H. KUTAY

METU 2007

STACKING SEQUENCE OPTIMIZATION OF A

COMPOSITE PRESSURE VESSEL

BY GENETIC ALGORITHM

HALİL KUTAY

SEPTEMBER 2007

STACKING SEQUENCE OPTIMIZATION OF A COMPOSITE PRESSURE VESSEL BY GENETIC ALGORITHM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HALİL KUTAY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
AEROSPACE ENGINEERING

SEPTEMBER 2007

Approval of the thesis:

STACKING SEQUENCE OPTIMIZATION OF A COMPOSITE PRESSURE VESSEL BY GENETIC ALGORITHM

submitted by **HALİL KUTAY** in partial fulfillment of the requirements for the degree of **Master of Science in Aerospace Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ______________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmail H. Tuncer ______________________
Head of Department, **Aerospace Engineering**

Assoc. Prof. Dr. Altan Kayran
Supervisor, **Aerospace Engineering Dept., METU** ______________________

Examining Committee Members:

Prof. Dr. Ozan TEKİNALP ______________________
Aerospace Engineering Dept., METU

Assoc. Prof. Dr. Altan Kayran ______________________
Aerospace Engineering Dept., METU

Prof. Dr. Polat M. Saka ______________________
Engineering Sciences Dept., METU

Dr. Güçlü Seber ______________________
Aerospace Engineering Dept., METU

Dr. Volkan Nalbantoğlu ______________________
Principal Controls Engineer, ASELSAN Inc.

Date: September 05, 2007

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name :

Signature :

ABSTRACT

STACKING SEQUENCE OPTIMIZATION OF A
COMPOSITE PRESSURE VESSEL
BY GENETIC ALGORITHM

Kutay, Halil
M.S., Department of Aerospace Engineering
Supervisor: Assoc. Prof. Dr. Altan Kayran

September 2007, 107 pages

Stacking sequence design is a combinatorial problem when the fiber orientations in each layer are restricted to certain angles. In addition, there often exist many optimal or near-optimal designs for the stacking sequence of a composite pressure vessel under different loading conditions. Genetic algorithms are quite well suited for finding the optimal designs for such a combinatorial problem. In this thesis, a genetic algorithm code is developed in Matlab, optimizing the stacking sequence of a composite pressure vessel subjected to internal and external pressures, axial load and body force due to rotation. For testing of the code and identification of the effects of optimization parameters, a problem, whose optimum solution is obvious, is defined and the optimum design is tried to be found by using the developed code. The results have shown that the code was quite successful in finding the best design. Afterwards, the code is used for the optimization of the stacking sequence of a composite pressure vessel under different loading conditions. Again the code has proven its reliability in finding the optimal designs. The developed genetic algorithm optimization code also has the infrastructure to be easily adapted to the solution of different combinatorial problems.

Keywords: genetic algorithms, composite, pressure vessel, stacking sequence.

ÖZ

KOMPOZİT BİR BASINÇLI KABIN KATMAN SIRASININ
GENETİK ALGORİTMA YÖNTEMİ İLE OPTİMİZASYONU

Kutay, Halil
Yüksek Lisans, Havacılık ve Uzay Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Altan Kayran

Eylül 2007, 107 sayfa

Fiber diziliş açıları belirli değerlerle kısıtlandığında, kompozit bir basınçlı kabın katman sırasının dizaynı bir kombinasyon problemi olarak ele alınabilir. Buna ek olarak, genellikle belirli yükler altındaki bir kompozit basınçlı kap için birçok optimum diziliş sırası dizaynı mevcuttur. Genetik algoritma bu tarz kombinasyon problemlerinin optimizasyonu için oldukça uygundur. Bu tezde, iç ve dış basınç, eksenel yük ve dönüden kaynaklanan kütlesel yüklere maruz kalan bir kompozit basınçlı kabın katman sırasını optimize eden bir genetik algoritma kodu Matlab ortamında geliştirilmiştir. Geliştirilen kodun testi ve optimizasyon parametrelerinin etkilerinin belirlenebilmesi için optimum çözümü aşikar olan bir dizayn problemi tanımlanmıştır. Elde edilen sonuçlar kodun en optimum dizaynı bulmakta oldukça başarılı olduğunu göstermiştir. Daha sonra geliştirilen kod, farklı yükler altındaki kompozit bir basınçlı kabın optimum katman sırasının bulunmasında kullanılmıştır. Yine kod optimum dizaynı bulma konusunda ne kadar güvenilir olduğunu ispatlamıştır. Geliştirilen genetik algoritma optimizasyon kodu farklı kombinasyon problemlerinin çözümü için de kolayca uyarlanabilecek bir yapıdadır.

Anahtar kelimeler: Genetik algoritma, kompozit, basınçlı kap, diziliş sırası.

To my wife

Aslı

In eternal gratitude

For the happy nest, sacrifice and love

ACKNOWLEDGMENTS

I express sincere appreciation to my advisor Assoc. Prof. Dr. Altan KAYRAN for his guidance, suggestions, comments and insight throughout the research. Again thanks to my advisor for his trust in me, for showing the right way which made this theses study possible.

To my peerless wife, Aslı, I offer eternal thanks for her love, unshakable faith in me and endless support. And thanks to my son, Metehan for being.

Finally, I want to express my eternal gratitude to my parents and brothers for their continuing endeavors starting from my first breath, which made all my achievements possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

N_{me}	Integer used in multiple elitist selection methods
Φ	Probability value used for parent selection
μ	Population size
α	Winding angle
ν	Poisson's ratio
E	Modulus of elasticity
G	Modulus of rigidity
$1,2,3$	Material coordinates
x,y,z	Rectangular coordinates
r,θ,z	Cylindrical coordinates
σ	Stresses
ε	Strains
γ	Shear strain
$\bar{U}$	Body force potential
ρ	Density of the material
ω	Rotational velocity
$F(r,\theta)$	Stress function
r_{int}, r_{ext}	Internal and external radius of the cylinder
P_{int}, P_{ext}	Internal and external pressures of the cylinder
b_{k-1}, b_k	Internal and external radius of the kth layer
q_{k-1}, q_k	Internal and external pressures of the kth layer
ξ_{ij}^k	Reduced strain coefficients of the kth layer

F_A	Axial force
u_r	Radial displacement
t	Thickness of the cylinder
R	Strength to stress ratio
$\sigma_{longitudinal,tensile}$	Longitudinal tensile strength
$\sigma_{longitudinal,compressive}$	Longitudinal compressive strength
$\sigma_{transverse,tensile}$	Transverse tensile strength
$\sigma_{transverse,compressive}$	Transverse compressive strength
σ_{shear}	Shear strength

CHAPTER 1

INTRODUCTION

In today's world, composite materials are taking a great role in aerospace structures. In addition to their high strength to weight ratio, they have the greatest advantage of tailoring the directional strength and stiffness of a material for a given loading environment [1]. Composite structures are typically made of laminates where thin layers (called plies) are stacked and held together with a matrix of a different material. Each ply has its greatest strength and stiffness along the direction of fibers. By changing the fiber orientation angles, the composite structures can be designed specifically for a given loading environment in order to have the maximum strength and stiffness properties at the high stress areas.

In the past decades, the designers have been unwilling to use composite structures in the aerospace applications where the strength to weight ratio is very important. Because, it was very difficult to estimate all the possible loading conditions that the structure will encounter during its whole service life, production cost was high and there was little information about the fatigue life or repair and maintenance procedures of composite structures. Boeing 777 is a good sample reflecting this idea. Only 9% by weight of this commercial transport aircraft consists of composite material [2].

However, as the studies on composite structures snowball, design and manufacturing capabilities are improving incredibly. The attack helicopter RAH-66 Comanche, built by Sikorsky Aircraft and the Boeing Co. proves that designers feel themselves more comfortable in using composites, because the helicopter is built almost entirely of composite materials. In addition to 20 – 30% savings in weight, the total construction cost is decreased considerably by utilizing advanced optimization and manufacturing techniques [3].

Composites are also widely used in space systems. Some of the most critical components of space systems are the propellant tanks and the helium gas bottles. Since early 1970s, due to their strength, weight and cost benefits, composite overwrapped pressure vessels (COPV) began to be used for such components instead of pressure vessels made up of high-strength metals such as steel, titanium and Inconel alloys [4]. As a result, the number of researches on COPV has been increasing. Thesken and his coworkers made a search on the mechanics and the stress rupture lifing of COPV [5]. Another study is made by Greene and his coworkers on the burst testing of Kevlar COPV [6]. It is clear that the future of composite materials is very bright.

1.1. Optimization of a Composite Structure

Design of a composite structure involves finding the optimum number of plies and ply angle orientations for a given loading condition in order to satisfy the desired objectives (e.g. minimum displacement, minimum weight) subject to the failure constraint. There have been many researches on the optimization of a composite structure. In early works, Foye has studied the stacking sequence optimization of a composite structure by a random search method under the strength and stiffness constraints [7]. Waddoups used the brute force method to find the optimum design [8]. In this method, all possible designs are evaluated in the design space.

Continuous optimization techniques were also used for the optimization of a composite structure. Schmit and Farshi have tried to optimize a composite structure with buckling constraints by using the ply thicknesses as continuous design variables [9], [10]. In these studies, buckling constraints were sequentially linearized with respect to the ply thicknesses. Pedersen have worked on the stacking sequence optimization of a balanced, symmetric and orthotropic angle-ply laminate for the maximum buckling load [11]. Ply orientations were treated as continuous design variables. The results show that the method has produced local optimum designs.

Branch and bound algorithm is another method used for the optimization of composite structures. Haftka and Walsh have tried to optimize the ply thicknesses in order to maximize the buckling load by using a branch and bound algorithm [12]. Todoroki and Hirano have developed a fractal branch and bound algorithm for the stacking sequence optimization of a composite delta wing for the buckling load maximization [13]. Similarly, Todoroki and Matsuzaki have developed a fractal branch and bound algorithm for the stacking sequence optimization of unsymmetrical laminates for the buckling load maximization [14]. Todoroki and his coworkers have worked on the stacking sequence optimization of a composite wing structure by using a modified fractal branch and bound algorithm [15]. The objective of the proposed problem was to minimize the bending and the torsion deformation of the structure.

Optimization techniques which use continuous design variables have several drawbacks. First, design and optimization of a composite structure includes highly nonlinear functions of the design variables (number of plies and orientation angles), hence linearization is required for solving the problem which increases the load of the solution method [3]. Second, there exist many optimal or near optimal designs, because the characteristic of a laminate performance depends on a smaller number of parameters when compared to the number of design variables [16]. Sequential linearization is used for solving non-linear problems, but the method often results in a local optimum design [8]. Another drawback is that the continuous optimization methods may produce local optimum or unfeasible designs when the design variables are rounded off [17]. Branch and bound algorithm is also not suitable for the composite optimization problems having large number of design variables, because the method is exponentially dependent on the number of design variables [16]. One of the most important drawback of the traditional methods is that these methods give only one optimum design and do not give alternative nearly optimum designs [16].

Due to the drawbacks mentioned in the previous paragraph, traditional methods such as gradient based continuous optimization methods or enumerative integer programming techniques are not well suited for the optimization of complex composite structures. Genetic algorithm methods are much more suitable for such problems due to their probabilistic characteristics. A genetic algorithm can reach any point of the design space and is not affected by the non-linearities; also its implementation is quite simple. There can be found wide variety of applications of genetic algorithms in this area. In early works, Callahan and Weeks used a GA to maximize strength and stiffness of a laminate under in-plane and flexural loads [18]. Nagendra, Haftka and Gürdal have made research on the stacking sequence optimization of a composite laminate under buckling and strength constraints [19], [20], [21]. Le Riche and Haftka have worked on the optimization of the composite structure for the buckling load maximization [22]. Harrison et al. [23] and Nagendra et al. [24] have studied the weight minimization of stiffened panels. Malott et al. [25], Gantovnik et al. [26] and Lin et al. [27] have performed researches on the optimal design of laminated composite sandwich panels. Walker and Smith have developed a technique for the multiobjective optimization of a composite structure by using finite element analysis [28]. Some other interesting studies on the stacking sequence optimization of composite laminates are made by Todoroki et al. [29], Liu et al. [30], Park et al. [31] and Soromekun et al. [32], [33].

Most of the works on this subject include the optimum design of composite plates. This work is focused on the development of an efficient genetic algorithm code for the optimization of composite pressure vessels instead of composite plates. In the proceeding sections of this chapter, brief information about the genetic algorithm is given. At the end of this chapter, the purpose of this work is explained.

1.2. Genetic Algorithm Method

Early implementation of the genetic search method is credited to Bremermann [34] and Rechenberg [35], although Holland's [36] work has formed the basis of most contemporary developments. Early applications of genetic algorithms to structural optimization are due to Goldberg and Samtani [37] and Hajela [38]. Genetic algorithms have since then been applied to numerous structural optimization problems (e.g. Furuya and Haftka [39], Powell and Skolnick [40], Shoenauer and Xanthakis [41], Watabe and Okina [42]).

A genetic algorithm is a guided random search technique that works on a population of designs. Each individual in the population represents a design (called chromosome), which is the stacking sequence in this context, coded in the form of a bit (called gene) string. The genetic algorithm begins with the random generation of an initial population of design alternatives. The creation of the initial population is a blind random search of the design space of the problem. After the generation of the initial population, each individual in the generation is evaluated, and a value of fitness is returned by a fitness function. The pool is sorted, with those having better fitness (representing better solutions to the problem) ranked at the top. "Better" in this context is relative, as initial solutions are all likely to be rather poor. Children are then created from the parents (members of the current population) by means of genetic operators to create a new population. These genetic operations are applied to individuals that are probabilistically selected from the parent population based on fitness. There are several well-defined parent selection methods; roulette wheel selection and tournament selection are the most popular methods. In this probabilistic selection process, better individuals are favored over inferior individuals. However, the best individual in the population is not necessarily selected and the worst individual in the population is not necessarily passed over. This is an advantage, since there is a possibility that the worst individual may guide the search in finding the best design. The next step is to choose the members of the new generation from the pool consisting of both the parents and the children. There are several selection methods. The selection methods used in this work is explained in **chapter 2** in detail.

The processes mentioned above ultimately result in the next generation population that is different from the initial generation. Generally the average fitness will be increased, since the better organisms from the first generation are selected for breeding. The entire process is repeated for the new generation. Each organism is evaluated and ranked according to their fitness value. Pairs (i.e. parents) are selected in order to create the child population and the members of the next generation are selected. This iterative process is repeated until a termination condition has been reached. Common terminating conditions are

- Fixed number of generations reached,
- Allocated budget (computation time/money) reached,
- An individual is found that satisfies minimum criteria,
- The highest ranking individual's fitness is reaching or has reached a plateau such that successive iterations do no longer produce better results,
- Manual inspection,
- Combinations of the above.

Genetic algorithms use objective function values (fitness), and do not need derivatives as some other optimization procedures do. Thus, they are applicable to the problems having discontinuous objective functions. They search several points of the design space at the same time. While a genetic algorithm explores a new design area, it also exploits the area of best designs found till then. There should be a balance between the exploitation and exploration character of the optimum search method based on genetic algorithm. A very exploitative genetic algorithm might get stuck to a local optimum solution and can not explore different design areas which might contain the best design. Similarly a genetic algorithm which explores too much might miss the best design in the search area due to poor exploitation. Genetic algorithms can rapidly locate good solutions, even for difficult search spaces when appropriately applied. They are computationally simple, but might be computationally expensive in accordance with the cost of the objective function computation. As the computers become faster, significance of the computational cost will decrease accordingly.

1.3. Purpose of the Work

The purpose of this work is to develop an efficient and reliable genetic algorithm code for the optimization of a composite pressure vessel under varying loading conditions. Two different optimization problems are defined. The objective of the first problem is to minimize the radial displacement at the outer surface of the vessel and the objective of the second problem is to minimize the weight of the vessel. Design variables are the ply orientation angles and the optimization problems are constrained to satisfy the required safety factor. In these problems vessel is subjected to internal and external pressures, axial load and body force due to rotation.

Identification of the effect of the genetic parameters (genetic operator probabilities, population size, parameters used in the selection methods …etc) on the efficiency and price of the optimization is another purpose. A test problem is designed and the developed code is adapted to the designed problem. This problem is designed for the identification of the effect of genetic parameters on the search.

The detailed description of the optimization problems for the composite pressure vessel is given in the first section of **chapter 2**. Afterwards, analytical derivation of the stresses and deformations of a layered composite pressure vessel is described. Then, the derived thick wall solution is compared with the thin wall solution. In section **2.5**, the failure theory used for the calculation of safety factor is explained.

Three different versions of a genetic algorithm code are developed in this study. There are small differences between these code versions. In **chapter 3**, representation of a design for each GA code is given; and the flowcharts of all three algorithms are described in detail. The first code (GAver1) is developed for testing the performance of the GA code. A test problem with a known solution is created; and many runs have been made by using GAver1 to identify the effect of the parameters (genetic operator probabilities, population size, parameters used in the selection methods …etc) on the efficiency and price of the optimization. The problem description and the results of GAver1 are given in **chapter 4**.

Code version 2 and 3 (called GAver2 and GAver3) are developed for the stacking sequence optimization of a composite pressure vessel. GAver2 is developed for the first optimization problem defined in section **2.1**; and GAver3 is developed for the second optimization problem defined in section **2.2**. These codes are collected, and a software called GACOM with a graphical user interface is developed in Matlab 7. In **chapter 5**, a brief description of GACOM is given first. In sections **5.2** and **5.3**, the results of GAver2 and GAver3 codes are given. These codes are experimented for the stacking sequence optimization of the layered pressure vessels under different load cases. Optimum solutions for the different load cases are determined, and comments are made on the optimum solutions by considering the individual load cases for which solutions are obtained. In **chapter 6**, conclusions are listed with regard to the overall work performed.

CHAPTER 2

OPTIMIZATION PROBLEMS
FOR A COMPOSITE PRESSURE VESSEL

In this chapter, the stacking sequence optimization problems for composite pressure vessels are explained in detail. There are two optimization problems with different objectives. The objective of the first problem is to minimize the radial displacement at the outer surface of the vessel; and the objective of the second problem is to minimize the weight of the vessel. In these problems, the vessel is subjected to internal and external pressures, axial load and body force due to rotation.

Detailed descriptions of the first and second optimization problems are given in section **2.1** and **2.2** respectively. In section **2.3**, the analytical formulation of the problem is given. Afterwards, the derived solution is compared with the thin-wall solution in section **2.4**. In section **2.5**, "3-D Quadratic Failure Theory" which is used to calculate the safety factor of the vessel is explained.

2.1. Optimization Problem for Minimum Radial Displacement

The objective of the first optimization problem is to minimize the radial displacement at the outer surface of the vessel. The design variables are the fiber orientation angles of the plies. The objective function is calculated in accordance with the fiber orientation angles and the material properties by using the equations given in section **2.3**. The optimization problem is constrained to satisfy the required safety factor. The formulation of the objective function is given in **2-1**. In the equation, n is the total number of plies, $R_{calculated}$ is the calculated safety factor and $R_{required}$ is the minimum required safety factor. As it can be seen from the equation that the objective function is penalized if the design does not satisfy the required safety factor criteria. Thus the search is forced to satisfy the safety factor constraint. In this problem, the thickness of the vessel is constant.

Hence, the search can not increase the number of plies if none of the possible stacking sequences satisfy the safety factor constraint. In that case, the search tries to increase the safety factor and decrease the radial displacement at the same time but of course the best design does not satisfy the constraint.

$$objective \quad function \quad = \begin{cases} u_r^n\left(b_n\right) & if \quad R_{calculated} >= R_{required} \\ u_r^n\left(b_n\right)\left(\dfrac{R_{required}}{R_{calculated}}\right)^2 & if \quad R_{calculated} < R_{required} \end{cases} \qquad \textbf{2-1}$$

2.2. Optimization Problem for Minimum Weight

The objective of the second optimization problem is to minimize the weight of the vessel. The design variables are again the fiber orientation angles of the plies. Similar to the first problem, the objective function is calculated by using the equations given in section **2.3**. The second optimization problem is also constrained to satisfy the required safety factor. The formulation of the objective function is given in **2-2**. The objective function is penalized similar to the first optimization problem if the design does not satisfy the required safety factor criteria. In the equation, ρ is the density of the composite, r_{int} and r_{ext} are the internal and the external radius of the vessel, $R_{calculated}$ is the calculated safety factor and $R_{required}$ is the minimum required safety factor. $R_{calculated}$ value should be greater than 1 in order to have a safe structure. Thus, the required safety factor value can not be smaller than 1.

$$objective\, function = \begin{cases} \rho\pi\left(r_{ext}^2 - r_{int}^2\right) & if \quad R_{calculated} >= R_{required} \\ \rho\pi\left(r_{ext}^2 - r_{int}^2\right)\left(\dfrac{R_{required}}{R_{calculated}}\right)^2 & if \quad R_{calculated} < R_{required} \end{cases} \qquad \textbf{2-2}$$

2.3. Analytical Formulation for Thick-Wall Solution

The analytical formulation of the deformation and stress analysis of a composite pressure vessel is given in this section [43].

The classical lamination theory is used to calculate the stresses and strains in each lamina. The assumptions of this theory can be summarized as follows:

- The laminate is thin and wide.
- All laminas are homogeneous and linearly elastic.
- Inter-ply bonding between the laminas is perfect.
- Strain distribution across the thickness is linear.

The elastic constants of an angle-ply lamina are calculated by using the following equations in Cartesian coordinates. In the equations, α is the fiber orientation angle with the positive x-axis.

$$\frac{1}{E_{xx}} = \frac{\cos^4\alpha}{E_{11}} + \frac{\sin^4\alpha}{E_{22}} + \left(\frac{1}{G_{12}} - 2\frac{\nu_{12}}{E_{11}}\right)\cos^2\alpha\sin^2\alpha \qquad \textbf{2-3}$$

$$\frac{1}{E_{yy}} = \frac{\sin^4\alpha}{E_{11}} + \frac{\cos^4\alpha}{E_{22}} + \left(\frac{1}{G_{12}} - 2\frac{\nu_{12}}{E_{11}}\right)\cos^2\alpha\sin^2\alpha \qquad \textbf{2-4}$$

$$\frac{1}{G_{xy}} = \frac{1}{E_{11}} + 2\frac{\nu_{12}}{E_{11}} + \frac{1}{E_{22}} - \left(\frac{1}{E_{11}} + 2\frac{\nu_{12}}{E_{11}} + \frac{1}{E_{22}} - \frac{1}{G_{12}}\right)\cos^2 2\alpha \qquad \textbf{2-5}$$

$$\nu_{xy} = E_{xx}\left[\frac{\nu_{12}}{E_{11}} - \left(\frac{1}{E_{11}} + 2\frac{\nu_{12}}{E_{11}} + \frac{1}{E_{22}} - \frac{1}{G_{12}}\right)\cos^2\alpha\sin^2\alpha\right] \qquad \textbf{2-6}$$

$$\nu_{yx} = \frac{E_{yy}}{E_{xx}}\nu_{xy} \qquad \textbf{2-7}$$

$$E_{zz} = E_{33} = E_{22} \qquad \textbf{2-8}$$

$$G_{yz} = G_{xz} = G_{xy} \qquad \textbf{2-9}$$

The elastic constants are converted from Cartesian coordinates x, y and z to the corresponding cylindrical coordinates z, θ and r by using the equations given in **2-10** and **2-11** (see **figure 2-1** for the coordinate representations).

$$\begin{aligned} E_{zz} &= E_{xx} \\ E_{rr} &= E_{22} \\ E_{\theta\theta} &= E_{yy} \end{aligned} \qquad \textbf{2-10}$$

$$\nu_{\theta r} = \frac{E_{yy}}{2G_{xy}} - 1 \quad , \quad \nu_{r\theta} = \frac{E_{22}}{E_{yy}} \nu_{\theta r}$$

$$\nu_{zr} = \frac{E_{xx}}{2G_{xy}} - 1 \quad , \quad \nu_{rz} = \frac{E_{22}}{E_{xx}} \nu_{zr} \qquad \textbf{2-11}$$

$$\nu_{z\theta} = \nu_{xy} \quad , \quad \nu_{\theta z} = \nu_{yx}$$

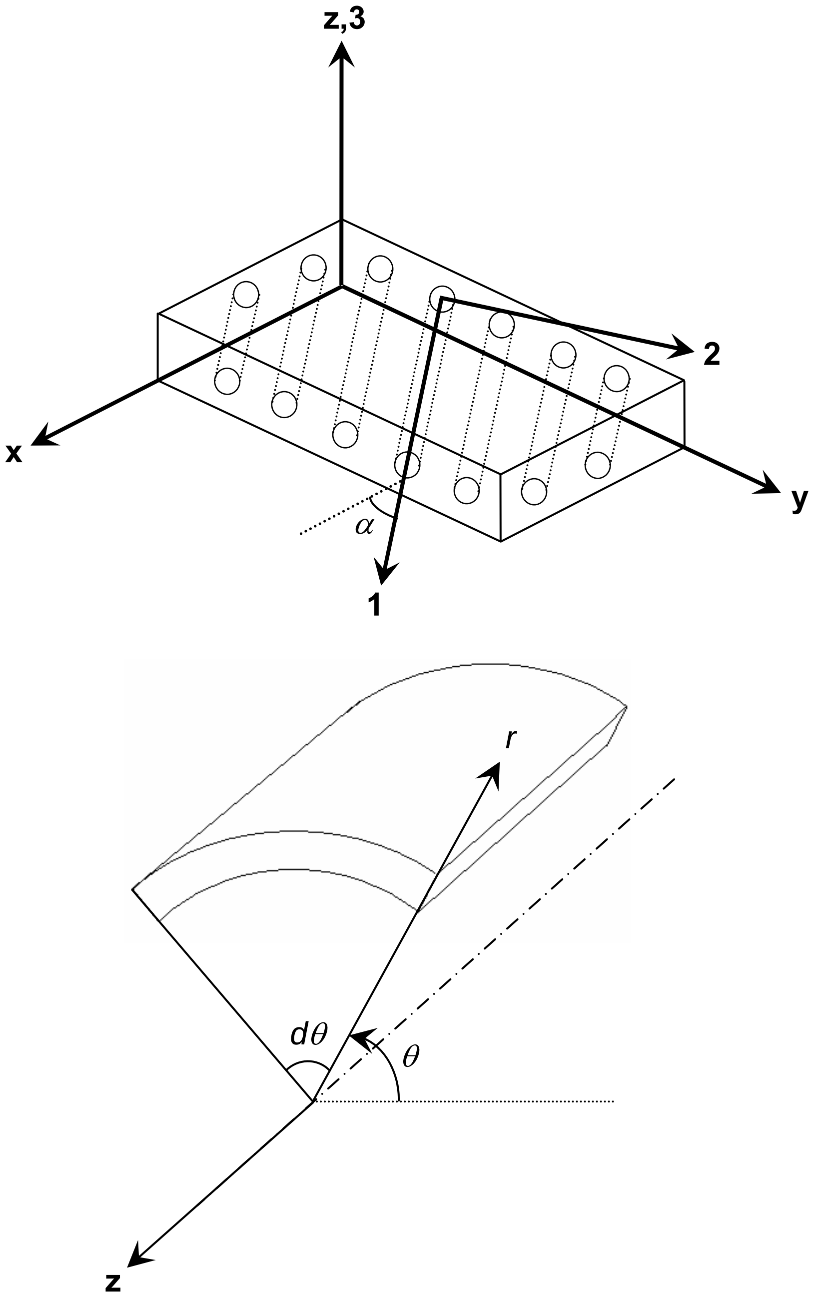


Figure 2-1 Coordinate representations

The analytical formulation for the thick-walled composite pressure vessel is based on the following assumptions:

- $\pm\alpha$ angle lay ups constitute the pressure vessel.
- The pressure vessel is cylindrically orthotropic.
- In the analysis of the plain strain problem, only small strains are considered.

The inverse form of the Hooke's law in cylindrical coordinates is given in equation **2-12**.

$$\begin{bmatrix} \varepsilon_{rr} \\ \varepsilon_{\theta\theta} \\ \varepsilon_{zz} \end{bmatrix} = \begin{bmatrix} a_{ij} \end{bmatrix} \begin{bmatrix} \sigma_{rr} \\ \sigma_{\theta\theta} \\ \sigma_{zz} \end{bmatrix} = \begin{bmatrix} \frac{1}{E_{rr}} & \frac{-\nu_{r\theta}}{E_{rr}} & \frac{-\nu_{rz}}{E_{rr}} \\ \frac{-\nu_{r\theta}}{E_{rr}} & \frac{1}{E_{\theta\theta}} & \frac{-\nu_{\theta z}}{E_{\theta\theta}} \\ \frac{-\nu_{rz}}{E_{rr}} & \frac{-\nu_{\theta z}}{E_{\theta\theta}} & \frac{1}{E_{zz}} \end{bmatrix} \begin{bmatrix} \sigma_{rr} \\ \sigma_{\theta\theta} \\ \sigma_{zz} \end{bmatrix} \qquad \textbf{2-12}$$

In the formulation, axial strain of the vessel is assumed to be constant. For an axisymmetric body having cylindrical anisotropy, the equilibrium equations, the equations of generalized Hooke's law (by disregarding σ_{zz}) and the strain-displacement relations can be written as:

$$\begin{aligned} &\frac{\partial \sigma_{rr}}{\partial r} + \frac{1}{r}\frac{\partial \sigma_{r\theta}}{\partial \theta} + \frac{\sigma_{rr} - \sigma_{\theta\theta}}{r} - \frac{\partial \bar{U}}{\partial r} = 0 \\ &\frac{\partial \sigma_{r\theta}}{\partial r} + \frac{1}{r}\frac{\partial \sigma_{\theta\theta}}{\partial \theta} + 2\frac{\sigma_{r\theta}}{r} - \frac{1}{r}\frac{\partial \bar{U}}{\partial \theta} = 0 \end{aligned} \qquad \textbf{2-13}$$

$$\begin{aligned} \varepsilon_{rr} &= \frac{1}{E_{rr}}\sigma_{rr} - \frac{\nu_{r\theta}}{E_{rr}}\sigma_{\theta\theta} \\ \varepsilon_{\theta\theta} &= -\frac{\nu_{r\theta}}{E_{rr}}\sigma_{rr} + \frac{1}{E_{\theta\theta}}\sigma_{\theta\theta} \\ \gamma_{r\theta} &= \frac{1}{G_{r\theta}}\sigma_{r\theta} \end{aligned} \qquad \textbf{2-14}$$

$$\begin{aligned} \varepsilon_{rr} &= \frac{\partial u_r}{\partial r} \\ \varepsilon_{\theta\theta} &= \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{u_r}{r} \\ \gamma_{r\theta} &= \frac{1}{r}\frac{\partial u_r}{\partial \theta} + \frac{\partial u_\theta}{\partial \theta} - \frac{u_\theta}{r} \end{aligned} \qquad \textbf{2-15}$$

In the equations, $\bar{U}$ refers to the body force potential.

The compatibility equation can be obtained by eliminating the displacements in the strain-displacement relations.

$$\frac{\partial^2 \varepsilon_{rr}}{\partial \theta^2} + r\frac{\partial^2 \left(r\varepsilon_{\theta\theta}\right)}{\partial r^2} - \frac{\partial^2 \left(r\gamma_{r\theta}\right)}{\partial r \partial \theta} - r\frac{\partial \varepsilon_{rr}}{\partial r} = 0 \qquad \textbf{2-16}$$

By introducing a stress function $F\left(r,\theta\right)$, the stresses can be rewritten as:

$$\begin{aligned} \sigma_{rr} &= \frac{1}{r}\frac{\partial F}{\partial r} + \frac{1}{r^2}\frac{\partial^2 F}{\partial \theta^2} + \bar{U} \\ \sigma_{\theta\theta} &= \frac{\partial^2 F}{\partial r^2} + \bar{U} \\ \sigma_{r\theta} &= -\frac{\partial^2}{\partial r \partial \theta}\left(\frac{F}{r}\right) \end{aligned} \qquad \textbf{2-17}$$

For a rotating disc, the body force potential is given by **2-18**, where ω is the angular velocity, ρ is the density of the material and r is the distance from the center of the disc.

$$\bar{U} = -\frac{\rho\omega^2}{2}r^2 \qquad \textbf{2-18}$$

By using the equations **2-14**, **2-16** and **2-17** and the fact that the problem is axisymmetric, the differential equation for kth layer can be obtained as follows:

$$\frac{1}{E_{\theta\theta}^k}r^4\frac{\partial^4 F}{\partial r^4} + \frac{2}{E_{\theta\theta}^k}r^3\frac{\partial^3 F}{\partial r^3} - \frac{1}{E_{rr}^k}r^2\frac{\partial^2 F}{\partial r^2} + \frac{1}{E_{rr}^k}r\frac{\partial F}{\partial r} = \left(\frac{3}{E_{\theta\theta}^k} - \frac{2\nu_{r\theta}^k + 1}{E_{rr}^k}\right)\rho\omega^2 r^4 \qquad \textbf{2-19}$$

The stress function $F\left(r\right)$ can be found by solving the above differential equation as:

$$F\left(r\right) = A^k + B^k r^2 + C^k r^{1+gl_k} + D^k r^{1-gl_k} + \frac{\rho_k\omega^2}{2}\left(3 - 2\nu_{\theta r}^k - gl_k^2\right)\frac{E_{rr}^k}{36E_{rr}^k - 4E_{\theta\theta}^k}r^4 \qquad \textbf{2-20}$$

where

$$gl_k = \sqrt{\frac{E^k_{\theta\theta}}{E^k_{rr}}} \qquad \textbf{2-21}$$

The stresses can be found by inserting equations **2-18** and **2-20** into equation **2-17**.

$$\sigma^k_{rr} = 2B^k + C^k(1+g_k)r^{g_k-1} + D^k(1-g_k)r^{-g_k-1} - \rho_k\omega^2\left(\frac{3+\nu^k_{\theta r}}{9-gl^2_k}\right)r^2 \qquad \textbf{2-22}$$

$$\sigma^k_{\theta\theta} = 2B^k + C^k(1+g_k)r^{g_k-1} + D^k(1-g_k)r^{-g_k-1} - \rho_k\omega^2\left(\frac{3\nu^k_{\theta r}+gl^2_k}{9-gl^2_k}\right)r^2 \qquad \textbf{2-23}$$

$$\sigma^k_{r\theta} = 0 \qquad \textbf{2-24}$$

The interface conditions between two adjacent layers can be written as:

$$\left.\begin{aligned} \sigma^{k-1}_{rr} = \sigma^k_{rr} &= q_{k-1} \\ u^{k-1}_r &= u^k_r \end{aligned}\right\} \quad for \quad r = b_{k-1} \qquad \textbf{2-25}$$

In the above equation, b_{k-1} and q_{k-1} refer to the internal radius and the pressure of the kth layer respectively. Similarly, b_k and q_k refer to the external ones (see **figure 2-2**).

