lambda )\)-ES selection to perform better on problems with an optimum moving over time, or on problems where the objective function is noisy.

In Step 1, for every offspring vector a temporary parent vector \( \tilde{X} = [\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n]^T \) is first built by means of recombination. For a continuous problem five recombination cases which can be used selectively are given in (Hoffmeister and Back, 1991; Schwefel, 1981):

\[
\tilde{x}_i = \begin{cases} 
  x_{a,i} \text{ or } x_{b,i} \text{ randomly} & \text{(A)} \\
  1/2(x_{a,i} + x_{b,i}) & \text{(B)} \\
  x_{b,j,i} & \text{(C)} \\
  x_{a,i} \text{ or } x_{b,j,i} \text{ randomly} & \text{(D)} \\
  1/2(x_{a,i} + x_{b,j,i}) & \text{(E)} 
\end{cases}
\]  

Where \( \tilde{x}_i \) is the \( i \)th component of the temporary parent vector \( \tilde{X} \), \( x_{a,i} \) and \( x_{b,i} \) are the \( i \)th components of the vectors \( X_a \) and \( X_b \) which are two parent vectors randomly chosen from the population. In case C of Equation (5), \( \tilde{x}_i = x_{b,j,i} \) means that the \( i \)th component of \( \tilde{X} \) is chosen randomly from the \( i \)th components of all \( \mu \) parent vectors (Figure 1). From the temporary parent \( \tilde{X} \) an offspring can be created in the same way as in two-membered ES (Equation (2)).

During the search, not only the design variables \( x_i \), but also the parameters, such as the deviations \( \sigma_i \), will be modified by the random operator mutation which replaces the \( \% \) success rule (Schwefel, 1981).
The modified evolution strategies

For the solution of discrete optimization problems modified evolution strategies are proposed by Thierauf and Cai (Cai, 1995; Cai and Thierauf, 1993). A discrete structural optimization problem can be formulated in the following form:

\[ \min f(\mathbf{X}), \quad \mathbf{X} = [x_1, x_2, \ldots, x_n]^T \]

subject to: \[ g_i(\mathbf{X}) \geq 0, \quad i = 1, 2, \ldots, m. \]
\[ x_j \in \mathbb{R}^d, \quad j = 1, 2, \ldots, n \]

(6)

Where \( \mathbb{R}^d \) is a given set of discrete design values. The design variables \( x_i \) (\( i = 1, 2, \ldots, n \)) can only take the discrete values of this set.

Mutation

The mutation operator, in the continuous version of the ES based optimization, produces a normally distributed random change vector \( \mathbf{Z}^{(g)} \). Each component of this vector has small standard deviation value \( \sigma \) and zero expectation. This means that there is a possibility that all components of a parent vector will need to be changed but the changes are usually small.

In the discrete version of the ESs we have to change the generator of the random vector \( \mathbf{Z}^{(g)} \) in order to produce a modified vector that leads from one discrete value to another adjacent one. The difference between any two adjacent values is usually not small, which is against the second requirement that arises by analogy to the natural evolution. For this reason, it is suggested that not all
of the \( n \) components of a parent vector, but only a few (say \( \ell \)) will be randomly changed every time. This means that \( (n - \ell) \) components of the randomly changed vector \( Z^{(g)} \) have zero value.

The components of the randomly changed vector \( Z^{(g)} \) have the form (Cai, 1995; Cai and Thierauf, 1993):

\[
Z_i^{(g)} = \begin{cases} 
(\kappa + 1)\delta x_i & \text{for } \ell \text{ randomly chosen components} \\
0 & \text{for } n - \ell \text{ other components}
\end{cases}
\]  

(7)

where \( \delta x_i \) is the current difference between two adjacent values in the discrete set and \( \kappa \) is a Poisson distributed integer random number with the following distribution

\[
p(\kappa) = \frac{(\gamma)^\kappa}{\kappa!} e^{-\gamma}
\]

(8)

in which \( \gamma \) is the deviation and the expectation of the random number \( \kappa \). For a very small \( \gamma \) (say 0.001) the number \( \kappa \) takes the value zero with probability greater than 99 per cent, and for \( \gamma = 0.05 \) the number \( \kappa \) can take the value zero with probability of 95 per cent and the value one with probability of 5 per cent. This means that the random change \( z_i^{(g)} \) is controlled by the parameter \( \gamma \).

A uniformly distributed random choice decides which \( \ell \) components should be changed according to Equation (7). For structural optimization problems, according to our experience, a suitable \( \ell \) value ranges from 8 to 12 (Cai, 1995).

Recombination

For discrete optimization problems the recombination cases (B) and (E) in Equation (5) are not suitable, because the mean value of two discrete values is usually not a discrete value from the given set. In this work the following recombination cases are employed:

\[
\bar{x}_i = \begin{cases} 
x_{a,i} \text{ or } x_{b,i} \text{ randomly} & \text{(A)} \\
x_{b,j,i} & \text{(B)} \\
x_{a,i} \text{ or } x_{b,j,i} \text{ randomly} & \text{(C)} \\
x_{m,i} \text{ or } x_{b,i} \text{ randomly} & \text{(D)} \\
x_{m,i} \text{ or } x_{b,j,i} \text{ randomly} & \text{(E)}
\end{cases}
\]

(9)
The recombination cases (A), (B) and (C) are the same as those in Equation (5).

