Harnpornchai Napat
College of Arts, Media, and Technology, Chiang Mai University, Chiang Mai, Thailand

Genetic algorithms-aided reliability analysis

Keywords

genetic algorithms, reliability analysis, simulation methods, complex systems, multiple failure modes

Abstract

A hybrid procedure of Genetic Algorithms (GAs) and reliability analysis is described, discussed, and summarized. The procedure is specifically referred to as a Genetic Algorithms-aided (GAs-aided) reliability analysis. Two classes of GAs, namely simple GAs and multimodal GAs, are introduced to solve a number of important problems in reliability analysis. The problems cover the determination of Point of Maximum Likelihood in failure domain (PML), the computation of failure probability using the GAs-determined PML, and the determination of multiple design points. The MCS-based method using the GAs-determined PML is specifically implemented in the so-called an Importance Sampling around PML (ISPML). The application of GAs to each respective problem is then demonstrated via numerical examples in order to clarify the procedures. With an aid from GAs, reliability analysis is possible even if there is no information about the geometry or landscape of limit state surfaces and the total number of crucial likelihood points. In addition, GAs significantly improve the computational efficiency and realize the analysis of rare events under constrained computational resources. The implementation of GAs to reliability analysis for building up the hybrid procedure is readily because of their algorithmic simplicity.

1. Introduction

Genetic Algorithms (GAs) are global search techniques that are based on evolutionary theory in biological sciences (see e.g. [4], [7], [9], [11],[15]). GAs have been employed in several engineering disciplines to obtain optimal solutions or optimal designs [7]. The application of GAs in context of reliability engineering is directed towards reliability-based optimization problems [7]. A most recent state-of-the-art survey [8] also shows only the application of GAs to reliability-based design optimization. While the design optimization is considered a conventional application for GAs, there is another potential application. The prospective application is relevant to reliability analysis. The application of GAs to reliability analysis appears to gain less interest and attention compared to the application to reliability-based design optimization. The purpose of this paper is to describe, discuss, and summarize the application of GAs to reliability analysis.

The procedure combines GAs with reliability analysis procedure, thus forming a hybrid procedure. The analysis which is based on the hybrid procedure will be hereinafter referred to as a Genetic Algorithms-aided (GAs-aided) reliability analysis. The application of GAs to reliability analysis is aimed at obtaining crucial information needed from the analysis. The crucial information includes Point of Maximum Likelihood in failure domain (PML) and failure probability for a given system or element. Another important application of GAs in context of reliability analysis is the determination of multiple design points in multiple failure modes.

The structure of the paper starts with the generic form of the optimization problems in reliability analysis. The next main sections contain the theoretical background of GAs which presents the description of both simple GAs and multimodal GAs, respectively. The application of GAs to each respective problem is then demonstrated via numerical examples in order to clarify the procedures. The simple GAs are used in the determination of PML. The GAs-determined PML is further employed in an MCS-based method which is specifically referred to as an Importance Sampling around PML (ISPML). The multimodal GAs are

employed in the determination of multiple design points. The crucial aspects of the paper will then be summarized at the end.

2. Generic form of optimization problems in reliability analysis

Optimization problems that appear in context of relia bility analysis include constrained and unconstrained optimization problems. Optimization problems gene rally aim at maximization or minimization of objecti ve functions. The constrained optimization problem for maximizing an objective function is expressed as

$$\text{Maximize} \quad O_1(\boldsymbol{x}) \tag{1}$$

$$\text{Subject to} \quad g_1(\boldsymbol{x}) \le 0 \tag{2.1}$$

$$\dots$$

$$g_j(\boldsymbol{x}) \le 0 \tag{2.j}$$

$$\dots$$

$$g_{NC}(\boldsymbol{x}) \le 0 \tag{2.NC}$$

, where $O_1(\boldsymbol{x})$ is the objective function of $\boldsymbol{x} = [x_1 \ \dots \ x_k \ \dots \ x_N]^T$. x_k is the kth design variable. N is total number of design variables. $g_j(\boldsymbol{x})$ is the jth constraint. NC is total number of constraints. The constrained maximization problem is found in the determination of PML and multiple design points.

3. Simple GAs

3.1. General on GAs

GAs are a stochastic search technique based on the mechanism of natural selection. It combines Darwin's principle of survival of the fittest and a structured information exchange using randomized operators to evolve an efficient search mechanism [9]. GAs have been utilized to successfully solve various optimization problems in which the optimal solutions are searched and determined by GAs [see e.g. [7], [4]. Major virtues of GAs are as follows, among others [9], [15], [7]. First, GAs are a population-based search and use probabilistic transition rules to direct the evolution of the search. In other words, GAs make a remarkable balance between the exploitation of the best solution and the exploration of the search space. The population-to-population approach and the probabilistic transition rules attempt to make the search escape from local optima. Correspondingly, the possibility of being trapped in local optima when searching for the design point can be reduced using GAs. Second, GAs use only the information of objective function, not the function derivatives or other auxiliary knowledge. The required information is the numerical value of the objective function. Third, GAs do not impose much mathematical requirement about the optimization problems. Yet, the algorithms are simple and readily to be implemented. Accordingly, GAs are robust and thus applicable whenever the numerical value of the objective function can be determined. The second and third virtues make GAs attractive to reliability analysis of complex systems where the associated Limit State Functions (LSFs) can be implicit, nonlinear, non-differentiable, and noisy. Those types of LSFs are thus characterized by numerical values only.

GAs procedure starts with an initial set of randomly selected trial solutions, namely population. Each individual in the population is encrypted and referred to as a chromosome which represents a possible solution to the optimization problem. The chromosomes evolve through successive iterations, called generations. In each generation, the fitness of each chromosome is evaluated. The fitness of each chromosome reflects the potential to be the optimal solution. Each chromosome is reproduced according to its fitness value. Fitter chromosomes have higher probabilities to be selected for reproduction whereas weaker chromosomes tend to die off. The chromosome selection and reproduction are carried out in a reproduction process. The chromosomes resulting from the reproduction process form a mating pool and are collectively referred to as offspring. The offspring are later undergone genetic operations. The exploration of search space is carried out through the genetic operations where genetic operators are applied to existing chromosomes and transform them into new chromosomes. The genetic operators-derived chromosomes represent new trial solutions in the search space. The resulting chromosomes then form the new generation of population. It should be noted that GAs work in two spaces alternatively. The selection process is performed in the space of original variables while the genetic operations are done in the space of coded variables. Both spaces are referred to as the solution and coding space, respectively [7]. The GAs search is terminated when a prescribed number of generations have elapsed. The procedure of GAs is summarized in *Figure 1*.