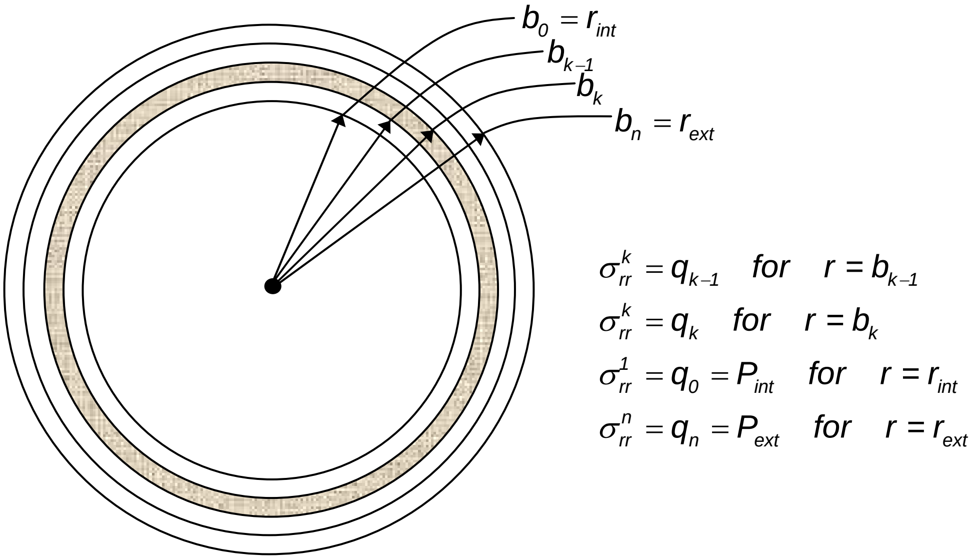


Figure 2-2 Coordinate representations

By applying the interface conditions, the coefficients of the stress function can be found as follows:

$$A^k = 0 \qquad B^k = 0$$

$$C^k = \frac{1}{1+g_k}\left[\begin{array}{l} \dfrac{b_k b_{k-1}^{-g_k}}{c_k^{g_k} - c_k^{-g_k}} q_k - \dfrac{\left[b_k^{g_k} b_{k-1}^{1-2g_k} + b_{k-1}^{1-g_k}\left(c_k^{g_k} - c_k^{-g_k}\right)\right]}{c_k^{g_k} - c_k^{-g_k}} q_{k-1} \\ +\rho_k \omega^2 \dfrac{3+\nu_{\theta r}^k}{9-gl_k^2}\left[\dfrac{b_k^{g_k} b_{k-1}^{3-2g_k} + b_k^3 b_{k-1}^{-g_k}}{c_k^{g_k} - c_k^{-g_k}} + b_{k-1}^{3-g_k}\right] \end{array}\right] \qquad \textbf{2-26}$$

$$D^k = \frac{1}{1-g_k}\left[\begin{array}{l} \dfrac{\left(q_{k-1} b_k^{g_k-1} - q_k b_{k-1}^{g_k-1}\right) b_k b_{k-1}}{c_k^{g_k} - c_k^{-g_k}} \\ -\rho_k \omega^2 \dfrac{3+\nu_{\theta r}^k}{9-gl_k^2} \dfrac{\left(b_{k-1}^3 b_k^{g_k} - b_k^3 b_{k-1}^{g_k}\right)}{c_k^{g_k} - c_k^{-g_k}} \end{array}\right]$$

where

$$c_k = \frac{b_{k-1}}{b_k} \quad and \quad g_k = \sqrt{\frac{\xi_{11}^k}{\xi_{22}^k}} \qquad \textbf{2-27}$$

The reduced strain coefficient ξ_{ij}^k proposed by Lekhnitskii [44] is given as:

$$\xi_{ij}^k = a_{ij}^k - \frac{a_{i3}^k a_{j3}^k}{a_{33}^k} \quad for \quad i,j = 1,2 \qquad \textbf{2-28}$$

a_{ij}^k in equation **2-28** is the corresponding element of the compliance matrix for k[th] layer (see equation **2-12**).

By inserting the coefficients into the equations **2-22** and **2-23**, the stresses can be found as:

$$\begin{aligned} \sigma_{rr}^k = {} & \rho_k \omega^2 b_k^2 \left(\frac{3+\nu_{\theta r}^k}{9-gl_k^2}\right)\left[\begin{array}{l} \left(\dfrac{1-c_k^{g_k+3}}{1-c_k^{2g_k}}\right)\left(\dfrac{r}{b_k}\right)^{g_k-1} \\ +\left(\dfrac{1-c_k^{g_k-3}}{1-c_k^{2g_k}}\right) c_k^{g_k+3}\left(\dfrac{b_k}{r}\right)^{g_k+1} - \left(\dfrac{r}{b_k}\right)^2 \end{array}\right] \\ & + \left(\frac{q_{k-1} c_k^{g_k+1}}{1-c_k^{2g_k}}\right)\left[\left(\frac{r}{b_k}\right)^{g_k-1} - \left(\frac{b_k}{r}\right)^{g_k+1}\right] \\ & + \left(\frac{q_k}{1-c_k^{2g_k}}\right)\left[c_k^{2g_k}\left(\frac{b_k}{r}\right)^{g_k+1} - \left(\frac{r}{b_k}\right)^{g_k-1}\right] \end{aligned} \qquad \textbf{2-29}$$

$$\sigma_{\theta\theta}^{k}=\left(\frac{\rho_k\omega^2 b_k^2}{9-gl_k^2}\right)\left\{\begin{array}{l}\left(3+\nu_{\theta r}^{k}\right)g_k\left[\begin{array}{l}\left(\dfrac{1-c_k^{g_k+3}}{1-c_k^{2g_k}}\right)\left(\dfrac{r}{b_k}\right)^{g_k-1}\\-\left(\dfrac{1-c_k^{g_k-3}}{1-c_k^{2g_k}}\right)c_k^{g_k+3}\left(\dfrac{b_k}{r}\right)^{g_k+1}\end{array}\right]\\-\left(gl_k^2+3\nu_{\theta r}^{k}\right)\left(\dfrac{r}{b_k}\right)^2\end{array}\right\}$$
$$+\left(\frac{q_{k-1}c_k^{g_k+1}g_k}{1-c_k^{2g_k}}\right)\left[\left(\frac{r}{b_k}\right)^{g_k-1}+\left(\frac{b_k}{r}\right)^{g_k+1}\right]$$
$$-\left(\frac{q_k g_k}{1-c_k^{2g_k}}\right)\left[\left(\frac{r}{b_k}\right)^{g_k-1}+c_k^{2g_k}\left(\frac{b_k}{r}\right)^{g_k+1}\right] \qquad \textbf{2-30}$$

By using equation **2-12**, σ_{zz}^{k} can be found as follows:

$$\sigma_{zz}^{k}=\frac{\varepsilon_{zz}^{0}}{a_{33}^{k}}-\frac{a_{13}^{k}\sigma_{rr}^{k}+a_{23}^{k}\sigma_{\theta\theta}^{k}}{a_{33}^{k}} \qquad \textbf{2-31}$$

where ε_{zz}^{0} is the axial deformation of all layers.

The axial equilibrium relation for closed end cylinder can be given as:

$$\sum_{k=1}^{n}2\pi\int_{b_{k-1}}^{b_k}\sigma_{zz}^{k}r\,dr=\pi\left(P_{int}r_{int}^2-P_{ext}r_{ext}^2\right)+F_A \qquad \textbf{2-32}$$

where F_A is the axial force.

ε_{zz}^{0} can be determined by using equations **2-29**, **2-30**, **2-31** and **2-32** as:

$$\varepsilon_{zz}=\frac{1}{\Delta}\left[\left(P_{int}r_{int}^2-P_{ext}r_{ext}^2\right)+\frac{F_A}{\pi}+\sum_{k=1}^{n}\left(q_{k-1}\delta_k+q_k\mu_k+\omega^2\psi_k\right)\right] \qquad \textbf{2-33}$$

where

$$\Delta = \sum_{k=1}^{n} \frac{b_k^2 - b_{k-1}^2}{a_{33}^k} \tag{2-34}$$

$$\delta_k = \frac{2c_k^{g_k+1}}{a_{33}^k\left(1-c_k^{2g_k}\right)} \left\{ \begin{array}{l} \dfrac{\left(a_{13}^k + g_k a_{23}^k\right)\left(b_k^2 - b_k^{1-g_k} b_{k-1}^{1+g_k}\right)}{1+g_k} \\ -\dfrac{\left(a_{13}^k - g_k a_{23}^k\right)\left(b_k^2 - b_k^{1+g_k} b_{k-1}^{1-g_k}\right)}{1-g_k} \end{array} \right\} \tag{2-35}$$

$$\mu_k = \frac{2}{a_{33}^k\left(1-c_k^{2g_k}\right)} \left\{ \begin{array}{l} \dfrac{-1}{1+g_k}\left(a_{13}^k + g_k a_{23}^k\right)\left(b_k^2 - b_k^{1-g_k} b_{k-1}^{1+g_k}\right) \\ +\dfrac{c_k^{2g_k}}{1-g_k}\left(a_{13}^k - g_k a_{23}^k\right)\left(b_k^2 - b_k^{1+g_k} b_{k-1}^{1-g_k}\right) \end{array} \right\} \tag{2-36}$$

$$\begin{aligned} \psi_k = \frac{-2\rho_k b_k^2\left(3+\upsilon_{\theta r}^k\right)}{9-gl_k^2} & \left\{ \left(\frac{a_{13}^k}{a_{33}^k} + \frac{a_{23}^k}{a_{33}^k} g_k\right) \left(\frac{\left(1-c_k^{g_k+3}\right)\left(b_k^{-g_k+1} b_{k-1}^{g_k+1} - b_k^2\right)}{\left(1-c_k^{2g_k}\right)\left(1+g_k\right)}\right) \right. \\ & + \left(\frac{a_{13}^k}{a_{33}^k} - \frac{a_{23}^k}{a_{33}^k} g_k\right) \left(\frac{\left(1-c_k^{g_k-3}\right) c_k^{g_k+3}\left(b_k^{g_k+1} b_{k-1}^{1-g_k} - b_k^2\right)}{\left(1-c_k^{2g_k}\right)\left(1-g_k\right)}\right) \\ & \left. + \left(\frac{a_{13}^k}{a_{33}^k} + \frac{a_{23}^k}{a_{33}^k} \frac{\left(gl_k^2 + 3\upsilon_{\theta r}^k\right)}{\left(3+\upsilon_{\theta r}^k\right)}\right) \left(\frac{1}{4b_k^2}\right)\left(b_k^4 - b_{k-1}^4\right) \right\} \end{aligned} \tag{2-37}$$

The displacements can be determined as follows:

$$\begin{aligned} u_r^k &= r\varepsilon_{rr} = r\left(a_{11}^k \sigma_{rr}^k + a_{12}^k \sigma_{\theta\theta}^k + a_{13}^k \sigma_{zz}^k\right) \\ &= r\left(a_{11}^k \sigma_{rr}^k + a_{12}^k \sigma_{\theta\theta}^k + a_{13}^k \left(\frac{\varepsilon_{zz}^0}{a_{33}^k} - \frac{a_{13}^k \sigma_{rr}^k + a_{23}^k \sigma_{\theta\theta}^k}{a_{33}^k}\right)\right) \\ &= r\left(\xi_{11}^k \sigma_{rr}^k + \xi_{12}^k \sigma_{\theta\theta}^k + \frac{a_{13}^k}{a_{33}^k} \varepsilon_{zz}^0\right) \end{aligned} \tag{2-38}$$

$$u_\theta^k = 0 \tag{2-39}$$

$$u_z^k = z\varepsilon_{zz}^0 \tag{2-40}$$

u_r^k can be found by inserting the equations **2-29**, **2-30**, and **2-33** into equation **2-38**. By using the derived expression for u_r^k and the interface condition:

$$u_r^k\left(b_k\right)=u_r^{k+1}\left(b_k\right) \tag{2-41}$$

the equation for the interface pressures, "q 's" can be obtained for each interface as:

$$\begin{aligned}&\phi_k q_{k+1}+\varphi_k q_k+\eta_k q_{k-1}+\omega^2\lambda_k-\left(\frac{\upsilon_{z\theta}^k-\upsilon_{z\theta}^{k+1}}{\Delta}\right)\left(\sum_{i=1}^{n}\left(q_{i-1}\delta_i+q_i\mu_i+\omega^2\psi_i\right)\right)\\&=\left(\frac{\upsilon_{z\theta}^k-\upsilon_{z\theta}^{k+1}}{\Delta}\right)\left[\left(P_{int}r_{int}^2-P_{ext}r_{ext}^2\right)+\frac{F_A}{\pi}\right]\end{aligned} \tag{2-42}$$

where

$$\phi_k=\frac{2g_{k+1}\xi_{22}^{k+1}c_{k+1}^{g_{k+1}-1}}{1-c_{k+1}^{2g_{k+1}}} \tag{2-43}$$

$$\varphi_k=-\xi_{12}^k-\frac{g_k\xi_{22}^k\left(1+c_k^{2g_k}\right)}{1-c_k^{2g_k}}+\xi_{12}^{k+1}-\frac{g_{k+1}\xi_{22}^{k+1}\left(1+c_{k+1}^{2g_{k+1}}\right)}{1-c_{k+1}^{2g_{k+1}}} \tag{2-44}$$

$$\eta_k=\frac{2g_k\xi_{22}^k c_k^{g_k+1}}{1-c_k^{2g_k}} \tag{2-45}$$

$$\begin{aligned}\lambda_k=&\,b_k^2\left\{\frac{\rho_k\xi_{22}^k}{9-gl_k^2}\left[g_k\left(3+\upsilon_{\theta r}^k\right)\left(\frac{1-2c_k^{g_k+3}+c_k^{2g_k}}{1-c_k^{2g_k}}\right)-\left(gl_k^2+3\upsilon_{\theta r}^k\right)\right]\right\}\\&-b_{k+1}^2\left\{\frac{\rho_{k+1}\xi_{22}^{k+1}}{9-gl_{k+1}^2}\left[\begin{array}{l}g_{k+1}\left(3+\upsilon_{\theta r}^{k+1}\right)\left(\dfrac{2c_{k+1}^{g_{k+1}-1}-c_{k+1}^{2g_{k+1}+2}c_{k+1}^2}{1-c_{k+1}^{2g_{k+1}}}\right)\\-\left(gl_{k+1}^2+3\upsilon_{\theta r}^{k+1}\right)c_{k+1}^2\end{array}\right]\right\}\end{aligned} \tag{2-46}$$

All the interface pressures can be determined by solving the set of equations (equation **2-42** for each interface) and the boundary conditions given in **figure 2-2**. After determining the interface pressures, all the other stresses, strains and displacements can be calculated by using the corresponding equations given in this section.

2.4. Comparison of Thick-Wall and Thin-Wall Solution

In this part, the results of the derived thick-wall solution will be compared with the thin-wall solution to establish the reliability of the derived solution. The equations of $\sigma_{\theta\theta}$ and σ_{zz} for thin-wall solution are given in equation **2-47** and **2-48** respectively.

$$\sigma_{\theta\theta} = \frac{Pr}{t} \qquad \textbf{2-47}$$

$$\sigma_{zz} = \frac{Pr}{2t} + \frac{F_A}{2\pi rt} \qquad \textbf{2-48}$$

where F_A is axial force, P is pressure, r is the radius and t is the thickness of the pressure vessel.

In the runs the used material is T300/N5208. The material properties are given in **table 2-1**.

Table 2-1 Material properties for T300/N5208

	T300/N5208
E_{11} (GPa)	181
E_{22} (GPa)	10.3
G_{12} (GPa)	7.17
ν	0.28
ρ (kg/m^3)	1600

Runs are made for three different cases. In the first case, only internal pressure of $0.6 \quad MPa$ acts on the pressure vessel and the fiber orientation angle, α is 90^o. The resultant graphs are given in **figure 2-3** and **2-4**. In the second case, again there is only internal pressure, but α is 45^o (see **figure 2-5** and **2-6** for results). In the last case, $50 \quad kN$ of axial force acts on the pressure vessel in addition to the internal pressure; and α is 45^o. The results are given in **figure 2-7** and **2-8**.

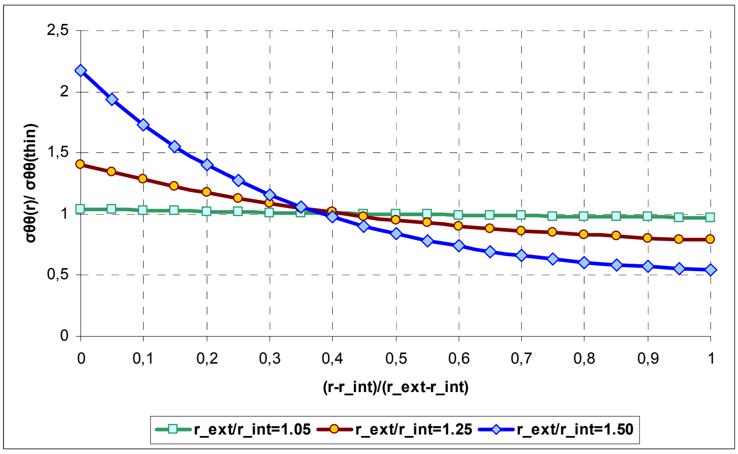


Figure 2-3 $\sigma_{\theta\theta}$ **graph (**$P_{int} = 0.6Mpa$**,** $\alpha = 90^0$**)**

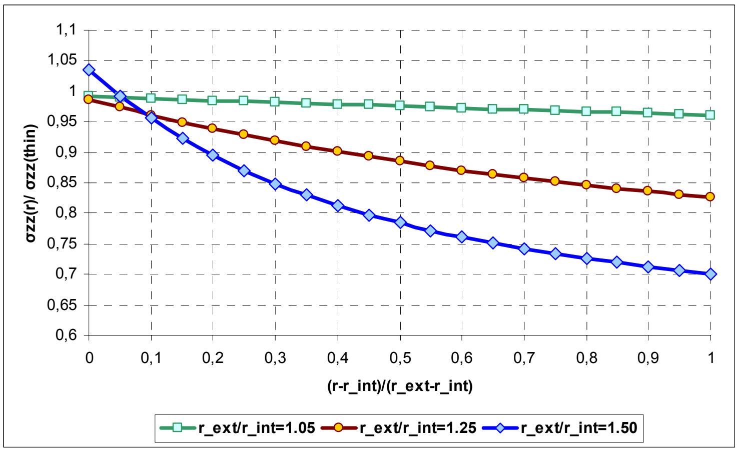


Figure 2-4 σ_{zz} **graph (**$P_{int} = 0.6Mpa$**,** $\alpha = 90^0$**)**

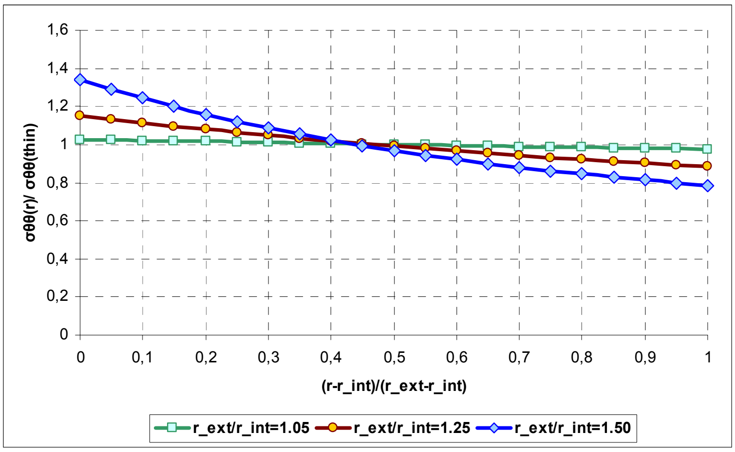


Figure 2-5 $\sigma_{\theta\theta}$ **graph (**$P_{int} = 0.6Mpa$**,** $\alpha = 45^0$**)**

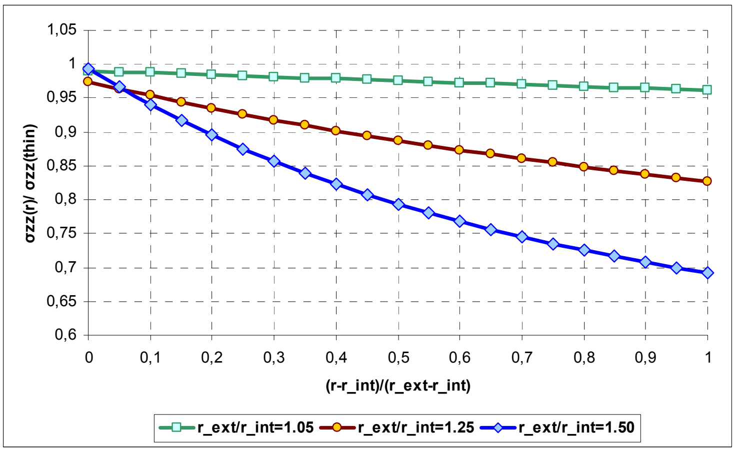


Figure 2-6 σ_{zz} **graph (** $P_{int} = 0.6Mpa$, $\alpha = 45^0$ **)**

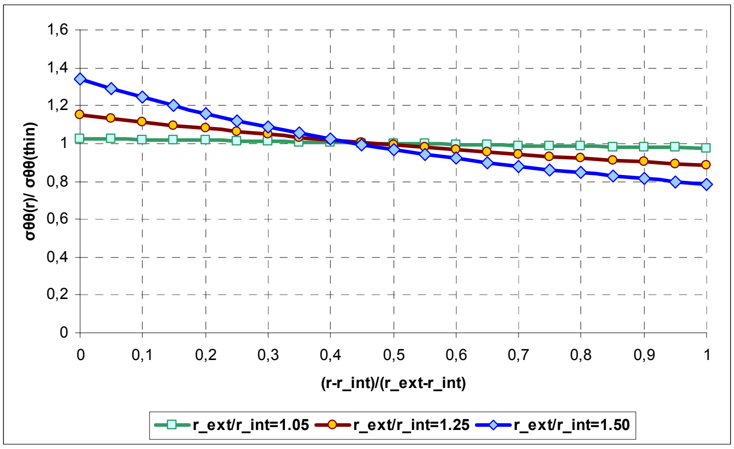


Figure 2-7 $\sigma_{\theta\theta}$ **graph (** $F_A = 50kN$, $P_{int} = 0.6Mpa$, $\alpha = 90^0$ **)**

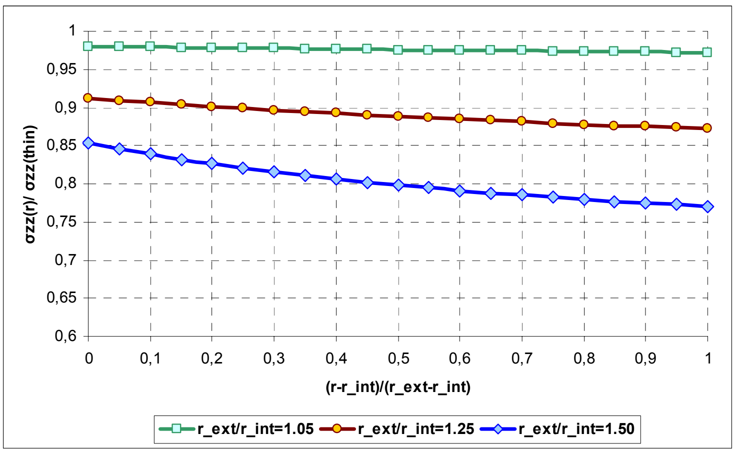


Figure 2-8 σ_{zz} **graph (** $F_A = 50kN$, $P_{int} = 0.6Mpa$, $\alpha = 90^0$ **)**

For different loading conditions and fiber orientation angles, it is seen from the graphs that as the thickness of the vessel decreases, the results of thick-wall and thin-wall solutions get closer. But as the thickness increases, the accuracy of the thin-wall solution decreases and as a result the difference between two solutions increases.

2.5. 3-D Quadratic Failure Theory

"3-D Quadratic Failure Theory" is used in failure analysis [45]. Failure criterion is given as:

$$\left\{F_{ij}\sigma_i\sigma_j\right\}R^2+\left\{F_i\sigma_i\right\}R-1=0$$

$$\left\{\begin{matrix} F_{11}^k\left(\sigma_{11}^k\right)^2+F_{33}^k\left\{\left(\sigma_{22}^k\right)^2+\left(\sigma_{33}^k\right)^2\right\} \\ +F_{ss}^k\left(\sigma_{12}^k\right)^2+2F_{13}^k\left(\sigma_{22}^k+\sigma_{33}^k\right)\sigma_{11}^k \\ +2F_{23}^k\sigma_{22}^k\sigma_{33}^k \end{matrix}\right\}R^2+\left\{\begin{matrix} F_1^k\sigma_{11}^k \\ +F_3^k\left(\sigma_{22}^k+\sigma_{33}^k\right) \end{matrix}\right\}R-1=0 \qquad \textbf{2-49}$$

where

$$\begin{aligned}
&\left[\sigma\right]_{max}=R\left[\sigma\right]_{applied} \\
&F_{11}^k=\frac{1}{\sigma_{longitudinal,tensile}^k\sigma_{longitudinal,compressive}^k} \\
&F_{33}^k=\frac{1}{\sigma_{transverse,tensile}^k\sigma_{transverse,compressive}^k} \\
&F_1^k=\frac{1}{\sigma_{longitudinal,tensile}^k}-\frac{1}{\sigma_{longitudinal,compressive}^k} \\
&F_3^k=\frac{1}{\sigma_{transverse,tensile}^k}-\frac{1}{\sigma_{transverse,compressive}^k}
\end{aligned} \qquad \textbf{2-50}$$

$$F_{ss}^k=\frac{1}{\left(\sigma_{shear}^k\right)^2} \qquad F_{13}^k=-0.5\left(F_{11}^kF_{33}^k\right)^{\frac{1}{2}} \qquad F_{23}^k=-0.5F_{33}^k$$

R is the strength to stress ratio in above equation. The positive root obtained by solving equation **2-49** under the defined loading condition is the safety factor of the system. Strength to stress ratio R should be greater than "**1**" for the system to be safe. GACOM gives R in the results text file.

CHAPTER 3

DEVELOPED GENETIC ALGORITHM CODE

In this chapter, the developed GA codes for different type of optimization problems are explained in detail. Actually there are not major differences between the versions of the code. Mainly, the GA code is developed for the optimization of composite pressure vessels under various loading conditions. But it is also aimed that the developed code can be adapted to different type of optimization problems with small modifications.

In section **3.1**, code version 1 (called GA*ver*1) which is developed for testing the performance of the GA code is described. The representation of a design is given first. Then steps of the algorithm are explained in detail.

Description of code version 2 (called GA*ver*2) is given in section **3.2**. This version is developed for the stacking sequence optimization of a composite pressure vessel. The objective is to minimize the radial displacement of a pressure vessel under defined loading conditions. Firstly, coding of a design is described. Then the differences and modifications made to GA*ver*1 are identified.

In section **3.3**, code version 3 (called GA*ver*3) is explained. Similar to GA*ver*2, GA*ver*3 is developed for the stacking sequence optimization of a composite pressure vessel, but the objective is to minimize the weight of the vessel. In this section, coding of a design and the modifications made to GA*ver*2 are given in detail.

3.1. GA Code version 1 (GA*ver*1) for Test Problem

A problem is created to test the performance of the GA code. There is a matrix of which elements consist of integers from 1 to 10. The objective of the problem is to minimize the product of the elements chosen from each column of the matrix. The detailed explanation of the problem is given in **chapter 4**.

Each gene is represented by an integer value from 1 to 10, which defines the row number in a column as seen in **figure 3-1**. Chromosome length is equal to the number of columns of the matrix. An example of a decoded design is given in **figure 3-2**.

$$
\begin{array}{cc}
1 & \rightarrow \\
2 & \rightarrow \\
3 & \rightarrow \\
4 & \rightarrow \\
5 & \rightarrow \\
6 & \rightarrow \\
7 & \rightarrow \\
8 & \rightarrow \\
9 & \rightarrow \\
10 & \rightarrow
\end{array}
\begin{bmatrix} 3 \\ 5 \\ 4 \\ 1 \\ 6 \\ 2 \\ 8 \\ 9 \\ 7 \\ 10 \end{bmatrix}
$$

Figure 3-1 Code key for GA*ver*1

$$
\begin{bmatrix}
2 & 7 & 1 & 5 & 10 & 7 & 10 & 1 & 5 & 2 \\
\boxed{5} & 3 & \boxed{4} & 3 & 1 & 6 & 1 & 9 & 6 & 7 \\
4 & \boxed{1} & 3 & 9 & 7 & 2 & \boxed{7} & 5 & 3 & 4 \\
8 & 10 & 8 & 6 & 3 & 3 & 6 & 2 & 8 & 6 \\
7 & 4 & 5 & \boxed{2} & 9 & 1 & 3 & 4 & 2 & 3 \\
1 & 6 & 2 & 1 & 5 & 4 & 5 & 3 & 10 & \boxed{8} \\
9 & 5 & 6 & 7 & \boxed{8} & \boxed{9} & 8 & 6 & 7 & 9 \\
6 & 8 & 9 & 4 & 2 & 5 & 2 & \boxed{8} & 4 & 1 \\
10 & 2 & 7 & 10 & 6 & 10 & 9 & 7 & 9 & 10 \\
3 & 9 & 10 & 8 & 4 & 8 & 4 & 10 & \boxed{1} & 5
\end{bmatrix}
$$

Sample code: $\begin{bmatrix} 2 & 3 & 2 & 5 & 7 & 7 & 3 & 8 & 10 & 6 \end{bmatrix}$

Figure 3-2 Sample design representation

3.1.1. GA Procedure

Main flowchart of the code is given in the following figure.

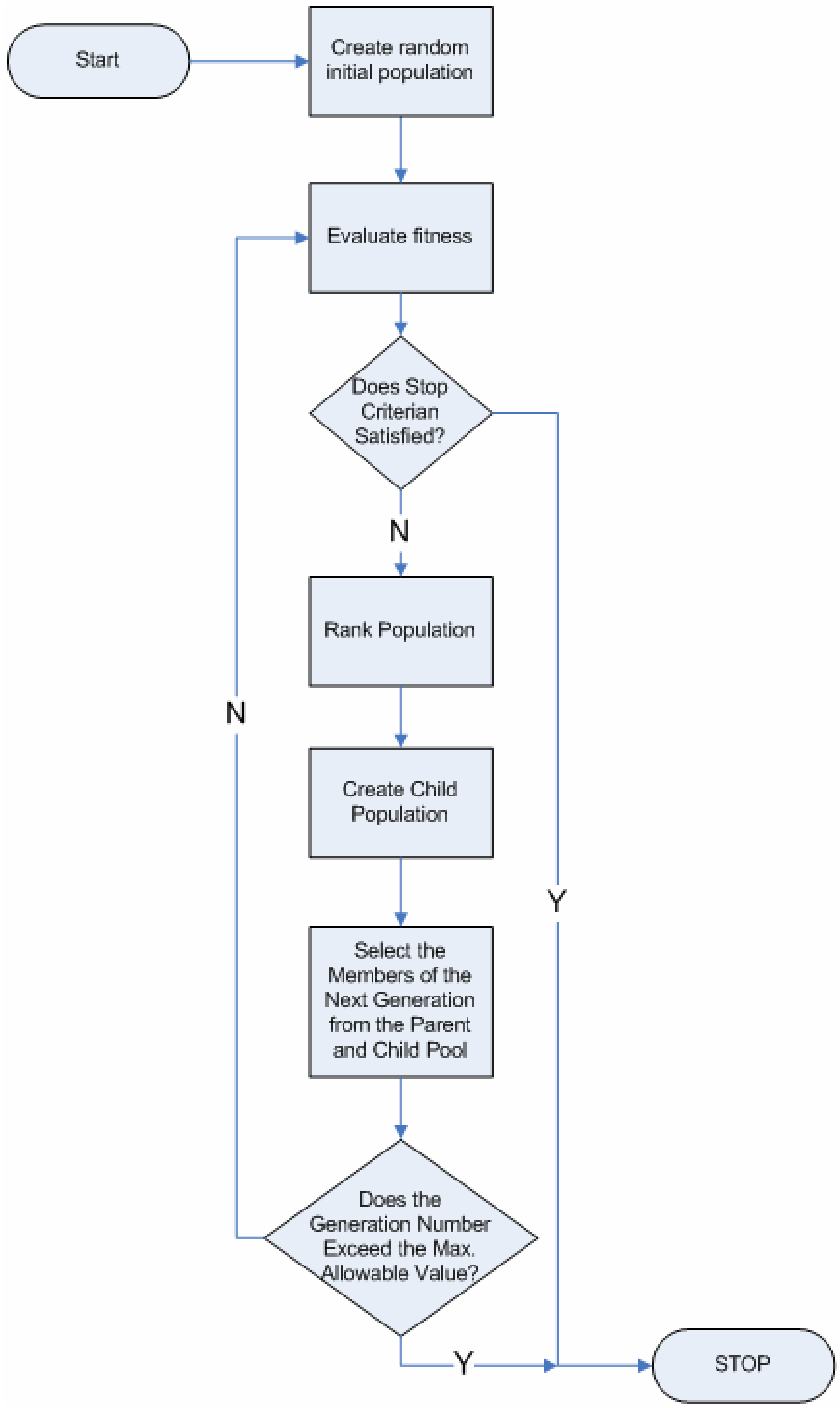


Figure 3-3 Main GA flowchart

Firstly, the initial population of designs is created by a random process, see section **3.1.2**. Population size is defined at the beginning and does not change for the entire optimization process. Members of the first generation are then placed in an environment where they compete and breed with other members of the population. At this point the main loop of the GA begins. The fitness values of each member in the generation are computed by a subroutine (see section **3.1.3**). If the best fitness value does not change for the last specified number of iterations, the program ends. If there is still improvement in the best fitness value, iteration continues. Next the population is ranked according to their fitness values and children population is created from the selected parents by using the genetic operators. Parent selection and the genetic operators used are described in sections **3.1.4.1** and **3.1.4.2** respectively. After the children population is created, members of the next generation are selected from the children and the parent pool. Detailed description of the selection schemes used is given in section **3.1.5**. This main loop continues until the maximum allowed number of generation is reached or the other stop criterion is satisfied.