The last two cases (D) and (E) in Equation (9) are suggested by Thierauf and Cai (Cai, 1995; Cai and Thierauf, 1993). The vector $X_m$ is not randomly chosen but is the best of the $\mu$ parent vectors in the $g$th generation. In cases (D) and (E), if the information from the best parent is used it can lead to a better convergence for many problems. From the temporary parent $X^T$ an offspring can then be created as in the two-membered type ES (Equation 2).

For discrete optimization the following termination criteria are suggested in (Cai, 1995; Thierauf and Cai, 1994), which can be used selectively:

1. if the best value of the objective function in the last $k_1 (>4n\mu/\lambda)$ generations has not been improved;
2. if the mean value of the objective values from all parent vectors in the last $k_{II} (>2n\mu/\lambda)$ generations has not been improved by less than a given value $e_{ii}$;
3. if the relative difference between the best objective function value and the mean value of the objective values from all parent vectors in the current generation is less than a given value $e_{ic} (= 0.0001)$;
4. if the ratio $\mu_b/\mu$ has reached a given value $e_{ij}$ (= 0.5 to 0.8) where $\mu_b$ is the number of the parent vectors in the current generation with the best value of objective function.

Hybrid solution methods

The algebraic definition of an evolution strategy procedure applied to a structural system with the finite element equation $K_u = f$ may be described as follows:

1. Initialization:
   selection of $X_i$ ($i = 1, 2, ..., \mu$) parent vectors of the design variables
   set $K_0 = K(X_1)$
   solve $K_0u_1 = f$
   $(K_0 + \Delta K (X_i))u_i = f$, ($i = 2, 3, ..., \mu$).
2. Constraints check.
3. Offspring generation:
   generate $X_j$, ($j = 1, 2, ..., \lambda$) offspring vectors of the design variables.
4. Solution step:
   $(K_0 + \Delta K (X_j))u_j = f$, ($j = 1, 2, ..., \lambda$).
5. Constraints check:
   if satisfied continue, else change $X_j$ and go to step 4.
(6) Selection step:
selection of the next generation parents according to $(\mu + \lambda)$ or $(\mu, \lambda)$ selection schemes.

(7) Convergence check:
if satisfied stop, else go to step 3.

It can be seen that the solution of at least $\lambda$ systems of finite element equations need to be solved at each generation, where $\Delta K(X_j)$ defines the modification of the stiffness matrix due to the changes on the design variables and is generally small compared to $K_0$.

The implementation of hybrid solution schemes, which are based on a combination of direct and preconditioned iterative methods, may drastically reduce the time required for the solution of the stiffness equations involved in the evolution strategy procedure, having an overall beneficial effect on the efficiency of the optimization procedure. In this work two hybrid methods are applied in the context of optimization problems solved by evolution strategies. The methods are properly modified to address the special features of the particular optimization problems at hand, while mixed precision arithmetic operations are proposed, resulting in additional savings in computer time and storage without affecting the accuracy of the solution.

The preconditioned conjugate gradient method
The optimization procedure, described in the previous section, requires a large number of consecutive linear solutions in which the overall stiffness matrix is slightly modified from one solution to the other. This special feature of the problem makes the preconditioned conjugate gradient method (PCG) very attractive for the solution of the linear problem. The PCG method is established as the more attractive iterative procedure for solving linear problems resulting from the finite element discretization. An important factor in the success of this method in solving large-scale finite element equations is the preconditioned technique used to improve the ellipticity of the coefficient matrix. This typically consists of replacing the original system $Ku = f$ by the equivalent system:

$$ R^{-1}Ku = R^{-1}f $$

(10)

where $R$ is the transformation or preconditioning matrix which is an approximation to $K$ and it is non-singular. The PCG algorithm, based on the most efficient conjugate gradient version in respect to computational labour, storage requirements and accuracy, is defined as follows for the untransformed variables:
At the heart of the PCG iterative procedure for solving $Ku = f$ is the determination of the residual vector and the selection of the preconditioning matrix. The accuracy achieved and the computational labour of the method is largely determined by how these two parameters are selected. A study performed in (Papadrakakis and Bitoulas, 1993) revealed that the computation of the residual vector from its defining formula $r(m) = K u(m) - f$ offers no improvement in the accuracy of the computed results. In fact, it was found that, contrary to previous recommendations, the calculation of the residuals by the recursive expression of algorithm (11) produces a more stable and well-behaved iterative procedure. Based on this observation a mixed precision PCG implementation is proposed in which all computations are performed in single precision, except for double precision computation of the matrix vector multiplication occurring during the recursive evaluation of the residual vector. This implementation is a robust and reliable solution procedure even for handling large and ill-conditioned problems, while it is also computer storage-effective. It was also proved to be more cost effective, for the same storage demands, than double precision calculations (Papadrakakis and Bitoulas, 1993).