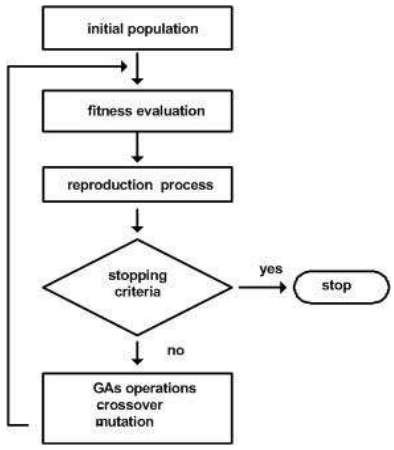


Figure 1. GAs search procedure [4], [7], [9], [15].

3.2. Chromosome representation

GAs encrypt each trial solution into a sequence of numbers or strings and denote the sequences as a chromosome. In this paper, a simple binary coding for real values as proposed by [15] is employed. According to the utilized coding scheme, each variable x_j in solution space is represented by a binary string as shown in *Figure 2*. The combination of these strings forms a chromosome in coding space. The evaluation of chromosome fitness is done in the solution space of x_j while the genetic operations are performed in the coding space of chromosome. The binary coding for real values will be briefly explained here. More details can be found in [15]. In context of GAs-aided reliability analysis, each variable value is corresponding to a realization of a random variable x_j. According to the binary coding for real values, the length of the binary strings depends on the required precision. When the domain of variable x_j is bounded by lower boundary lb_j and the upper boundary ub_j, and the required precision needs ζ_j places after the decimal point, the range of the domain of each variable should be divided into at least $(ub_j - lb_j)\times 10^{\zeta_j}$ size ranges. The required bits l_j for the variable is then obtained from

$$2^{l_j-1} < (ub_j - lb_j)\times 10^{\zeta_j} \le 2^{l_j} \qquad (3)$$

The encoding, i.e. from a real number to a binary string, and the decoding follow the relation

$$x_j = lb_j + decimal(substring_j)\times \frac{ub_j - lb_j}{2^{l_j}-1} \qquad (4)$$

, where *decimal* ($substring_j$) represents the decimal value of $substring_j$ for variable x_j in the solution space. The decimal value is also referred to as the decimal number. The obtained decimal number is then transformed into the binary number.

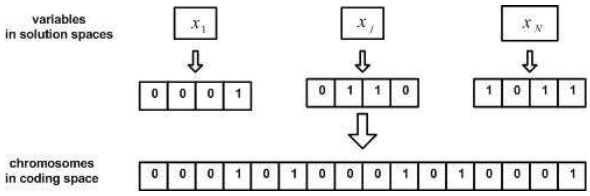


Figure 2. Chromosome representation using binary coding for real values [15].

As an example, the design variables are x_1 and x_2, both of which have the same domain boundaries [-1,1]. Suppose that the desired precision is three decimal places for each variable. Therefore, the required number of bits for each variable is 11 and the total length of the binary string is thus 22 bits. The decoding of a binary-coded chromosome according to this example is illustrated in *Table 1* and *Figure 3*, respectively. η_k is a binary string representing the *k*th chromosome.

Table 1. Binary numbers and their corresponding decimal numbers.

Variable	Binary Number	Decimal Number
x_1	11110001001	1929
x_2	01001101110	622

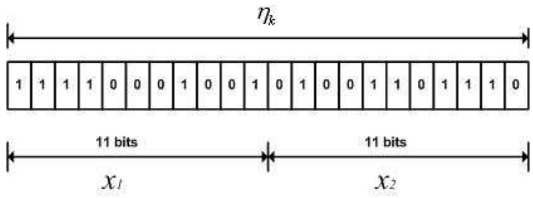


Figure 3. A binary-coded chromosome for real values.

3.3. Reproduction process

Reproduction in GAs is a process in which individual chromosomes are copied according to their fitness values. Copying chromosomes according to their fitness values implies that a chromosome with higher fitness value has a higher probability of contributing one or more offspring in the next generation. This operation imitates the survival of the fittest or the natural selection as used by Darwin in [3]. Fitness in natural population is determined by the ability of a creature to survive predators, pestilence, and the other obstacles to adulthood and subsequent reproduction. Fitness in an optimization by GAs is defined by a fitness function. Based on the optimization problem as described by Eq. (1) and the

set of constraints (2), the fitness function $F(\boldsymbol{x})$ of a chromosome representing a vector $\boldsymbol{x}$ of variables in the solution space is defined as

$$F(x)=\begin{cases} O_1(\boldsymbol{x}) & ;\boldsymbol{x}\text{ is feasible} \\ O_1(\boldsymbol{x})-\sum_{j=1}^{NC}k_j v_j(\boldsymbol{x}) & ;\boldsymbol{x}\text{ is infeasible} \end{cases} \quad (5)$$

An adaptive penalty scheme which is introduced by [1] and improved by [17] will be employed to handle the constraints. The improved adaptive penalty scheme shows its excellent capability in handling a very large number of constraints [10]. This adaptive scheme is given by

$$k_j=\left|max\left(O_1^{\inf}(\boldsymbol{x})\right)\right|\frac{<v_j(\boldsymbol{x})>}{\sum_{l=1}^{NC}[<v_l(\boldsymbol{x})>]^2} \quad (6)$$

, where $max(O_1^{\inf}(\boldsymbol{x}))$ is the maximum of the objective function values in the current population in the infeasible region, $v_j(\boldsymbol{x})$ is the violation magnitude of the jth constraint. $<v_j(\boldsymbol{x})>$ is the average of $v_j(\boldsymbol{x})$ over the current population. k_j is the penalty parameter for the jth constraint defined at each generation. The violation magnitude is defined as

$$v_l(\boldsymbol{x})=\begin{cases} |g_l(\boldsymbol{x})| & ;g_l(\boldsymbol{x})>0 \\ 0 & ;\text{otherwise} \end{cases} \quad (7)$$

The reproduction operator may be implemented in a number of ways. The easiest and well-known approach is the roulette-wheel selection (see e.g.[4], [9]). According to the roulette-wheel scheme, the jth chromosome will be reproduced with the probability of

$$P_k=\frac{F_k}{\sum_{l=1}^{NPop}F_l} \quad (8)$$

, in which N_{Pop} is the population or sample size. The fitness value F_k is obtained from Eq. (5). On passing, it should be noted that GAs utilize only the numerical values of the objective function and of its associated constraints for the evaluation of the chromosome fitness (confer Eqs. (5) to (7)). This advantageous feature makes GAs readily applicable to real-world problems where the LSFs are generally implicit with respect to random variables.

