Journal of Universal Computer Science, vol. 14, no. 15 (2008), 2566-2581
submitted: 18/1/08, accepted: 30/6/08, appeared: 1/8/08

GADYM - A Novel Genetic Algorithm in Mechanical Design Problems

Khadiza Tahera
(Monash University, Melbourne, Australia
Khadiza.Tahera@eng.monash.edu.au)

Raafat N. Ibrahim
(Monash University, Melbourne, Australia
Raafat.Ibrahim@eng.monash.edu.au)

Paul B. Lochert
(Monash University, Melbourne, Australia
plochert@bigpond.net.au)

Abstract: This paper proposes a variant of genetic algorithm – GADYM, Genetic Algorithm with *Gender-Age structure, DYnamic parameter tuning and Mandatory self perfection scheme.* The motivation of this algorithm is to increase the diversity throughout the search procedure and to ease the difficulties associated with the tuning of GA parameters and operators. To promote diversity, GADYM combines the concept of gender and age in individuals of a traditional Genetic Algorithm and implements the self perfection scheme through sharing. To ease the parameter tuning process, the proposed algorithm uses dynamic environment in which heterogeneous crossover and selection techniques are used and parameters are updated based on deterministic rules. Thus, GADYM uses a combination of genetic operators and variable parameter values whereas a traditional GA uses fixed values of those. The experimental results of the proposed algorithm based on a mechanical design problem show promising result.

Keywords: Genetic Algorithm, Optimization, Search
Categories: F.2.0, G.1.6, I.2.8

1 Introduction

Originally developed by Holland [Holland, 75], a genetic algorithm (GA) is a robust technique, based on the natural selection and genetic production mechanism. The genetic algorithm works with a group of possible solutions within a search space instead of working with a single solution as is seen in gradient optimization methods. In addition, the search in GA is probability guided and stochastic, rather than deterministic or random searching, which distinguishes it from traditional methods.

The use of genetic algorithm requires the choice of a set of genetic operations among many possibilities. The users need to set parameters, which are often determined from a set of experiments. For example, the user needs to define genetic parameters such as values of population size, crossover rate and mutation rate. Additionally, the user needs to select genetic operators such as selection type, crossover type and mutation type. This choice can be effective for one type of GA but worse for another. Thus, selecting the appropriate parameters and genetic operators becomes a complex permutation problem and is very time consuming. Nevertheless, the success of the GA depends on the correct choice of parameter and genetic operators. In most cases, these parameters are set in advance. However, setting the GA parameters in advance entails the following problems: (1) normally the good enough values for the GA parameters depend on the particular problem; and (2) the genetic operators and GA parameter settings are likely to be different for each problem or even for each individual. Thus, a better performance might be achieved by adjusting the parameters of a GA.

In general, the choice of genetic parameters has been debated in both analytical and empirical investigations. The trade-offs that arise in this issue are:

- Increasing the crossover rate increases recombination of building blocks, but it will also increases the disruption of good strings.
- Increasing the mutation rate tends to transform the genetic search into a random search, but it also helps to reintroduce lost genetic material.

There has been much research on adaptive GAs that allows dynamic adjustment during execution [Lobo, 2000; Vasconcelos, 2001; Hatta, 2001 and Cervantes, 2006]. Lobo [Lobo, 2000] used an adaptive population size which increased continuously by trying to reach the right population size. However, the termination of this growing process of the population was left to the user. Vascobcelos et al. [Vasconcelos, 2001] investigated dynamic adaptation of crossover and mutation probabilities in a GA using three analytical test functions. When the genetic diversity is smaller, then they reduced the crossover rate and augmented the mutation rate by a coefficient value. In the other case, when the genetic diversity is higher, then they augmented the crossover rate and reduced the mutation rate by a coefficient value. The results revealed that dynamic adaptability of crossover and mutation probabilities, the reduction space and the global elitism enhance the performance of GA. In other study [Cervantes, 2006], the mutation rate was calculated based on the string length; the mutation rate was decreased with increasing number of generations or based on the Hamming distance between strings. Hatta et al. [Hatta, 2001] used an elite degree as a measure of individual's latent fitness. This measure was employed to switch among crossover options and adjust mutation probabilities.