3.1.2. Creation of the Initial Population

Flowchart of the creation of random initial population section is given in **figure 3-4**. In this section of the code, members of the initial population are created until the defined population size is reached. Genes of the chromosomes are chosen randomly from integers 1 to 10. After the creation of each chromosome, its uniqueness is tested in order to avoid having the same chromosomes in the first generation. After the initial population is created the main loop of GA begins.

3.1.3. Fitness Evaluation and Population Ranking

Main loop begins with the fitness evaluation, see **figure 3-5**. The fitness values of the members of the last generation are evaluated by a function. Fitness function evaluation is explained in **chapter 4**. After fitness evaluation, the best, average and the worst fitness values of the generation are identified. If the best fitness value does not change for the last specified maximum number of generations, program ends the iteration process. If a variable multiple elitist (VE) selection is chosen, and the best fitness value does not change for the last specified number of generations (specified number referred herein is maximum allowed number of generations that the best fitness value does not improve, and this number is valid for VE method), the value of N_{me} is set to the population size.

N_{me} is a number which is used in multiple and variable multiple elitist selection methods. In section **3.1.5**, selection schemes and N_{me} will be described in detail.

Next, the population is ranked according to their fitness values (see **figure 3-6**), and Φ values which are required for the parent selection are evaluated. Φ values are the probability interval limits used to select a parent according to the generated random number. Φ is defined in the interval 0 and 1. Parent selection method used in GA and calculation of Φ values are described in detail in section **3.1.4.1**.

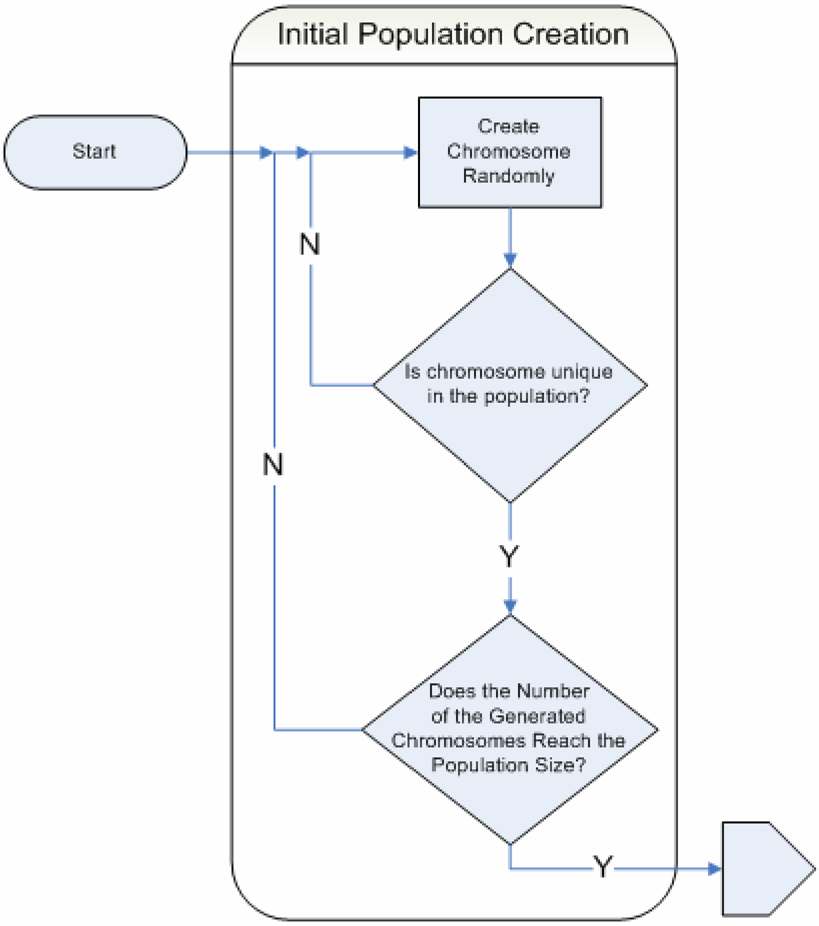


Figure 3-4 Creation of the random initial population (GA*ver*1)

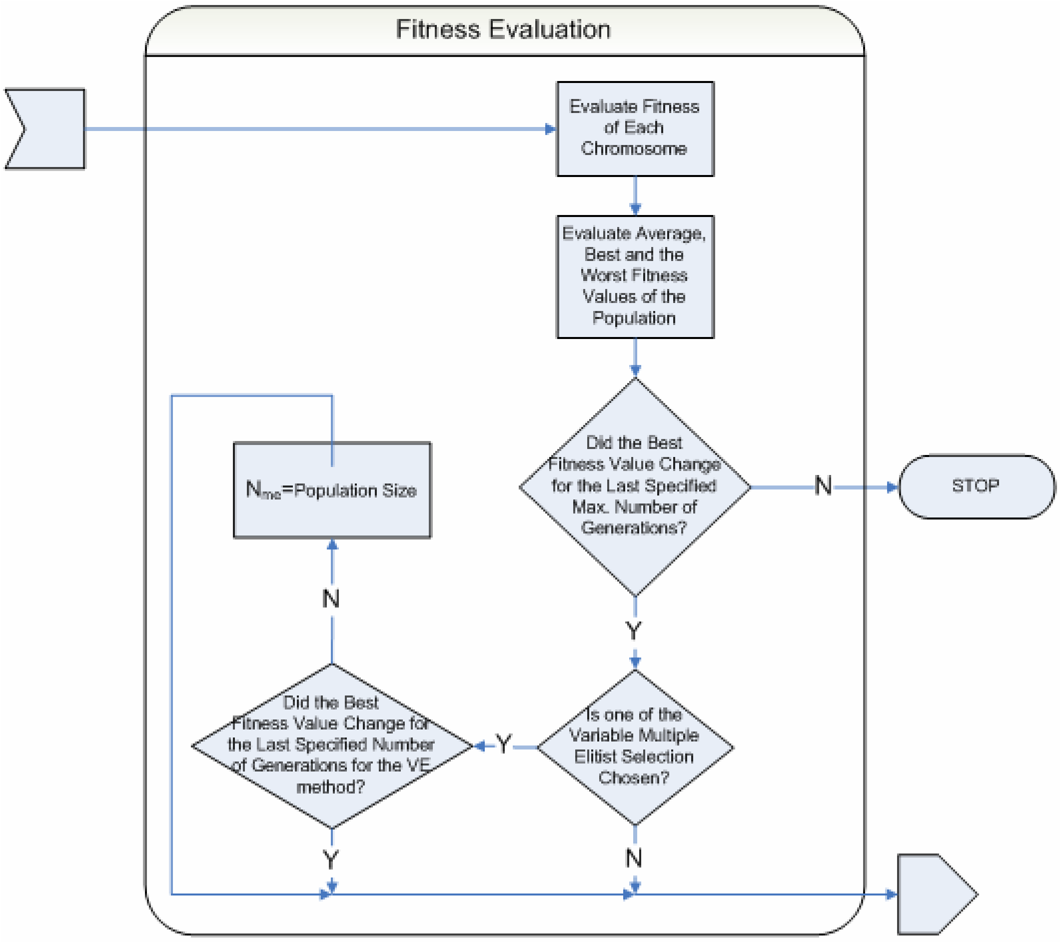


Figure 3-5 Fitness evaluation

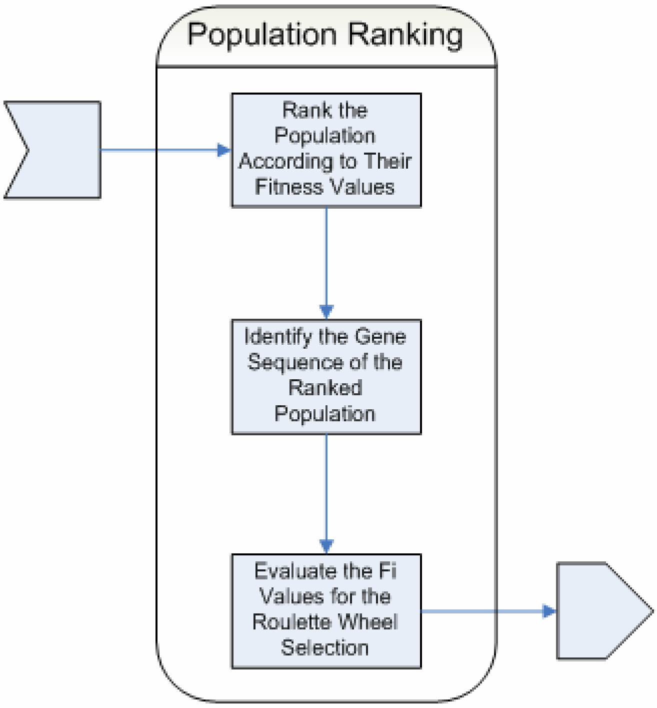


Figure 3-6 Population ranking

3.1.4. Creation of Child Population

After fitness evaluation and population ranking, new members can be created by random selection of the parents. The idea of parent selection is described in section **3.1.4.1**. Children are created from the selected parents by using genetic operators (section **3.1.4.2**). Flowchart for creation of child population is given in **figure 3-7**.

Every created child is checked for uniqueness in both the whole parent population and the child population which is created till then. If it is not unique, child is created again.

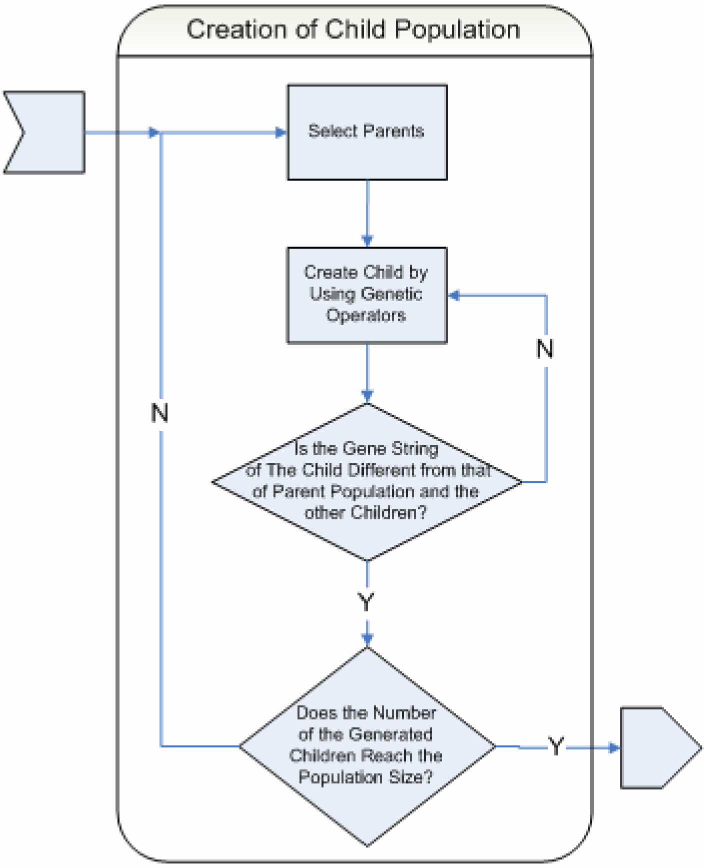


Figure 3-7 Creation of child population (GA*ver*1)

3.1.4.1. Parent Selection

A parent selection method determines the probability of an individual to become a parent and produce an offspring. In GA, every single member of the parent population has a chance to become a parent. In order to have better members in the next generation, fitter members should have a higher probability to become a parent whereas members having worse fitness should have small probabilities to be chosen.

In this GA, roulette wheel concept is used for the parent selection. The selection probabilities are not calculated directly from the fitness value; instead it is calculated from ranking. The parent population is ranked first, and then the Φ values are calculated as given in equation **3-1** where μ is the population size and i is the rank ($i = 1, 2, \ldots, \mu$) [3].

$$\Phi_i = \Phi_{i-1} + \frac{2(\mu+1-i)}{\mu(\mu+1)} \qquad \textbf{3-1}$$

A random number which is uniformly distributed in the interval $[0,1)$ is generated. If the generated random number is in the interval $[\Phi_{i-1},\Phi_i)$, i^th^ best design is chosen as a parent. As an example, the selection probability and the required value of the random number for the selection of each member of a population having 5 individuals is given in **figure 3-8**. For instance, if the values of the generated random numbers are 0.35 and 0.86 respectively, then 2^nd^ and 4^th^ ranked members are selected as parents. As seen from the graph, the better ranked individuals have a higher probability to become a parent.

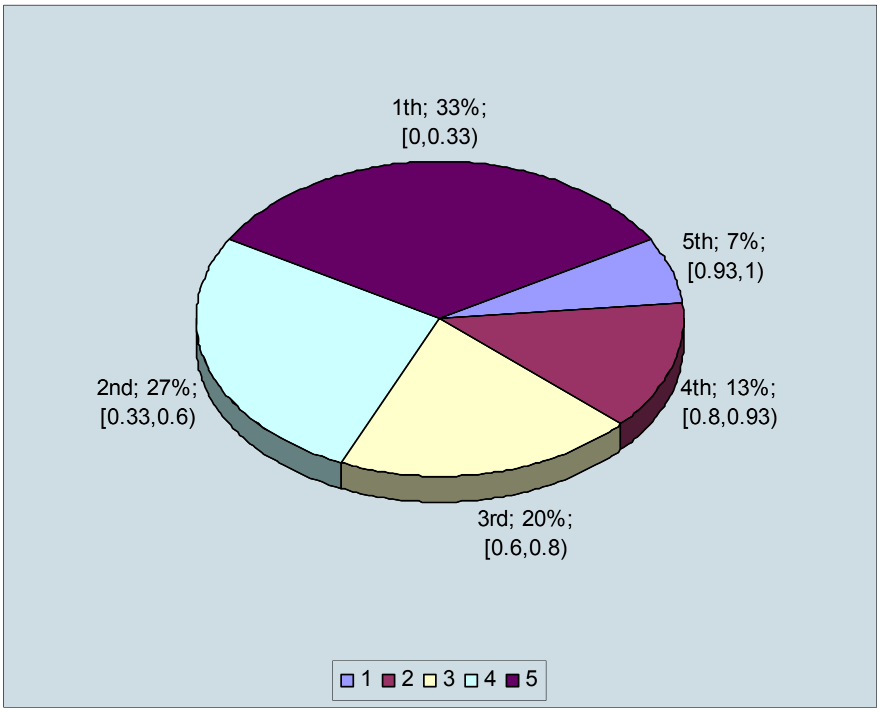


Figure 3-8 Selection probabilities (population size: 5)

Flowchart of the parent selection part is given in **figure 3-9**. First the parents are selected as described above. Then, it is checked whether same ranked parents are chosen or not. Initial developed code was checking the uniqueness according to only the rankings. But it is seen that after some runs different ranked individuals might have the same gene sequence which means they are same. This situation has a very small probability which depends on the population size, the number of generations and the child creation procedure, but in such case the GA code gets stuck into a region and finds the local optimum. In order to avoid this problem, uniqueness of the gene sequences of the selected parents are also checked.

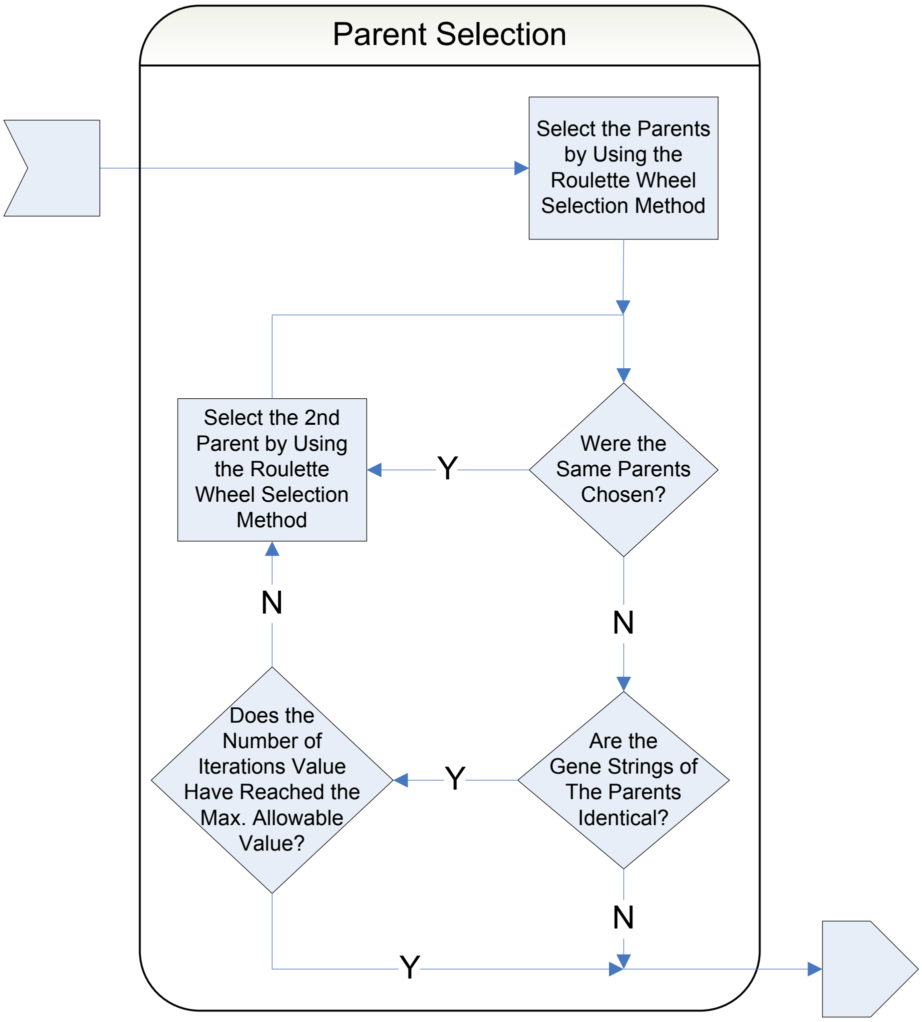


Figure 3-9 Parent selection

3.1.4.2. Genetic Operators

After the parent selection, a child is created using the genetic operators (see **figure 3-10**). In this problem type, crossover, mutation and gene swap are used as genetic operators which are explained in sections ***3.1.4.2.1***, ***3.1.4.2.2*** and ***3.1.4.2.3*** respectively. Each genetic operator has an assigned probability which is identified in the interval $[0,1)$ at the beginning of the optimization process. A uniformly distributed random number is generated for each genetic operator. If the generated random number is smaller than the identified value than the genetic operator is applied.

In **chapter 4**, the effects of the change in the genetic operator probability values are examined, and optimum value ranges are tried to be identified.

3.1.4.2.1 Crossover

Crossover is the most important genetic operator of a GA. There are several versions of this operator. The main idea is recombination of the parents to form a new individual which is called child. In the developed GA, one-point crossover is used [16]. This operator is applied with a certain probability which is identified before the initialization of the optimization process. First, a random number is generated to decide whether crossover will be applied or not. A uniformly distributed random integer is generated in the interval $[1,N-1]$, where N is the chromosome length. Generated random integer identifies the crossover point. At that point, the gene strings of both parents split into two pieces. The left portion of the first parent and the right portion of the second parent are combined to constitute the child (see **figure 3-11**). If the child's gene string is identical to one of the parents then crossover point is changed and new offspring is created.

Crossover point

Parent I : [**2 3 8 9** | 4 10 2 2 5 7]

Parent II : [3 7 5 5 | **6 9 1 8 6 10**]

$\Longrightarrow$ Child : [**2 3 8 9 6 9 1 8 6 10**]

Figure 3-10 Crossover operator

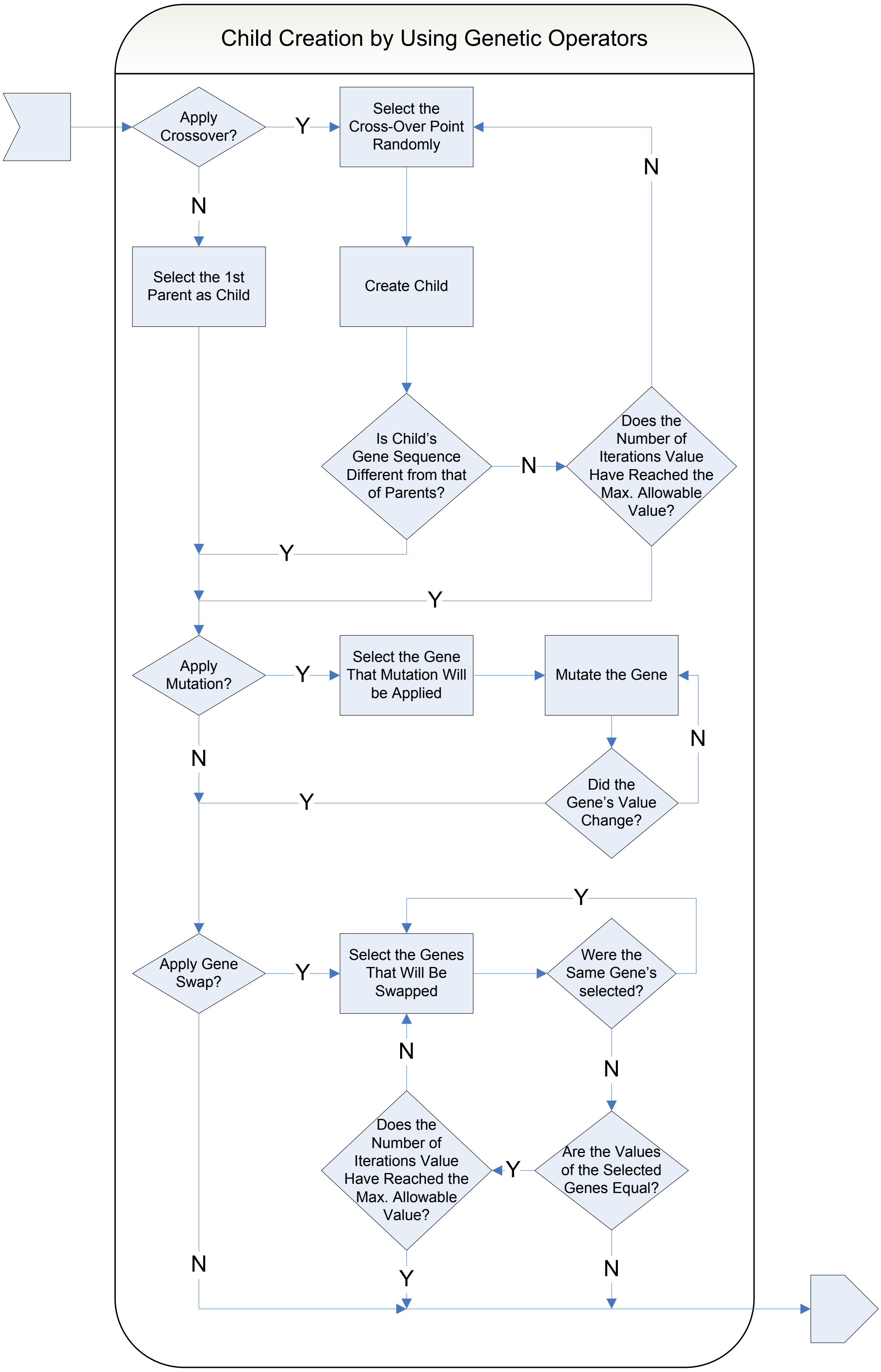


Figure 3-11 Child creation by genetic operators

3.1.4.2.2 Mutation

After child creation, mutation is applied to the created offspring usually with a small probability. In this problem type only gene flipping is used as a mutation operator [16]. A gene is chosen and its value is changed to another permissible value randomly (see **figure 3-12**).

Before Mutation: **After Mutation:**

Child : [2 3 8 9 6 9 $\boxed{\mathbf{1}}$ 8 6 10] $\Longrightarrow$ Child : [2 3 8 9 6 9 $\boxed{\mathbf{7}}$ 8 6 10]

Figure 3-12 Mutation operator

3.1.4.2.3 Gene-Swap

Similar to other operators, gene swap is applied to the offspring with a certain probability. Two genes in the string are chosen randomly and replaced with each other as seen in **figure 3-13** [3].

Before Gene Swap: **After Gene Swap:**

Child : [2 $\boxed{\mathbf{3}}$ 8 9 6 9 7 $\boxed{\mathbf{8}}$ 6 10] $\Longrightarrow$ Child : [2 $\boxed{\mathbf{8}}$ 8 9 6 9 7 $\boxed{\mathbf{3}}$ 6 10]

Figure 3-13 Gene swap operator

3.1.5. Selection Schemes

After the creation of the child population, members of the next generation are selected from the pool that consists of both parent and child population. A selection scheme identifies how much information from the last generation is carried through to the next generation, which directly affects the explorative and exploitative characteristics of GA. Hence, there should be an optimization for the passing information. There have been used several number of selection schemes. In this GA, elitist, multiple elitist (I and II), and variable multiple elitist (I and II) selection schemes which are explored by Soremekun [3] are used. Selection schemes are described in section **3.1.5.1** and **3.1.5.2**. Main flowchart of the selection schemes part is given in **figure 3-14**.

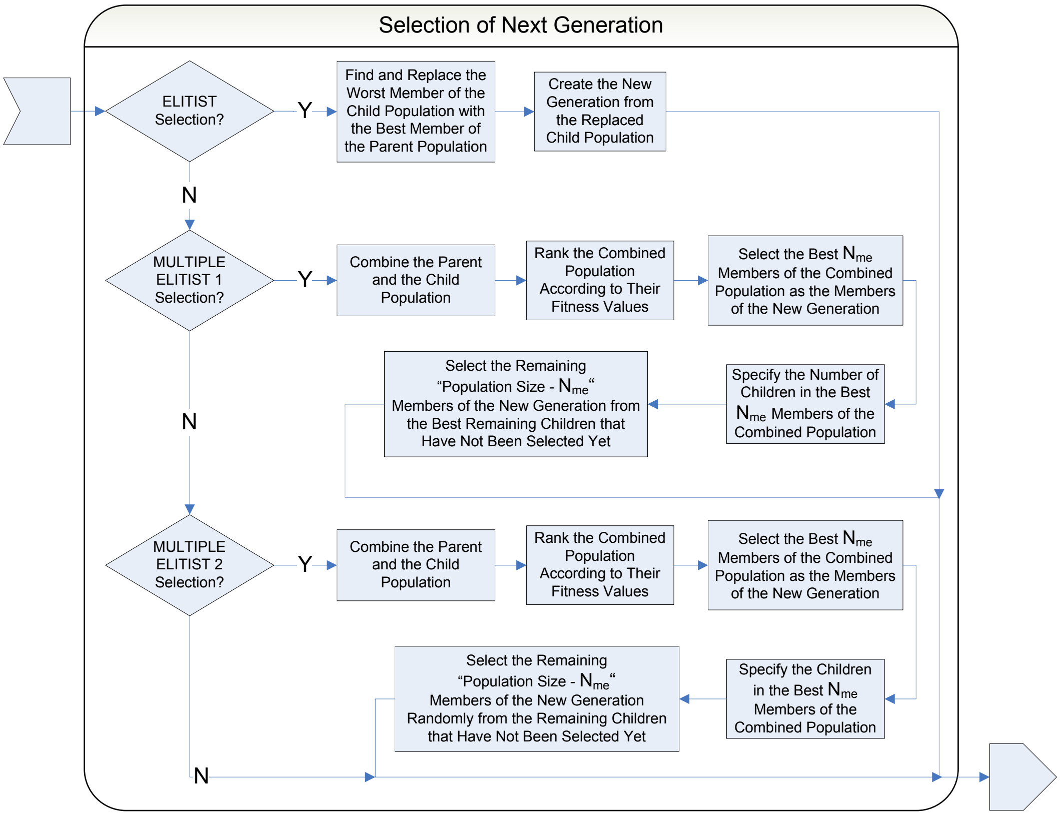


Figure 3-14 Selection of next generation

3.1.5.1. Elitist Selection (EL)

In elitist selection, whole child population constitutes the new generation except that the worst member of the child population is replaced with the best member of the parent population. In this method, the best ever found individual is never lost, but all the other members in the parent population that might contain important genetic information is lost. If this important genetic information is conserved, GA may converge to a better design faster and create richer populations. In **figure 3-15**, the schematic description of *EL* method is given. In this example, objective is to minimize the fitness value.

Member	Fitness
PI	3
PII	15
PIII	37
PIV	43
PV	65

Ranked
Parent Population

Create
Child
Population

Member	Fitness
CI	4
CII	17
CIII	22
CIV	39
CV	45

Ranked
Child Population

Next
Generation

Member	Fitness
NI	3
NII	4
NIII	17
NIV	22
NV	39

Next
Generation

Figure 3-15 Elitist (*EL*) selection method

3.1.5.2. Multiple Elitist Selections

Multiple elitist selections are designed to decrease the potential loss of important genetic information that some individuals in the parent population may contain [3]. The amount of preserved information identifies the search characteristics of a GA. If very small amount of information is preserved from the parent population, GA becomes quite explorative. The algorithm continuously searches for the attractive areas and may miss the optimum design points. On the other hand, as the amount of preserved information increases, GA becomes more exploitative. Genetic search is focused on a specific design area which is occupied by the best individuals found till then. In such case, GA may find the local optimum point due to inadequate exploration. It is very important to identify the optimum amount of preserved information in order to find the best design quicker. In multiple elitist selection schemes, the amount of preserved information can be adjusted. Detailed descriptions of multiple elitist selection schemes are given in the following sub-sections.

3.1.5.2.1. Multiple Elitist 1 Selection (ME1)

In *ME1*, the parent and child populations both having μ individuals are combined and ranked from best to worst. The best N_{me} individuals of the combined population are carried over to the next generation and the remainder of the next generation is filled with top $\mu - N_{me}$ best members of the child population which have not been ranked in the best N_{me} individuals of the combined population. The integer N_{me} takes a value in the interval $[1, \mu]$. When $N_{me} = 1$ and the best member of the combined population comes from the parent population, *ME1* is identical to *EL*. *ME1* scheme is shown in **figure 3-16**.

Member	Fitness
PI	3
PII	15
PIII	37
PIV	43
PV	65

Parent Population

Member	Fitness
CI	4
CII	17
CIII	22
CIV	39
CV	45

Child Population

Member	Fitness
CmI(PI)	3
CmII(CI)	4
CmIII(PII)	15
CmIV(CII)	17
CmV(CIII)	22
CmVI(PIII)	37
CmVII(CIV)	39
CmVIII(PIV)	43
CmIX(CV)	45
CmX(PV)	65

Combined Population

$N_{me}=3$

Member	Fitness
NI	3
NII	4
NIII	15
NIV	17
NV	22

Next Generation

Figure 3-16 Multiple elitist 1 (*ME1*) selection method

3.1.5.2.2. Multiple Elitist 2 Selection (ME2)

ME2 is quite similar to *ME1*. As in *ME1*, parent and child populations are combined and ranked from best to worst; and the best N_{me} individuals of the combined population are carried over to the next generation. The difference is how the remainder of the next generation is filled. The remainder $\mu - N_{me}$ members are chosen randomly from the children that have not been ranked in the best N_{me} individuals. Every remaining child has equal chance to be selected for the next generation. In the code, uniqueness of each randomly chosen child is checked. *ME2* scheme is demonstrated in **figure 3-17.**

Member	Fitness
PI	3
PII	15
PIII	37
PIV	43
PV	65

Parent Population

Member	Fitness
CI	4
CII	17
CIII	22
CIV	39
CV	45

Child Population

Member	Fitness
CmI(PI)	3
CmII(CI)	4
CmIII(PII)	15
CmIV(CII)	17
CmV(CIII)	22
CmVI(PIII)	37
CmVII(CIV)	39
CmVIII(PIV)	43
CmIX(CV)	45
CmX(PV)	65

$N_{me} = 3$

Member	Fitness
NI	3
NII	4
NIII	15
NIV	22
NV	45

Next Generation

Figure 3-17 Multiple elitist 2 (*ME2*) selection method

3.1.5.3. Variable Multiple Elitist Selections

In multiple elitist selections N_{me} is constant throughout the search. Variable multiple elitist selection 1 and 2 (*VME1*, *VME2*) are identical to *ME1* and *ME2* respectively, except that N_{me} can be varied during the search process. The value of N_{me} identifies the amount of information preserved from the parent population, i.e. search characteristics of the GA. Search is quite explorative for small values of N_{me}. As the value of N_{me} increases, search becomes more exploitative, thus less explorative. Having the capability of varying the value of N_{me} during the search allows changing the explorative and exploitative characteristics of the GA at different stages. At the initial stage of the search N_{me} value is kept small and the interesting areas of the design space are searched roughly. After predefined number of generations without improvement in the best fitness value, the value of N_{me} is set equal to the population size (μ) and the most interesting area ever found is searched rigorously to find the optimum design. This capability increases the possibility of finding the better designs in less time compared to the corresponding multiple elitist selection methods.

3.2. GA Code version 2 (GA*ver*2): Stacking Sequence Optimization of a Composite Pressure Vessel for Minimum Radial Displacement

In this code version, the aim is finding the optimum stacking sequence for a composite pressure vessel under the defined loading conditions. The objective of the optimization is to minimize the radial displacement. The analytical procedure for the objective function (fitness) evaluation is given in **Chapter 2**.

In this study, the composite pressure vessel is symmetrically laminated and consists of two-ply stacks having $\pm\alpha$ ply angles. There are three different codings developed for the identification of a design. Different codings of ply angles are given in **figure 3-18**. In the first coding, ply angles are constricted to the orientation angles $0^o, \pm 45^o, 90^o$. Second coding allows $0^o, \pm 15^o, \pm 30^o, \pm 45^o, \pm 60^o, \pm 75^o, 90^o$ ply angles; and wider range of ply angle orientations ($0^o, \pm 10^o, \pm 20^o, \pm 30^o, \pm 40^o, \pm 50^o, \pm 60^o, \pm 70^o, \pm 80^o, 90^o$) is coded in the third one. A sample representation of a stacking sequence is given in **figure 3-19**.