3.4 Genetic operators

Cross-over and mutation are genetic operators. Crossover operates on two chromosomes at a time and results in two new chromosomes. A simple binary crossover with two cut point is illustrated in *Figure 4*.

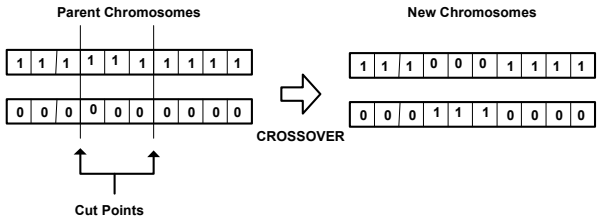


Figure 4. Crossover of two chromosomes.

Mutation operates on a chromosome by mutate a gene in a chromosome to produce new chromosome. A binary mutation converts the randomly selected genes from 0 to 1 or from 1 to 0 (see *Figure 5*).

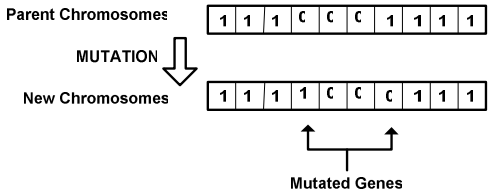


Figure 5. Mutation on a chromosome.

4. Multimodal GAs

4.1. General

Simple GAs perform well in locating a single optimum but face difficulties when requiring multiple optima [5], [13], [14], [16], [18]. Even there exist multiple optima in the search space, the simple GAs will converge to a single optimal point. This is the result of genetic drift, which is the tendency of the GAs to converge over time to one optimal point within the search space [16]. The term genetic drift explains the effect of a loss of population diversity that occurs due to the stochastic nature of selection in a finite population [12]. As to avoid the solution population to converge to a single optimal point, mechanisms of diversification have been proposed to force GAs to maintain a diverse population [5], [13], [14]. Niching methods extend the simple GAs to maintain the population diversity and provide the stability of subpopulations in the vicinity of optimal solutions in a multimodal domain [5], [13], [14]. Niching method thus can identify the multiple solutions with certain extent of diversity [16]. Among niching methods, Standard Crowding Genetic Algorithms (SCGAs) [5] and Deterministic Crowding Genetic Algorithms (DCGAs) [13], [14]

have been commonly used in multimodal functions optimization. These two methods will be used as tools for locating multiple design points herein. It should be noted that both SCGAs and DCGAs, however, are originally designed for unconstrained optimization problems. To handle the constraint (2), the adaptive penalty described in the previous section will be used in both SCGAs and DCGAs.

4.2. Standard crowding genetic algorithms (SCGAs)

SCGAs was proposed by De Jong [5]. The intention of the methodology is to preserve diversity and slow down convergence on multimodal functions, specifically Shekels Foxholes multimodal function. Premature convergence is reduced in SCGAs by minimizing changes in the overall population distribution between generations [16]. According to SCGAs, the procedure revises the population by replacing similar parent. The replacement for each offspring produced is considered individually. For each such individual, a sample of crowding factor (*CF*) individuals are randomly drawn from the parent population and searched for the most similar bit-string to the offspring in question. Similarity is measured as the number of point differences between the equal length bit-strings, called the Hamming distance. The most similar individual from the small sample, i.e. from *CF*, is then directly replaced in the population by the offspring, without regard for fitness [2]. The following provides the pseudo code of SCGAS [2].

G : Generational gap; ratio of the reproduced population in each generation.
NPop : Population size.
CF : Crowding Factor; the size of sample taken from population and searched for the most similar.

1. Randomly initialize population.
2. Evaluate fitness of the population.
3. Loop until stop condition:
   a. Select population set of size $G\times NPop$ by fitness proportion.
   b. Crossover to generate $G\times NPop$ offspring.
   c. Evaluate fitness of offspring.
   d. Loop for each offspring:
      i. Randomly select sample size of *CF* from the parent population.
      ii. Search for most similar in sample in comparison with the offspring.
      iii. Replace most similar in population with offspring irrespective of fitness.

4.2. Deterministic crowding niche genetic algorithms (DCGAS)

Mahfoud [13], [14] proposed a simple multimodal GAs and is known as Deterministic Crowding Niche Genetic Algorithms (DCGAS). DCGAS work as follows. First all population elements are grouped into *N/2* pairs, where *N* is number of population. The crossover and mutation are the applied to all pairs. Each offspring competes against one of the parents that produced it. For each pair of offspring, two sets of parent-child tournaments are possible. DCGAS hold the set of tournaments that forces the most similar elements to compete. Like in sharing, similarity can be measured using either genotype or phenotype distance. The DCGAS is indeed a special case of SCGAS where the crowding factor, *CF*, equals to 2. They were developed to improve De Jong's basic crowding scheme. The following provides a pseudo code of DCGAS [2].

NPop : Population size.
$d(x, y)$: Distance between individuals x and y.
$F(x)$: Fitness of individual population member.

1. Randomly initialize population.
2. Evaluate fitness of population.
3. Loop until stop condition:
   a. Shuffle the population.
   b. Crossover to produce *NPop*/2 pairs of offspring.
   c. Apply mutation (optional).
   d. Loop for each pair of offspring:
      i. If
      (d(parent1,child1)+d(parent2,child2)) $\leq$ (d(parent2,child1)+d(parent1,child2)).
         1. If F(child1) > F(parent1), child1 replaces parent1.
         2. If F(child2) > F(parent2), child2 replaces parent2.
      ii. Else
         1. If F(child1) > F(parent2), child1 replaces parent2.
         2. If F(child2) > F(parent1), child2 replaces parent1.

Instead of using De Jong's crowding factor, DCGAS method compares the new offspring directly to their parents. The parents are replaced only if the children have higher fitness [2].

5. Applications of algorithms to reliability analysis

5.1. Determination of PML

5.1.1. Problem formulation

Since PML is the point of highest probability density function in the failure domain, the PML $\boldsymbol{x}^*$ can be obtained from solving the following optimization problem:

$$\text{Maximize} \quad O_3(\boldsymbol{x}) = f_{\boldsymbol{X}}(\boldsymbol{x}) \tag{9}$$

$$\text{Subject to} \quad g_1(\boldsymbol{x}) \le 0 \tag{10.1}$$

$$\cdots$$

$$g_j(\boldsymbol{x}) \le 0 \tag{10.j}$$

$$\cdots$$

$$g_{NC}(\boldsymbol{x}) \le 0 \tag{10.NC}$$

in which $f_{\boldsymbol{X}}(\boldsymbol{x})$ is the Joint Probability Density Function (JPDF) of $\boldsymbol{X} = [X_1 \ldots X_N]^{\mathrm{T}}$ and $g_j(\boldsymbol{x})$ (j = 1, …, NC) is the jth LSF. N and NC are the total number of basic random variables and the total number of limit state functions, respectively. The corresponding fitness function is

$$F(x) = \begin{cases} O_3(\boldsymbol{x}) & ; \boldsymbol{x} \text{ is feasible} \\ O_3(\boldsymbol{x}) - \sum_{j=1}^{NC} k_j v_j(\boldsymbol{x}) & ; \boldsymbol{x} \text{ is infeasible} \end{cases} \tag{11}$$

, where k_j and v_j(x) are defined as in Eqs. (6) and (7), respectively.