In this context, a *Genetic Algorithm with* Gender-Age structure, DYnamic parameter tuning and Mandatory self perfection scheme (GADYM), in which the genetic operators and parameters setting are varied during GA execution, has been proposed. The rest of this paper is outlined as follows. Section 2 describes the proposed GADYM. Section 3 describes the experiment of a mechanical design optimization problem for this study. The experimental result and relevant analysis are presented in Section 4. Section 5 concludes this paper.

2 GADYM – A Variant of GA

The goals of GADYM are to maintain diversity in the population, prevent GA to converge prematurely to local minima and ease the parameter tuning process. The approach is based on the following fundamental factors:

- The reproduction is only permitted to opposite gender and the operation produces one child or two children depending on the fertility rate of the parents.
- When the crossover is not carried out, the parents may adopt child depending on their random decision.
- The gender of the child is determined by a threshold limit, γ which considers the gender density of the population.
- The mutation is biased on the group's gender; thus the mutation rates for male and female are different
- The genetic operators (i.e. selection type, crossover type) and the genetic parameters (i.e. crossover rate, mutation rate) are changed randomly to represent the dynamic environment. Thus, the algorithm utilizes the strengths of a group of genetic operators.

2.1 The Steps of GADYM

As a general example, consider minimizing the following objective function:

$$f = f(x_1, x_2.....x_n) \qquad (1)$$
$$a_i \le x_i \le b_i \quad i = 1,2,3,...n$$

Where x_i is the i-th variable and a_i and b_i are the limits of that variable. The function variables could be either integer or real. The steps to implement GADYM are the following.

2.1.1 Initial Generation

An initial generation is composed of a number of individuals which undergo different genetic operations. GADYM considers each individual consist of chromosome and other features as opposed to chromosomes of a standard genetic algorithm. The formation of an individual in the initial generation is given below.

(a) Coding

The simplified coding is a one-variable, one-code system. Each variable (or gene) is randomly generated (x'_i) which is subject to uniform (0, 1) distribution and coded according to the lower (a_i) and upper (b_i) limit. The coded variable is given by,

$$x_i = a_i + x'_i(b_i - a_i) \qquad (2)$$

This coding offers the possibility of directly programming the real location of the parameter in its feasible range.

(b) Gene generation

Instead of binary coding, GADYM uses real coding to represent a chromosome. We randomly select a distinct point for each gene, (i.e. variable) of the problem from the search space. Each point is selected with equal probability and is coded using Eqn.

(2). Thus, this identifies the first point (i.e. chromosome) in the search space. The range of each variable depends on a given problem. The chromosome will look like:

$$chromosome = x_1, x_2 .....x_n$$

If an invalid gene is formed which does not satisfy a constraint of the given problem, we ignore the whole chromosome and generate another new chromosome with valid genes. Thus, no repairing algorithm is used in GADYM. This coding may also be adopted in an Evolutionary Algorithm.

(c) Gender and Age Assignment

Each individual is randomly assigned a gender (either male or female) that is not changed throughout the process.

A zero age is given to the individual at the initial generation. The death (or lethal) age has a major effect on the performance of GADYM. We can control the diversity of population using death age. If the death age is small then the individuals will die soon and new individuals will be created. Thus the algorithm approaches more to exploratory nature rather than exploitative nature. The death age corresponds to the control parameter temperature in the simulated annealing optimisation strategy. Unlike other aged Genetic Algorithms in the literature, the death age of each individual in GADYM is different and is determined randomly.

$$\text{Death Age of individual} = \text{Rnd() X Common Death Age of the population} \qquad (3)$$

where $0 \le \text{Rnd()} \le 1$

(d) Fertility Rate

The fertility rate defines the number of children to be produced in a crossover operation. The fertility rate incorporates the age effect. In biological populations, it will depend in a complex manner on perceived likelihoods of producing healthy offspring. For our purpose, a simple triangular function of width around the age of maximum fertility proves adequate. Thus, the fertility rate increases up to the maximum fertility age and then decrease according to the width of the triangle. Figure 1 shows the triangular concept of age in calculating fertility rate.