This code version is very similar to the first one. The main GA procedure and all the other procedures described in the preceding sub-sections are identical except the creation of the initial population (see **figure 3-20**) and the creation of the child population (see **figure 3-21**). These differences are due to contiguity constraint. Three or more 0_2^o and 90_2^o contiguous stacks are not allowed in order to avoid matrix cracking [3].

Code	Definition
1	two-ply stack oriented at 0^o
2	two-ply stack oriented at $\pm 45^o$
3	two-ply stack oriented at 90^o

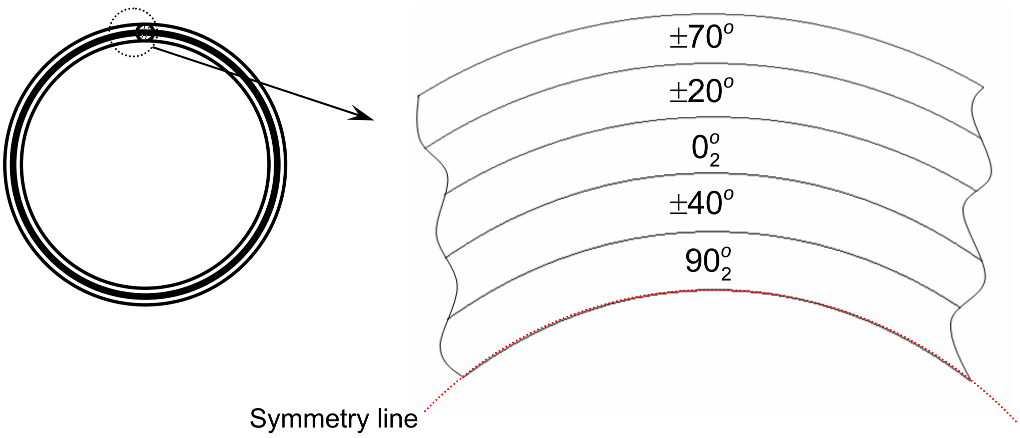
Figure 3-18a Ply-angle code I

Code	Definition
1	two-ply stack oriented at 0^o
2	two-ply stack oriented at $\pm 15^o$
3	two-ply stack oriented at $\pm 30^o$
4	two-ply stack oriented at $\pm 45^o$
5	two-ply stack oriented at $\pm 60^o$
6	two-ply stack oriented at $\pm 75^o$
7	two-ply stack oriented at 90^o

Figure 3-18b Ply-angle code II

Code	Definition
1	two-ply stack oriented at 0^o_2
2	two-ply stack oriented at $\pm 10^o$
3	two-ply stack oriented at $\pm 20^o$
4	two-ply stack oriented at $\pm 30^o$
5	two-ply stack oriented at $\pm 40^o$
6	two-ply stack oriented at $\pm 50^o$
7	two-ply stack oriented at $\pm 60^o$
8	two-ply stack oriented at $\pm 70^o$
9	two-ply stack oriented at $\pm 80^o$
10	two-ply stack oriented at 90^o_2

Figure 3-18c Ply-angle code III



Stacking Sequence: $[\pm70^o, \pm20^o, 0^o_2, \pm40^o, 90^o_2, 90^o_2, \pm40^o, 0^o_2, \pm20^o, \pm70^o]$

GA Coding: $[8\ 3\ 1\ 5\ 10]_s$

Figure 3-19 Sample design representation

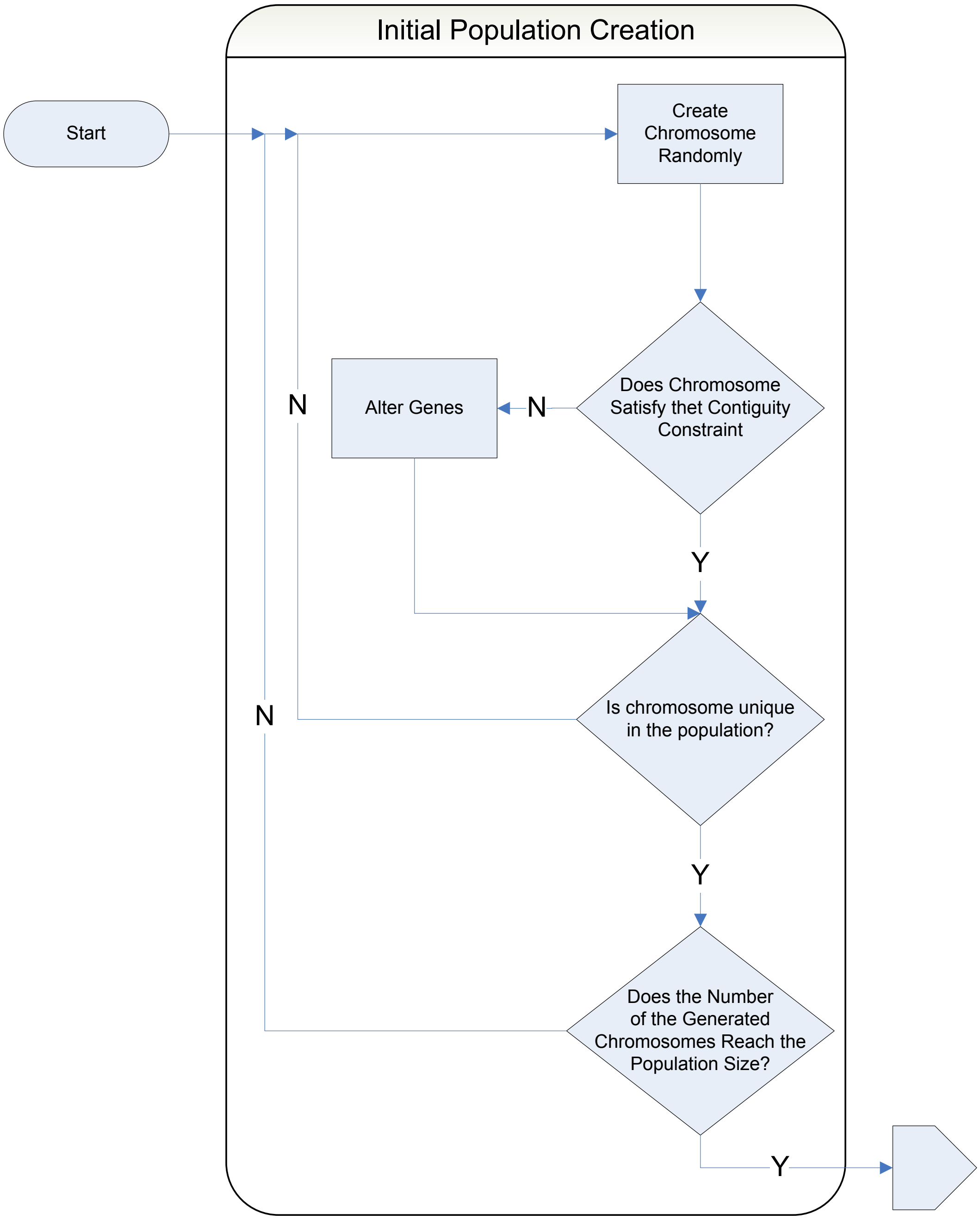


Figure 3-20 Creation of the random initial population (GA*ver*2)

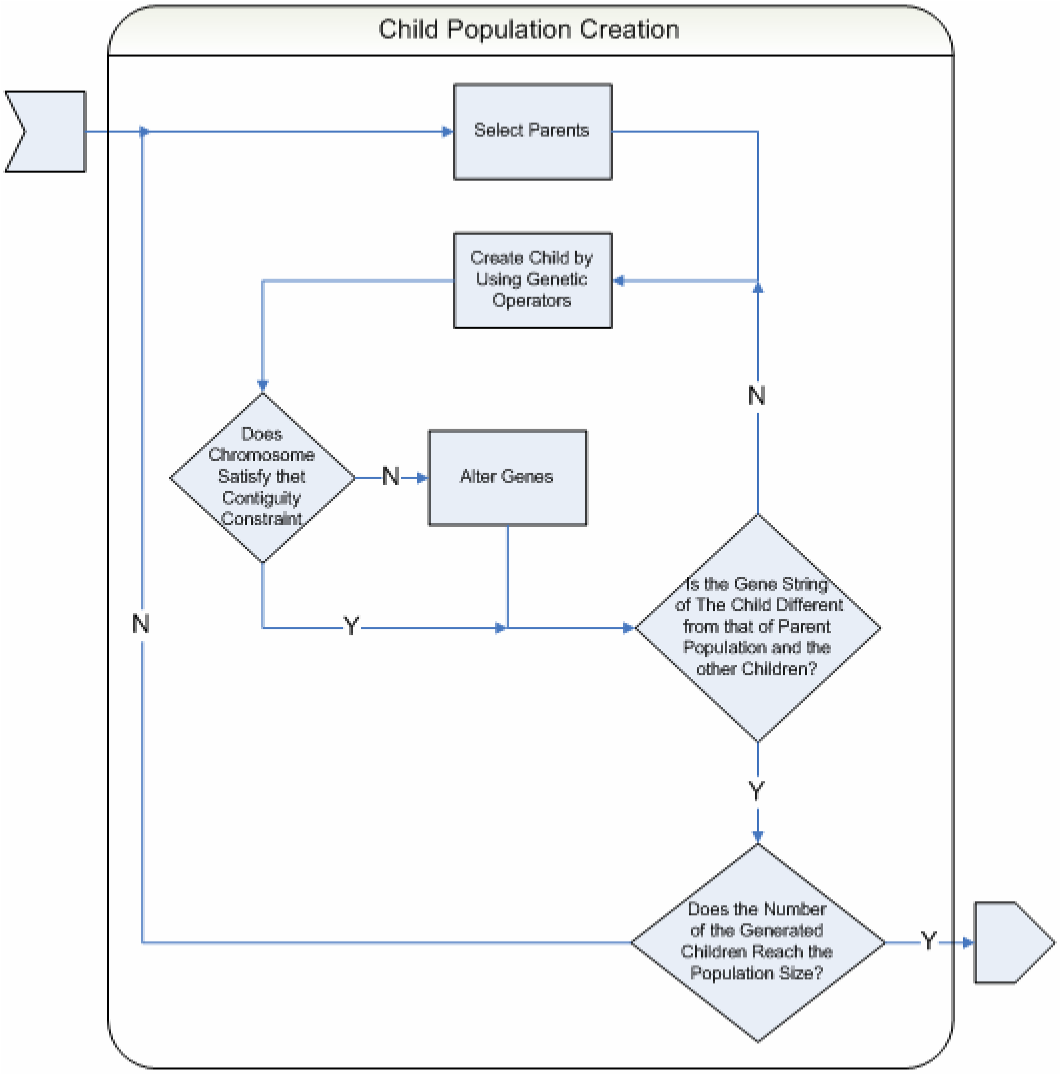


Figure 3-21 Creation of the child population (GA*ver*2)

3.3. GA Code version 3 (GA*ver*3): Stacking Sequence Optimization of a Composite Pressure Vessel for Minimum Thickness

This code version is similar to the second version. The aim is again finding the optimum stacking sequence, but the objective is to minimize the thickness, thus weight. The analytical procedure for the objective function (fitness) evaluation is same as the procedure given in **chapter 2**.

The coding for the ply angles is also same as the coding defined in **figure 3-18**. In addition, “0” code is added to define the empty stacks. The main GA procedure and all the other procedures described in section **3.2** are same except the creation of the initial population (see **figure 3-22**) and the additive genetic operators.

Crossover, mutation and gene-swap operators are same. The only difference is in crossover operator. In the crossover operator, the generated random integer which defines the crossover point is in the interval $\left[K_{empty}, N-1\right]$, where K_{empty} is the maximum number of empty stacks in the parents and N is the chromosome length. This limit is chosen to avoid excessive exploration in the design space.

The additive operators in this code version are ply-deletion and ply-addition. In ply-deletion operator, a non-empty stack is randomly chosen and deleted. Then the stacking sequence re-organized in order to have the empty stacks at the outer-most plies (see **figure 3-23**). In ply-addition operator, a new non-empty stack is added to the sequence in place of the innermost empty stack (see **figure 3-24**). Flowchart of the genetic operators is given in **figure 3-25**.

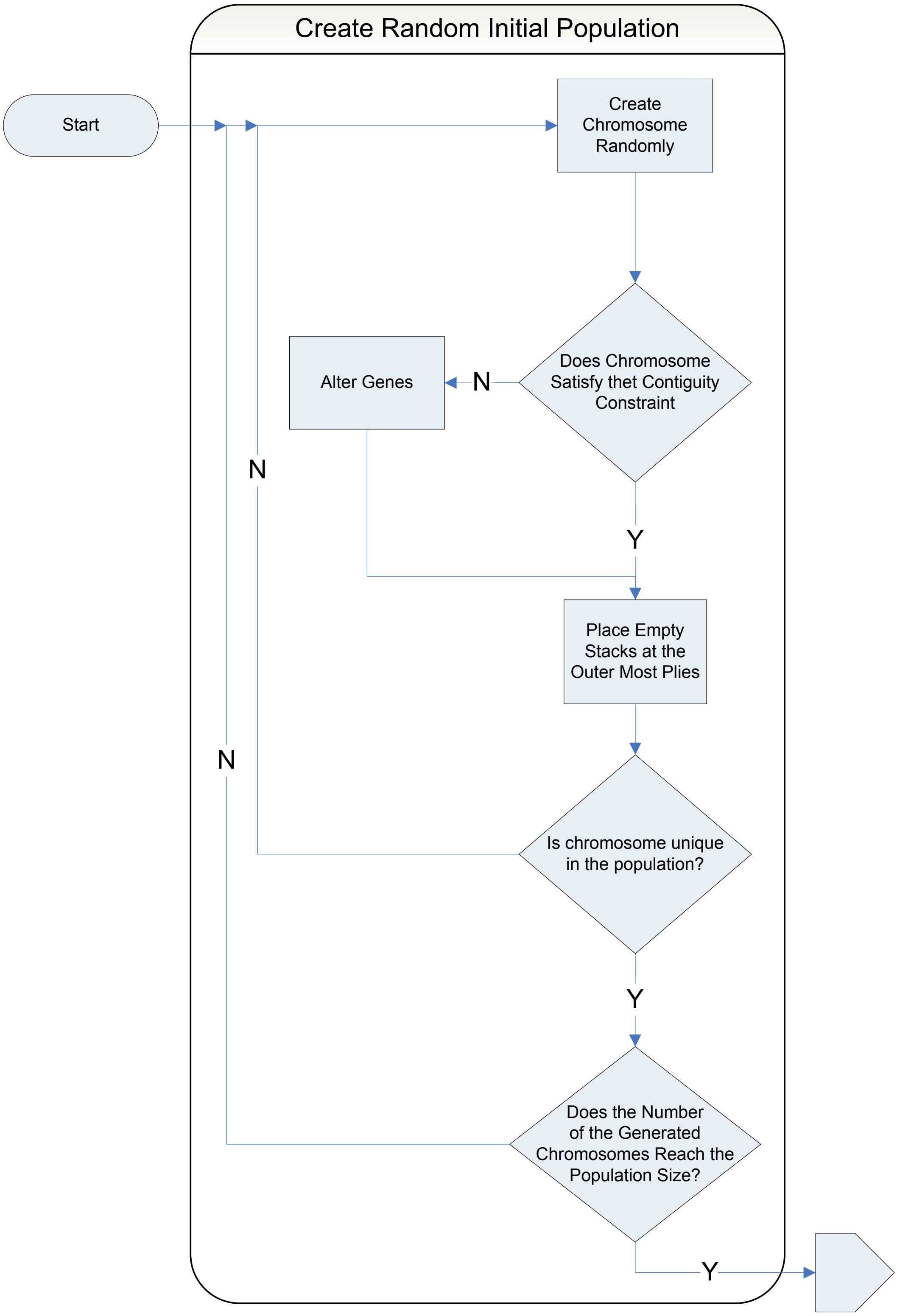


Figure 3-22 Creation of the random initial population (GA*ver*3)

Stacking Sequence:

Before Ply-Deletion: [$Empty_2$,$Empty_2$, $\pm 20^o, \pm 70^o, \boxed{\pm 20^o}, 0^o_2, \pm 40^o, 90^o_2]_s$

After Ply-Deletion: [$Empty_2$,$Empty_2$, $\pm 20^o, \pm 70^o, \boxed{Empty_2}, 0^o_2, \pm 40^o, 90^o_2]_s$

Re-organize: [$Empty_2$,$Empty_2$, $Empty_2$, $\pm 20^o, \pm 70^o, 0^o_2, \pm 40^o, 90^o_2]_s$

GA Coding:

Before Ply-Deletion: $[0\ 0\ 3\ 8\ 3\ 1\ 5\ 10]_s$

After Ply-Deletion: $[0\ 0\ 3\ 8\ 0\ 1\ 5\ 10]_s$

Re-organize: $[0\ 0\ 0\ 3\ 8\ 1\ 5\ 10]_s$

Figure 3-23 Ply-deletion operator

Stacking Sequence:

Before Ply-Addition: [$Empty_2$,$Empty_2$, $Empty_2$, $\pm 20^o, \pm 70^o, 0^o_2, \pm 40^o, 90^o_2]_s$

After Ply- Addition: [$Empty_2$,$Empty_2$, $\pm 60^o, \pm 20^o, \pm 70^o, 0^o_2, \pm 40^o, 90^o_{2\,s}$

GA Coding:

Before Ply- Addition: $[0\ 0\ 0\ 3\ 8\ 1\ 5\ 10]_s$

After Ply- Addition: $[0\ 0\ 7\ 3\ 8\ 1\ 5\ 10]_s$

Figure 3-24 Ply-addition operator

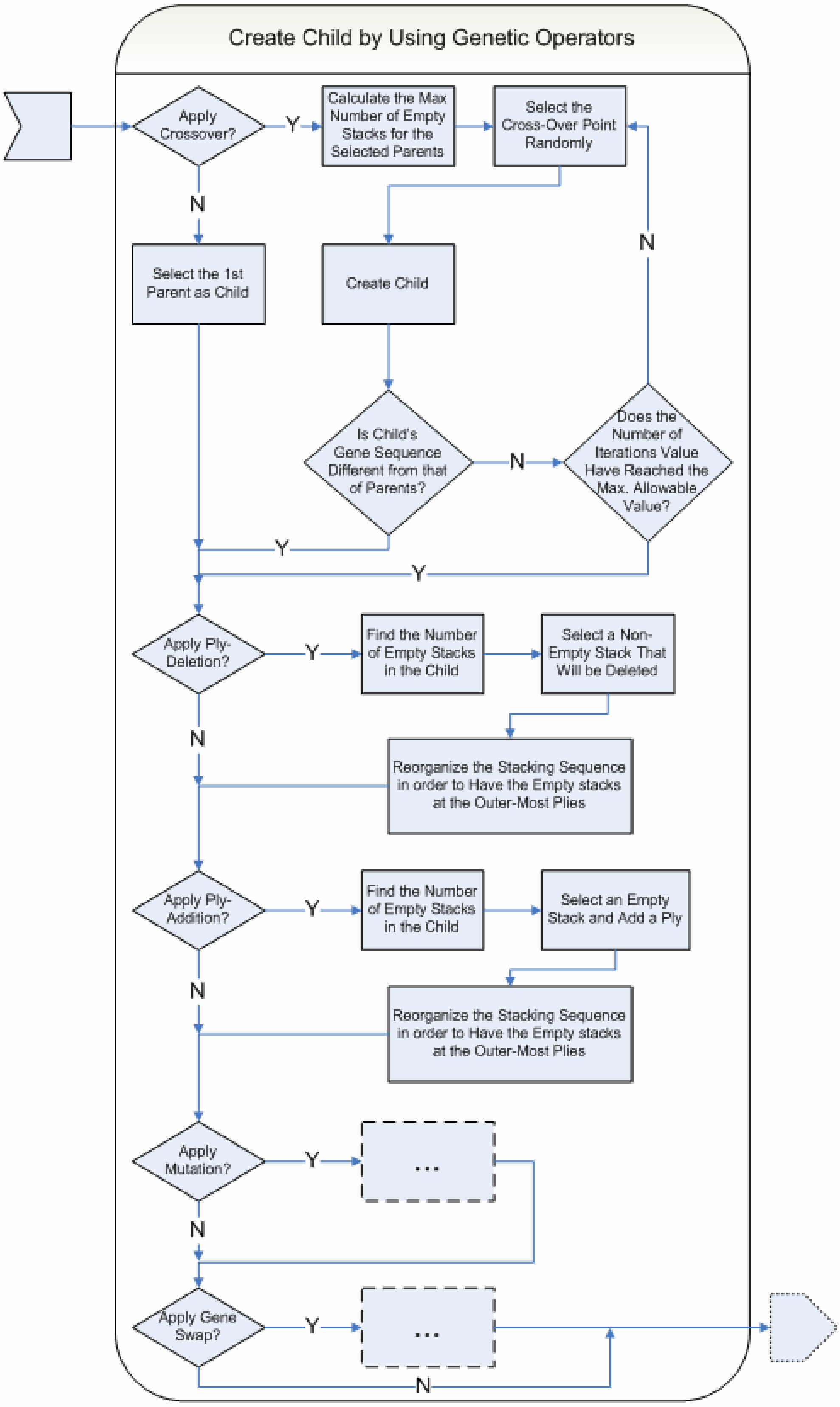


Figure 3-25 Genetic operators (GA*ver*3)

CHAPTER 4

GA CODE VERSION 1 (GAver1) FOR TEST PROBLEM

In order to test the performance of the developed code and identify the effects of the optimization parameters, a test problem is designed and the code is adapted to the designed problem (Ga*ver*01). The detailed description of the problem and the results are given in the following sub-sections.

4.1. Description of the Test Problem

In this problem, there is a matrix of which elements consist of integers from 1 to 10. The objective of the problem is to minimize the product of the elements chosen from each column of the matrix. The chosen matrix size is 10 by 50. Since the size is quite big, the product may take a value up to 10^{50}. In order to decrease the value range of the objective function to [1,101], the matrix is converted to a new matrix of which elements are dependent on the elements of the original matrix by the formula given in **4-1**.

$$newA_{i,j} = 0.0107440471414871 \times \left(oldA_{i,j} - 1 \right) + 1 \qquad \textbf{4-1}$$

The matrix used in the runs is given in **figure 4-1**. The elements of the matrix are randomly chosen from integers 1 to 10, and an integer can exist only once in a column. Each bit value in the gene string is equal to the row number in the corresponding column. In this case, the bit value range is [1,10], and the chromosome length is 50.

The minimum value that the objective function can get is “1” which corresponds to the multiplication of the column elements which are equal to “1”. The best design coding is given in **figure 4-2**.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
10	8	6	2	**1**	7	5	5	5	7	6	4	6	7	8	9	4
3	**1**	9	10	8	4	4	9	3	4	3	5	8	8	6	7	9
6	2	3	6	10	6	6	8	7	2	9	6	10	4	7	3	2
5	3	10	5	5	5	2	7	8	6	8	7	3	3	2	5	3
9	6	8	4	3	**1**	**1**	4	4	8	5	2	7	10	5	2	10
8	5	2	3	7	2	9	2	**1**	10	10	9	9	5	4	6	5
1	9	**1**	8	4	3	3	3	9	3	2	8	5	9	9	4	7
7	7	4	7	9	8	7	**1**	6	**1**	4	3	4	2	10	**1**	8
2	4	7	**1**	2	9	10	10	10	5	7	10	**1**	6	3	8	**1**
4	10	5	9	6	10	8	6	2	9	**1**	**1**	2	**1**	**1**	10	6

Figure 4-1a Test matrix (columns 1 to 17)

18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
3	6	5	5	3	5	8	9	6	3	4	8	3	2	10	4	5
8	10	9	4	4	6	4	6	7	5	2	9	2	**1**	7	3	9
7	8	3	9	**1**	3	6	7	8	6	5	4	**1**	9	8	**1**	8
6	9	7	7	9	**1**	2	8	5	8	9	3	8	3	3	2	7
9	2	10	3	6	8	3	2	3	9	3	6	9	10	6	5	**1**
1	7	4	10	2	2	7	**1**	2	**1**	**1**	2	4	7	9	8	2
4	3	6	2	7	9	**1**	5	4	10	6	7	7	5	5	9	3
2	5	2	**1**	10	4	5	3	9	7	8	**1**	10	4	4	10	4
5	**1**	**1**	8	8	7	9	4	**1**	4	7	10	5	6	2	7	6
10	4	8	6	5	10	10	10	10	2	10	5	6	8	**1**	6	10

Figure 4-1b Test matrix (columns 18 to 34)

35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
4	2	10	10	5	2	10	4	7	3	7	9	2	3	5	**1**
3	6	2	3	2	5	6	7	10	8	2	5	8	5	10	4
6	4	8	6	10	4	7	5	6	2	5	**1**	4	2	7	8
2	9	4	7	4	8	3	2	8	6	6	7	5	8	9	3
5	3	5	5	8	7	8	3	**1**	4	3	4	**1**	**1**	8	6
9	8	**1**	8	3	3	9	10	3	9	**1**	6	3	6	4	9
8	10	6	2	7	10	2	**1**	9	7	4	2	10	7	6	10
7	**1**	3	**1**	9	**1**	4	8	5	**1**	8	8	9	10	**1**	7
1	5	9	4	**1**	6	5	9	4	10	9	10	6	4	2	5
10	7	7	9	6	9	**1**	6	2	5	10	3	7	9	3	2

Figure 4-1c Test matrix (columns 35 to 50)

7 2 7 9 1 5 5 8 6 8 10 10 9 10 10 8 9

Figure 4-2a Best design code (columns 1 to 17)

6 9 9 8 3 4 7 6 9 6 6 8 3 2 10 3 5

Figure 4-2b Best design code (columns 18 to 34)

9 8 6 8 9 8 10 7 5 8 6 3 5 5 8 1

Figure 4-2c Best design code (columns 35 to 50)

Several runs have been made to identify the effects of the optimization parameters. In order to compare the results, the reliability and the price of the optimization are calculated for each run. The equation used to calculate the reliability is given in **4-2.** Here, fitness refers to the mean objective function value of the runs. Price of the search is equal to the number of objective function evaluations required to converge to the best design divided by the reliability of the search. The evaluation of the results will be based on reliability and the price of the search.

$$Reliability = \left(\frac{(101 - fitness)}{100}\right)^{50} \qquad \textbf{4-2}$$

4.2. Results

There are significant number of parameters that affect the performance of the optimization. In order to find the optimum ones, the runs have been made in different groups. In each group, all the parameters except the examined one are kept constant in order to make the comparisons healthier. The results will be given for each examined parameter in the corresponding sub-section. In section **4.2.1.**, the parent selection methods are compared, and the optimum range of N_{me} parameter used in multiple elitist selection methods are investigated. The optimum probability value ranges for the genetic operators (mutation and swap) are examined in sections **4.2.2.** and **4.2.3.**. The optimum value range for cross-over probability is not investigated; and in all the runs, the cross-over probability value is equal to 1.0. Because the child takes the all the information from one parent (the first parent in this code), if cross-over is not applied. In section **4.2.4.**, the optimum population size range as a percent of the chromosome length is sought.

4.2.1. Parent Selection Methods and N_{me} Parameter

In this section, elitist (EL), multiple elitist 1 (ME1) and multiple elitist 2 (ME2) selections are compared. The runs have been made for different population sizes. First, all three selection methods are compared for single run. In this run, the population size is 40 and the probabilities for swap, cross-over and mutation operators are 0.8, 1.0 and 0.1 respectively. The number of generations for the search is limited to 1000. The results are given in **figure 4-3** and **figure 4-4**. As it can be seen from figures, using ME1 and ME2 selection methods gives better results when compared to the EL method in all fields. The price of the search for EL method is higher and the best, average and the worst reliability of the search is worse than ME1 and ME2 methods. Thus, the remainder of the runs will focus on ME1 and ME2 methods and the N_{me} value used in these methods. The high fluctuations in the figures are due to single run and the random nature of the search. As the number of runs increases, the comparisons become healthier. The remainder results will be given for 100 runs each.

In the following part of this section only ME1 and ME2 methods are compared. Multiple runs (100 runs) are made for population sizes 10, 20 and 40. For all three population sizes, used probabilities for the cross-over, swap and mutation operators are again 1.0, 0.8 and 0.1 respectively.

The results for population size 10 are given in **figures 4-6** to **4-10**. In **figure 4-6**, fitness versus generation number plots are given for both ME1 and ME2 methods and for different N_{me} values. In the graphs, best, average and the worst fitness values are the average of the corresponding fitness values of 100 runs. To have a better understanding of the meaning of the fitness values given in the plots, a sample multiple run (5 runs) problem is given in **figure 4-5**. The population size of the sample runs is chosen to be 10.

It is seen from the plots of **figure 4-6** that, as the N_{me} value increases, the best and the worst fitness values get closer for both selection methods. In **figure 4-7**, the best fitness value range of the runs versus generation number plots are given (see **figure 4-5** for the definition of the best fitness value range of a multiple run). To some point, as the N_{me} value increases, the best fitness value gets better. But after that point it becomes to get worse. It means that there is an optimum value of N_{me} in order to have a better best fitness value. As the N_{me} gets closer to the population size, the limits of the best fitness range of the runs narrow which means that the best fitness values of the runs get closer to each other. Thus the search loses the explorative capability as N_{me} increases too much. The only advantage of a very high N_{me} value is that the average fitness of the generations increases which makes the population richer and there exists more alternative members having quite good fitness values.

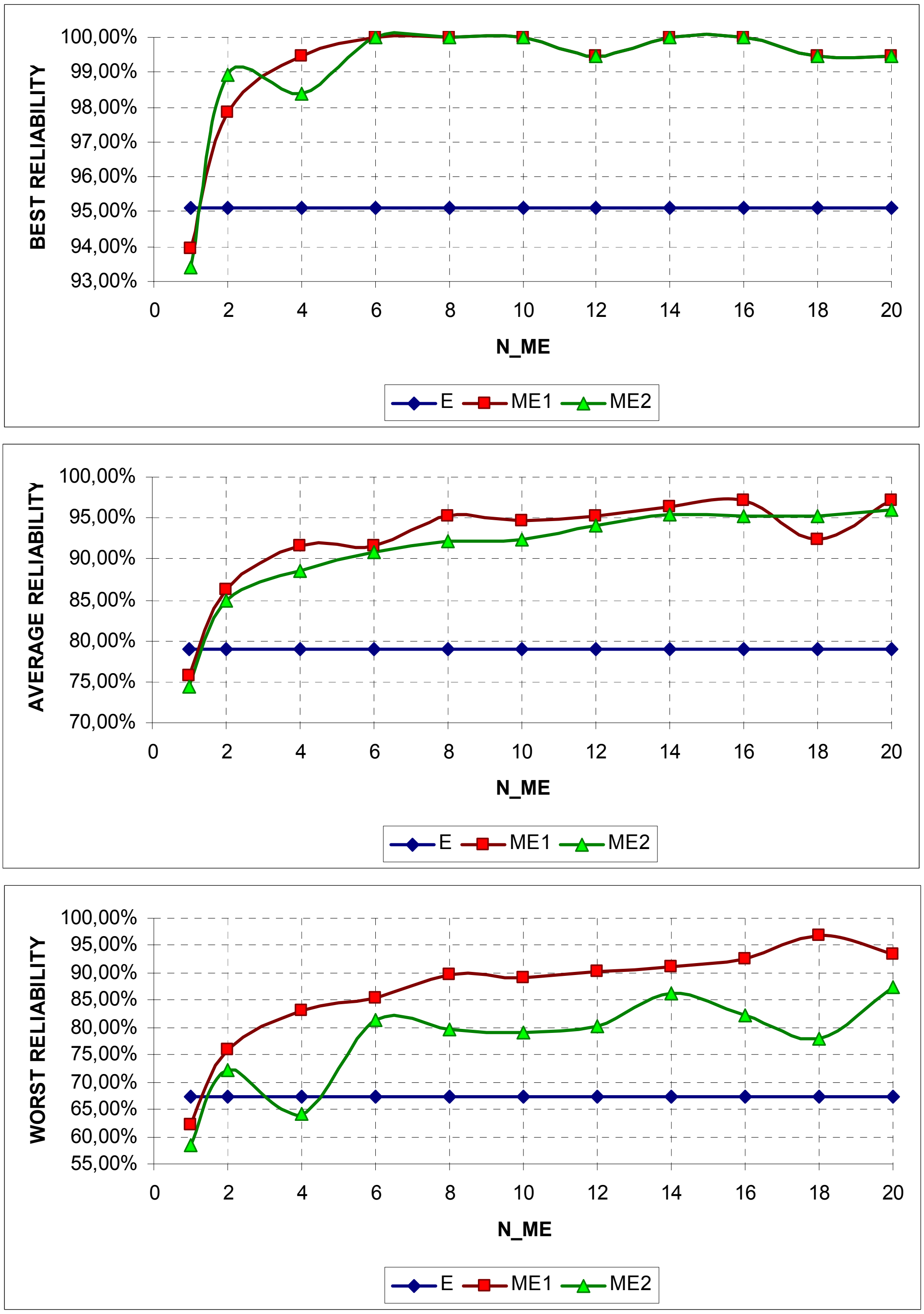


Figure 4-3 Reliability vs N_{me} for Population Size : 40 (single run)

In **figure 4-8**, reliability versus N_{me} graphs are given. The best, average and the worst reliability values are calculated from the corresponding fitness values at the end of the search. The maximum generation number of the search is 1000. It is seen that the best reliability value increases as N_{me} increases from 1 to 5. After that point, the best reliability value decreases. But the average and the worst reliability values do not have a peak point. Both increase till the end. The peak point of the best reliability is reached for N_{me} is equal to the 50 percent of the population size in ME1 method whereas it is reached for N_{me} is equal to the 70 percent in ME2 method.

In **figures 4-9** and **4-10**, ME1 and ME2 methods are compared. It is easily seen that for significant range of N_{me} values, ME1 method gives better results when compared to ME2 method. In ME1, the best reliability of the search is better whereas the price of the search is less. But when N_{me} gets closer to the population size the difference decreases.