5.1.2. Numerical example 1: a plate with an edge crack [19]

Consider a plate with an edge crack. When the cracked plate is loaded in combined tensile and bending, the total stress intensity factor $K_{\text{I–total}}$ is given by:

$$\begin{aligned} K_{\text{I-total}} &= K_{\text{I-tension}} + K_{\text{I-bending}} \\ &= \left[S_{\text{t}} F(A/t)_{\text{tension}} + S_{\text{b}} F(A/t)_{\text{bending}} \right] \sqrt{\pi A} \end{aligned} \tag{12}$$

where $K_{\text{I–tension}}$ is the tensile stress intensity factor and $K_{\text{I-bending}}$ is the bending stress intensity factor. S_{t} is the tensile stress, S_{b} is the outer-fiber bending stress, A is the crack depth and t is the plate thickness. A, S_{t}, and S_{b} are statistically independent random variables. The crack depth A is modelled by the exponential random variable whose PDF is

$$f_A(a) = \frac{1}{\mu_A} exp\left(-\frac{a}{\mu_A} \right) \tag{13}$$

, where the mean crack depth μ_A is 6 mm. The thickness of the plate is constant and deterministic. The thickness in this example is set equal to 100 mm. The tensile stress S_{t} and the bending stress S_{b} are modelled by normal random variables. Accordingly, the PDF of S_{t} is

$$f_{S\text{t}}(s_{\text{t}}) = \frac{1}{\sqrt{2\pi}\sigma_{S\text{t}}} exp\left(-\frac{(s_{\text{t}} - \mu_{S\text{t}})^2}{2\sigma_{S\text{t}}^2} \right) \tag{14}$$

where $\mu_{S\text{t}}$ and $\sigma_{S\text{t}}$ is the mean and standard deviation of S_{t}, respectively. Similarly, the PDF of S_{b} is

$$f_{S\text{b}}(s_{\text{b}}) = \frac{1}{\sqrt{2\pi}\sigma_{S\text{b}}} exp\left(-\frac{(s_{\text{b}} - \mu_{S\text{b}})^2}{2\sigma_{S\text{b}}^2} \right) \tag{15}$$

in which $\mu_{S\text{b}}$ and $\sigma_{S\text{b}}$ is the mean and standard deviation of S_{b}, respectively. The mean tensile stress $\mu_{S\text{t}}$ is 20 MPa and the associated Coefficient of Variation (COV) is 0.10. The mean bending stress $\mu_{S\text{b}}$ is 10 MPa and the corresponding COV is 0.20.
The fracture toughness K_{Ic} is modelled by three-parameter Weibull random variable with the Cumulative Distribution Function (CDF) [20].

$$F_{K\text{Ic}}(k_{\text{Ic}}) = 1 - exp\left[-\left(\frac{k_{\text{Ic}} - k_{\min}}{k_{\text{o}} - k_{\min}} \right)^b \right] \tag{16}$$

, where $F_{K\text{Ic}}(k_{\text{Ic}})$ is the cumulative distribution function of fracture toughness, $k_{\min}$ is the location parameter, k_{o} is the scale parameter, and b is the shape parameter. Mean toughness in terms of the Weibull distribution parameters is

$$\mu_{\text{KIc}} = k_{\min} + (k_{\text{o}} - k_{\min}) \Gamma\left(1 + \frac{1}{b} \right) \tag{17}$$

in which $\Gamma(.)$ is the gamma function. Standard deviation is then equal to

$$\sigma_{K\text{Ic}} = \frac{\mu_{K\text{Ic}} - k_{\min}}{\Gamma\left(1 + \frac{1}{b} \right)} \sqrt{\Gamma\left(1 + \frac{2}{b} \right) - \left[\Gamma\left(1 + \frac{1}{b} \right) \right]^2} \tag{18}$$

The mean toughness $\mu_{K\text{Ic}}$ is 200 $\text{MPa}\sqrt{\text{m}}$. b = 4 and $k_{\min}$ = 20 $\text{MPa}\sqrt{\text{m}}$. The corresponding PDF of K_{Ic} is

$$f_{K\mathrm{Ic}}(k_{\mathrm{Ic}}) = \frac{b}{k_{\mathrm{o}} - k_{\min}} \left(\frac{k_{\mathrm{Ic}} - k_{\min}}{k_{\mathrm{o}} - k_{\min}} \right)^{b-1} e^{-\left(\frac{k_{\mathrm{Ic}} - k_{\min}}{k_{\mathrm{o}} - k_{\min}} \right)^{b}} \quad (19)$$

All random variables including the associated mean and COV values are summarized in *Table 2*. The original JPDF is thus

$$f_{\boldsymbol{X}1}(\boldsymbol{x}_1) = f_A(a) f_{\mathrm{St}}(s_{\mathrm{t}}) f_{\mathrm{Sb}}(s_{\mathrm{b}}) f_{K\mathrm{Ic}}(k_{\mathrm{Ic}}) \quad (20)$$

, where $\boldsymbol{X}_1 = [A \quad S_{\mathrm{t}} \quad S_{\mathrm{b}} \quad K_{\mathrm{Ic}}]^T$ is the vector of all random variables.

Table 2. Description of random variables in Example 1.