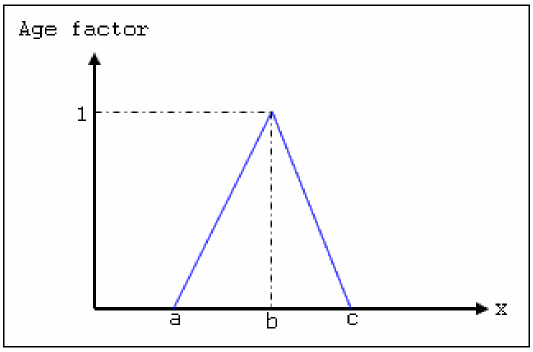

Figure 1: Triangular age concept

$$\begin{aligned} A &= \frac{1+age(y)-a}{b-a}; && a \le age(y) < b-1 \\ A &= 1; && age(y) = b-1 \\ A &= \frac{c-age(y)-1}{c-b}; && b < age(y) < c-1 \\ A &= 0; && otherwise \end{aligned} \tag{4}$$

Where: A = Age factor
b = Age at maximum fertility
c-a = Triangle width

Figure 2 shows the features of an individual.

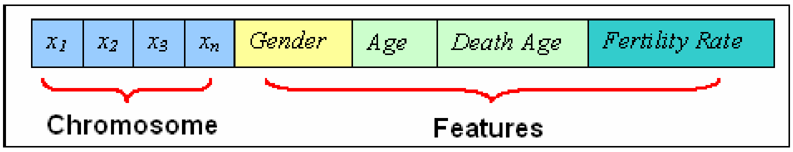

Figure 2: Chromosome and features of an individual

We repeat this procedure for N times for each of the individual. Therefore, the initial population ($P_{i=0}$) is constructed with the population size of N.

2.1.2 Fitness Evaluation

The objective function values of the points in the population (P_i) are calculated by using Equation. 1. This value represents how fit (i.e. good) an individual is.

2.1.3 Dynamic Parameter Tuning

The genetic operators and parameters can be changed either at each generation or at some specific generation calculated as:

$$t_m = t + rand(t_{total} - t) \tag{5}$$

Where: t_m = the generation to change parameters
t = the current generation
t_{total} = total generation

The parameters values are changed according to some deterministic rules which are.

- The selection, crossover and mutation technique are chosen from a group of options based on a random integer.
- The crossover and mutation rates are changed as follow. In nature, the mutation rate of male and female are different. Thus GADYM considers a different mutation rate

for male and female. The crossover and mutation methods are discussed detail in next sections.

$$p_{ct} = \frac{p_c}{\sqrt{t}}$$

$$m_y = \sqrt{t} \times (Rnd() \times 0.1)$$

$$m_x = \sqrt{t} \times (Rnd() \times (0.01 - m_y) + m_y) \qquad (6)$$

Where: p_c = the initial crossover rate

p_{ct} = the crossover rate at t

m_y = the mutation rate of male

m_x = the mutation rate of female

t = the current generation

2.1.4 Pre-selected Population for the Next Generation

The population for the next generation is created after genetic operations of crossover and mutation. The number of times to perform the crossover operation needs to be decided beforehand. In GADYM we adopt the following strategy as shown in Figure 3.

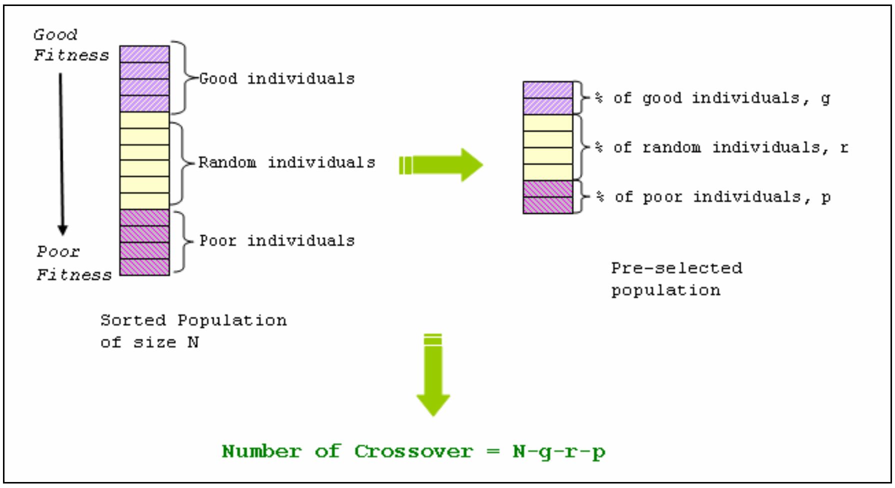

Figure 3: Determining the number of crossover operation in a generation

First we choose a preselected population for the next generation. To keep diversity, we pick a proportion of good individuals (g), poor individuals (p) and random individuals (r) from the sorted population. The number of crossover (ψ) in a generation will be:

$$\psi = N - g - r - p \tag{7}$$

Where: N = the population size.