The preconditioned matrix $R$ has to be selected appropriately so that the eigenvalues of $R^{-1}K$ are spread over a much narrower range than those of $K$ (Bitoulas and Papadrakakis, 1994). For the type of problems considered in this study $K_0$ is taken as the preconditioning matrix. The diagonal and triangular factors of the LDLT factorization of $K_0$ are stored in single precision arithmetic, while the frequency of updating the preconditioning matrix is controlled either

\[
\alpha_m = \frac{(r(m), z(m))}{(d(m), Kd(m))}
\]

\[
u(m+1) = u(m) + \alpha_m d(m)
\]

\[
\begin{align*}
\beta_m &= \frac{(r(m+1), z(m+1))}{(r(m), z(m))} \\
d(m+1) &= -z(m+1) + \beta_m d(m)
\end{align*}
\]

\[
\text{if } \frac{\|r(m+1)\|}{\|f\|} \leq \varepsilon_i \text{ then stop}
\]

\[
z(m+1) = R^{-1}r(m+1)
\]

\[
\text{with }\ r^{(o)} = Ku^{(o)} - f, \quad z^{(o)} = R^{-1}r^{(o)}, \quad d^{(o)} = z^{(o)}
\]

(11)
Advanced solution methods

The solution of a typical reanalysis problem

\[(K_0 + \Delta K)u = f\]  (12)

yields

\[u = (I + K_0^{-1} \Delta K)^{-1} K_0^{-1} f\]  (13)

The term in parentheses can be expressed in a Neumann expansion giving

\[u = (I - P + P^2 - P^3 + \ldots) K_0^{-1} f\]  (14)

with \(P = K_0^{-1} \Delta K\). The solution vector can now be represented by the following series

\[u = u_0 - Pu_0 + P^2 u_0 - P^3 u_0 + \cdots\]  (15)

or

\[u = u_0 - u_1 + u_2 - u_3 + \cdots\]  (16)

The series solution can also be expressed by the following recursive equation

\[K_0 u_i = \Delta K u_{i-1}\quad i = 1, 2, \ldots\]  (17)

The advantage of this expression is that the stiffness matrix has to be factorized once while the additive terms \(u_i\) to the solution of Equation (16) can be computed by successive backward and forward substitutions. The series may be truncated after a fixed number of terms or according to some error norm given by...
or
\[ \| r_i \| / \| f \| \leq \varepsilon_2 \]  
where \( r_i = f - (K_0 + \Delta K)u_i \). The first criterion is most frequently used since it is computationally more efficient. The second criterion requires the evaluation of the residual vector which involves an additional matrix vector multiplication for each Neumann term.

In order to improve the quality of the preconditioning matrix \( R \) used in the PCG method, the Neumann series expansion is implemented for the calculation of the preconditioned vector \( z^{(m)} = R^{-1}r^{(m)} \) of the PCG algorithm. The preconditioning matrix is now defined as the current global stiffness matrix \( K_0 + \Delta K(X_j) \), but the solution for \( z \) is performed approximately using a truncated Neumann series expansion. Thus, the preconditioned vector \( z \) of the PCG algorithm is obtained at each iteration by
\[ z = z_0 - z_1 + z_2 - z_3 + \ldots \]  
where superscript \( m \) is dropped for clarity and \( z_0, z_i \) are given by
\[ z_0 = K_0^{-1}r \]  
and
\[ K_0z_i = \Delta K(X_j)z_{i-1} \quad i = 1, 2, \ldots \]  
The incorporation of the Neumann series expansion in the preconditioned step of the PCG algorithm can be seen from two different perspectives. From the PCG point of view an improvement of the quality of the preconditioning matrix is achieved by computing a better approximation to the solution of \( u = (K_0 + \Delta K(X_j))^{-1}f \) than the one provided by the preconditioning matrix \( K_0 \). From the Neumann series expansion point of view, the inaccuracy entailed by the truncated series is alleviated by the conjugate gradient iterative procedure. It remains to be seen, however, whether the anticipated improvement of the convergence properties of the PCG method or of the Neumann series expansion implies also a reduction on the overall computational effort by counteracting the additional cost involved at each iteration.

The storage requirements for NCG in mixed precision arithmetic are the following: the compact stiffness \( K \) is stored in double precision arithmetic and the factorized skyline stiffness and \( \Delta K \) are stored in single precision arithmetic, while \( 2^{\left[ \left\lfloor N/2 \right\rfloor + 1 \right]} \) short integers are needed for the addressings.
A parallel implementation of the solution schemes

In evolution strategies, at each design optimization iteration (generation) the evaluation of the objective function involves a large number of finite element analyses of the structure (one analysis for each member of the population). An important characteristic of ESs that differ from other conventional optimization algorithms is that in place of a single design point the ESs work simultaneously with a population of design points in the space of variables. This allows for a natural implementation of the evolution procedure in a parallel computer environment. Since a number of finite element analyses of the structure can be performed independently and concurrently, a complete finite element analysis can be assigned to a processor without any need of inter-processor communication during the solution phase. Therefore the parallelization of the ESs is based on the fundamental premise that each individual in the population of the offsprings represents an independent group of all design variables, and therefore its function evaluation can be done independently and concurrently.