Random Variable	Distribution Type	Mean	COV
A	Exponential	$6.00\text{x}10^{-3}$ m	1
S_{t}	Normal	20 MPa	0.10
S_{b}	Normal	10 MPa	0.20
K_{Ic}	3-parameter Weibull	200 $\mathrm{MPa}\sqrt{\mathrm{m}}$	Eqs. (18) and (19)

Tada [21] gives several formulas for $F(A/t)$. The following formulas are used in this example and they are applicable for any A/t.

$$F(A/t)_{\mathrm{tension}} = \frac{0.752 + 2.02(A/t) + 0.37\left[1 - \sin\left(\frac{\pi A}{2t}\right)\right]^3}{\cos\left(\frac{\pi A}{2t}\right)} \times \sqrt{\frac{2t}{\pi A}\tan\left(\frac{\pi A}{2t}\right)} \quad (21.1)$$

$$F(A/t)_{\mathrm{bending}} = \frac{0.923 + 0.199\left[1 - \sin\left(\frac{\pi A}{2t}\right)\right]^4}{\cos\left(\frac{\pi A}{2t}\right)} \times \sqrt{\frac{2t}{\pi A}\tan\left(\frac{\pi A}{2t}\right)} \quad (21.2)$$

The LSF is defined as

$$g(K_{\mathrm{I-total}}, K_{\mathrm{Ic}}) = K_{\mathrm{Ic}} - K_{\mathrm{I-total}} \quad (22.1)$$

or $$g(A, S_{\mathrm{t}}, S_{\mathrm{b}}, K_{\mathrm{Ic}}) = K_{\mathrm{Ic}} - K_{\mathrm{I-total}}(A, S_{\mathrm{t}}, S_{\mathrm{b}}) \quad (22.2)$$

in which K_{Ic} is the fracture toughness.
GAs have been applied to determine PML first. The objective function according to Eq. (9), is

$$O(a, s_{\mathrm{t}}, s_{\mathrm{b}}, k_{\mathrm{Ic}}) = f_A(a) f_{\mathrm{St}}(s_{\mathrm{t}}) f_{\mathrm{Sb}}(s_{\mathrm{b}}) f_{K\mathrm{Ic}}(k_{\mathrm{Ic}}) \quad (23)$$

The magnitude of the constraint violation, according to Eqs. (7) and (22), is

$$v(a, s_{\mathrm{t}}, s_{\mathrm{b}}, k_{\mathrm{Ic}}) = \begin{cases} \left| k_{\mathrm{Ic}} - k_{\mathrm{I-total}}(a, s_{\mathrm{t}}, s_{\mathrm{b}}) \right| & ; g(a, s_{\mathrm{t}}, s_{\mathrm{b}}, k_{\mathrm{Ic}}) > 0 \\ 0 & ; \text{otherwise} \end{cases} \quad (24)$$

It is noted from Eq. (12) that $K_{\text{I-total}}$ is the function of A, S_{t}, and S_{b}. The fitness function is thus

$$F(a, s_{\mathrm{t}}, s_{\mathrm{b}}, k_{\mathrm{Ic}}) = \begin{cases} O(a, s_{\mathrm{t}}, s_{\mathrm{b}}, k_{\mathrm{Ic}}) & ; g(a, s_{\mathrm{t}}, s_{\mathrm{b}}, k_{\mathrm{Ic}}) \le 0 \\ O(a, s_{\mathrm{t}}, s_{\mathrm{b}}, k_{\mathrm{Ic}}) - kv(a, s_{\mathrm{t}}, s_{\mathrm{b}}, k_{\mathrm{Ic}}) & ; g(a, s_{\mathrm{t}}, s_{\mathrm{b}}, k_{\mathrm{Ic}}) > 0 \end{cases} \quad (25)$$

Table 3: PML in Example 1.

PML	Magnitude at PML
a^*	$84.4\text{x}10^{-3}$ m
s_{t}^*	20.0 MPa
s_{b}^*	10.0 MPa
k_{Ic}^*	205 $\mathrm{MPa}\sqrt{\mathrm{m}}$

GAs search employs the population size of 100. The number of generations used in the search is 100. A two-point crossover is utilized with the crossover rate of 0.8. The mutation rate is taken as 0.002. *Figure 6* shows the history of the average fitness of the feasible chromosomes. The resulting PML is shown in *Table 3*.Although the LSFs in the example is explicit, the numeric -based feature of GAs naturally enable the algorithms applicable to the case of implicit LSFs.
With respect to this feature, the approximation of implicit LSFs is not necessary. The error from such an approximation is thus not encountered.Based on the introduced concept and implementation, GAs is readily applicable to higher dimensional problems. Due to the algorithmic simplicity, the implementation of GAs does not require any additional effort.

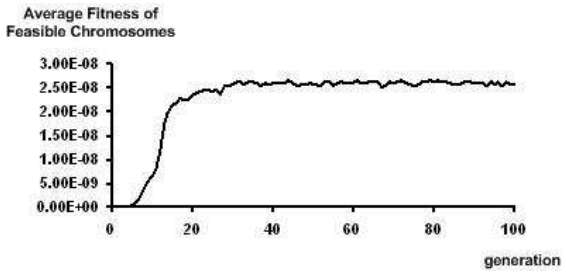


Figure 6. The history of the average fitness of the feasible chromosomes.

5.2. GAs-aided importance sampling

5.2.1. Principles

The probability p_F of a failure event F is obtained from

$$p_F = \int_{D_F} f_X(\boldsymbol{x})d\boldsymbol{x} \tag{26}$$

, where D_F is the subspace corresponding to the failure event F in a multidimensional space of $X_1,\ldots, X_N$ and will be referred to as the event or failure domain. $f_X(\boldsymbol{x})$ is the JPDF of $X_1,\ldots, X_N$. Using the importance sampling technique, Eq. (26) is modified to

$$p_F = \int I(\boldsymbol{y})\frac{f_X(\boldsymbol{y})}{h_X(\boldsymbol{y})}h_X(\boldsymbol{y})d\boldsymbol{y} \tag{27.1}$$

,or $$p_F = \int I(\boldsymbol{y})\frac{f_X(\boldsymbol{y})}{h_X(\boldsymbol{y})}h_X(\boldsymbol{y})d\boldsymbol{y} \tag{27.2}$$

Note that the subscript h signifies that the expectation E is taken with respect to an importance sampling PDF or Importance Sampling Function (ISF) $h_X(\boldsymbol{x})$. The failure probability is estimated as

$$P_F = \frac{1}{Nsim}\sum_{j=1}^{Nsim} I(\boldsymbol{Y}_j)\frac{f_X(\boldsymbol{Y}_j)}{h_X(\boldsymbol{Y}_j)} \tag{28}$$

in which $\boldsymbol{Y}_j$ is the j-th sample from the ISF $h_X(\boldsymbol{x})$ and N_{sim} is the sample size.