For instance, consider population size of 64, g=10%, p=10% and r=20%, we take preselected population for next generation as follow. We sort the population based on fitness value. Then we choose 10% good individual (i.e. 6 individuals from the top), 10% poor individuals (i.e. 6 individuals from the bottom) and 20% random individuals (i.e. 12 individuals) from the sorted population to make the preselected population of 24 individuals. This preselected population will be added to the children generated in crossover operation. We calculate the number of crossover in a generation by Equation (7).

$$\psi = 64 - 6.4 - 6.4 - 12.8 = 38.4 = 38$$

Thus we perform the crossover operation ψ times in a generation to produce the children.

2.1.5 Parent Selection

In the genetic crossover, we need to select parents who qualify for reproduction. During the parent selection process it is ensured that the mating is between a male (m_i) and a female (f_i). In this algorithm, the males and the females are grouped separately. Then the first candidate is selected from the males' pool (m) and the second candidate is selected from the females' pool (f). Thus, the mating of opposite genders is ensured. Mathematically, in a population P,

$$m_i \in m \quad \text{and} \quad f_i \in f$$
$$m_i \in P \quad \text{and} \quad f_i \in P$$
$$P = m \cup f \quad \text{and} \quad m \cap f = 0 \tag{8}$$

In each pool, the selection is done based on the selection technique adopted in that generation.

2.1.6 Crossover

(a) *Crossover:* The parents go through the crossover operation to reproduce child. Based on the environment, the crossover technique depends on the crossover technique adopted in that generation. Thus, the algorithm may use a combination of crossover techniques in one particular run.

(b) Number of children: If the total fertility rate of mom and dad is greater than a fertile limit, ζ, then the crossover operation reproduces 2 children; otherwise it reproduces 1 child. ζ is age dependent which is computed as follows:

$$\xi = \frac{\xi_0}{\sqrt{t}} \tag{9}$$

Where: ξ_0 = the initial fertile limit (0 to 1)

ξ = the fertile limit at t

t = the current generation

(c) Adopting child: When the crossover is not performed between parents due to low crossover probability then a random number is generated to decide whether or not to take an adopted child based on adopting probability, α(α=0 to 1). If the parents decide to take an adopted child then a random immigrant is inserted in the population. This adopted child helps to introduce more diversity in the population. The number of adoption can be controlled by controlling α. For example, a high value of α restricts the parents to adopt a child i.e. the algorithm behaves exploitative nature. On the other hand, a low value of α encourages parents to adopt a child i.e. the algorithm tends to behave explorative nature.

(d) Gender Assignment: The gender of the children is assigned after creating the population for the next generation. The gender assignment of each child is based on monitoring the number of males and females in the population. If the number of a particular gender is reduced below a threshold limit l, the offspring's gender is assigned to that particular gender otherwise the gender is randomly determined. Therefore, this approach avoids the chance of possible no-regeneration.

(e) Updating age status: The status of the individuals selected to reproduce is updated as "parents" and the new born children after crossover as "child". In a discrete aged Genetic Algorithm [Kubota, 1994], the parents will die. Therefore, parents and children can not coexist. However, in a continuous aged Genetic Algorithm [Kubota, 1994], the parent does not die and thus parents and children coexist in a population. GADYM considers the continuous aged model.

These children are now combined with the pre-selected population formed in step 2.1.4. The mutation process is performed on this new population.

2.1.7 Mutation

Occasionally with a small probability p_m, we alter the population (i.e. population of the parents and children) to P_i'. Two different mutation approaches are taken based on a random number: (a) normal mutation and (b) dominant mutation. In a normal mutation, we use two different mutation rates for each gender and apply the gender information in computing the mutation rate (p_m).

$$p_m = \chi_y m_y + (1-\chi_y) m_x \tag{10}$$

where χ_y represents the proportion of males in the population, m_y the mutation rate of male and m_x = the mutation rate of female

In a dominant mutation, we apply mutation on each of the two gender group based on their respective mutation rate. As the mutation process is applied separately on two gender groups, thus this is named as dominant mutation.