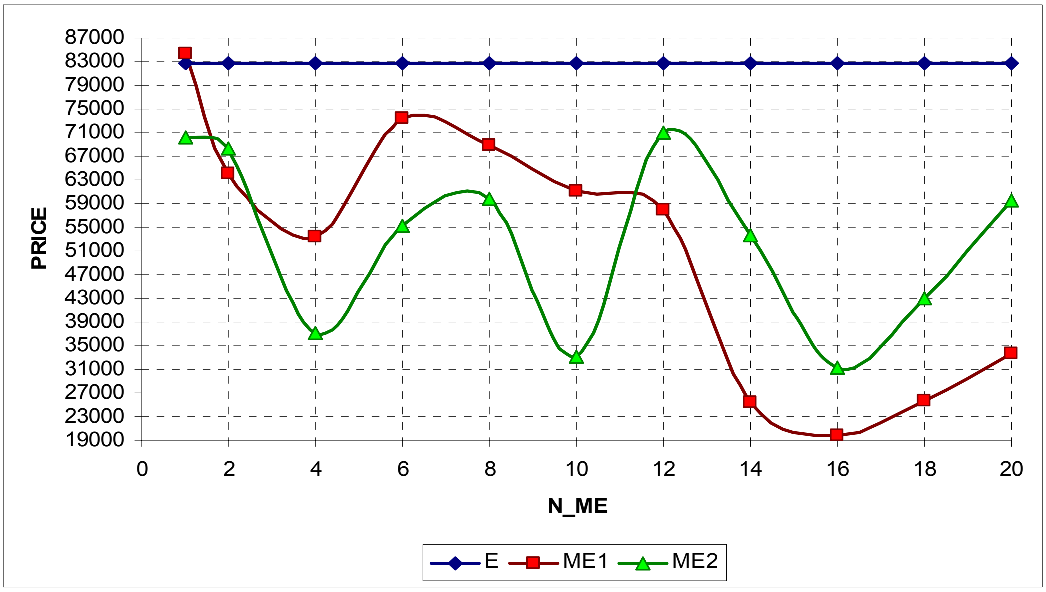


Figure 4-4 Price vs N_{me} for Population Size : 40 (single run)

Fitness Values for Generation Number: i

Run # : 1	**Run # : 2**	**Run # : 3**	**Run # : 4**	**Run # : 5**
12	8	15	9	10
20	10	18	13	16
21	23	20	16	23
23	31	24	32	39
38	35	32	39	43
42	39	36	43	45
55	42	43	48	49
58	59	49	57	58
73	66	53	63	67
86	89	71	79	81

Best Fitness Values for Generation Number: i

Run # : 1	**Run # : 2**	**Run # : 3**	**Run # : 4**	**Run # : 5**
12	8	15	9	10

Average Fitness Values for Generation Number: i

Run # : 1	**Run # : 2**	**Run # : 3**	**Run # : 4**	**Run # : 5**
42.8	40.2	36.1	39.9	43.1

Worst Fitness Values for Generation Number: i

Run # : 1	**Run # : 2**	**Run # : 3**	**Run # : 4**	**Run # : 5**
86	89	71	79	81

Best Fitness Value of Multiple Run = Average of Best Fitness Values : 10.8

Average Fitness Value of Multiple Run = Average of Average Fitness Values : 40.42

Worst Fitness Value of Multiple Run = Average of Worst Fitness Values : 81.2

Lower Limit of Best Fitness Value : 8

Upper Limit of Best Fitness Value : 15

Figure 4-5 Fitness values of a sample multiple run

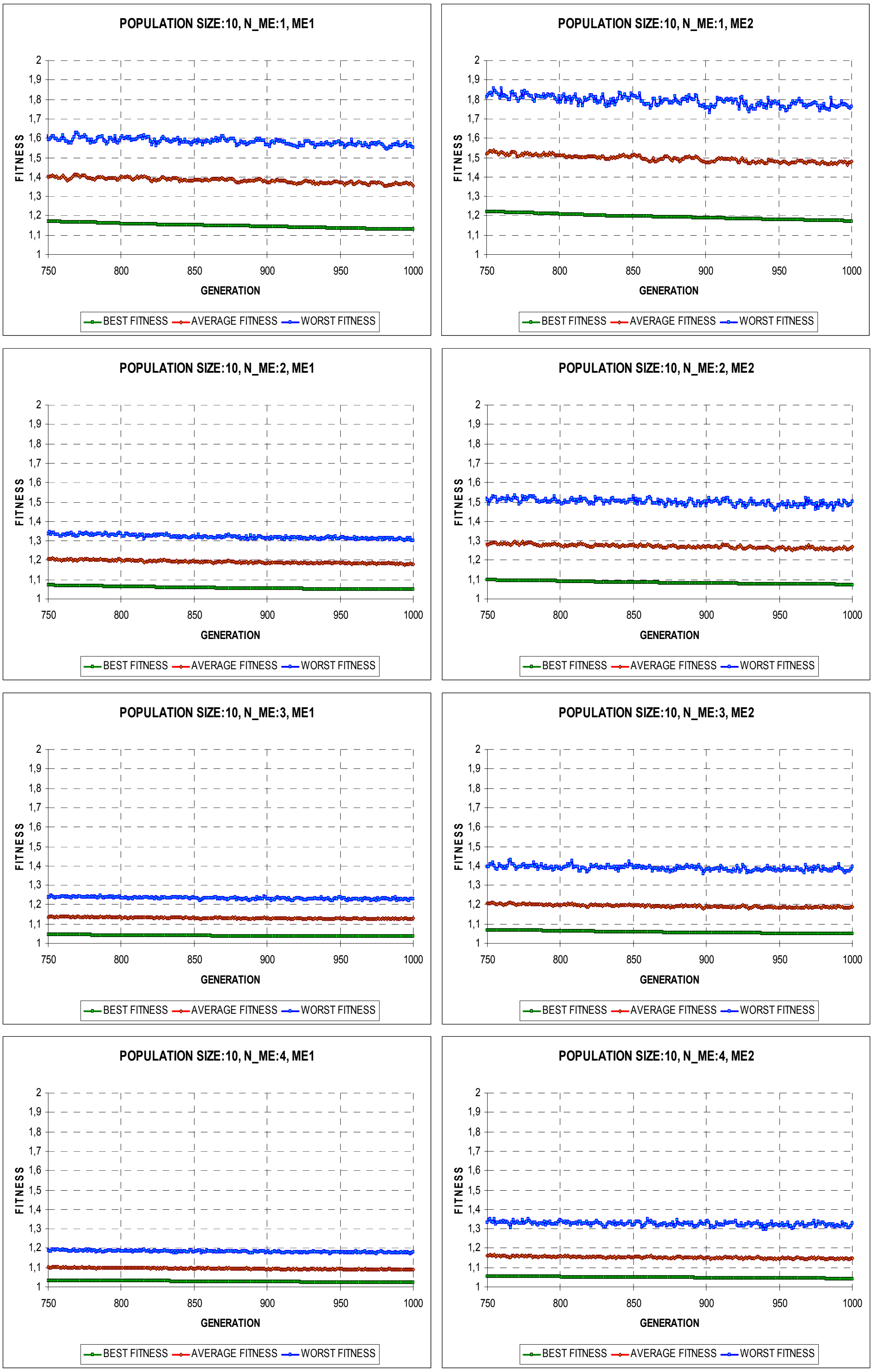


Figure 4-6a Fitness vs Generation # for Population Size : 10 (100 runs)

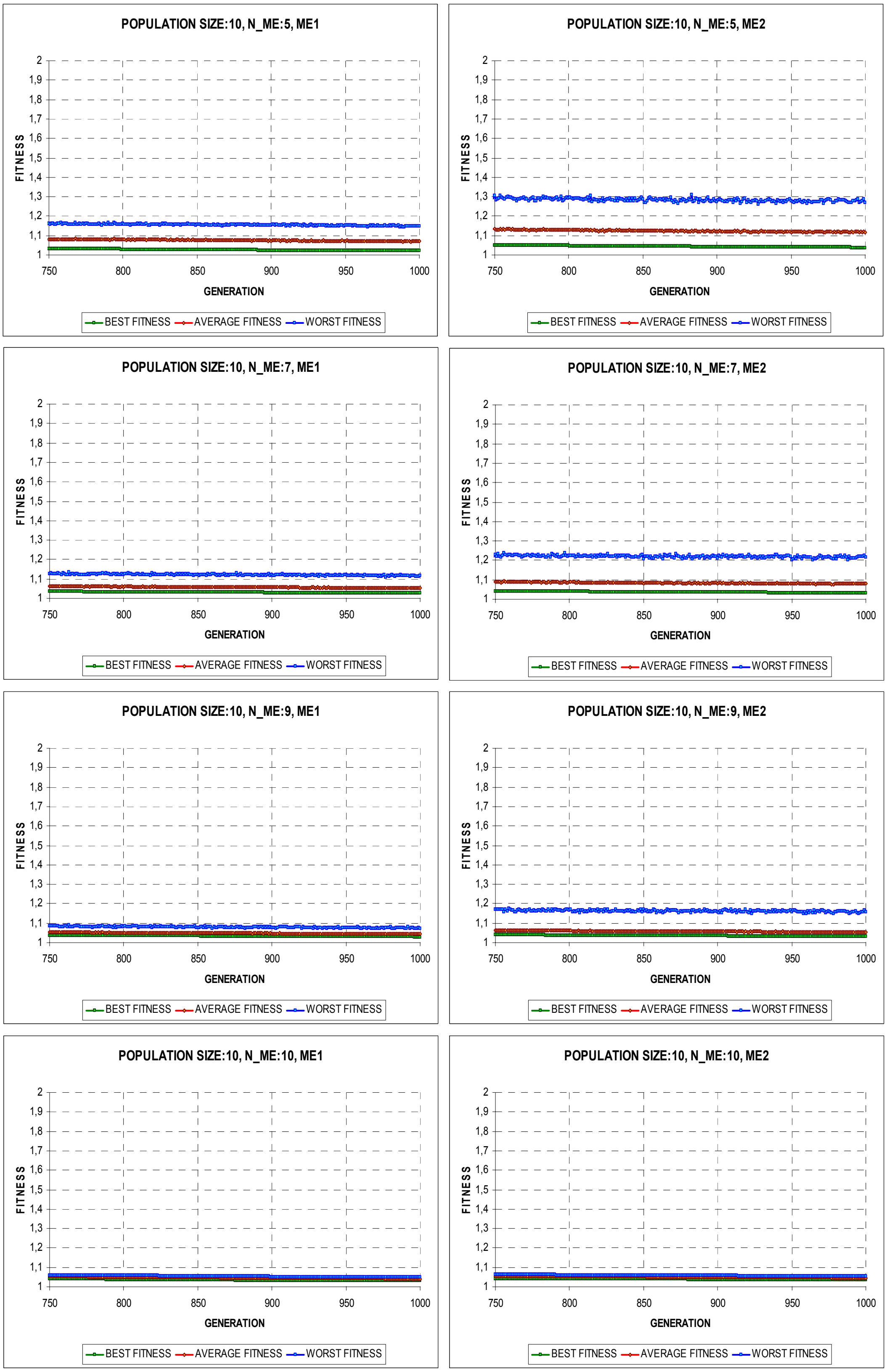


Figure 4-6b Fitness vs Generation # for Population Size : 10 (100 runs)

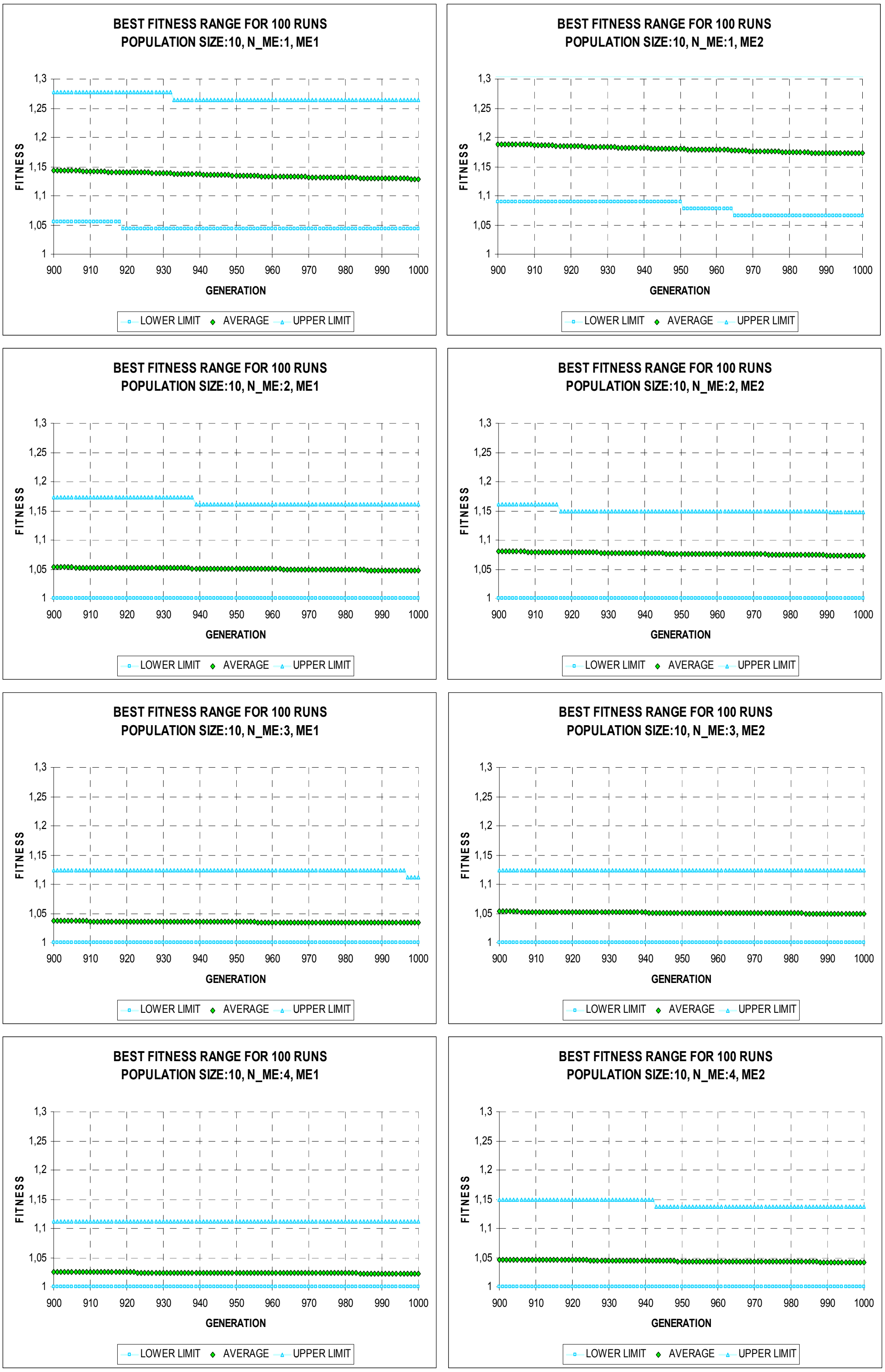


Figure 4-7a Best Fitness Range vs Generation # for Population Size : 10 (100 runs)

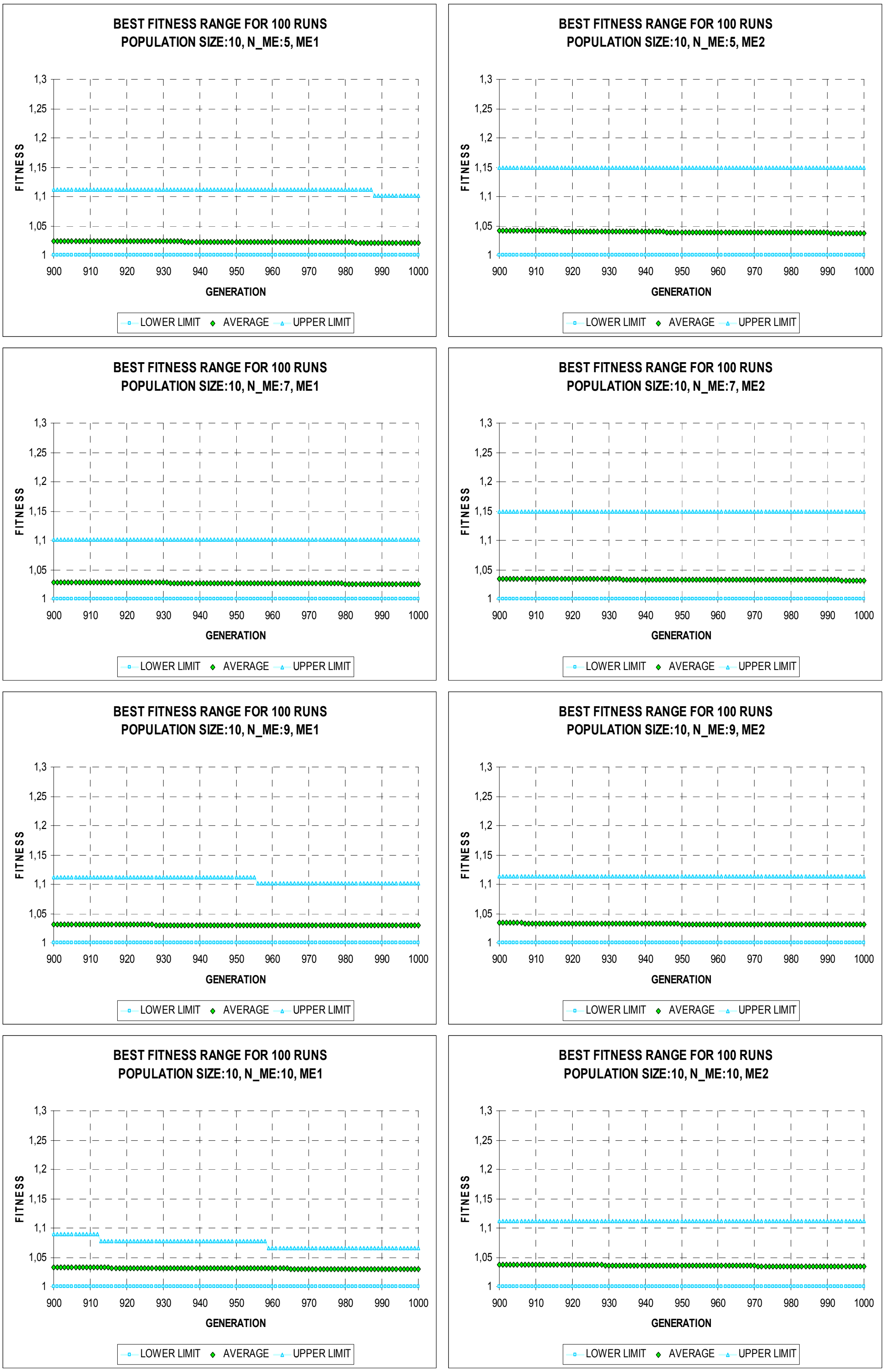


Figure 4-7b Best Fitness Range vs Generation # for Population Size : 10 (100 runs)

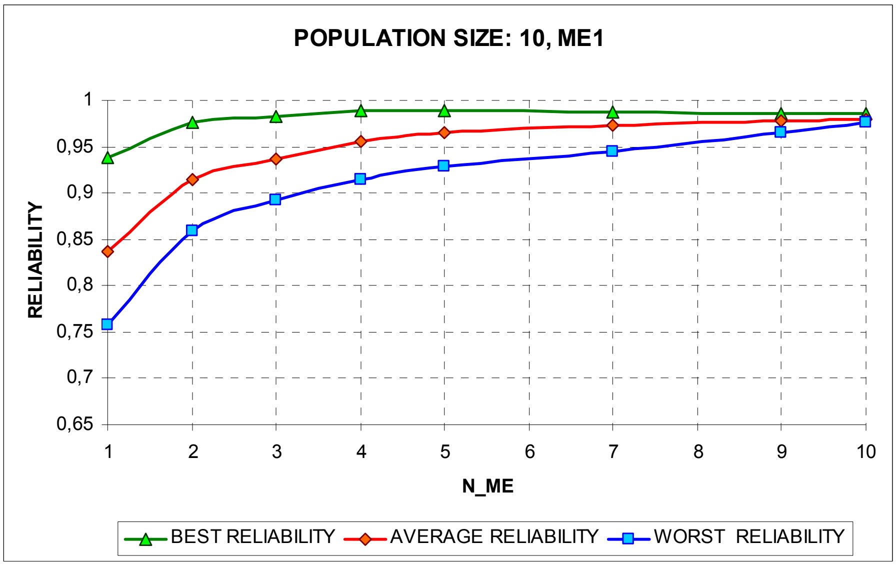


Figure 4-8 Reliability vs N_{me} for Population Size : 10 (100 runs)

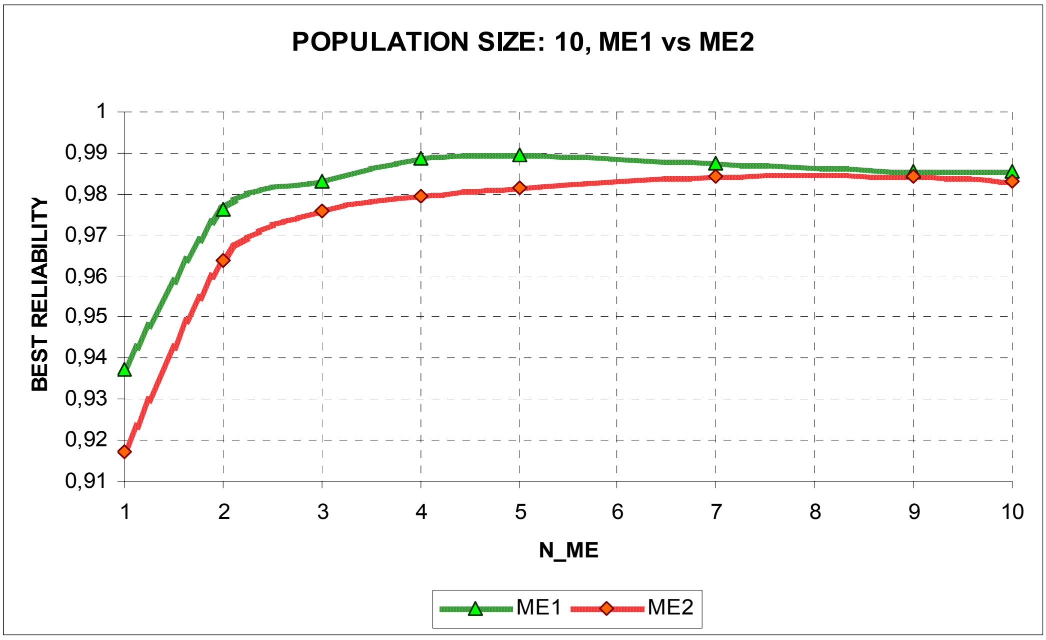


Figure 4-9 Best Reliability vs N_{me} for Population Size : 10 (100 runs)

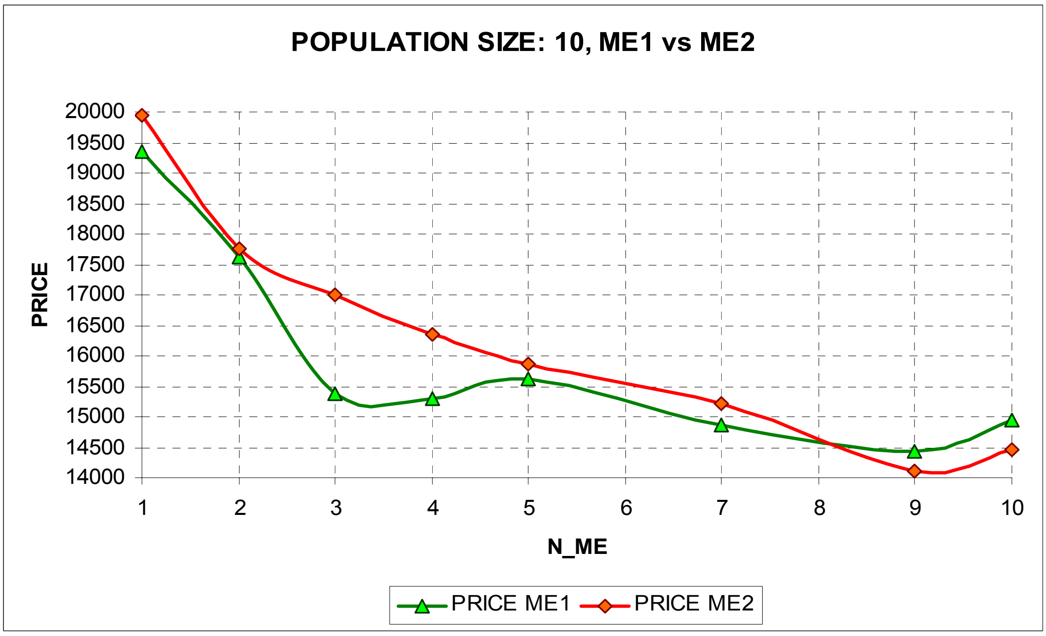


Figure 4-10 Price vs N_{me} for Population Size : 10 (100 runs)

The resultant graphs for population size 20 are given in **figures 4-11** to **4-15**. As seen in **figure 4-11**, both methods tend to have a closer best and worst fitness values as N_{me} gets closer to the population size. But in ME2 method, the difference between the best and the worst fitness values is bigger when compared to ME1 method.

Best fitness value limits of the runs versus generation number plots (see **figure 4-12**) are similar to the graphs given for population size 10. For both methods, best of the average of the best fitness value is reached for a specific N_{me} value smaller than the population size. As seen in the graph sets for both population sizes, the worst of the best fitness value among 100 runs do not get better as N_{me} increases. This is not expected since the average of the average fitness value gets better as N_{me} increases for population size 10 and same case is also seen for population size 20 (**figures 4-6** and **4-11**). This means that the difference between the fitness values of the members in the generation shall decrease as N_{me} increases. The unexpected result is due to the fact that the limits shown in **figures 4-7** and **4-12** stand for a single run. As mentioned before, due to the random characteristics of the problem a single run might show unexpected characteristics. For multiple runs, expected characteristics of the search can be observed better and the effect of the optimization parameters can be identified more clearly.

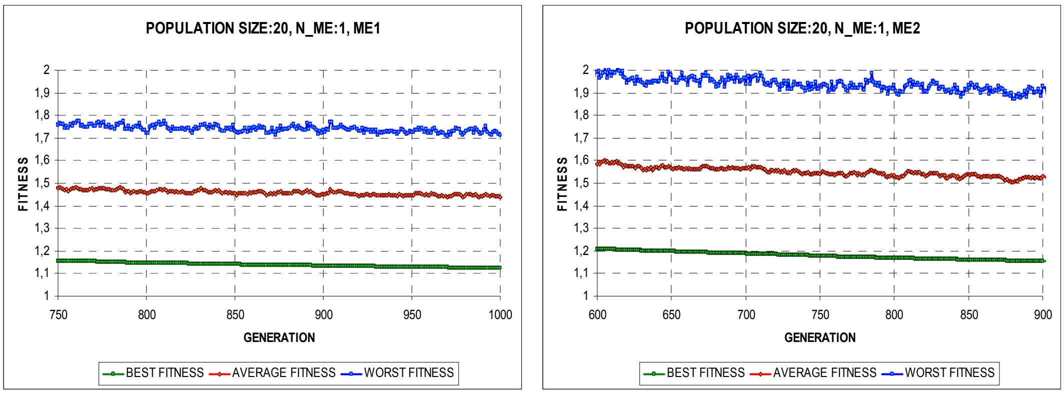


Figure 4-11a Fitness vs Generation # for Population Size : 20 (100 runs)

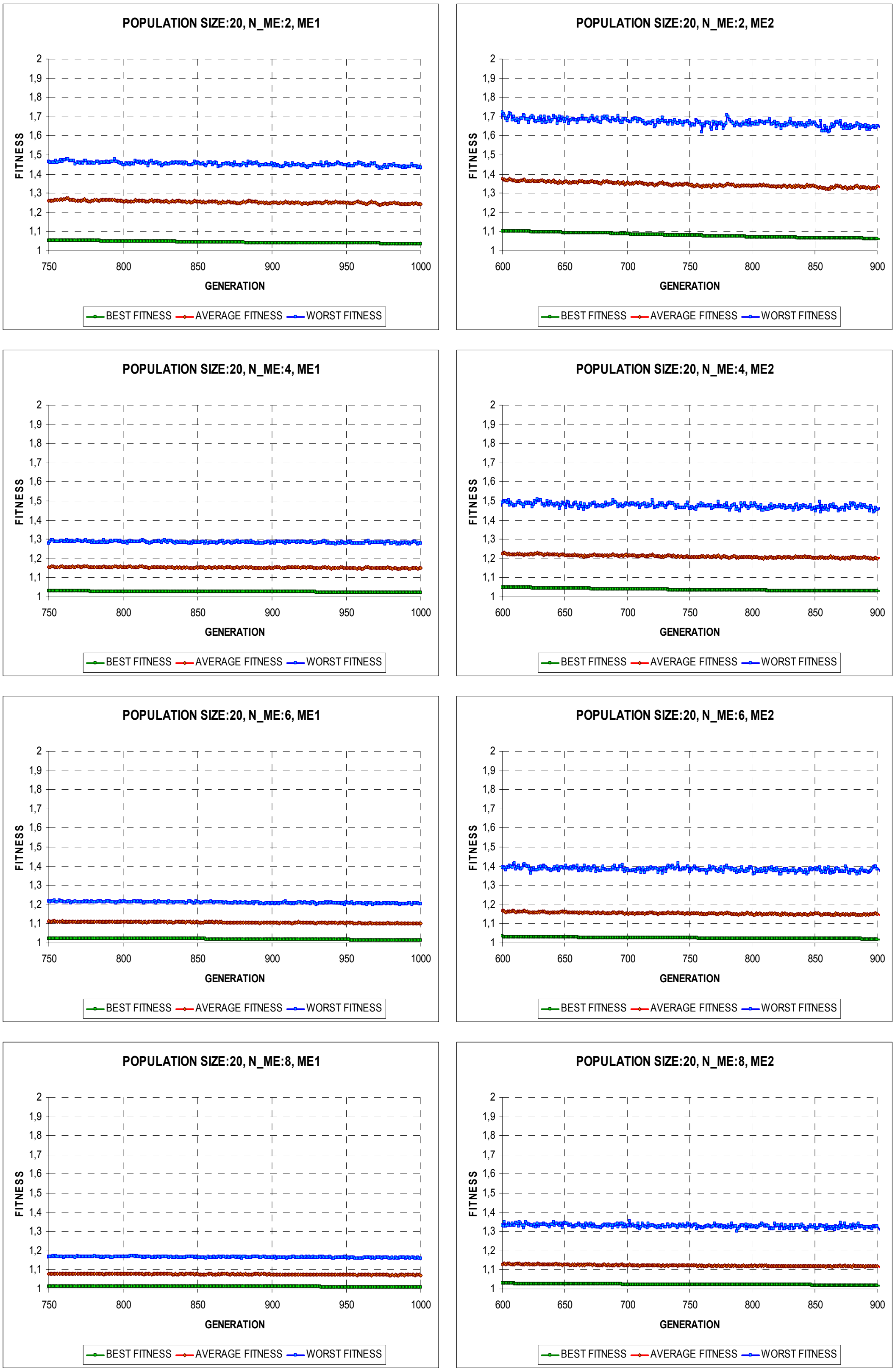


Figure 4-11b Fitness vs Generation # for Population Size : 20 (100 runs)

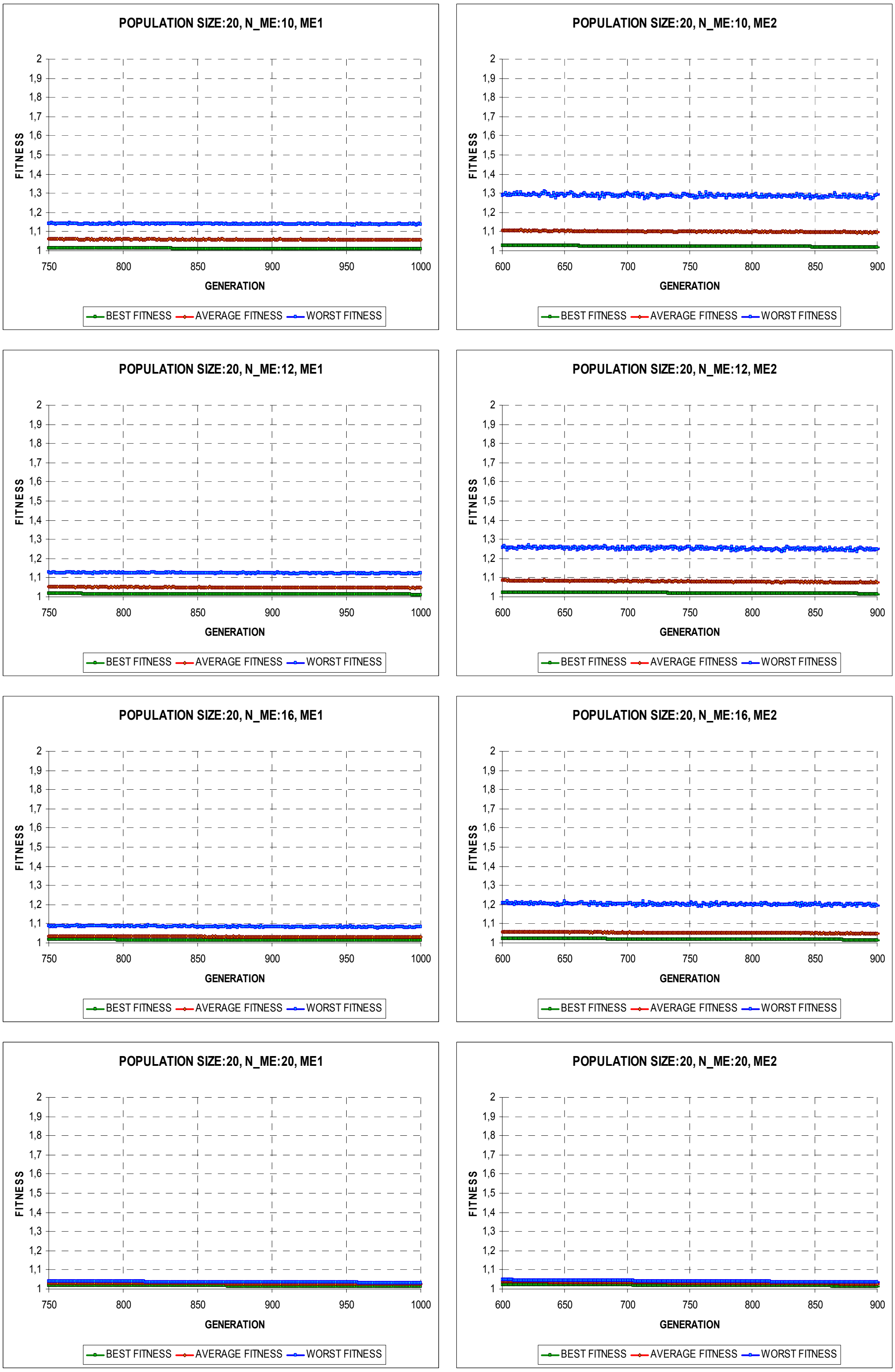


Figure 4-11c Fitness vs Generation # for Population Size : 20 (100 runs)