The PML obtained from GAs search can enhance the efficiency of Monte Carlo Simulation (MCS). The efficiency enhancement is accomplished by employing the GAs-searched PML as the sampling center of the ISF $h_X(\boldsymbol{x})$. This sampling scheme is denoted as an Importance Sampling around PML (ISPML). For the purpose of procedure clarity, the original JPDF $f_X(\boldsymbol{x})$ will be rewritten as $f_X(\boldsymbol{x}|\boldsymbol{\mu}=\boldsymbol{\mu}_o)$ in which $\boldsymbol{\mu}$ denotes the mean vector. $\boldsymbol{\mu}_o$ is the original mean vector. According to the ISPML, the ISF $h_X(\boldsymbol{x})$ takes the form

$$h_X(\boldsymbol{x}) = f_X(\boldsymbol{x} \mid \boldsymbol{\mu} = \boldsymbol{x}^*) \tag{29}$$

, where $\boldsymbol{x}^*$ is PML. That is the ISF has the same functional form as the original JPDF. The mean vector of the ISF, however, is different from that of the original JPDF and takes the PML as the mean vector. Consequently, the estimate of the failure probability is

$$P_F = \frac{1}{Nsim}\sum_{j=1}^{Nsim} I(\boldsymbol{Y}_j)\frac{f_X(\boldsymbol{Y}_j \mid \boldsymbol{\mu} = \boldsymbol{\mu}_o)}{f_X(\boldsymbol{Y}_j \mid \boldsymbol{\mu} = \boldsymbol{x}^*)} \tag{30}$$

5.2.2. Numerical example 2: a plate with an edge crack [19]

Based on the GAs-searched PML, the ISF takes the form defined by Eq. (29) with the mean equal to PML, i.e.

$$\boldsymbol{\mu} = \begin{bmatrix} a^* & s_t^* & s_b^* & k_{Ic}^* \end{bmatrix}^T \tag{31}$$

The ISF as defined by Eq. (29) is used for computing the failure probability according to the LSF (22). The results are compared with MCS and shown in *Figures 7, 8,* and *Figure* 9. The estimate of the failure probability in each MCS and ISPML methodology is based on 10 independent runs. The sample sizes shown in all figures are the values used in each respective run.

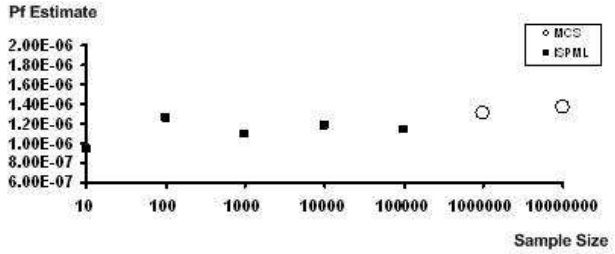


Figure 7. Estimates of failure probability from MCS and ISPML.

Figure 7 shows that the estimated failure probability is at the order of 10^6. The theoretical sample sizes required by MCS for the estimation are at least 10^6. The same figure also shows the sample sizes used by ISPML in order to compute the failure probability. It is obvious that the sample sizes used by ISPML are much smaller than the theoretically required sample sizes for MCS. *Figure 8* shows the variation of COV with respect to different sample sizes in both MCS and ISPML cases. The rate of COV reduction with respect to sample size in case of ISPML is significantly higher that that in case of MCS.

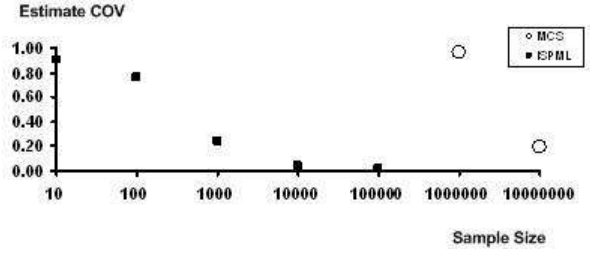


Figure 8. Effect of sample size on estimate COV.

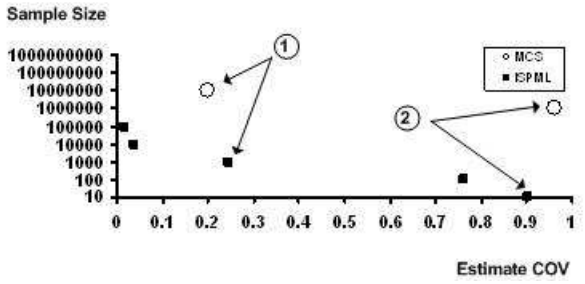


Figure 9. Sample sizes used by MCS and ISPML, with respect to the estimate COVs.

The same information in *Figure 8* is rearranged and plotted in *Figure 9*. Symbols ① and ② in *Figure 9* indicate that MCS employs the sample sizes approximately 1,000 times larger than ISPML uses in order to attain the same confidence level of estimate or estimate COV. ISPML is thus more efficient than MCS with respect to the quality of the estimate and the computational resource consumption. In this respect, GAs significantly contributes to the reduction in computational complexity. GAs also realizes the estimation of failure probability in the situation where the size of sampling is limited by the availability of computational resources. According to the numerical results, the computation of failure probability can demand considerable computation resource although the system dimension is large.
In summary, GAs can be used as tool for enhancing the efficiency in risk analysis by providing such crucial information as the PML which is then employed in the analysis procedure.

5.3. Determination of multiple design points

5.3.1. Problem characteristics

Design point is the point on the limit state surface that is nearest to the origin in a standard normal space. In optimization context, the design point is the global minimum obtained from solving a constrained optimization problem. However, it is possible that there are other local minima whose distances to the origin are of similar magnitudes to the global minimum. The global minimum and local minima with similar magnitudes lead to the situation of multiple design points. When multiple design points exist, the reliability analysis based only on any single design point among the multiple design points may result in an underestimation of failure probability. Determination of global optimum as well as local optima belongs to a multimodal function optimization. The following section intends to demonstrate the application SCGAs and DCGAs to the determination of multiple design points. Since SCGAs and DCGAs are originally designed for unconstrained optimization problems, it is necessary to provide means of handling constraints. The adaptive penalty technique as proposed in [17] and described in Section 3.3 will be combined with SCGAs and DCGAs for handling constraints.

5.3.2. Problem formulation

From the definition of the design point, the design point $\boldsymbol{U}^*$ is obtained from solving the following constrained optimization problem:

$$\text{Minimize} \qquad O(\boldsymbol{U}) = \|\boldsymbol{U}\| \tag{32}$$

$$\text{Subject to constraints} \qquad g(\boldsymbol{U}) = 0 \tag{33}$$

in which $\boldsymbol{U} = [U_1 \ \ldots \ U_N]^{\mathrm{T}}$ denotes the vector of standard normal variables. $g(\boldsymbol{U})$ is the LSF. N is the total number of basic random variables. $g(\boldsymbol{U}) = 0$ is the limit state surface and $g(\boldsymbol{U}) \leq 0$ indicates the failure state corresponding to the LSF. For the constrained optimization above, there may exist several optima whose objective function values are of similar magnitudes to the global maximum. Erroneous results occur if the optima from which the neighborhoods have significant contributions to the reliability assessment have not been located and so have been neglected. It is, therefore, necessary to locate global optimum as well as local optima. This can be achieved by solving a constrained multimodal optimization problem of the form (32) and (33).