Next, we generate a random number r. If $r \le p_m$ then we do the implicit mutation as follow: First: we decide the number of total mutations (z) by using a random number; Second: We randomly select the gene from the population (in case of normal mutation) or the gender groups of male and female (in case of dominant mutation) by using the following equation.

$$\begin{aligned} row &= Round(Rnd() \times (N-1)) \\ column &= Round(Rnd() \times (n-1)) \end{aligned} \tag{11}$$

Where: n = the chromosome size

N = the population size

p = the male and female group size [in case of dominant mutation, we replace N by p]

Third: We replace the value of the particular gene by another uniformly distributed random number; Finally, we repeat this process for z times.

2.1.8 Aging

After the mutation process, the age of each individual is incremented by one in each generation. The child produced after the crossover and mutation operation is assigned to be the age of 0. The age structure is considered as a continuous alternation model of generation. In other words, parents can coexist with their children.

2.1.9 Replacement Strategy

If an individual reaches the death (or lethal) age, then it is removed from the population i.e. it dies. It is noteworthy that the population size is not constant throughout the algorithm. If the population size reduces than the initial population size then random immigrants are inserted in the population. The immigrants are normal individuals which are generated randomly.

2.1.10 Mandatory Self Perfection Strategy

A new sharing scheme is implemented in GADYM. The individuals share information with a group of local best individuals and compete to be the best individual in a population. The sharing scheme utilizes the crossover concept with the difference that individuals do not produce any child, rather they update themselves to be fitter in the current environment. If the sharing does not raise their solution accuracy (i.e. fitness), they exist with their own values. The pseudo code is given in Figure 4.

We repeat STEP 2.1.2 to STEP 2.1.10 until the termination condition is met, which is to run the algorithm for a fixed number of generations. The fittest solution found during the procedure is the optimum solution of the algorithm. The steps of the proposed genetic algorithm are shown in Figure 5.

3 Design of a Pressure Vessel Problem

A cylindrical vessel is capped at both ends by hemispherical heads [Kannan, 1994]. The objective is to minimize the total cost, including the cost of the material, forming and welding. There are four design variables: $T_S(x_1)$ (thickness of the shell), $T_h(x_2)$ (thickness of the head), $R(x_3)$ (inner radius) and $L(x_4)$ (length of the cylindrical section of the vessel, not including the head). T_S and T_h are integer multipliers of 0.0625 in, which are the available thicknesses of rolled steel plates and R and L are continuous. (See Figure 6). The problem can be stated as a minimization of function *f*(*x*):

$$f(x) = 0.6334x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \tag{12}$$

Subject to: $g_1(x) = -x_1 + 0.0193x_3 \leq 0$ $\quad g_2(x) = -x_2 + 0.00954x_3 \leq 0$

$$g_3(x) = -\pi x_3^2 x_4 - \tfrac{4}{3}\pi x_3^3 + 1296000 \leq 0 \quad g_4(x) = x_4 - 240 \leq 0$$

```
Function: mandatory self perfection

Select a pool size of m with g% good individuals from the population
        POP of size N
Individual, i = m+1
while i < N do
        Randomly select an individual, r from the pool
        Perform crossover between i and r and produce 1 child, o

        If fitness_O > fitness_i then
                Update current position of i with o's position
        Else
                Restore original position of i
        End if

end while
```