In a distributed memory computing environment this implementation can be realised provided that there is enough memory at each processor to accommodate the matrices and vectors required by the solution algorithm. In a shared memory environment, however, it is the number of processors to be employed that is affected by the storage limitations. In our case the computations were performed on the SGI Power Challenge shared memory computer where parents and offsprings are of equal number \( \mu = \lambda \) and the number of processors is taken \( p = \mu \). There is no need for a host processor to accumulate all information from the other \( p - 1 \) processors, as in distributed computing environment, in order to select the \( \mu \) parents of the next generation.

In order to use the hybrid solution methods in the parallel version of the evolution strategies an initial \( K_0^{(i)} \), \( i = 1, 2, \ldots, p \) is assigned to each processor. The factorization is performed simultaneously in all processors when one of the following factorization criteria is satisfied in more than 80 per cent of the processors:

1. the number of PCG iterations per analysis is greater than \( m \), where \( m \) is the ratio of the factorization time over the time required for a forward and backward substitution;
2. the improvement of the objective function becomes less than a given value \( \varepsilon_2 \)

\[
\frac{f_j - f_{j+1}}{f_j} \leq \varepsilon_2
\]  

(23)

where \( f_j \) is the objective function at generation \( j \). Each time the criterion (2) is satisfied the tolerance parameter \( \varepsilon_2 \) is divided by 2. The initial value is taken \( \varepsilon_2 = 0.001 \).
Numerical tests

Three test examples have been considered to illustrate the efficiency of the discrete ESs in structural design problems. In all examples the modulus of elasticity is 200 GPa (29,000 ksi) and the yield stress is \( \sigma_y = 250 \text{ MPa (36 ksi)} \). The cross section of each member is assumed to be a W-shape (AISC, 1989) and for each member two design variables are considered as shown in Figure 2. The objective function of the problems is the weight of the structure. The constraints are the member stresses and inter-storey drifts. For rigid frames in rolled W-shapes, under allowable stress design requirements specified by (AISC, 1989), the stress constraint is the non-dimensional ratio \( q \) of the AISC interaction formulas:

\[
q = \frac{f_a}{F_a} + \frac{f_b}{F_b} \leq 1.0 \quad \text{if} \quad \frac{f_a}{F_a} \leq 0.15
\]

\[
q = \frac{f_a}{0.60 \cdot \sigma_y} + \frac{f_b}{F_b} \leq 1.0 \quad \text{if} \quad \frac{f_a}{F_a} > 0.15
\]

where \( f_a \) is the computed compressive axial stress, \( f_b \) is the computed bending stress, \( F_a \) is the allowable compressive axial stress, \( F_b \) is the allowable bending stress and \( \sigma_y \) is the yield stress of the steel. The allowable inter-storey drift is limited to 1.5 per cent of the height of each storey. One load case is considered in all examples.

A compact storage scheme is used for PCG and NCG methods to store the stiffness matrix in a mixed precision arithmetic mode, where all operations are performed in single precision except for double precision computation of the matrix-vector multiplication occurring during the evaluation of the residual vector. The termination parameter \( \epsilon_1 \) of the algorithm (11) is taken \( \epsilon_1 = 0.1 \). All
test results shown in this section were performed on the SG Power Challenge Shared memory computer.

Example 1
The first example is a six storey space frame, first analysed by (Orbinson et al., 1982) with 63 elements and 180 degrees of freedom. The beams have $L_1 = 7.32m$ and the columns $L_2 = 3.66m$. The loads consisting of 19.16kPa gravity load on all floor levels and a lateral load of 109kN applied at each node in the front elevation in the $z$ direction. The direct skyline method is used for the required finite element analysis since this example is very small to access the efficiency of the solution procedures described in the previous sections. The element members are divided into five groups shown in Figure 3 and the total number of design variables is ten. Table I shows the performance of the two types of the multimembered ESs, namely $(\mu + \lambda)$ - ES and $(\mu, \lambda)$ - ES, for different numbers of parents and offsprings. Table II depicts the influence of the deviation parameter $\lambda$, which controls the generation of the random vector $Z^{(g)}(\text{Equations (7),(8)})$, in the case of $(\mu + \lambda)$ - ES. The influence of the initial design and the type of termination criterion used for discrete optimization is demonstrated in Tables III and IV, respectively.
Example 2

The second example is the 20-storey space frame first analysed by (Orbinson et al., 1982) with 460 columns and beams and 1,200 degrees of freedom (Tables V and VI). The loads considered here are uniform vertical forces applied at

<table>
<thead>
<tr>
<th>ES type</th>
<th>Number of FE analyses</th>
<th>Number of generations</th>
<th>Value of the objective function</th>
</tr>
</thead>
<tbody>
<tr>
<td>(5 + 5) – ES</td>
<td>171</td>
<td>45</td>
<td>880</td>
</tr>
<tr>
<td>(10 + 10) – ES</td>
<td>595</td>
<td>83</td>
<td>871</td>
</tr>
<tr>
<td>(20 + 20) – ES</td>
<td>1,122</td>
<td>74</td>
<td>858</td>
</tr>
<tr>
<td>(5 + 30) – ES</td>
<td>187</td>
<td>10</td>
<td>896</td>
</tr>
</tbody>
</table>

Table I. Example 1 – Performance of the two types of ES for different numbers of parents and offspring.