5.3.3. Numerical example 3: a highly non-linear limit state function

This example considers the following complicate LSF as introduced in [17].

$$g(U_1, U_2) = 6 - \cos(5U_1) - U_2 \tag{34}$$

where U_1 and U_2 are independent standard normal variables. Although the problem is of low dimension, the purpose of this problem is to illustrate the capability of multimodal GAs in determining multiple design points under situation of highly nonlinear LSF. *Figure 10* displays the plot of the limit state surface in 2-dimensional space. The plot of the limit state surface shows that all local optima are at the base or bottom of the wavy limit state surface.

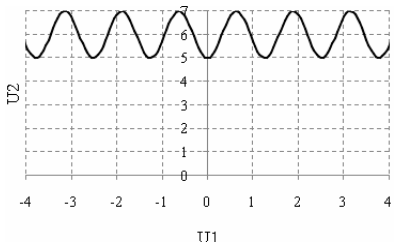


Figure 10. The limit state surface as in Eq. (34).

The equality constraint (34) is modified to an equality constraint for the purpose of its handling. The resulting equality constraint is

$$g_3(U_1,U_2)=\left|6-cos(5U_1)-U_2\right|-\delta \tag{35}$$

,where δ is the tolerance of being null. The value of δ is set to a small value to ensure that the obtained solution is closed to the limit state surface as much as possible.

Since GAs are originally designed for maximization problems, the fitness function according to the expressions (32) and (33) is defined as

$$F(U_1,U_2)=\frac{1}{H(U_1,U_2)} \tag{36}$$

$$H(U_1,U_2)=\begin{cases} O(U_1,U_2) & ;g_3(U_1,U_2)\le 0 \\ O(U_1,U_2)+kv(U_1,U_2) & ;g_3(U_1,U_2)>0 \end{cases} \tag{37}$$

$$O(U_1,U_2)=\sqrt{U_1^2+U_2^2} \tag{38}$$

The magnitude of the constraint violation, according to Eqs. (7) and (35), is

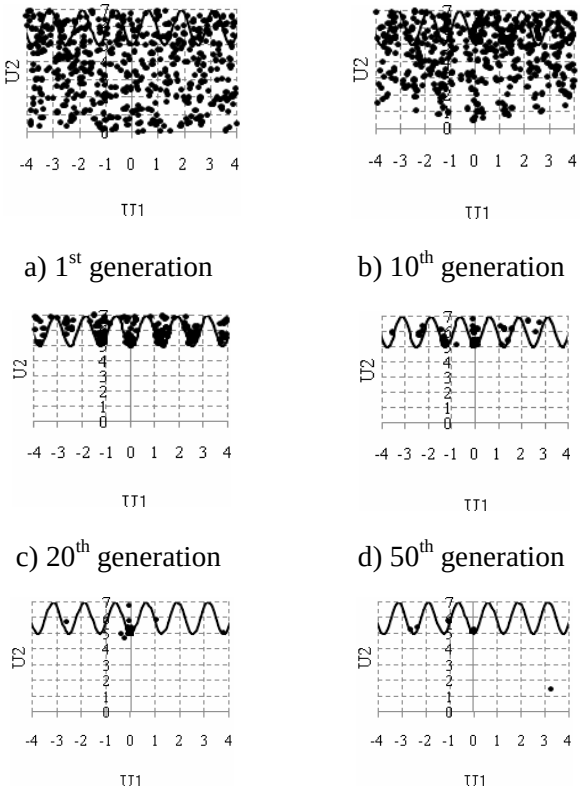
$$v(U_1,U_2)=\begin{cases} \left|g_3(U_1,U_2)\right| & ;g_3(U_1,U_2)>0 \\ 0 & ;\text{otherwise} \end{cases} \tag{39}$$

Both SCGAs and DCGAs are employed to search for design points in the domain of [-4,4]×[0,7]. The search results from each respective algorithm are separately shown in the following subsections. It should be noted in Eq. (37) that the penalty term is added to the objective function (38), instead of subtracting it, because the problem is a minimization problem.

5.3.3.1. SCGAs

The parameters for the SCGAs are given in *Table 4.*

The evolutions of the search using SCGAs approach are shown in *Figure 11*. The evolution also shows the so-called genetic drift effect.



a) 1st generation b) 10th generation

c) 20th generation d) 50th generation

e) 80th generation f) 100th generation

*Figure 11.*Chromosomes distribution at various generations by SCGAs in Example 3.

The genetic drift normally happens with the niche methods such as SCGAs and DCGAs.The genetic drift leads to the possible lost of the already captured optima. The optima found by SCGAs are also shown in *Table 5*. δ in Eq. (35) is equal to 0.01.

Table 4. SCGAs parameters for Example 3.

Parameters	Value
NPop	500
P_c	1.00
G	0.1
CF	0.2
Number of Generations	100

5.3.3.2. DCGAs

The parameters for the DCGAs are given in *Table 6*. The search results by DCGAs at different

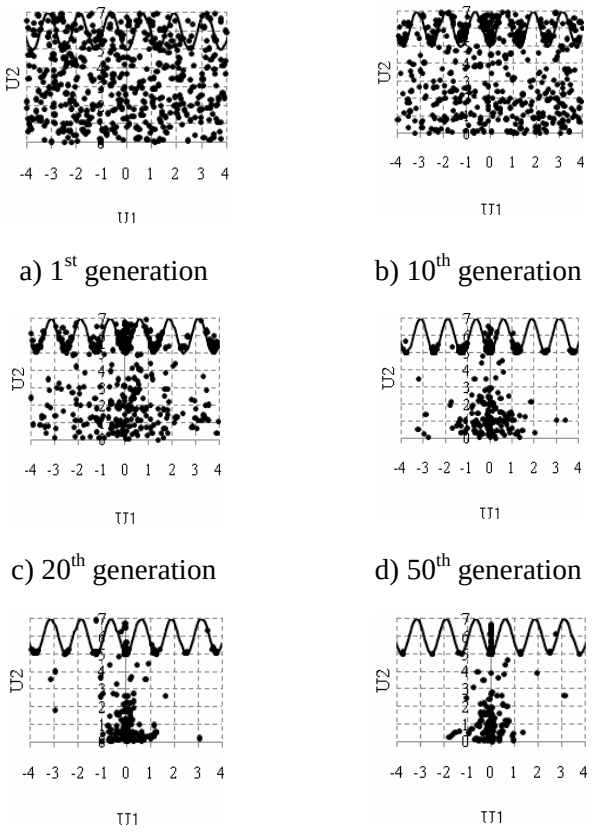
generations are shown in *Figure 12*. δ in Eq. (35) is equal to 0.01. The optima are summarized in *Table 5*. The GAs results explicate that both SCGAs and DCGAs have noticeable capabilities in simultaneously locating several design points under the LSFs with high non-linearity and irregular geometry. Regarding to two utilized multimodal GAs, it is possible that the found design points in early generations can be lost due to the inherent genetic drift. When considering from the plots of population distributions at different generations, DCGAs are more stable in maintaining the found results than SCGAs. The numerical results also suggest that DCGAs provide better quality of solutions than SCGAs. It should be noted that the multimodal GAs work in a different manner from the sequential search methods, such as the gradient-based methods, where the decision on the numbers of the desired design points must be made by the user. It is possible that some significantly contributing design points may be ignored if the search is stopped before total numbers of the influential design points are obtained. In that case, the failure probability is certainly underestimated. The multimodal GAs, therefore, naturally circumvents such a difficulty.