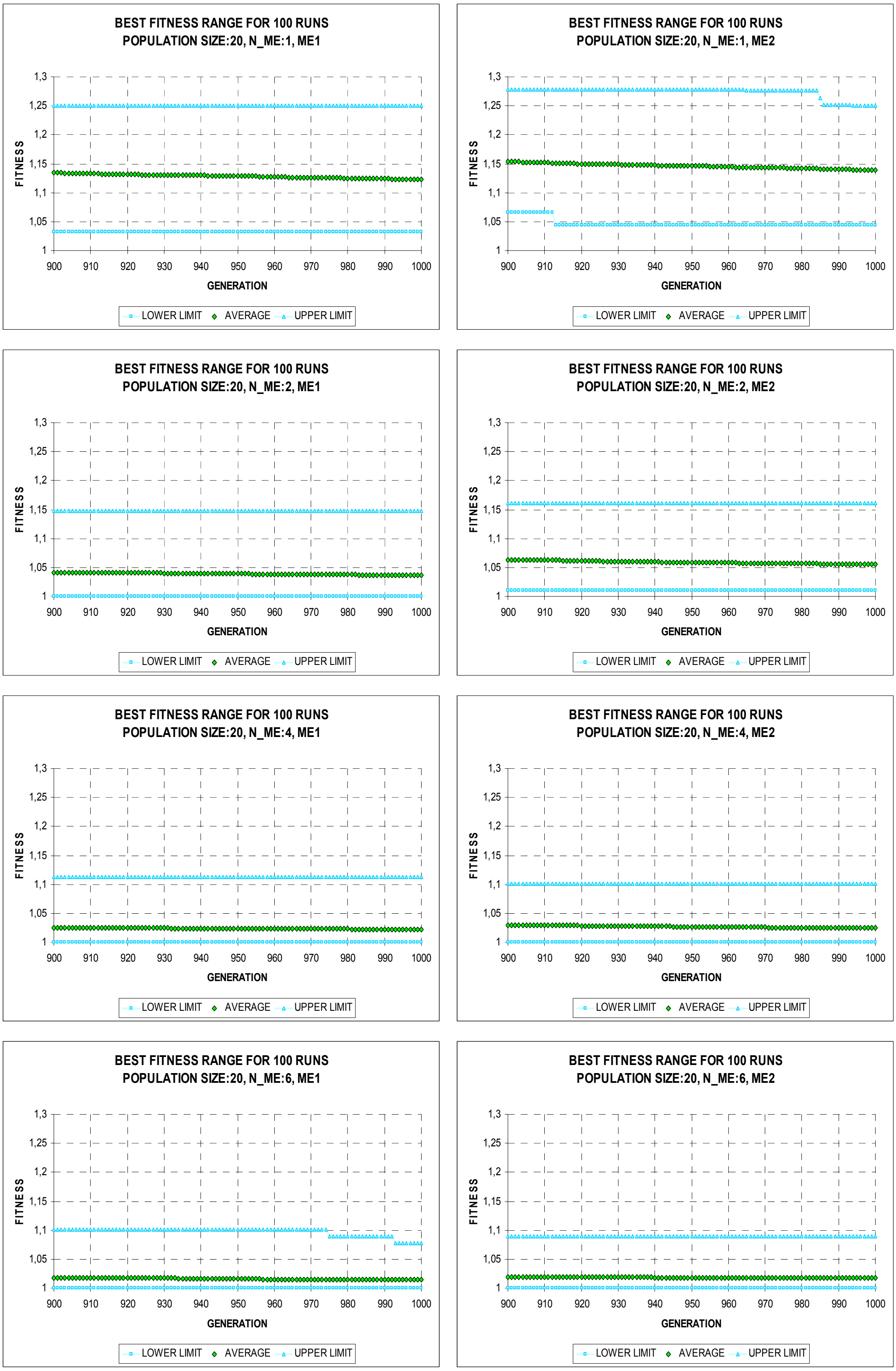


Figure 4-12a Best Fitness Range vs Generation # for Population Size : 20 (100 runs)

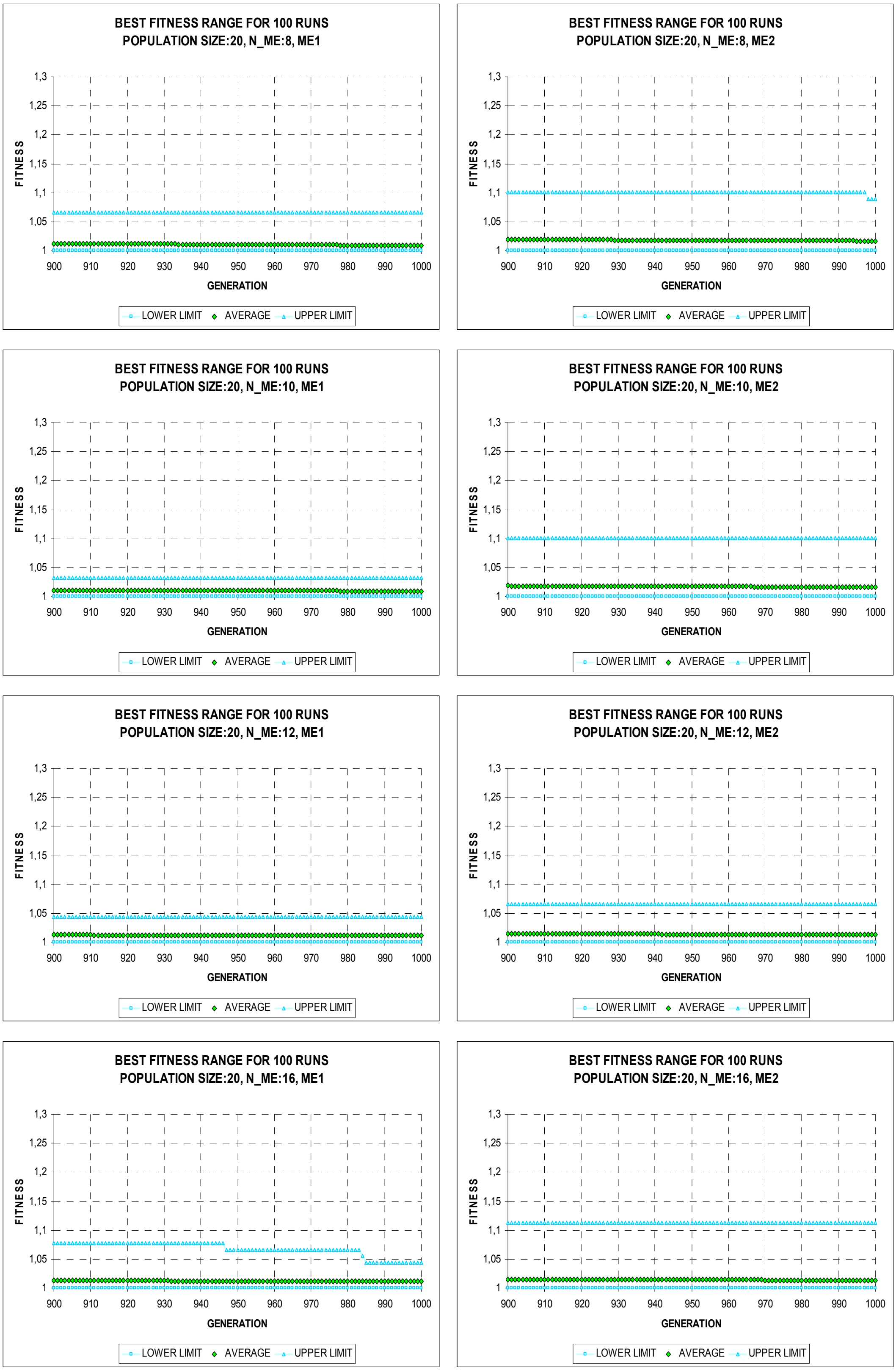


Figure 4-12b Best Fitness Range vs Generation # for Population Size : 20 (100 runs)

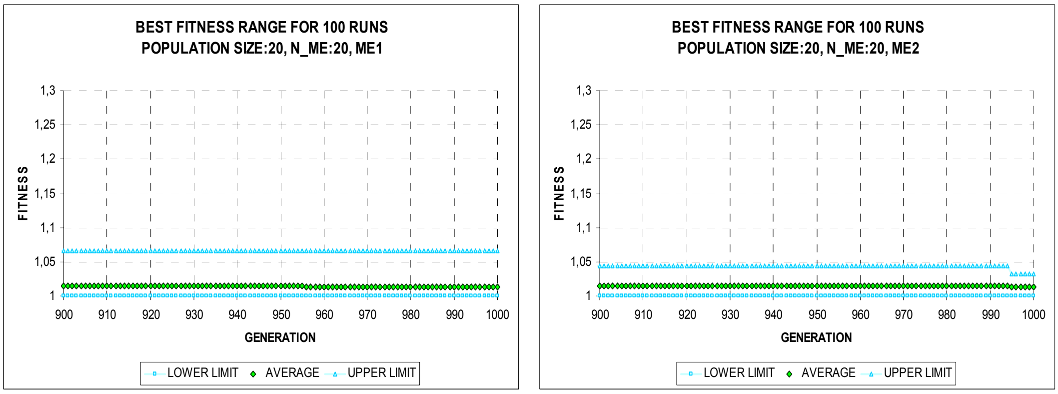


Figure 4-12c Best Fitness Range vs Generation # for Population Size : 20 (100 runs)

Reliability versus N_{me} graphs are given in **figure 4-13**. For population size 20, the best reliability value increases till a specific value of N_{me} which equals to 8 in ME1 method and 12 in ME2 method. Thus for population size 20, the optimum value of N_{me} is equal to 40 percent of the population size in ME1 method and 60 percent of the population size in ME2 method. For bigger N_{me} values, the best reliability value decreases as seen for population size 10. Again the average and the worst reliability values increase till N_{me} equals to population size and ME1 method gives better results.

The best reliability and the price of the selection methods are given in **figures 4-14** and **4-15** respectively. It is seen that for all N_{me} values, ME1 method has a better best reliability value than ME2 method. In addition, the price of ME1 is less for all N_{me} values except for N_{me} equals to 16.

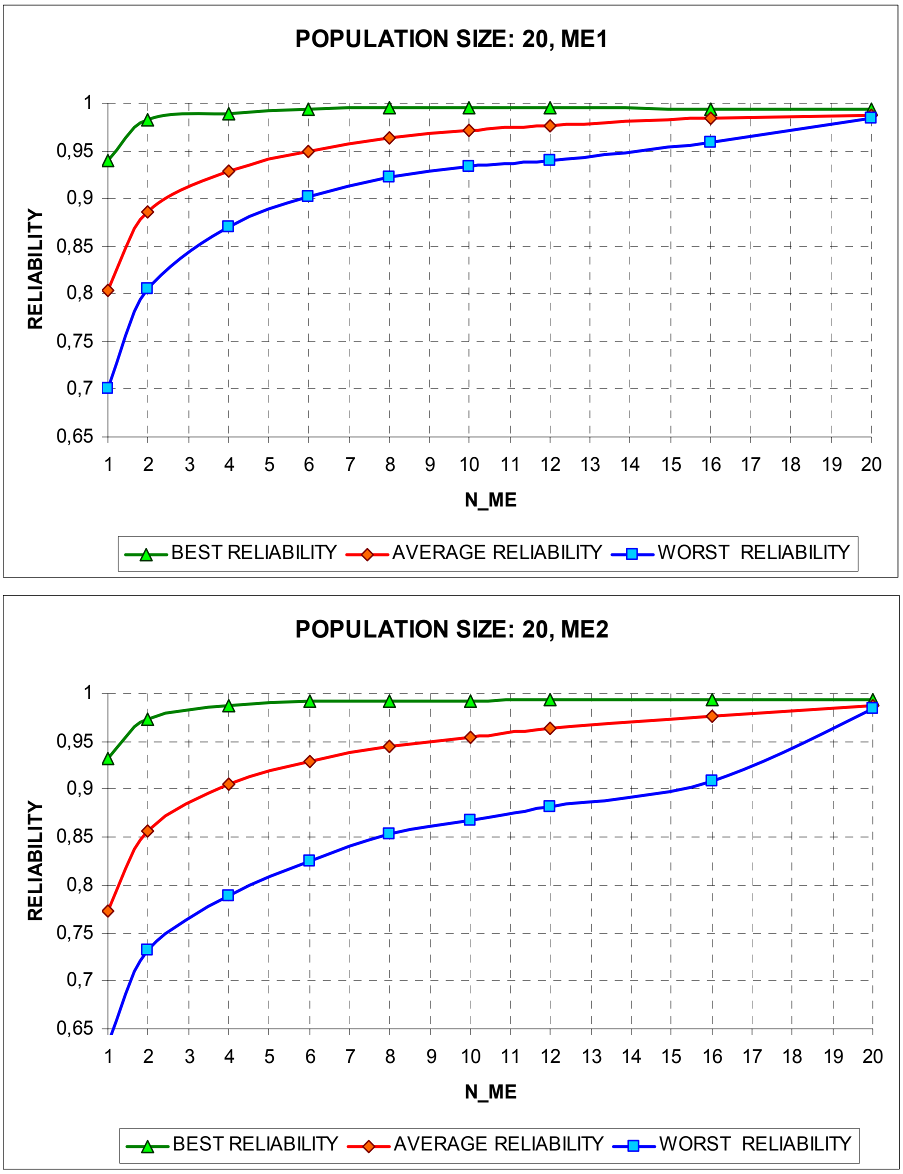


Figure 4-13 Reliability vs N_{me} for Population Size : 20 (100 runs)

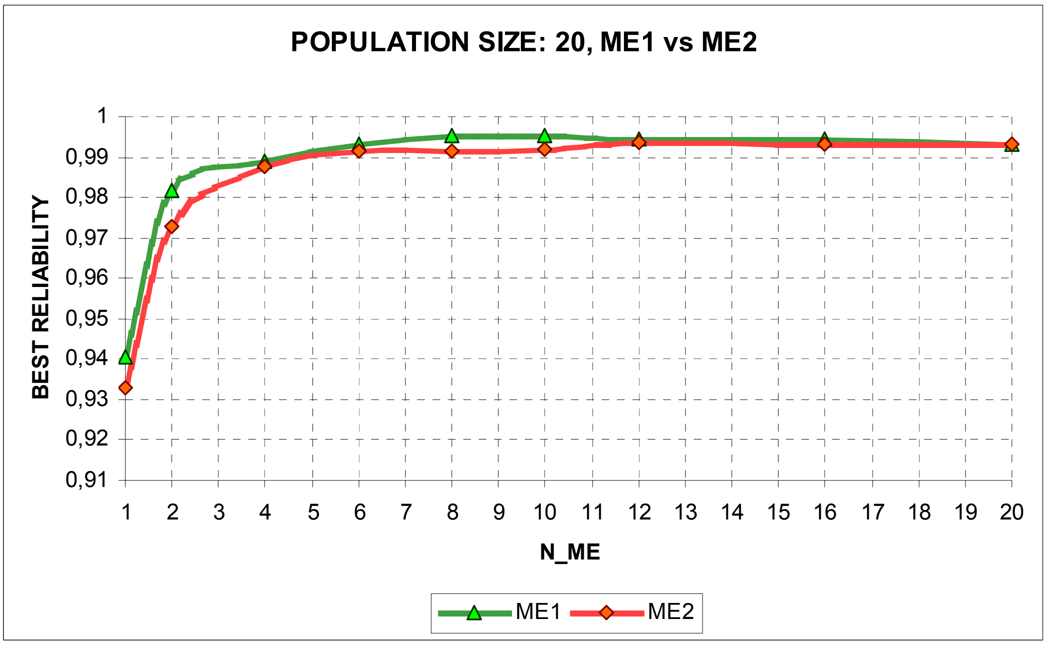


Figure 4-14 Best Reliability vs N_{me} for Population Size : 20 (100 runs)

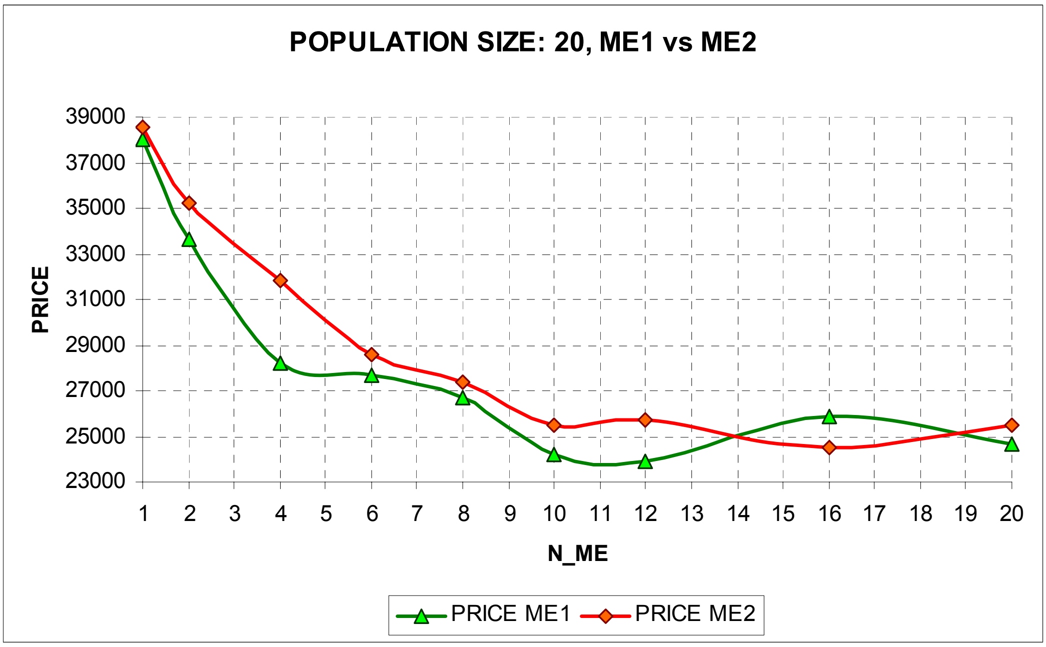


Figure 4-15 Price vs N_{me} for Population Size : 20 (100 runs)

It is seen for both methods that, using a very high N_{me} value is not advantageous and ME1 method gives better results than ME2 method. Thus in the runs for population size 40, N_{me} value is limited to the advantageous interval valid for ME1 method which is maximum 50 percent of the population size (in this case N_{me} =20).

The best, average and the worst fitness versus generation number graphs for population size 40 are given in **figures 4-16**. The results are not surprising. Both methods tend to have a closer best and worst fitness values as N_{me} gets closer to the population size and in ME2 method, the difference between the best and the worst fitness values is bigger. It is seen that as the population size increases, the best fitness value gets better whereas there is no significant difference in the average fitness value. This means that the generations have a greater variety and greater reliability at the same time, i.e. the search becomes more explorative and exploitative at the same time. The disadvantage of increasing population size is the increasing price. However, there shall be an optimization in population size as will be discussed in section **4.2.4**.

The best, average and the worst reliability versus N_{me} graphs for population size 40 are given in **figure 4-17**. As expected, all reliability values are better for ME1 method and the best and the worst reliability values get closer as N_{me} increases. In ME1 method, the best reliability value is reached for N_{me} value that is equal to 18 (45 percent of the population size). For ME2 method, the optimal value of N_{me} is 14 (35 percent of the population size) in order to have the best reliability value.

In the first plot of **figure 4-17**, best reliability values obtained from ME1 and ME2 methods are compared. As expected, best reliability values are better for ME1 method for all N_{me} values. The difference between the best reliability values is quite small for N_{me} values between 40 and 50 percent of the population size. As seen for other population sizes, the price of the search is considerably less for ME1 when compared to ME2 method (see **figure 4-18**).

Figure 4-16a Fitness vs Generation # for Population Size : 40 (100 runs)

Figure 4-16b Fitness vs Generation # for Population Size : 40 (100 runs)

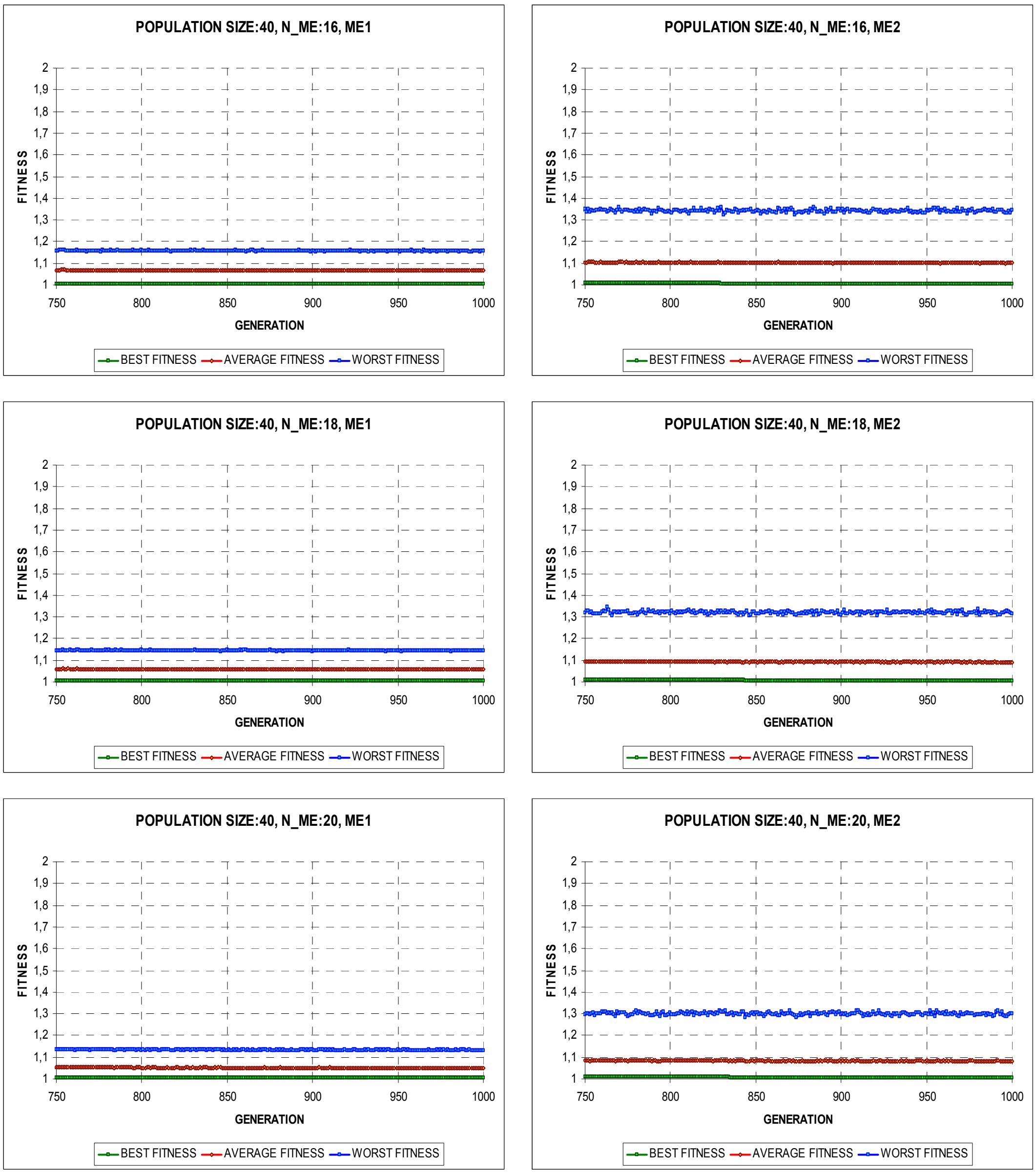


Figure 4-16c Fitness vs Generation # for Population Size : 40 (100 runs)

All the results show that ME1 is a better selection method than ME2. The best and the average reliability values are better and the price is less. As N_{me} gets closer to the population size, the best and the worst reliability values get close to each other which means that the difference between the fitness values of the members decrease. If N_{me} is equal to the population size ME1 and ME2 methods are equivalent, thus the results obtained for N_{me} equals to population size are expected to be same for both methods.

The optimum interval for N_{me} value is found to be around 40-50 percent of the population size for ME1 method. But for ME2 method, the optimum value of N_{me} decreased from 70 to 35 percent as population size increased. In ME2 method, the best N_{me} members of the combined population are carried over to the next generation and the remaining members are chosen from the child population randomly. First chosen N_{me} members identify the exploitative capability and the remaining members identify the explorative capability due to random selection. When the population size is 10, N_{me} value shall be greater in order to have the optimum exploitative capability due to small population size. If N_{me} has a small percentage value, a few number of best designs are carried over to the next generation and the search can not find the optimal point in the search area, instead it continuously looks for different interesting areas. But for the same percent value, as the population size increases, the number of the best designs carried over also increases which results in increased exploitative capability and better reliability. This situation is not valid for ME1 method. In ME1 method, the first best N_{me} members are chosen from the combined population similar to ME2, but the remaining members are chosen from the best members of the child population. This difference increases the exploitative capability of the remaining chosen members in ME1 method. Thus, the optimum percent value of N_{me} is smaller in ME1 method for small population sizes.

The last multiple run is made for population size 80, using only ME1 method. The used values for N_{me} are chosen to be the optimal values which are found from the previous runs. Chosen values are 32 and 40 (40 and 50 percent of the population size). For N_{me} equals to 32, 93 of 100 runs have found the best design; and for N_{me} equals to 40, 91 of 100 runs have found the best design. These results show that the performance of the GA code is quite good if the optimal selection method and N_{me} values are used.

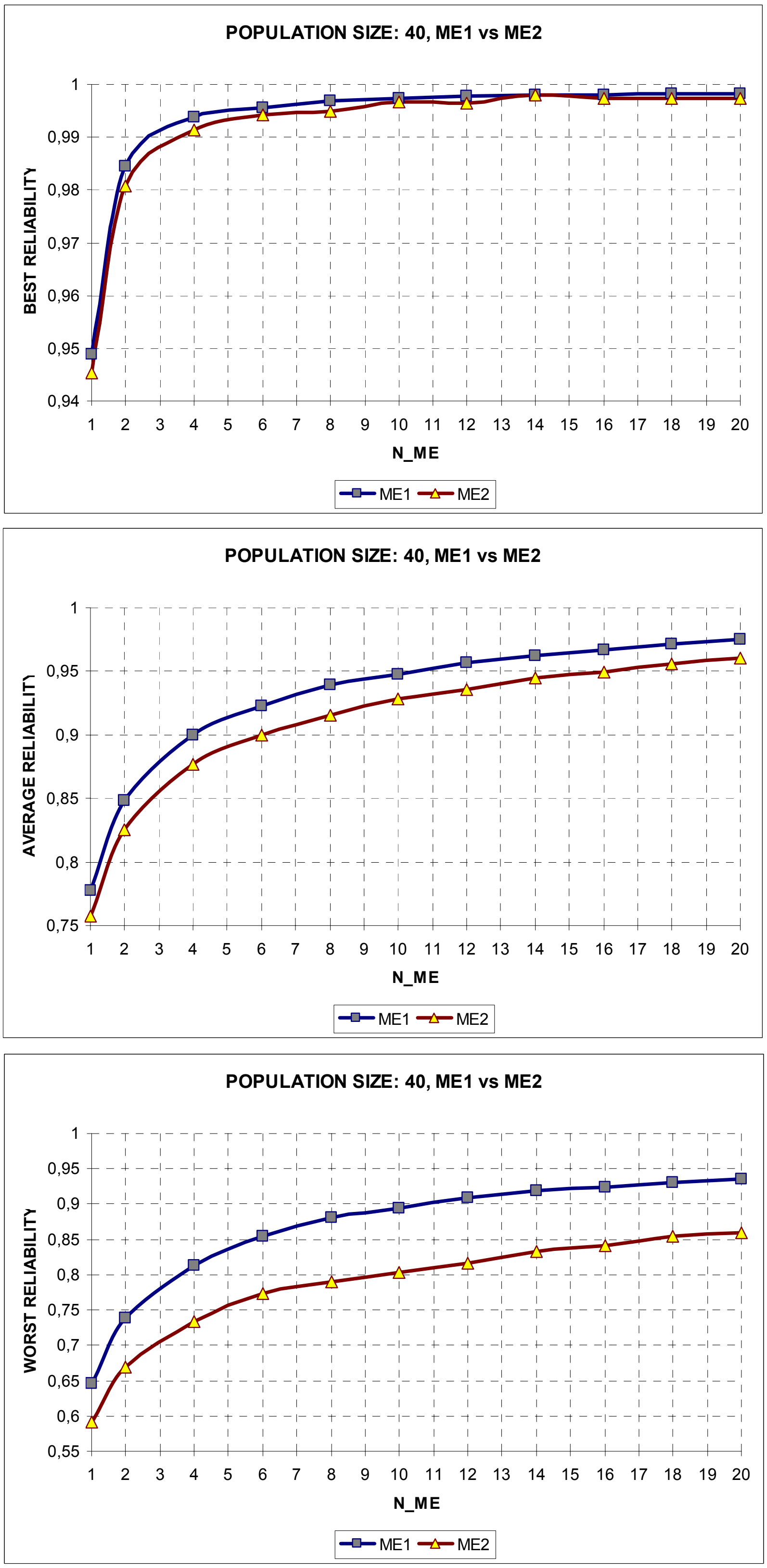
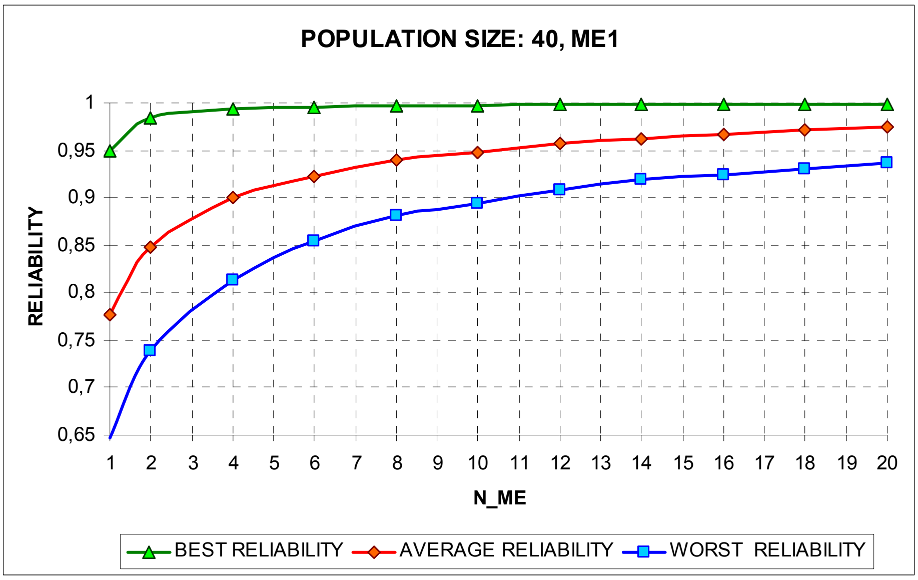


Figure 4-17a Reliability vs N_{me} for Population Size : 40 (100 runs)



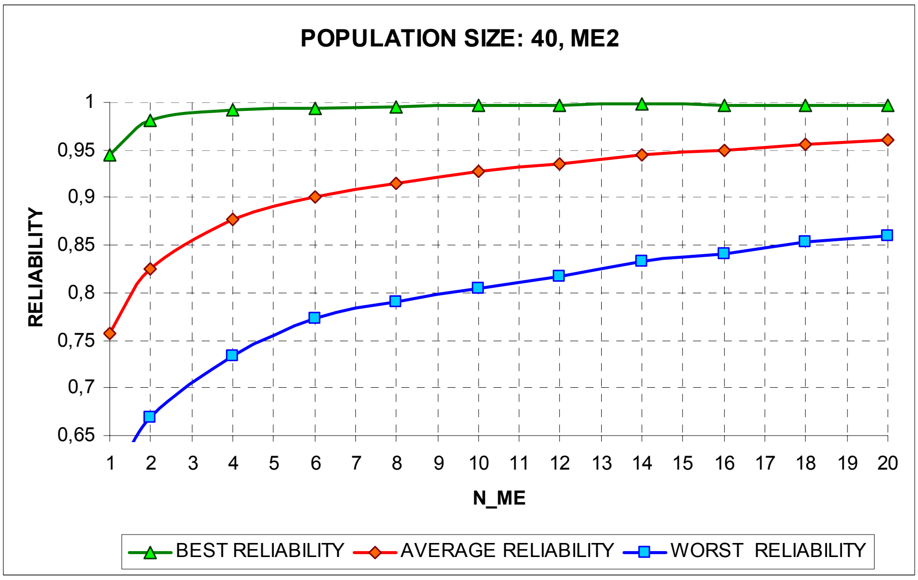


Figure 4-17b Reliability vs N_{me} for Population Size : 40 (100 runs)

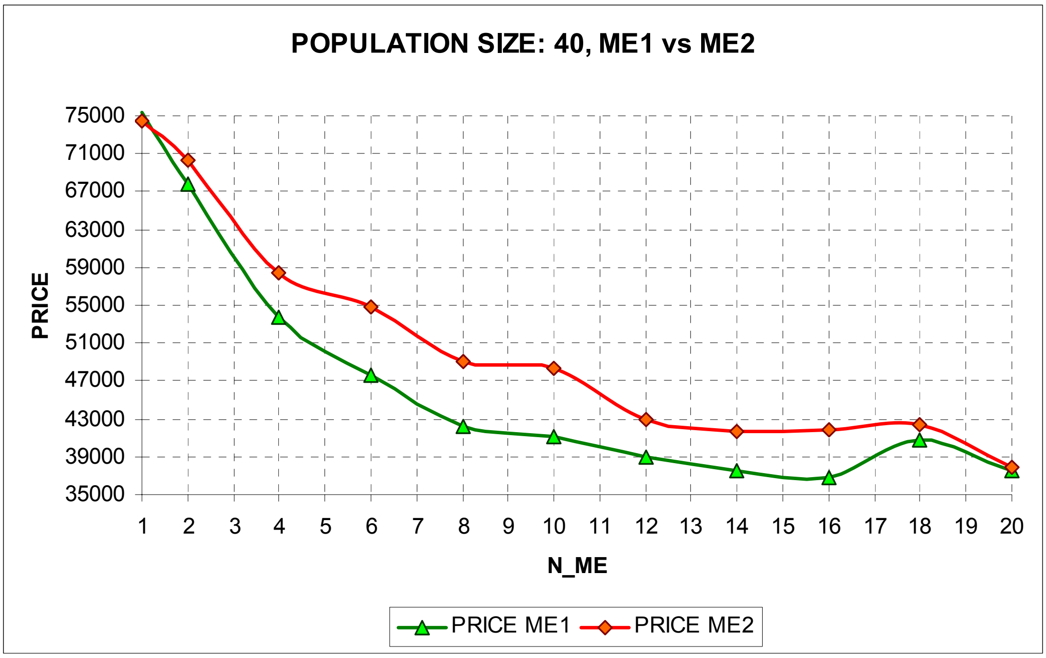


Figure 4-18 Price vs N_{me} for Population Size : 40 (100 runs)

4.2.2. Swap Probability

In this section, the optimal value range of the swap probability is investigated. The population size used in the multiple runs is 20 and ME1 method is used with a N_{me} value of 8. Number of generations is limited to 500. In all 100 runs, the used probability values for cross-over and mutation operators are 1.0 and 0.1 respectively.