<table>
<thead>
<tr>
<th>Deviation ( \gamma )</th>
<th>Number of FE analyses</th>
<th>Number of generations</th>
<th>Value of the objective function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>114</td>
<td>27</td>
<td>872</td>
</tr>
<tr>
<td>0.5</td>
<td>186</td>
<td>52</td>
<td>869</td>
</tr>
<tr>
<td>0.1</td>
<td>171</td>
<td>45</td>
<td>800</td>
</tr>
<tr>
<td>0.05</td>
<td>276</td>
<td>76</td>
<td>859</td>
</tr>
<tr>
<td>0.01</td>
<td>199</td>
<td>50</td>
<td>877</td>
</tr>
</tbody>
</table>

Table II. Example 1 – The influence of the deviation parameter \( \gamma \), required for the generation of the random vector \( Z(\gamma) \), on the performance of (5 + 5)-ES.

<table>
<thead>
<tr>
<th>Initial objective function ( f_0 )</th>
<th>Number of FE analyses</th>
<th>Number of generations</th>
<th>Value of the objective function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,191</td>
<td>281</td>
<td>87</td>
<td>867</td>
</tr>
<tr>
<td>1,141</td>
<td>136</td>
<td>39</td>
<td>874</td>
</tr>
<tr>
<td>1,078</td>
<td>114</td>
<td>27</td>
<td>872</td>
</tr>
</tbody>
</table>

Table III. Example 1 – The influence of the initial design on the performance of (5 + 5) – ES.

<table>
<thead>
<tr>
<th>Termination criterion</th>
<th>Number of FE analyses</th>
<th>Number of generations</th>
<th>Value of the objective function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>122</td>
<td>28</td>
<td>881</td>
</tr>
<tr>
<td>2</td>
<td>136</td>
<td>34</td>
<td>872</td>
</tr>
<tr>
<td>3</td>
<td>114</td>
<td>27</td>
<td>872</td>
</tr>
<tr>
<td>4</td>
<td>116</td>
<td>28</td>
<td>875</td>
</tr>
</tbody>
</table>

Table IV. Example 1 – Influence of the type of the termination criterion on the performance of (5 + 5) – ES (\( f_0 = 1,078kN \)).
joints and equivalent to uniform load of 4.788 kPa and horizontal forces equivalent to uniform forces of 0.956 kPa on the largest surface. The direct skyline method is used for the solution of the finite element equations. The element members are divided into 11 groups shown in Figure 4 (plan (a)) and the total number of design variables is 22. Table V shows the performance of the two types of the multi-membered ES, namely $(\mu + \lambda) - ES$ and $(\mu, \lambda) - ES$, for different numbers of parents and offsprings. Table VI depicts the influence of the deviation parameter $\gamma$ in the case of $(\mu + \lambda) - ES$, while the influence of the initial design and the type of termination criterion is demonstrated in Tables VII and VIII, respectively.

Example 3

The third example is the 20-storey space frame, analysed by (Papadrakakis and Papadopoulos, 1995) shown in Figure 4 (plan (b)) with 1,020 members and 2,400
degrees of freedom. The loads for this example are taken as in Example 2. The element members are divided into 11 groups shown in Figure 4 (plan (b)) and the total number of design variables is 22. Table IX shows the performance of the two types of the multi-membered ES, namely ($\mu + \lambda$) - ES and ($\mu, \lambda$) - ES, for different numbers of parents and offsprings. Table X depicts the influence of the

<table>
<thead>
<tr>
<th>ES type</th>
<th>Number of FE analyses</th>
<th>Number of generations</th>
<th>Value of the objective function</th>
</tr>
</thead>
<tbody>
<tr>
<td>(5 + 5) - ES</td>
<td>220</td>
<td>58</td>
<td>3.143</td>
</tr>
<tr>
<td>(10 + 10) - ES</td>
<td>581</td>
<td>91</td>
<td>3.064</td>
</tr>
<tr>
<td>(20 + 20) - ES</td>
<td>1,085</td>
<td>83</td>
<td>2.988</td>
</tr>
<tr>
<td>(5 + 30) - ES</td>
<td>461</td>
<td>37</td>
<td>2.987</td>
</tr>
</tbody>
</table>

**Table V.**

Example 2 – Performance of the two types of ES for different numbers of parents and offsprings