Figure 4: Function to mandatory self-perfection approach

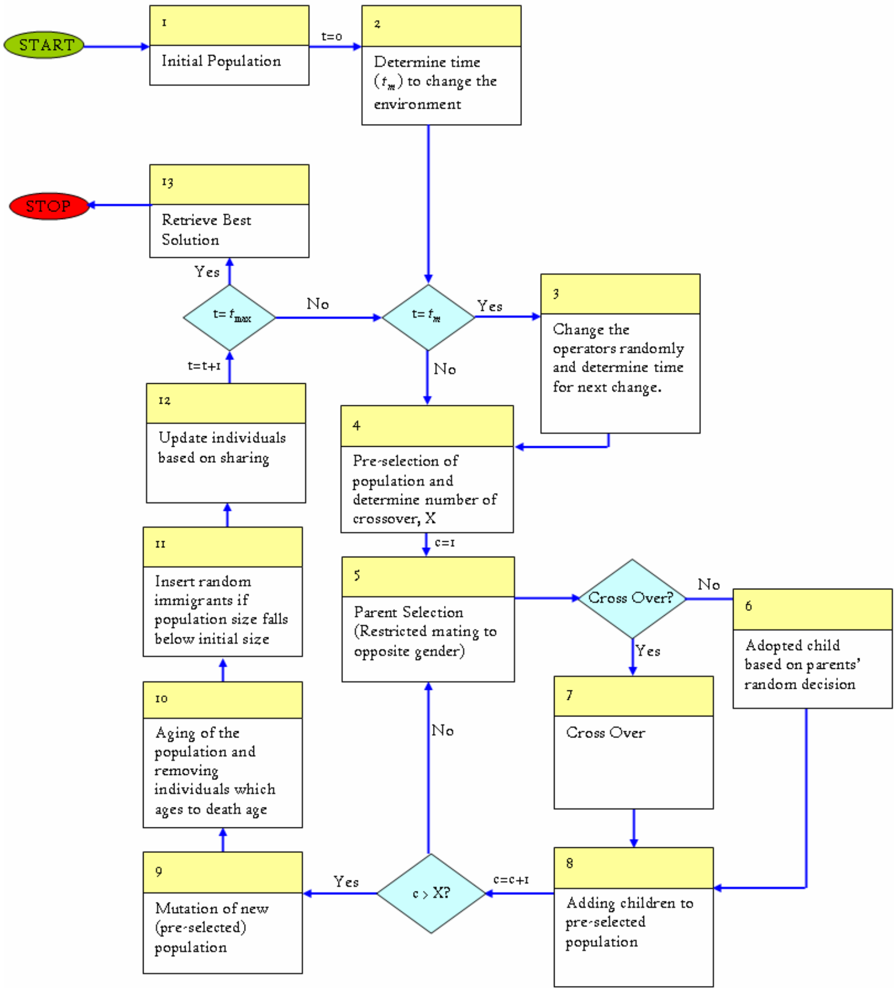

Figure 5: The steps of GADYM

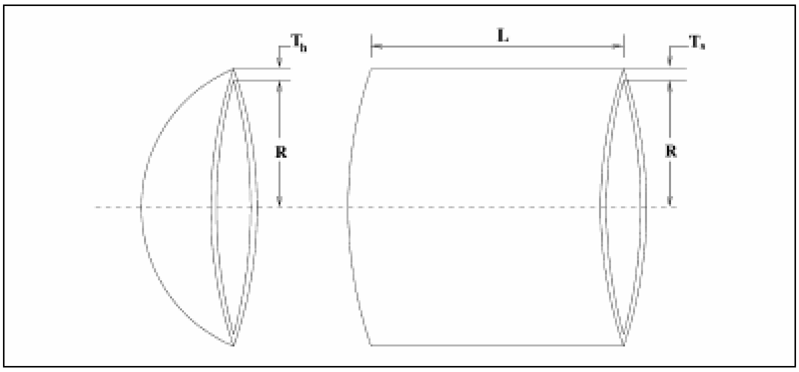

Figure 6: Pressure vessel

4 Experimental Study

GADYM is applied to the mechanical design problem. For comparison purpose, the results of four variants of Genetic Algorithm are presented. These are: standard Genetic Algorithm, gendered Genetic Algorithm [Drezner, 2006], Aged Genetic Algorithm [Kubota, 1994], gendered and aged genetic algorithm and GADYM. The parameters of each of the algorithms are given in Table 1 where the first three variants use the same parameters in (a) and GADYM uses the parameters in (b).

We assess and compare the performance of these algorithms by using solution quality, De-Jong's [De Jong, 1975] online and offline measure. Due to the stochastic nature of Genetic Algorithms, hundred trials are conducted and the best of all these trials is considered as the optimum solution. De-Jong's online measure is defined as

$$X_e(h) = \frac{1}{T}\sum_{t=1}^{T} f_e(t)$$

Where: $f_e(t)$ is the fitness value of the best individual among the candidate solutions in generation t. This measure is thus the average of the best individuals in each generation iterated in the algorithm. De-Jong's offline measure is defined as

$$X_e^*(h) = \frac{1}{T}\sum_{t=1}^{T} f_e^*(t)$$

Where $f_e^*(t) = best\{f(a_1), f(a_2),... f(a_n)\}$ at generation t and T is the total number of generations. $f(a_1), f(a_2),... f(a_n)$ are the best solutions in generations $1,2,.....,n$. This measure is thus the average of the best individuals over all the generations iterated in the algorithm.