The best, average and the worst reliability values versus swap probability graph is given in **figure 4-19**. It is seen that the reliability values decrease as the swap probability increase from 0.3 to 1.0. The slope is quite small, especially for the best reliability value, till swap probability equals to 0.8, but after that point differences become greater as swap probability increases. In **figure 4-20**, price versus swap probability value is given. The price of the search does not change too much for the swap probability values between 0.1 and 0.5. But for greater values the price increases significantly.

As a result of the graphs, it is not logical to use a swap probability greater than 0.8. Since the search loses exploitative capability and can not find optimal solutions due to too much alteration in the genes. A probability value between 0.3 and 0.5 seems to be logical. Although the best reliability value seems to be higher for a lower swap probability, the explorative capability is lower which may cause to converge to a local optimum point. As the number of generations increases, the best reliability value might become bigger for a bigger swap probability value. An exploitative search converges earlier when compared to an explorative search.

Figure 4-19 Reliability vs Swap Probability

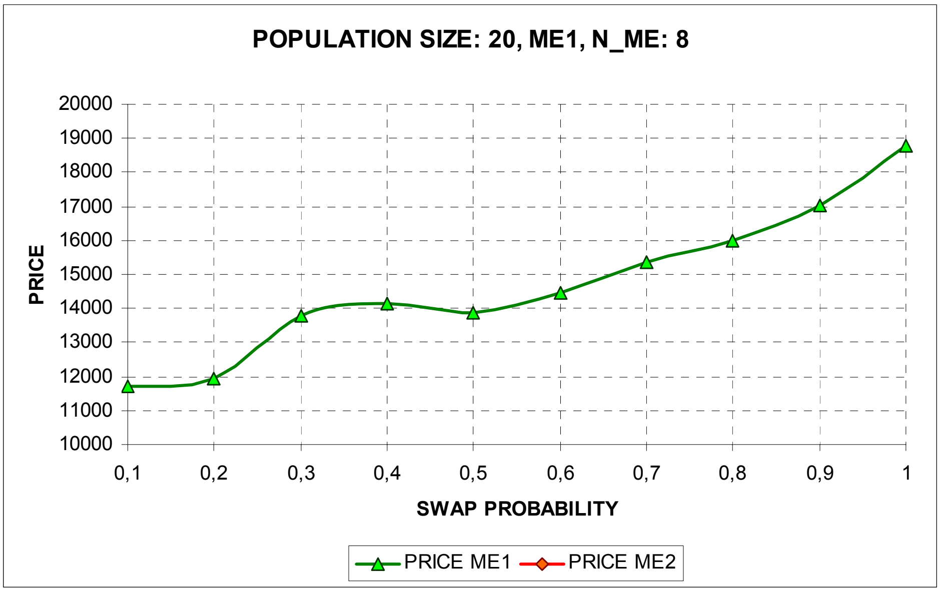


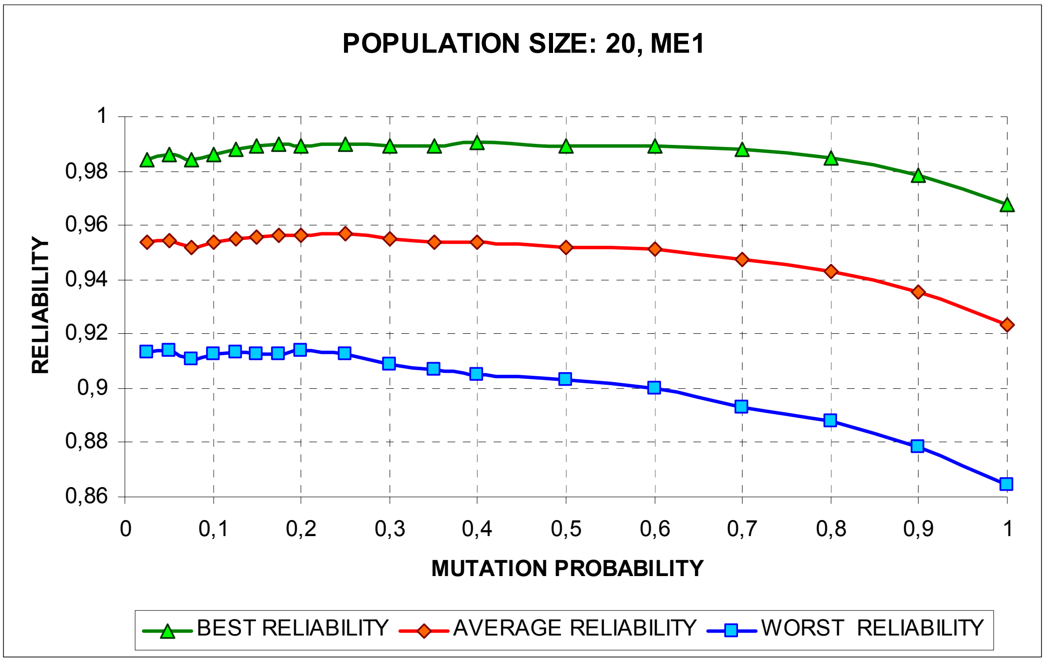
Figure 4-20 Price vs Swap Probability

4.2.3. Mutation Probability

The optimal value range of the mutation probability is examined in this section. The search parameters of the multiple runs are similar to the parameters used in the optimal swap probability runs. In all 100 runs, the population size is 20 and again ME1 method is used with N_{me} value of 8. Number of generations is limited to 500. The used probability values for cross-over and swap are 1.0 and 0.8 respectively. The swap probability value is not chosen to be in the best optimal range found in the previous section, but the chosen value is still in the logical range limit. Because, it is assumed that there is no dependency of the optimum value of any genetic parameter on the other. Hence, the swap probability value is chosen to be equal to 0.8 which is equal to the swap probability value used in the runs given in section **4.2.1**.

Reliability versus mutation probability graphs are given in **figure 4-21**. In the first graph, all reliability values are given together; and in the second graph, only the best reliability value is given in order to see the differences between the best reliability values better. As seen in the graphs, the reliability values do not change much for the mutation probability values up to 0.6. Especially the best reliability value is almost constant for the mutation probabilities between 0.3 and 0.6. But for greater values of mutation probability, the reliability values decrease significantly.

There is slight variations in the price of the search (see **figure 4-22**) for mutation probability values up to 0.5. After that point, price increases as the mutation probability increases. When the price and the reliability values are taken into account, the optimal range for mutation probability value seems to be between 0.3 and 0.5. If only the best reliability value is taken into account, the range expands to the range 0.2 to 0.6.



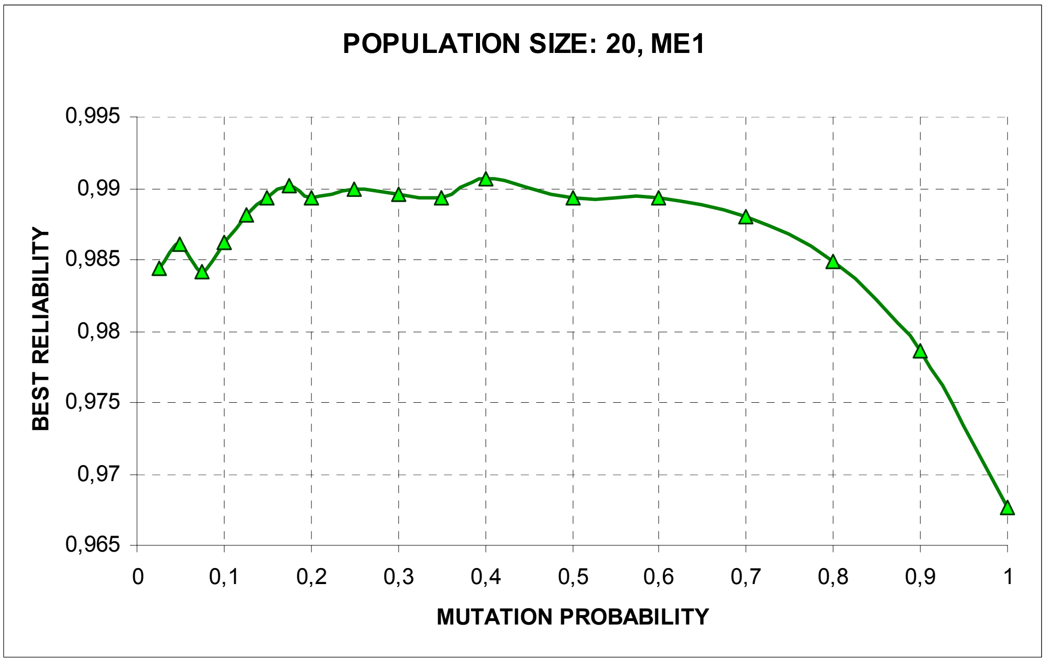


Figure 4-21 Reliability vs Mutation Probability Graphs

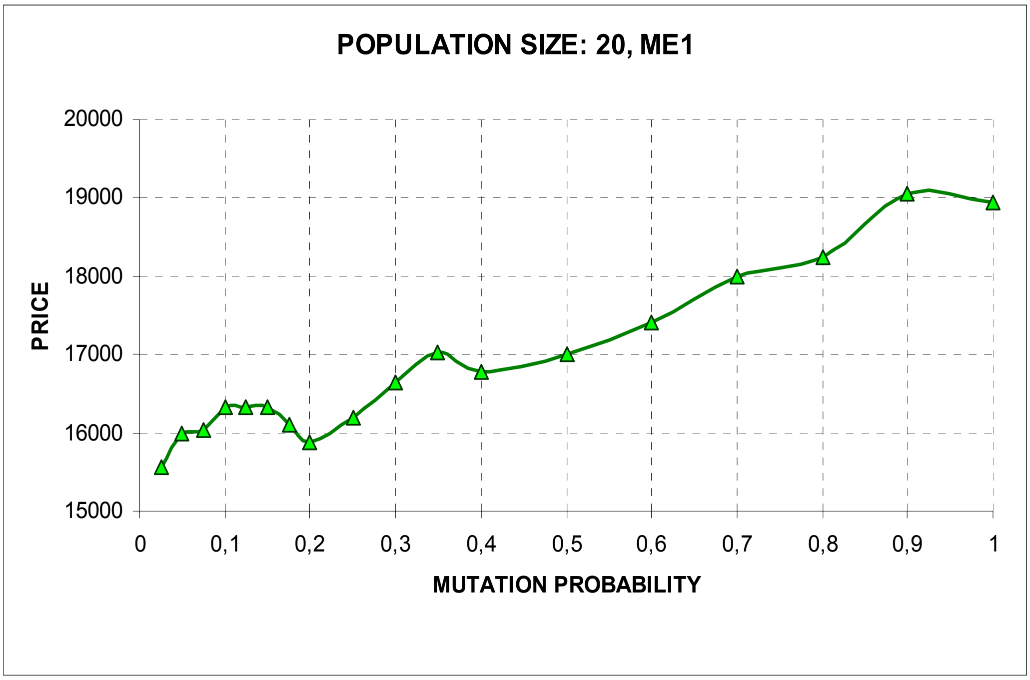


Figure 4-22 Price vs Mutation Probability

4.2.4. Population Size

Increasing the population size always gives better results for best reliability. But after an optimal point, as the population size increases, the price becomes too much. In this part of the work, the optimal population size limit for this problem is tried to be found. 100 runs are made for each population size and each run is continued until the best design is found. ME1 method is used with a N_{me} value 40% of the population size. The probability values of cross-over, mutation and swap operators used in the runs are 1.0, 0.3 and 0.5 respectively. The number of generations required to converge to the best design is saved for each search and the average value is calculated for each population size. Using these values the required price for finding the best design is calculated. Required price versus population size graph is given in **figure 4-23**. It is easily seen from the graph that the best value for the population size is 80 for this problem. With an acceptable tolerance, the optimum range for the population size can be found to be between 40 and 100, which is 80% to 200% of the chromosome length. The optimum population size limit is only given as a percent of the chromosome length. Actually, the number of possible bit values of a gene (10 for this problem) is also important for finding the optimal population size. But in this study, possible bit values of a gene do not change too much (between 3 and 10, 3 is for $0^o_2, \pm 45^o, 90^o_2$ and 10 is for $0^o_2, \pm 10^o, \pm 20^o, \ldots, \pm 80^o, 90^o_2$) and thus not taken into account. Because, the possible bit value of the chosen test problem is 10 which is equal to the maximum possible bit value for the optimization of the composite pressure vessels in this study. For a smaller possible bit values, it is expected to have a smaller optimum population size values. Thus for the bit values smaller than 10, the optimum population size found as a percentage of the chromosome length might be a little bit greater than the actual optimum one. Choosing a greater population size increases the price of the search, but the search capability of the search also increases. For a possible bit value range 3 to 10, it is assumed that the change in the optimum population size value is negligible.

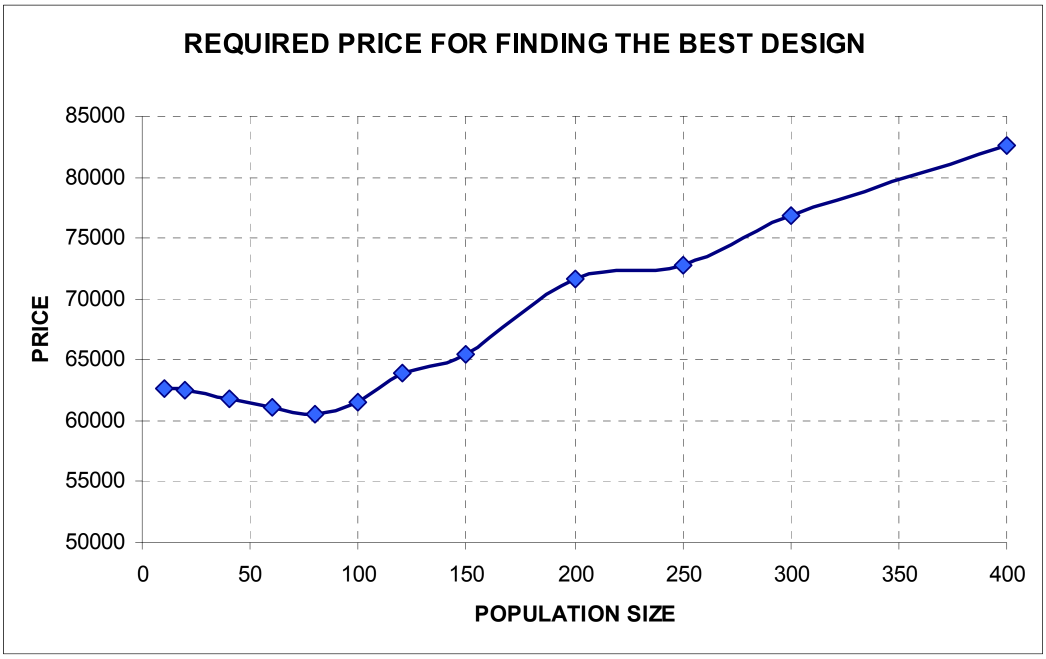


Figure 4-23 Required Price vs Population Size For Convergence to the Best Design

CHAPTER 5

RESULTS FOR COMPOSITE PRESSURE VESSEL OPTIMIZATION PROBLEMS

Code versions 2 and 3 (GAver2 and GAver3) which are developed for the stacking sequence optimization of a composite pressure vessel are combined together and a program called GACOM (**G**enetic **A**lgorithm Optimization of **COM**posites) is created in MATLAB 7 with a graphical user interface. The important features of GACOM are given in **section 5.1**. In the proceeding sections of this chapter, the results of the composite pressure vessel optimization problems are given. The detailed description of the optimization problems are given in **chapter 2**. GAver2 is developed to minimize the radial displacement at the outer surface of the vessel and GAver3 is developed to minimize the weight of the vessel under the defined loading condition.

In section **5.2**, results for GAver2 are given. The minimum displacement is searched for different loading conditions and different selection methods are compared for the combined loading condition case. The results for GAver3 are given in section **5.3**. The minimum weight is searched for the combined loading condition. Optimum ply-addition and ply-deletion probability values are examined.

5.1. GACOM Software

GACOM is a program with a graphical user interface created in Matlab 7, which contains GAver2 and GAver3 codes. Defining and running an optimization problem becomes much easier with GACOM. A snapshot of GACOM is given in **figure 5-1**. The program has significant advantageous features. Some of them are listed below.

- GACOM has physical property data for five different composite materials; and the user can also define a new material.
- Definition of each variable is clearly given in the interface. Edit boxes and popup boxes allows the user easily initialize the problem.

- User can save the input data to a file and load the data from the appropriate file. If the file is invalid, the warning message appears.
- GACOM warns the user about the invalid entry by making the corresponding box red and displays a message about the reason of the error.
- Only required input variables can be entered in the interface according to the chosen options and problem type. The unnecessary boxes are disabled according to the situation.
- The program has a progress bar that shows how many percent of the run is accomplished. By this way, user is informed about the current situation of the run and finish time of the run.
- After the run finishes, GACOM saves the results to a mat file and a text file. User can see the results of the last run by clicking the “show results” button; and make further calculations by using the “results.mat” file.

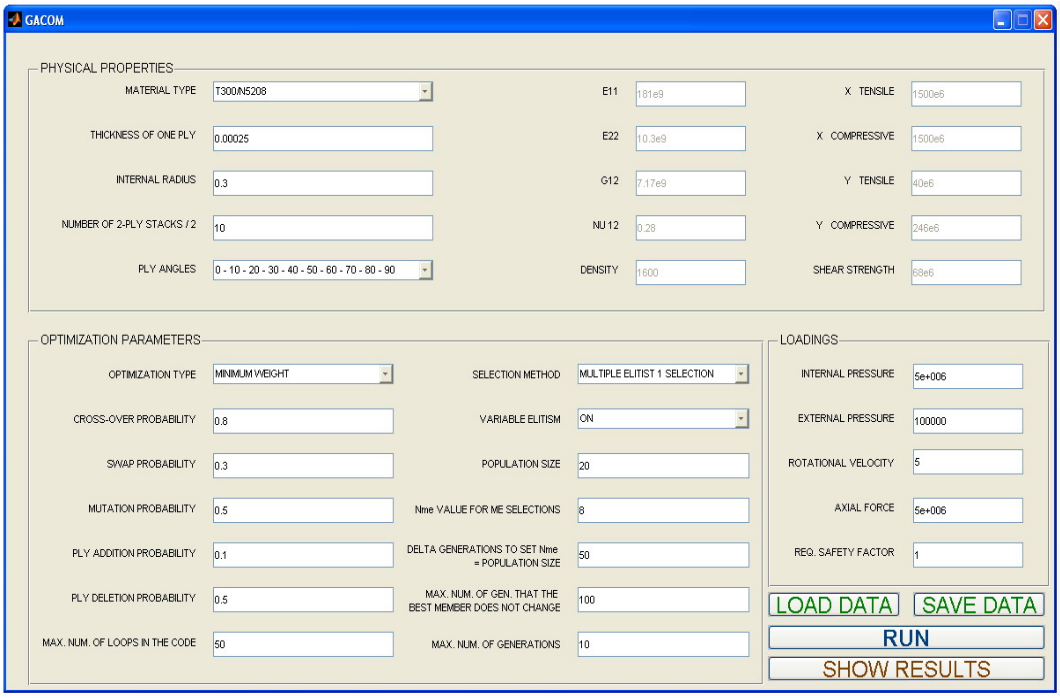


Figure 5-1 GACOM graphical user interface

5.2. Results for GA Code version 2 (GAver2): Composite Pressure Vessel Optimization for Minimum Radial Displacement

GAver2 is developed for the optimization of stacking sequence of a composite pressure vessel for the minimum radial displacement at the outer surface. The detailed description of the problem is given in section **2.1**.

In the analysis, required input parameters can be grouped in three categories which are physical properties, optimization parameters and the loading conditions (see **figure 5-1**). The detailed descriptions of the entered parameters are given below.

- **Physical properties**
  - **Material type:** In GACOM, five materials (T300/N5208, B(4)/N5505, AS 3501, E-Glass Epoxy and KEV49 Epoxy) are predefined. User can also define a new material. The required input parameters are:
    - **E11:** Modulus of elasticity in fiber direction.
    - **E22:** Modulus of elasticity perpendicular to fiber direction.
    - **G12:** Shear modulus.
    - ν **:** Poisson ratio.
    - ρ **:** Density.
    - $\sigma_{longitudinal,tensile}$ **:** Tensile strength in fiber direction.
    - $\sigma_{longitudinal,compressive}$ **:** Compressive strength in fiber direction.
    - $\sigma_{transverse,tensile}$ **:** Tensile strength perpendicular to fiber direction.
    - $\sigma_{transverse,compressive}$ **:** Compressive strength perpendicular to fiber direction.
    - σ_{shear} **:** Shear strength.
  - **Thickness of one ply:** Thickness of a single ply is entered.
  - **Number of 2-ply stacks / 2:** Quarter of the total number of plies is entered in this edit box due to symmetry and $\pm\alpha$ ply orientation restrictions.
  - **Internal radius:** Internal radius of the pressure vessel is entered.
  - **Ply angles:** Possible ply orientation angles are chosen from this combo box. Possible ply orientation angles are:
    - $0^0, \pm 45^o, \pm 90^o$

- $0^0, \pm 15^o, \pm 30^o, \pm 45^o, \pm 60^o, \pm 75^o, \pm 90^o$
- $0^0, \pm 10^o, \pm 20^o, \pm 30^o, \pm 40^o, \pm 50^o, \pm 60^o, \pm 70^o, \pm 80^o, \pm 90^o$

– **Optimization Parameters:**
  - **Optimization type :** Type of the optimization problem is chosen from this combo box. For GA*ver*02, "minimum radial displacement" is chosen.
  - **Cross-over probability:** Probability of the cross-over operator is entered.
  - **Swap probability:** Probability of the swap operator is entered.
  - **Mutation probability:** Probability of the mutation operator is entered.
  - **Selection method:** Selection method type is chosen from this combo box. The selection methods are elitist, multiple elitist 1 and multiple elitist 2 selection methods.
  - **Variable elitism:** This is a switch whether variable elitism is "on" or "off". The switch is enabled only for multiple elitist selections.
  - **Population size:** The population size of the GA search is entered.
  - N_{me} **:** This parameter is required only for multiple elitist selection methods (see **section 2.1.5.2.** for detailed description of N_{me}).
  - **Delta generations to set** N_{me} **= population size:** This edit box is enabled only for variable multiple elitist selection methods. N_{me} is set to the population size in case of no improvement in the best fitness value for the entered number of generations.
  - **Maximum allowed number of generations with no improvement in best fitness value:** The search is stopped if the best fitness value does not change for the entered number of generations.
  - **Maximum number of generations:** Maximum number of allowed generations for the search is entered in this edit box.

– **Loadings:** The loadings acting on the pressure vessel is entered in the corresponding edit boxes. The loadings are:
  - **Internal pressure**
  - **External pressure**
  - **Rotational velocity**
  - **Axial force**
  - **Required safety factor:** Minimum required safety factor of the vessel under the defined loading condition is entered in this edit box.

GACOM results are given for different type of loading conditions. Search parameters are chosen to be in the optimum range which was tried to be found in **chapter 4**.

5.2.1. Pure internal pressure

In first case, only 5 MPa of internal pressure is applied to the pressure vessel. The input parameters can be summarized as follows:

- Material : T300/N5208
- Ply thickness : 0.00025 m
- Internal radius : 0.3 m
- Half of the number of 2-ply stacks : 20
- Ply angles : $0^0, \pm 10^o, \pm 20^o, \pm 30^o, \pm 40^o, \pm 50^o, \pm 60^o, \pm 70^o, \pm 80^o, \pm 90^o$
- Cross-over probability : 0.8
- Swap probability : 0.3
- Mutation probability : 0.5
- Selection method : ME1
- Population size : 30
- N_{me} : 12
- Maximum allowed number of generations with no improvement in best fitness value : 100
- Maximum number of generations : 500
- Required safety factor : 1

After 49 generations, the best design is found to be:

$$\left\{\begin{matrix} 10 & 10 & 9 & 10 & 10 & 9 & 10 & 10 & 9 & 10 & 10 & 9 & 10 & 10 & 9 & 10 & 10 & 9 & 10 & 9 \\ 90_2^o & 90_2^o & \pm 80^o & 90_2^o & 90_2^o & \pm 80^o & 90_2^o & 90_2^o & \pm 80^o & 90_2^o & 90_2^o & \pm 80^o & 90_2^o & 90_2^o & \pm 80^o & 90_2^o & 90_2^o & \pm 80^o & 90_2^o & \pm 80^o \end{matrix}\right\}_s$$

Minimum displacement versus generation number plot is given in **figure 5-2**. The best fitness value (minimum displacement) and the safety factor of the best design is found to be 0.128 mm and 1.194 respectively. It is seen from the results that the best fiber orientation angle for the pure internal pressure loading condition is 90_2^o as expected. Due to the contiguity constraint, no more than two 90_2^o stacks can be combined. Thus, there exist some $\pm 80^o$ plies. If there was not any contiguity constraint, the fiber orientation angle for all the plies would be 90_2^o.

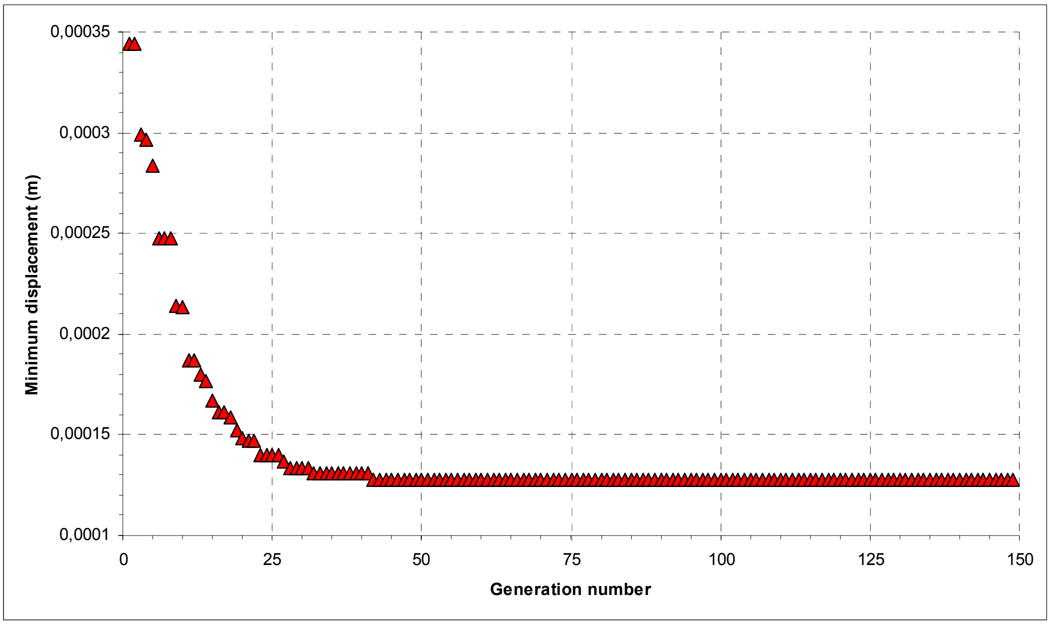


Figure 5-2 Minimum displacement vs generation number (pure internal pressure)

5.2.2. Pure rotational velocity

All the input parameters are same as the parameters of the previous case except for the loading. In this case, vessel has a constant rotational velocity of 600 rad/sec (approximately 95.5 rps). Minimum displacement variation is given in **figure 5-3**.

Search is stopped after 214 generations. The best design found is:

$$\left\{ \begin{matrix} 10 & 1 & 10 & 1 & 10 & 10 & 1 & 10 & 10 & 1 & 10 & 10 & 1 & 10 & 10 & 1 & 10 & 10 & 1 & 10 \\ 90_2^o & 0_2^o & 90_2^o & 0_2^o & 90_2^o & 90_2^o & 0_2^o & 90_2^o & 90_2^o & 0_2^o & 90_2^o & 90_2^o & 0_2^o & 90_2^o & 90_2^o & 0_2^o & 90_2^o & 90_2^o & 0_2^o & 90_2^o \end{matrix} \right\}_s$$

Displacement of the best ever found design is 6.2645e-005 m and the corresponding safety factor is 10.112. It is seen that most of the ply orientation angles are 90_2^o , but there are some 0_2^o stacks. To test whether the 0_2^o stacks are optimum, a set of analytic runs are made in which the orientation angle of 0_2^o stacks in the optimum design change from 0_2^o to 90_2^o as shown below. The variation of the displacement at the outer surface of the vessel and the safety factor with the orientation angle is given in **figure 5-4** and **figure 5-5** respectively.

$$\{10\ \ 1\ \ 10\ \ 1\ \ 10\ \ 10\ \ 1\ \ 10\ \ 10\ \ 1\ \ 10\ \ 10\ \ 1\ \ 10\ \ 10\ \ 1\ \ 10\ \ 10\ \ 1\ \ 10\}_s$$
$$\{10\ \ 2\ \ 10\ \ 2\ \ 10\ \ 10\ \ 2\ \ 10\ \ 10\ \ 2\ \ 10\ \ 10\ \ 2\ \ 10\ \ 10\ \ 2\ \ 10\ \ 10\ \ 2\ \ 10\}_s$$
$$\ldots$$
$$\{10\ \ i\ \ 10\ \ i\ \ 10\ \ 10\ \ i\ \ 10\ \ 10\ \ i\ \ 10\ \ 10\ \ i\ \ 10\ \ 10\ \ i\ \ 10\ \ 10\ \ i\ \ 10\}_s$$
$$\ldots$$
$$\{10\ \ 10\ \ 10\ \ 10\ \ 10\ \ 10\ \ 10\ \ 10\ \ 10\ \ 10\ \ 10\ \ 10\ \ 10\ \ 10\ \ 10\ \ 10\ \ 10\ \ 10\ \ 10\ \ 10\}_s$$

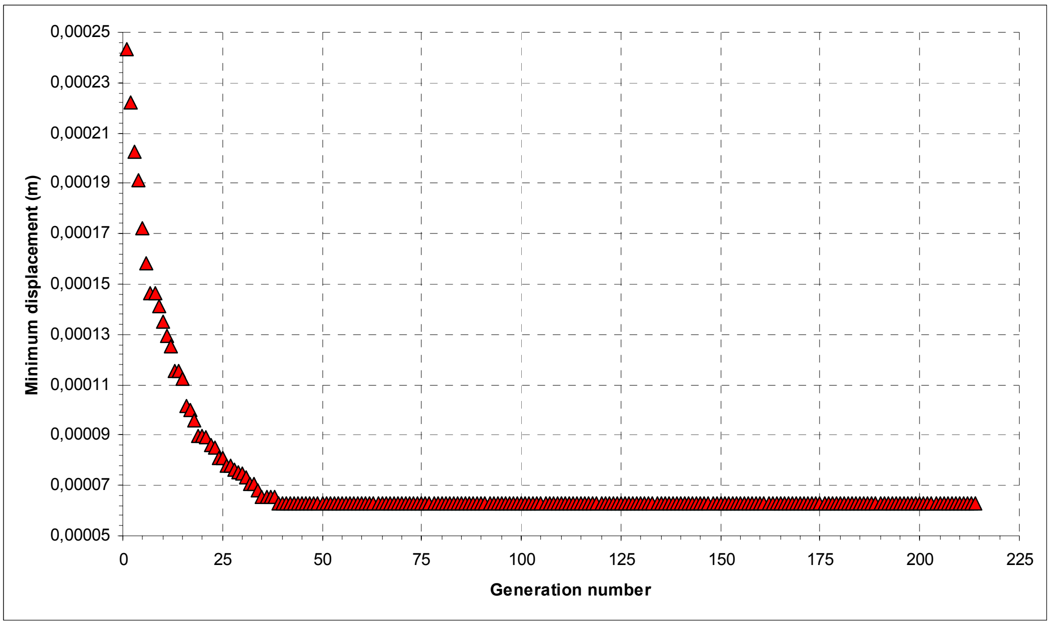


Figure 5-3 Minimum displacement vs generation number (pure rotation)

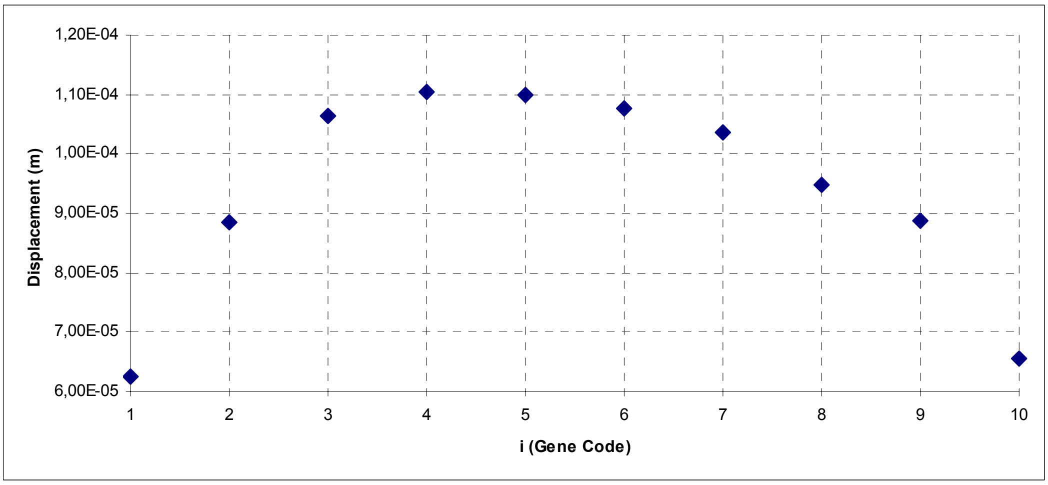


Figure 5-4 Displacement vs orientation angle code (i) (pure rotation)

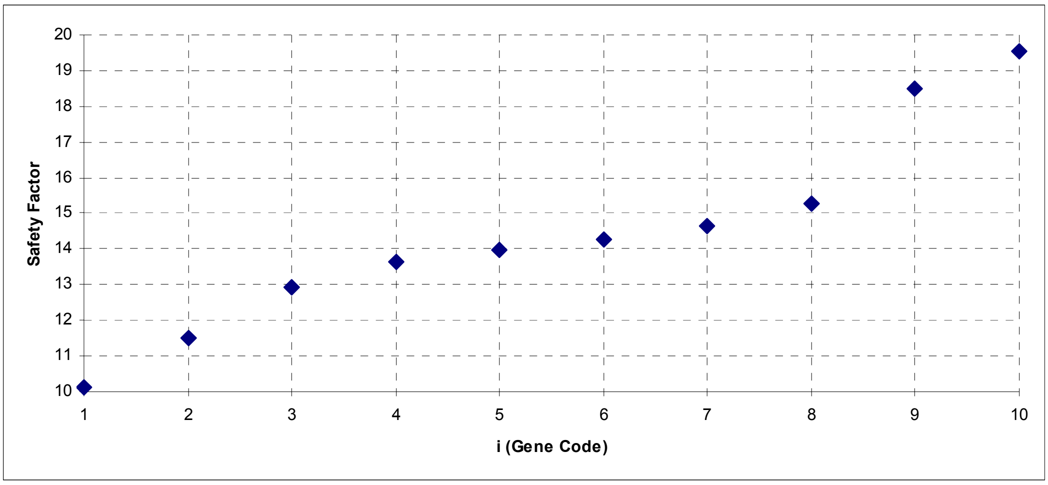


Figure 5-5 Safety factor vs orientation angle code (i) (pure rotation)

It is seen from the graphs that the minimum displacement is achieved if 0_2^o stacks are used. But it is also seen that the safety factor of the design is worst when compared to the other designs. Since the safety factor of the best design is still good enough there is no effect on the objective function. Thus the best found design seems to be optimal for the defined objective function.