<table>
<thead>
<tr>
<th>Deviation $\gamma$</th>
<th>Number of FE analyses</th>
<th>Number of generations</th>
<th>Value of the objective function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>269</td>
<td>20</td>
<td>3.037</td>
</tr>
<tr>
<td>0.5</td>
<td>365</td>
<td>29</td>
<td>3.017</td>
</tr>
<tr>
<td>0.1</td>
<td>461</td>
<td>37</td>
<td>2.987</td>
</tr>
<tr>
<td>0.05</td>
<td>440</td>
<td>32</td>
<td>3.031</td>
</tr>
<tr>
<td>0.01</td>
<td>482</td>
<td>37</td>
<td>3.082</td>
</tr>
</tbody>
</table>

**Table VI.**

Example 2 – The influence of the deviation parameter $\gamma$, required for the generation of the random vector $Z^{(g)}$, on the performance of (5 + 30) – ES

<table>
<thead>
<tr>
<th>Initial objective function $f_0$</th>
<th>Number of FE analyses</th>
<th>Number of generations</th>
<th>Value of the objective function</th>
</tr>
</thead>
<tbody>
<tr>
<td>5,973</td>
<td>485</td>
<td>41</td>
<td>3.087</td>
</tr>
<tr>
<td>5,709</td>
<td>428</td>
<td>35</td>
<td>3.017</td>
</tr>
<tr>
<td>3,620</td>
<td>365</td>
<td>29</td>
<td>3.017</td>
</tr>
</tbody>
</table>

**Table VII.**

Example 2 – the influence of the initial design on the performance of (5 + 30) – ES

<table>
<thead>
<tr>
<th>Termination criterion</th>
<th>Number of FE analyses</th>
<th>Number of generations</th>
<th>Value of the objective function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>330</td>
<td>27</td>
<td>3.010</td>
</tr>
<tr>
<td>2</td>
<td>387</td>
<td>31</td>
<td>3.017</td>
</tr>
<tr>
<td>3</td>
<td>365</td>
<td>29</td>
<td>3.017</td>
</tr>
<tr>
<td>4</td>
<td>382</td>
<td>31</td>
<td>3.106</td>
</tr>
</tbody>
</table>

**Table VIII.**

Example 2 – Influence of the type of the termination criterion on the performance of ES ($f_0 = 3,620kN$)
deviation parameter $\gamma$ in the case of $(\mu + \lambda) - ES$, while the influence of the initial design and the type of termination criterion is demonstrated in Tables XI and XII, respectively. The iteration history of the value of the objective function at each generation and at each FE analysis is shown in Figures 5 and 6, respectively.

Table XIII shows the performance of the proposed solution methods, incorporated into $(10 + 10) - ES$ where the deviation $\gamma = 0.1$ and the termination criterion (2) are adopted. A first-order Neumann expansion is chosen for NCG method. The objective function corresponding to the initial weight of the structure is $f_0 = 8,059$ kN. After 51 generations and 374 finite element analyses
a final design with a weight of $f_{51} = 5,635$ kN and an improvement of 30 per cent is obtained.

In this example the proposed solution schemes are compared to the conventional direct skyline solution (DIR) of the finite element equations at each search step. A modified Cholesky factorization is also implemented in an effort

| Table IX. Example 3 – Performance of the two types of ES for different numbers of parents and offsprings |
|-----------------------------------------------|------------------|------------------|------------------|
| ES type | Number of FE analyses | Number of generations | Value of the objective function |
| (5 + 5) - ES | 316 | 89 | 5,813 |
| (10 + 10) - ES | 554 | 83 | 5,611 |
| (20 + 20) - ES | 1,505 | 108 | 5,623 |
| (5 + 30) - ES | 477 | 32 | 5,657 |
| (5,5) - ES | 346 | 93 | 5,899 |
| (10,10) - ES | 556 | 83 | 5,697 |
| (20,20) - ES | 1,395 | 98 | 5,704 |
| (5,30) - ES | 541 | 41 | 5,696 |

| Table X. Example 3 – The influence of the deviation parameter $\gamma$ required for the generation of the random vector $Z(\gamma)$, on the performance of (10 + 10) – ES |
|-----------------------------------------------|------------------|------------------|------------------|
| Deviation $\gamma$ | Number of FE analyses | Number of generations | Value of the objective function |
| 1.0 | 567 | 81 | 5,628 |
| 0.5 | 694 | 106 | 5,642 |
| 0.1 | 554 | 83 | 5,611 |
| 0.05 | 934 | 149 | 5,548 |
| 0.01 | 912 | 142 | 5,548 |

| Table XI. Example 3 – The influence of the initial design on the performance of (10 + 10) – ES |
|-----------------------------------------------|------------------|------------------|------------------|
| Initial objective function $f_0$ | Number of FE analyses | Number of generations | Value of the objective function |
| 12,685 | 1,566 | 245 | 5,430 |
| 9,758 | 1,105 | 172 | 5,571 |
| 8,059 | 554 | 83 | 5,611 |

| Table XII. Example 3 – Influence of the type of the termination criterion on the performance of (10 + 10) – ES ($f_0 = 8,059$ kN) |
|-----------------------------------------------|------------------|------------------|------------------|
| Termination criterion | Number of FE analyses | Number of generations | Value of the objective function |
| 1 | 265 | 32 | 5,674 |
| 2 | 374 | 51 | 5,635 |
| 3 | 554 | 83 | 5,611 |
| 4 | 396 | 53 | 6,476 |
Advanced solution methods