Table 5: Comparison of exact multiple design points with the design points from SCGAs and DCGAs.

Optima	X_1			X_2		
	Exact	SCGAs	DCGAs	Exact	SCGAs	DCGAs
1	-3.75	-3.71	-3.74	5.00	5.19	5.01
2	-2.50	-2.65	-2.51	5.00	5.54	5.00
3	-1.25	-1.21	-1.26	5.00	5.10	5.00
4	0.00	-0.07	0.01	5.00	5.09	5.00
5	1.25	1.18	1.25	5.00	5.09	5.00
6	2.50	2.58	2.50	5.00	5.10	5.00
7	3.75	3.75	3.75	5.00	5.05	5.03

Table 6. DCGAs parameters for Example 3.

Parameters	Value
Npop	500
P_c	1.00
Number of Generations	100



a) 1st generation b) 10th generation

c) 20th generation d) 50th generation

e) 80th generation f) 100th generation

Figure 12. Chromosomes distribution at various generations by DCGAs in Example 3.

6. Conclusion

Application of Genetic Algorithms (GAs) to reliability analysis is described, discussed, and summarized in this paper. Simple GAs and Multimodal GAs are addressed. GAs are applied to solve key problems in reliability analysis. The problems include the determination of the Point of Maximum Likelihood in failure domain (PML), the computation of failure probability using the so-called Importance Sampling around PML (ISPML), and the determination of multiple design points. GAs-aided reliability analysis requires no approximation of Limit State Functions (LSFs) although the LSFs are implicit, non-linear, non-differentiable, and noisy. Consequently, the computational modeling errors of LSFs are circumvented. The numeric-based feature of GAs also makes the requirement on the knowledge and visualizability of the geometry or landscape of limit state surfaces become unnecessary. The capability in the reliability analysis of complex systems, where their knowledge is not

always given and their visualizability are generally not possible, is thus enhanced. From the viewpoint of computational performance, the computational efficiency with respect to the sample size is significantly improved via the utilization of GAs-searched PML in ISPML and the substantial reduction in sample size. The reliability analysis of rare events can be realized even if the computational resources are limitedly provided. In case of multiple design points, the population-based nature of GAs can be advantageously employed to automatically detect those crucial information points simultaneously. Unlike sequential search algorithms, the information about the number of design points is not necessary. The multimodal GAs, therefore, reduces the possibility of missing influential design points. Finally, the algorithmic simplicity of GAs in implantation is considered a highly attractive feature from the viewpoint of software engineering and practical application.

References

[1] Barbosa, H. & Lemonge, A. (2003). A new adaptive penalty scheme for genetic algorithms. *Information sciences*, 156, 215-251.

[2] Brownlee, J. (2004). *Parallel Niching Genetic Algorithms: A Crowding Perspective*. Thesis in Master of Information Technology, Centre for Intelligent Systems and Complex Processes, School of Information Technology, Swinburne University of Technology, Australia.

[3] Darwin, Ch. (1869). *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. 5th ed., John Murray, London.

[4] Deb, K. (1995). *Optimization for Engineering Design Algorithms and Examples*. Prentice-Hall, New York.

[5] De Jong, K.A. (1975). *An analysis of the behavior of a class of genetic adaptive systems*. Ph.D. Thesis, Department of Computer Science, University of Michigan, Ann Arbor, MI, USA.

[6] Fishman, G. (1996). *Monte Carlo: Concepts, Algorithms, and Applications*. Springer-Verlag, New York.

[7] Gen, M. & Cheng, R. (1997). *Genetic algorithms and engineering design*. John Wiley and Sons, New York.

[8] Gen, M. & Yun, Y.S. (2006). Soft computing approach for reliability optimization: State-of-the-art survey. *Reliability Engineering and System Safety*, 91, 1008–1026.

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

[10] Harnpornchai, N., Chakpitak, N., Chandarasupsang, T., Tuang-Ath Chaikijkosi. & Dahal, K. (2007). Dynamic adjustment of age distribution in Human Resource Management by genetic algorithms. *Evolutionary Computation, CEC 2007, IEEE Congress* on 25-28 Sept. 2007, 1234 – 1239

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

[12] Kimura, M. (1983). *The Neutral Theory of Molecular Evolution*. Cambridge University Press.

[13] Mahfoud, S.W. (1995). A comparison of parallel and sequential niching methods. *International Conference on Genetic Algorithms*, 136-143.

[14] Mahfoud, S.W. (1995). *Niching methods for genetic algorithms*. Ph.D. dissertation, Univ. of Illinois, Urbana-Champaign.

[15] Michalewicz, A. (1996). *Genetic + Data Structures = Evolution Programs*. Second ed. Berlin, Springer.

[16] Miller, B.L & Shaw, M.J. (1995). *Genetic algorithms with dynamic niche sharing for multi-modal function optimization.* IlliGAL Report no. 9510.

[17] Obadage, A.S. & Hampornchai, N. (2006). Determination of point of maximum likelihood in failure domain using genetic algorithms. *Int J Press Vessels Pipe*, 83(4), 276-82.

[18] Qing, L., Gang, W., Zaiyue, Y. & Qiuping, W. (2008). Crowding clustering genetic algorithm for multi-modal function optimization,. *International Journal of Applied Soft Computing*, Volume 8, Issue 1, 88-95.

[19] Sprung, I. (2003). Invariance of safety factor in probabilistic fracture mechanics analysis. *Int J Press Vessels Pipe*, 80, 367–378.

[20] Tada, H. (1978). *The stress analysis of cracks handbook*. Del Research Corporation.

[21] Wallin, K. (1984). The scatter in KIc results. *Engng Fract Mech*,19,1085–93.