GA PARAMETERS	Value	Fixed/Variable
Population Size	64	Fixed
Total Generation	50	Fixed
Total Trial	100	Fixed
Crossover Rate	0.8	Fixed
Mutation Rate	0.08	Fixed

GENETIC OPERATOS	Value	Fixed/Variable
Selection	Tournament Selection	Fixed
Crossover	Single Point Crossover	Fixed
Mutation	Random Mutation	Fixed

GA PARAMETERS IN GADYM	Value	Fixed/Variable
Population Size	64	Variable
Total Generation	50	Fixed
Total Trial	100	Fixed
Crossover Rate	-	Variable
Mutation Rate	-	Variable

GENETIC OPERATOS	Value	Fixed/Variable
Selection	-	Variable
Crossover	-	Variable
Mutation	-	Variable

Table 1: GA Parameters used in mechanical design optimisation problem

	Best Solution found			
Design Variables	Coello	Deb	Kannan	Sandgren
x1	0.8125000	0.9375000	1.1250000	1.1250000
x 2	0.4375000	0.5000000	0.6250000	0.6250000
x 3	42.0973980	48.3290000	58.2910000	47.7000000
x4	176.6540470	112.6790000	43.6900000	117.7010000
Constraints				
g1	-0.0000202	-0.0047503	0.0000163	-0.2043900
g 2	-0.0358908	-0.0389413	-0.0689039	-0.1699420
g 3	-546.5323390	-4175.9878717	-542.8693940	-467.3929114
g 4	-63.3459530	-127.3210000	-196.3100000	-122.2990000
Objective Function				
f(x)	6059.9464093	6410.3811385	7198.0428258	8129.1036010

Table 2: Comparison of the results for optimisation of a pressure vessel

4.1 Experimental Results and Analysis

The design of a pressure vessel problem has been solved by Sandgren [Sandgren, 1988] using a branch-and-bound approach, by Kannan and Kramer [Kannan, 1994] using an augmented Lagrangian Multiplier approach, by Deb [Deb, 1997] using GeneAS (genetic adaptive search), and by Coello et al [Coello, 2002] using Genetic Algorithm with a dominance-based Tournament Selection approach. Their results are shown in Table 2. It is noticed from the table that the design variables found by [Kannan, 1994] slightly violates a constraint.

Table 3 shows the results of the variants of Genetic Algorithms. It is seen from Table 3 that the performance of GADYM is better than those of the other variants of Genetic Algorithms in terms of the best solution and offline measure. The standard genetic algorithm is better in terms of online measure. A popular optimisation algorithm Generalised Reduced Gradient (GRG) is also applied to the problem in order to compare the performance in Table 3. However, the offline and online measures of GRG are not available.

It could be argued that the performance of Genetic Algorithm [Coello, 2002], as can be seen from Table 2, is even better than GADYM. However, GADYM attempts to implement a different version of GA using gender based approach. Coello et al. [Coello, 2002] used dominance based tournament selection in a standard genetic algorithm whereas the proposed algorithm uses normal tournament selection in a gender based genetic algorithm. In either case, the results in Table 2 and Table 3 summarises that Genetic Algorithm is a better optimisation algorithm than the other conventional algorithms in mechanical design of pressure vessel.

It could be noticed from Tables 2 and 3 that conventional genetic algorithms have already been reported in previous work and GADYM presented a similar performance in mechanical design optimization problem. However, GADYM mimics the nature more closely by the introduction of “true” randomness in terms of using different genetic operators in each generation, different mutation rates for male and female, age dependent fertility rate, variable number of children and self-learning strategy. In doing so, GADYM improves the solution quality over each generation.

5 Conclusions

The parameter setting in a genetic algorithm becomes a complex permutation problem and the same set of parameter values may not work efficiently in all problems. Therefore, finding the genetic parameters in an adaptive environment becomes an active research area. In an adaptive environment, the genetic operators and GA parameters change dynamically during the processing. This situation is opposite to a steady state environment where a fixed value of GA operators and parameters are used throughout the operation.

This paper proposes a self adaptive genetic algorithm GADYM where the genetic operators are chosen randomly from a pool of options. Thus, a number of genetic operators can be used instead of using a fixed type of genetic operator. Thus, the proposed approach uses inhomogeneous crossover and selection techniques.