31

to improve the direct skyline solver. During the factorization phase the alterations to the factorized stiffness matrix are confined to the bottom right-hand corner starting from the first node with a change in the stiffness matrix. Consequently the time-consuming factorization part need not be repeated but only the steps after the smallest degree of freedom which is affected by the change of stiffness matrix and downwards. Thus, the stiffness matrix is not refactorized at each step but is partially refactorized starting with the least degree of freedom affected by the change of the cross-section of a member. This technique is referred to as a modified complete factorization (DIR-M).

A modified implementation of PCG and NCG solution schemes is also tested. This consists in updating the preconditioning matrix either when the number of PCG and NCG iterations exceed ten and five, inside a finite element analysis respectively, or when there is no significant improvement of the objective function. The number ten, corresponds to the ratio of the required factorization time over one forward and backward substitution for this example. Thus, a refactorization is performed every time the number of PCG or NCG iterations becomes greater than ten or five, respectively. The number five corresponds to the maximum allowable NCG iterations since a first-order Neumann expansion is used.

Table XIV depicts the performance of the methods when the values of the parents design variables of the first generation are poor giving an initial weight of 12,685kN compared to the improved design which is achieved after 195 generations.

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of iterations</th>
<th>Factorization time(s)</th>
<th>Substitutions time(s)</th>
<th>Total optimization times(s)</th>
<th>Storage (Mbytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIR</td>
<td>-</td>
<td>2,784</td>
<td>95</td>
<td>5,954</td>
<td>2,500</td>
</tr>
<tr>
<td>DIR-M</td>
<td>-</td>
<td>2,773</td>
<td>95</td>
<td>5,892</td>
<td>2,500</td>
</tr>
<tr>
<td>PCG</td>
<td>3,064</td>
<td>7</td>
<td>662</td>
<td>3,276</td>
<td>1,370</td>
</tr>
<tr>
<td>NCG</td>
<td>1,615</td>
<td>7</td>
<td>1,063</td>
<td>3,518</td>
<td>1,410</td>
</tr>
<tr>
<td>PCG1</td>
<td>1,623</td>
<td>57</td>
<td>319</td>
<td>2,921</td>
<td>1,410</td>
</tr>
<tr>
<td>NCG1</td>
<td>852</td>
<td>51</td>
<td>537</td>
<td>3,107</td>
<td>1,410</td>
</tr>
</tbody>
</table>

Table XIV. Example 3 – Sequential performance of solution methods after 51 generations ((10 + 10) – ES, γ = 0.1, termination criterion (2) 374 FE analyses, $f_0 = 8,059kN$, $f_{51} = 5,635kN$)

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of iterations</th>
<th>Factorization time(s)</th>
<th>Substitutions time(s)</th>
<th>Total optimization times(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIR</td>
<td>-</td>
<td>9,619</td>
<td>326</td>
<td>20,568</td>
</tr>
<tr>
<td>DIR-M</td>
<td>-</td>
<td>9,538</td>
<td>331</td>
<td>20,329</td>
</tr>
<tr>
<td>PCG</td>
<td>22,456</td>
<td>7</td>
<td>15,394</td>
<td>15,494</td>
</tr>
<tr>
<td>NCG</td>
<td>13,352</td>
<td>7</td>
<td>9,490</td>
<td>18,913</td>
</tr>
<tr>
<td>PCG1</td>
<td>5,445</td>
<td>130</td>
<td>1,092</td>
<td>10,965</td>
</tr>
<tr>
<td>NCG1</td>
<td>3,238</td>
<td>117</td>
<td>1,740</td>
<td>13,743</td>
</tr>
</tbody>
</table>

Table XIII. Example 3 – Sequential performance of solution methods after 195 generations ((10 + 10) – ES, γ = 0.1, termination criterion (2) 1,297 FE analyses, $f_0 = 12,685kN$, $f_{195} = 5,430kN$)
generations and 1,297 finite element analyses. The achieved weight by ES is 5,430kN (57 per cent improvement).

The performance of the parallel implementation of the solution schemes is demonstrated in Tables XV and XVI, where close and crude initial values of parents design variables are selected respectively. The numbers after the abbreviated names of the methods correspond to the number of processors used. The maximum number of processors is affected by the storage requirements of each method. In all parallel tests a \((p + p)\) – ES is adopted, where \(p\) is the number of processors employed. In this case, the updating of the preconditioning matrix in the modified implementation of PCG and NCG solution schemes is performed when the factorization criteria are satisfied in more than 80 per cent of the processors.

### Table XV.
Example 3 – Parallel performance of solution methods \((f_0 = 8,059kN)\)

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of FE analyses</th>
<th>Number of generations</th>
<th>Value of the objective function (kN)</th>
<th>Number of iterations</th>
<th>Total sequential time(s)</th>
<th>Total parallel time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIR-6</td>
<td>277</td>
<td>63</td>
<td>5,725</td>
<td>–</td>
<td>4,598</td>
<td>1,077</td>
</tr>
<tr>
<td>DIR-M-6</td>
<td>277</td>
<td>63</td>
<td>5,725</td>
<td>909</td>
<td>4,469</td>
<td>1,053</td>
</tr>
<tr>
<td>PCG1-6</td>
<td>277</td>
<td>63</td>
<td>5,725</td>
<td>2,204</td>
<td>2,752</td>
<td>739</td>
</tr>
<tr>
<td>PDG-9</td>
<td>372</td>
<td>56</td>
<td>5,609</td>
<td>1,824</td>
<td>4,036</td>
<td>674</td>
</tr>
<tr>
<td>NCG-9</td>
<td>372</td>
<td>56</td>
<td>5,609</td>
<td>1,259</td>
<td>5,662</td>
<td>944</td>
</tr>
<tr>
<td>PCG1-9</td>
<td>372</td>
<td>56</td>
<td>5,609</td>
<td>621</td>
<td>3,784</td>
<td>630</td>
</tr>
</tbody>
</table>

### Table XVI.
Example 3 – Parallel performance of solution methods \((f_0 = 12,685kN)\)

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of FE analyses</th>
<th>Number of generations</th>
<th>Value of the objective function (kN)</th>
<th>Number of iterations</th>
<th>Total sequential time(s)</th>
<th>Total parallel time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIR-6</td>
<td>850</td>
<td>229</td>
<td>5,547</td>
<td>–</td>
<td>14,081</td>
<td>3,810</td>
</tr>
<tr>
<td>DIR-M-6</td>
<td>850</td>
<td>229</td>
<td>5,547</td>
<td>–</td>
<td>13,912</td>
<td>3,749</td>
</tr>
<tr>
<td>PCG1-6</td>
<td>850</td>
<td>229</td>
<td>5,547</td>
<td>2,761</td>
<td>8,387</td>
<td>2,435</td>
</tr>
<tr>
<td>PDG-9</td>
<td>928</td>
<td>152</td>
<td>5,435</td>
<td>16,921</td>
<td>10,511</td>
<td>1,841</td>
</tr>
<tr>
<td>NCG-9</td>
<td>928</td>
<td>152</td>
<td>5,435</td>
<td>10,127</td>
<td>13,716</td>
<td>2,406</td>
</tr>
<tr>
<td>PCG1-9</td>
<td>928</td>
<td>152</td>
<td>5,435</td>
<td>2,835</td>
<td>9,428</td>
<td>1,654</td>
</tr>
<tr>
<td>NCG1-9</td>
<td>928</td>
<td>152</td>
<td>5,435</td>
<td>2,255</td>
<td>10,231</td>
<td>1,798</td>
</tr>
</tbody>
</table>

### Conclusions
The two types of the multi-membered ESs discussed in this work, namely \((\mu + \lambda)\) - ES and \((\mu,\lambda)\) - ES, appear not to have a significant influence on the improved design achieved by ES optimization procedure. There is only a slight superiority of the \((\mu + \lambda)\) - ES over the \((\mu,\lambda)\) - ES in all cases considered. The
computational efficiency, however, is strongly affected by the number of parents and offspring. Large numbers of parents and offspring produce a computational overhead without a significant improvement on the values of the objective function.

For the test examples performed in this study the influence of the deviation parameter $\gamma$ for the generation of the random vector $Z^{(g)}$ over the final design achieved appears to be significant. Values of $\gamma$ close to one required the least number of finite element analyses giving a satisfactory global optimum. The improved design produced by the ES, on the other hand, is not affected substantially by the initial design point. This is not the case for the computational effort, which is strongly affected by the starting values of the design parameters. The four termination criteria tested gave similar results both in terms of the final design achieved and the required computational effort.

The modified factorization procedure, in which alterations to the factorized matrix are confined to the right hand bottom, has only a marginal influence on the performance of the direct method owing to the random selection of the design variables at each search step.

The improvement of the quality of the preconditioning matrix with a first order Neumann series expansion of the inverse of the stiffness matrix, although it results on a reduction of approximately 45 per cent in the number of iterations compared to a PCG version, it requires slightly more computing time owing to one additional matrix-vector multiplication and forward-backward substitution required by NCG. This superiority of the PCG version becomes more pronounced in the parallel implementation of the methods.

The modified versions of PCG and NCG solution schemes, where the preconditioning matrix is not kept fixed throughout the evolution procedure, has beneficial effects on the performance of the methods, resulting in up to 30 per cent reduction in the total computing time.

Finally, both storage requirements and computing time with PCG and NCG methods compared to the direct skyline routine are reduced almost by half in sequential as well as in parallel implementations. The improvement achieved in the total solution time with the parallel implementation of the methods is reduced by a factor of four in the direct skyline solver, using six processors, and by a factor of six in PCG method using nine processors.

References


Holland, J. (1975), Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI.