The goal of GADYM is to improve the performance of the genetic algorithm by enhancing the diversity of the population. To increase the diversity, auxiliary information of gender and age are incorporated to each individual. The fertility rate incorporates the age information and is used to determine the number of children to be produced during the crossover operation. Random immigrants are introduced when the crossover operations cannot produce a child. A mandatory self perfection scheme is adopted in this approach. GADYM was applied to mechanical design problem and it shows promising result.

Best Solution

	GA					GRG2
Var.	Standard GA	Gen. GA	Aged GA	Gen. &Aged	GADYM	
x1	0.9375	0.9375	0.9375	0.9375	0.8125	1.0000
x 2	0.5000	0.4375	0.5000	0.4375	0.4375	1.0000
x 3		45.3333	45.6809	45.6809	42.0976	37.6920
x4		140.6831	142.3777	140.4018	176.7685	240.0000
Constraints						
g1	-0.056	-0.063	-0.056	-0.056	0.000	-0.273
g 2	-0.064	-0.005	-0.064	-0.002	-0.036	-0.640
g 3	-27348.157	-3062.917	-37213.584	-24254.864	-1196.515	0.0000
g 4	-99.127	-99.317	-97.622	-99.598	-63.232	0.0000
Objective Function						
f(x)	6798.739	6502.028	6843.020	6552.952	6062.652	9664.086
Offline	10816.80	9465.83	10586.61	9338.42	8213.47	-
Online	9143.03	8565.68	8875.10	8361.49	13998.08	-

Table 3: Results of GA variants

Acknowledgments

The authors will like to acknowledge the support received from CRCIEAM and the postgraduate scholarship to carry out the research work.

References

[Cervantes, 2006] Cervantes J. and Stephens C. R.: Optimal mutation rates for genetic search, Proceedings of the 2006 Genetic and Evolutionary Computation Conference, Washington, USA, July, 2006, 1313 - 1320.

[Coello, 2002] Coello C. A. C. and Mpntes E. M.: Constraint handling in genetic algorithms through the use of dominance-based Tournament Selection, Journal of Advanced Engineering Informatics, 2002, 16, 193-203.

[Deb, 1997] Deb K.: GeneAS: a robust optimal design technique for mechanical component design, In: Dasgupta D, Michalewicz Z editors, Evolutionary algorithms in engineering applications, Berline: Springer, 1997, 497-514.

[De Jong, 1975] De Jong K. A.: Analysis of the Behaviour of a Class of Genetic Adaptive Systems, PhD thesis, Department of Computer and Communication Science, University of Michigan, 1975.

[Drezner, 2006] Drezner T. and Drezner Z.: Gender specific Genetic Algorithms, INFOR, 2006, 44(2), 117-127.

[Hatta, 2001] Hatta K.: Wakabayashi S. and Koide T. Adaptation of genetic operators and parameters of a genetic algorithm based on the elite degree of an individual, Systems and Computers in Japan, 2001, 31, 1, 29 – 37.

[Holland, 1975] Holland: Adaptation in natural and artificial systems, Ann Arbor, MI: University of Michigan Press, 1975.

[Kannan, 1994] Kannan BK and Kramer S. N.: An augmented Lagrange multiplier based method for mixed integer discrete continuous optimisation and its applications to mechanical design, Journal of Mechanical Design Transactions ASME, 1994, 116, 318-320.

[Kubota, 1994] Kubota N., Fukuda T., Arai F. and Shimojima K.: Genetic Algorithm with age structure and its application to self-organising manufacturing system, IEEE Symposium on Engineering Technology and Factory Automation, 1994, 472-477.

[Lobo, 2000] Lobo F. G.: The parameter-less Genetic Algorithm: Rational and Automated Parameter Selection for simplified Genetic Algorithm operation, Ph.D. theses, Universidade Nova de Lisboa, Lisboa, 2000.

[Sandgren, 1988] Sandgren E.: Nonlinear integer and discrete programming in mechanical design, Proceedings of the ASME Design Technology Conference, Kissimine FL, 1988, 95-105.

[Vasconcelos, 2001] Vasconcelos J. A., Ramfrez J.Am Takahashi R. H. C. and Saldanha R. R.: Improvements in Genetic Algorithms, IEEE Transactions on Magnetics, 2001, 37, 5, 3414-3417.