5.2.3. Pure axial force

In this case, only 5000 kN of axial force acts on the vessel. The remaining input parameters are again same as the parameters of the previous cases. Minimum displacement versus generation number plot is given below.

Best design found after 208 generations is:

$$\begin{Bmatrix} 1 & 1 & 10 & 1 & 1 & 10 & 1 & 1 & 10 & 1 & 10 & 10 & 1 & 10 & 10 & 1 & 10 & 10 & 1 & 10 \\ 0_2^o & 0_2^o & 90_2^o & 0_2^o & 0_2^o & 90_2^o & 0_2^o & 0_2^o & 90_2^o & 0_2^o & 90_2^o & 90_2^o & 0_2^o & 90_2^o & 90_2^o & 0_2^o & 90_2^o & 90_2^o & 0_2^o & 90_2^o \end{Bmatrix}_s$$

Radial displacement at the outer surface of the vessel under the defined axial load and the safety factor are calculated to be 1.32e-005 m and 5.31 respectively. The best design has mostly 0_2^o stacks at the outer sides and 90_2^o stacks at the middle. Similar to the previous case, displacement and the safety factor of the designs that have different orientation angles are calculated by using the analytic function. 90^o_2 stacks of the best found design are changed from 0^o_2 to 90^o_2 as shown below. The variation of the displacement and the safety factor with the orientation angle is given in **figure 5-7** and **figure 5-8** respectively.

$$\begin{array}{l}
\{1\ \ 1\ \ 10\ \ 1\ \ 1\ \ 10\ \ 1\ \ 1\ \ 10\ \ 1\ \ 10\ \ 10\ \ 1\ \ 10\ \ 10\ \ 1\ \ 10\ \ 10\ \ 1\ \ 10\}_s \\
\{1\ \ 1\ \ 9\ \ 1\ \ 1\ \ 9\ \ 1\ \ 1\ \ 9\ \ 1\ \ 9\ \ 9\ \ 1\ \ 9\ \ 9\ \ 1\ \ 9\ \ 9\ \ 1\ \ 9\}_s \\
\qquad\qquad\qquad\qquad\qquad \ldots \\
\{1\ \ 1\ \ i\ \ 1\ \ 1\ \ i\ \ 1\ \ 1\ \ i\ \ 1\ \ i\ \ i\ \ 1\ \ i\ \ i\ \ 1\ \ i\ \ i\ \ 1\ \ i\}_s \\
\qquad\qquad\qquad\qquad\qquad \ldots \\
\{1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\}_s
\end{array}$$

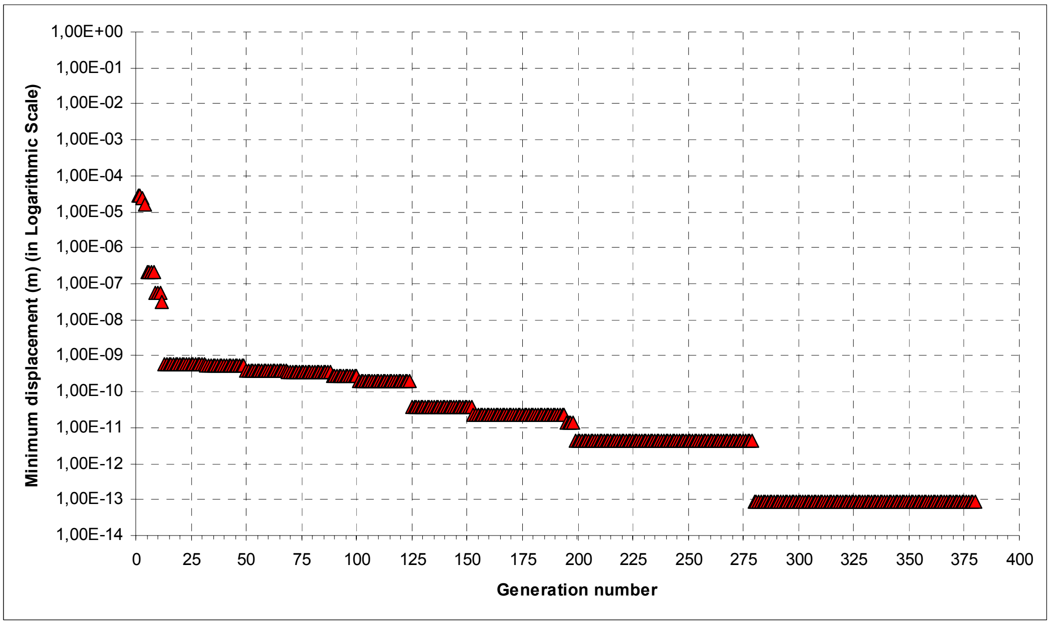


Figure 5-6 Minimum displacement vs generation number (pure axial force)

The results are similar to the previous case. The minimum displacement is achieved by using 90^o_2 stacks, but the safety factor of the design is the worst when compared to other designs. Although the best found design does not have the best safety factor, it still satisfies the safety factor constraint and is better than the other designs due the smaller displacement value.

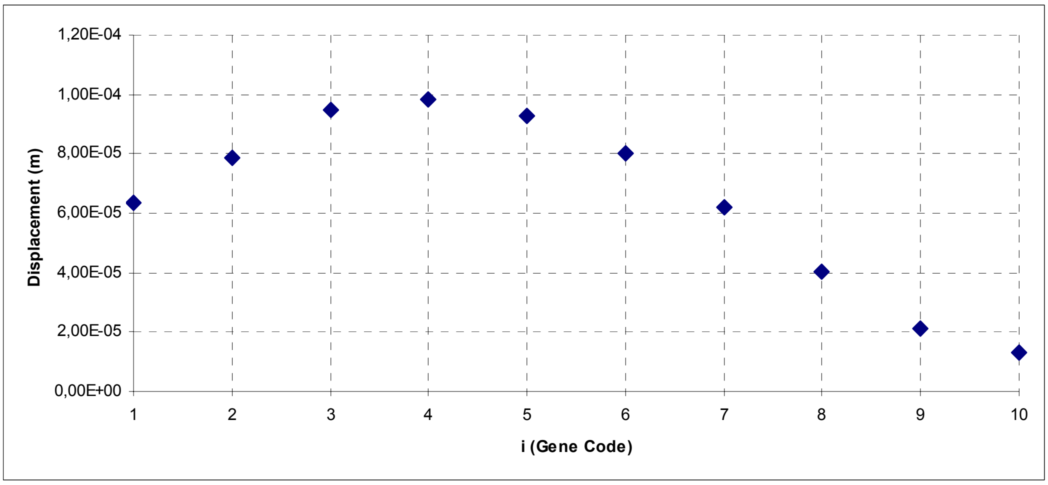


Figure 5-7 Displacement vs orientation angle code (i) (pure axial force)

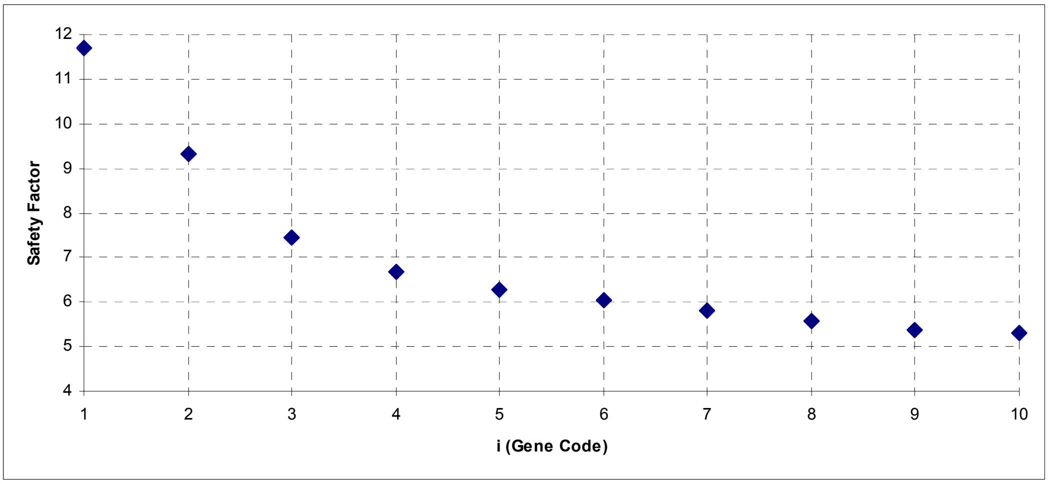


Figure 5-8 Safety factor vs orientation angle code (i) (pure axial force)

5.2.4. Combined loading

In the last case, best design is searched under the combined loading condition. The operator probability values are equal to the probability values of the previous cases. The loadings are given below:

- Internal pressure: 500 kPa
- External pressure: 100 kPa
- Rotational velocity: 60 rad/s
- Axial Force: 7000 kN

In this case, best design is searched for different selection methods and different N_{me} values to see the behavior of the search. The variation of the radial displacement is given in **figure 5-9**.

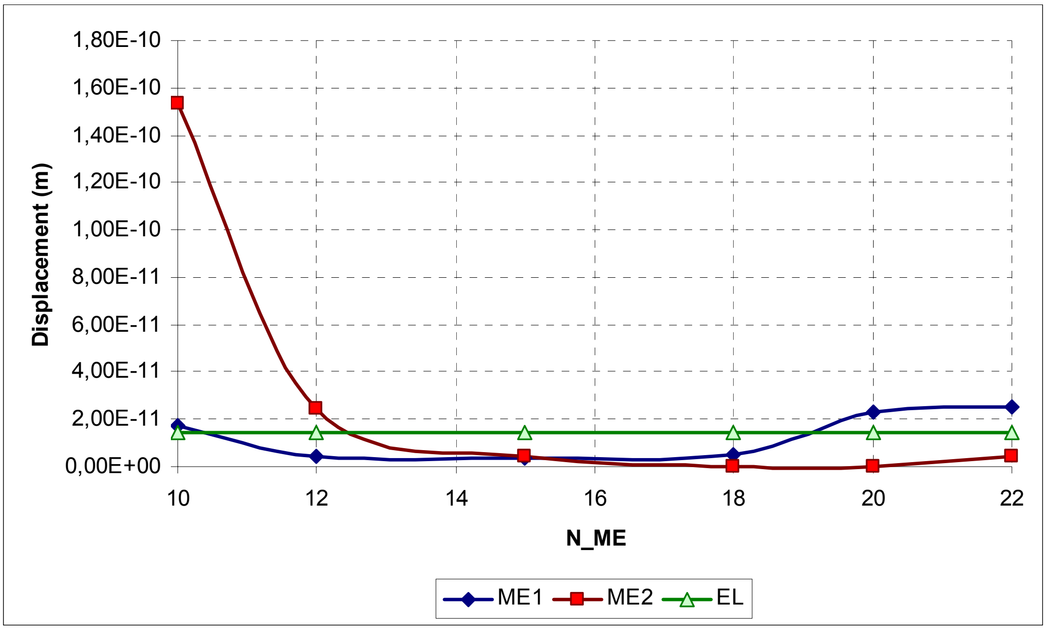


Figure 5-9 Minimum displacement vs N_{me} for different selection methods

As seen from figure, ME1 and ME2 methods are better than EL method when N_{me} value is in the optimal range. For ME1 method, the optimal range of N_{me} is found to be between 12 and 18 (40% to 60% of population size) where the best result is obtained for N_{me} equal to 15 (50% of the population size). For ME2 method, the optimal range of N_{me} is found to be between 15 and 22 (50% to 73% of population size) where the best result is obtained for N_{me} equal to 18 (60% of the population size). The results of ME1 and ME2 methods are quite close in the optimal N_{me} value range, but ME2 method gives better results. The best design is found by using ME2 method with a N_{me} value of 18. Minimum displacement versus generation number plot is given in **figure 5-10**. Displacement axis of the plot is chosen to be in logarithmic scale due to big differences in displacement. Search has converged after 280 generations and stopped at generation number 380. The stacking sequence of the best design is found to be:

$$\left\{\begin{array}{cccccccccc} 2 & 2 & 1 & 5 & 4 & 1 & 1 & 7 & 9 & 8 \; \ldots \\ \pm 10^o & \pm 10^o & 0_2^o & \pm 40^o & \pm 30^o & 0_2^o & 0_2^o & \pm 60^o & \pm 80^o & \pm 70^o \; \ldots \\ 2 & 9 & 1 & 1 & 3 & 2 & 2 & 4 & 3 & 8 \\ \pm 10^o & \pm 80^o & 0_2^o & 0_2^o & \pm 20^o & \pm 10^o & \pm 10^o & \pm 30^o & \pm 20^o & \pm 70^o \end{array}\right\}_s$$

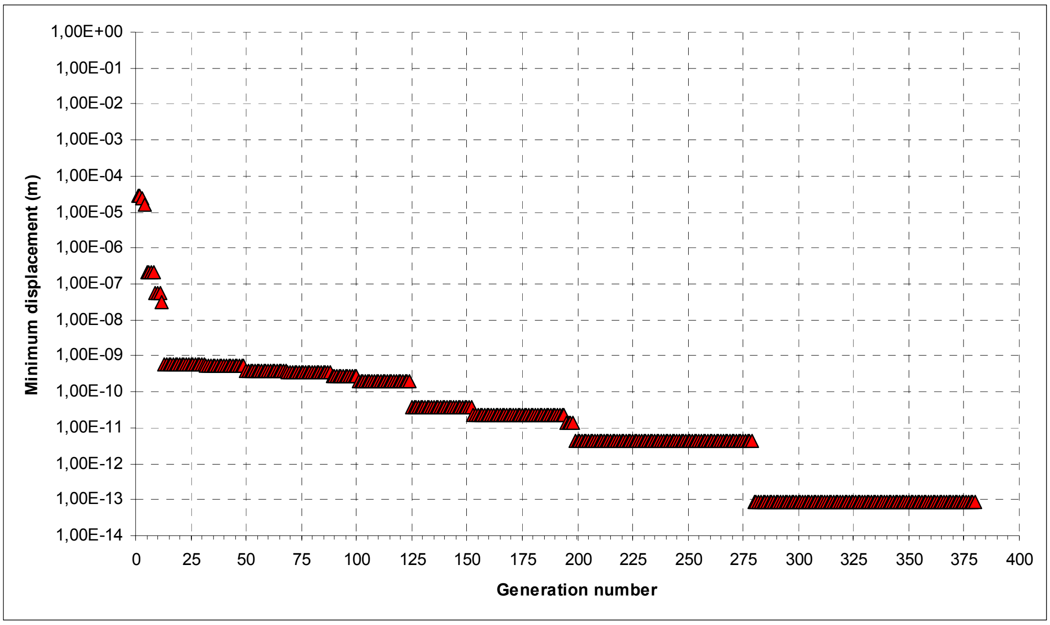


Figure 5-10 Minimum displacement vs generation number (combined loading)

As it can be seen from the results, it is quite difficult to predict the optimum design for the combined loading condition. In such cases, it is the best way to use genetic algorithm optimization tools to find the optimal design. The displacement of the best ever found design under the defined loading condition is 8.84e-014 m. The safety factor of the design is found to be 1.35. It is concluded from the results that under such a combined loading condition, GACOM can find the optimal designs easily for the minimum radial displacement of the vessel at the outer surface.

5.3. Results for GA Code version 3 (GAver3): Composite Pressure Vessel Optimization for Minimum Weight

GAver3 is developed for the optimization of stacking sequence of a composite pressure vessel for the minimum weight. The detailed description of the problem is given in section **2.2**.

Required input parameters are quite similar to the parameters of the GAver2 code. There are few additional input parameters in this case, which are listed below.

- **Optimization Parameters:**
    - **Optimization type:** For GAver3, “minimum weight” is chosen.
    - **Ply addition probability:** Probability of the ply addition operator is entered.
    - **Ply deletion probability:** Probability of the ply deletion operator is entered.

The best design for the problem given in section **5.2.4** is searched in this part, but the objective is to minimize the weight instead. The combined loading condition is given in section **5.2.4** and all the other input parameters are given in section **5.2.1.** Several runs are made for different ply addition and ply deletion probabilities.

All the runs have converged to the same optimal result. The stacking sequence of the best design found by all the runs is:

$$\left\{\begin{matrix} 1 & 1 & 2 & 1 \\ 0^o_2 & 0^o_2 & \pm 10^o & 0^o_2 \end{matrix}\right\}_s$$

The results show that 16 plies are enough for the structure to be safe under the defined loading conditions. The safety factor of the best design is found to be 1.0032 which satisfies the safety factor requirement.

The variation of required number of generations for convergence versus ply addition and ply deletion probabilities are given in **figure 5-11** and **figure 5-12** respectively. For variable ply addition probability runs, the ply deletion probability value is chosen to be equal to 0.3; and for variable ply deletion probability runs, the ply addition probability value is chosen to be equal to 0.3.

It is seen from the graph that the best optimal value range for the ply addition probability is found to be between 0.1 and 0.3; whereas the values between 0.3 and 0.6 still can be used. But very big operator probability values may increase the explorative capability which may result in missing the best design in the search area. In order to have the balanced explorative and exploitative capability for the search, 0.3 seems to be the best value for the ply addition probability.

For the ply deletion probability, the optimal value range seems to be between 0.3 and 0.9. The optimal range is found to be a little big, because the difference between the starting chromosome length (20) and the optimal chromosome length (4) is quite big in this problem. Thus using big ply deletion probability values seem to be logical for this case. But in general, it might increase the explorative capability of the search too much. In that case, the search might not find the best design in the search area.

A single run has been made using ply deletion and ply addition probabilities both equal to 0.3. The variation of the weight per unit length is given in **figure 5-13**. The search has found the best design after 33 generations. It is seen from the graph that the search continues to have the explorative capability although it has converged to the optimal design. This capability avoids the search to get stuck into a local optimal design.

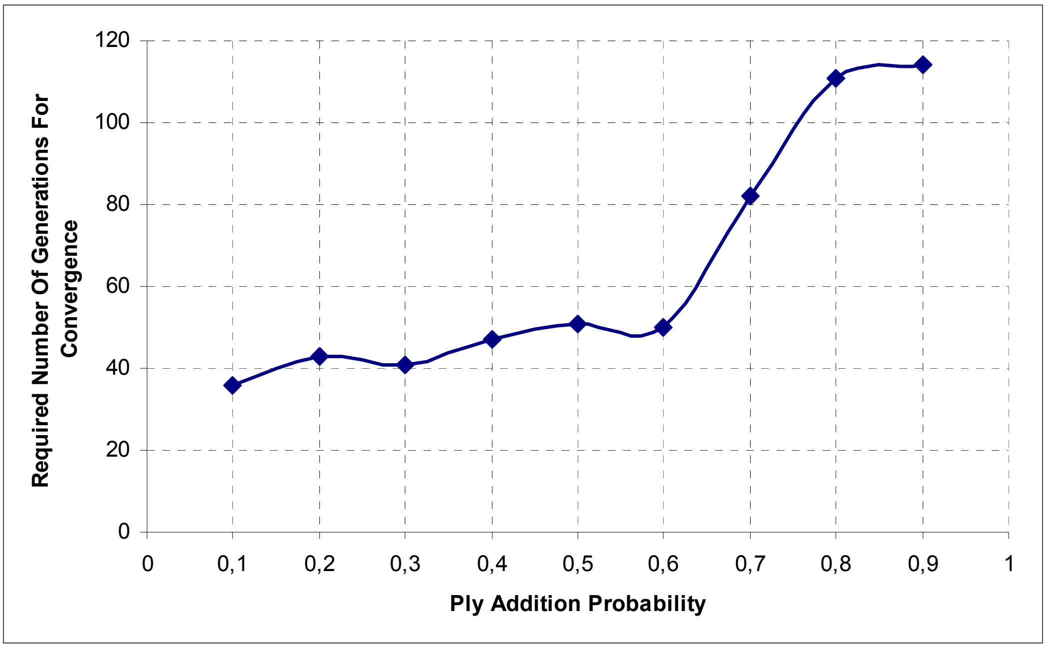


Figure 5-11 Required number of generations for convergence vs ply addition probability

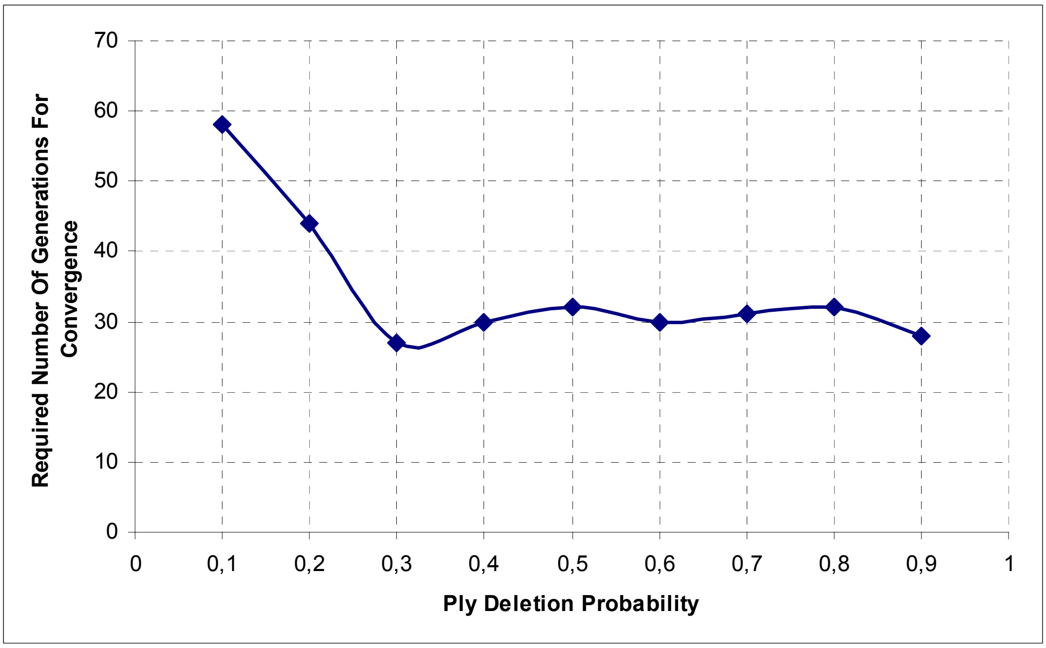


Figure 5-12 Required number of generations for convergence vs ply deletion probability

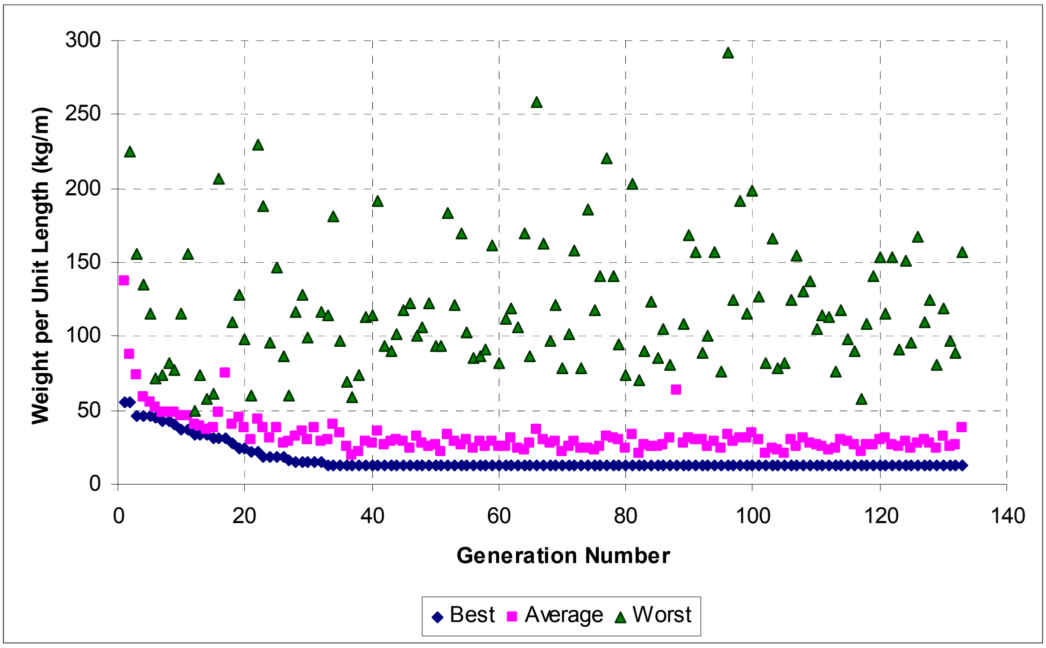


Figure 5-13 Weight per unit length vs generation number

The fiber orientation angles of the best found design are changed as shown below and the safety factor of each design is calculated (see **figure 5-14**).

$$\{1 \quad 1 \quad 2 \quad 1\}_s$$
$$\{2 \quad 2 \quad 2 \quad 2\}_s$$
$$\ldots$$
$$\{i \quad i \quad 2 \quad i\}_s$$
$$\ldots$$
$$\{10 \quad 10 \quad 2 \quad 10\}_s$$

It can be concluded from **figure 5-14** that the found design is the best among others, since the safety factor values of the other designs are quite low which means that all the other designs fail under the defined loading condition and does not satisfy the safety factor constraint.

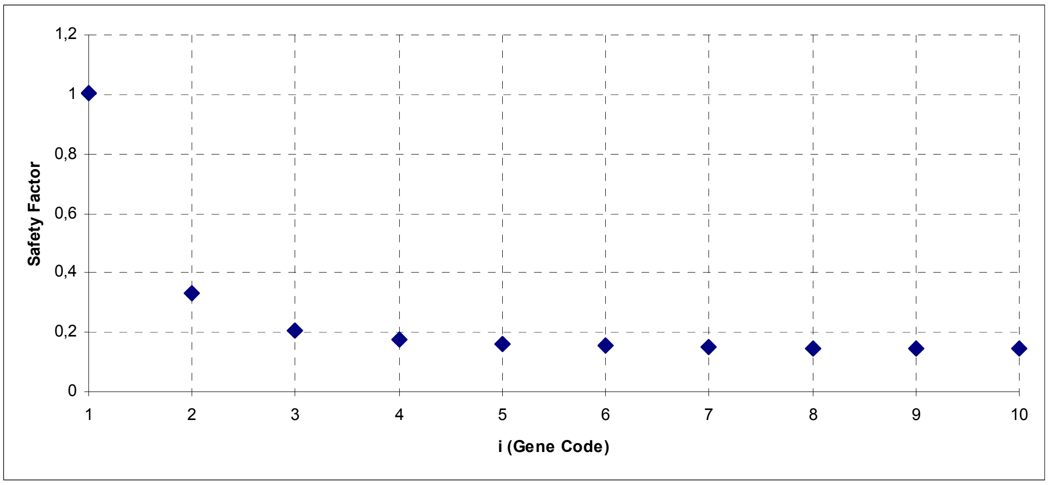


Figure 5-14 Safety factor vs orientation angle code (i)

A single run has been made with the same inputs except that the required safety factor is made 3. The search has converged to the optimal design after 38 generations and stopped at generation number 138. The results of the last generation written in the text file by GACOM are given in **figure 5-15**.

It is seen that GACOM has found optimal designs satisfying the safety factor constraint. In the last generation, there are alternative designs which are equal in weight and have very close safety factor values. There are also some designs which do not satisfy the safety factor constraint, since the objective function value is good enough due to smaller weight.

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  Ranked gene codes for each member of Generation Number : 138
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Rank No.  R          gene code
   1      3.147485   0  0  0  1  1  2  1  1  2  1  1  2  1  1  10
   2      3.078365   0  0  0  1  1  2  1  1  2  1  1  2  1  1  9
   3      3.078311   0  0  0  1  1  2  1  1  2  1  2  1  1  9  1
   4      3.078298   0  0  0  1  1  2  1  2  1  1  2  1  1  9  1
   5      3.046340   0  0  0  1  1  2  1  1  2  1  1  10 1  9  1
   6      3.039777   0  0  0  1  1  2  1  1  2  1  1  2  1  2  10
   7      3.039756   0  0  0  1  1  2  1  2  1  1  2  1  1  2  10
   8      3.039717   0  0  0  1  1  2  2  1  1  2  1  2  1  1  10
   9      3.039706   0  0  0  1  1  2  2  1  2  1  1  2  1  1  10
   10     3.025485   0  0  0  1  1  2  1  1  2  1  1  10 1  1  8
   11     3.025479   0  0  0  1  1  2  1  1  8  1  1  2  1  1  10
   12     3.016839   0  0  0  1  1  9  1  1  2  1  1  2  1  1  9
   13     2.980157   0  0  0  1  1  2  1  1  2  1  1  2  9  1  2
   14     2.930154   0  0  0  1  1  2  2  1  2  1  2  1  1  2  10
   15     2.785331   0  0  0  0  1  1  2  1  2  1  2  1  1  9  1
   16     2.897245   0  0  0  1  1  4  1  1  2  1  1  10 1  1  7
   17     3.025782   0  0  1  1  2  2  1  2  1  1  2  1  1  8  3
   18     3.448885   0  0  1  1  2  1  1  2  1  1  2  1  1  10 1
   19     3.021172   0  0  1  1  2  2  1  1  2  1  10 1  1  8  3
   20     2.714125   0  0  0  0  1  1  2  1  1  8  1  1  2  1  10
   21     2.707439   0  0  0  0  1  1  2  1  1  2  1  3  1  1  9
   22     2.909000   0  0  1  1  4  1  1  2  9  1  10 1  1  2  3
   23     2.767854   0  0  0  1  1  2  2  1  2  1  1  2  8  1  9
   24     2.623967   0  0  0  0  1  1  2  10 1  2  1  2  1  1  10
   25     2.621384   0  0  0  0  1  1  2  2  1  2  1  2  1  2  10
   26     2.584148   0  0  0  0  1  1  4  1  1  2  1  1  9  1  9
   27     3.650495   1  1  2  1  1  2  1  1  2  1  2  1  1  9  5
   28     2.667189   0  0  0  1  1  5  1  1  2  1  2  1  2  10 9
   29     2.875890   0  1  1  2  2  1  2  1  1  2  1  8  4  5  6
   30     2.532885   0  0  0  0  1  1  2  1  2  1  2  1  3  2  10
```

Figure 5-15 Results of last generation (a part of GACOM output text file)

CHAPTER 6

CONCLUSION

Stacking sequence design can be considered to be a combinatorial problem when the fiber orientations in each layer are restricted to certain angles. Genetic algorithms are quite suitable for such a combinatorial problem, and can find many optimal or near-optimal designs. This work is focused on the development of an efficient and reliable genetic algorithm code for the optimization of a composite pressure vessel subjected to internal and external pressures, axial load and body force due to rotation. In this study, three different versions of a genetic algorithm code are developed in Matlab 7. The differences between the codes are the objective function evaluation. It was seen that the code was easily adapted to different type of optimization problems.

The first code (GAver1) is developed for testing the performance of the code. The designed problem has a known best solution which enables to calculate the reliability and the cost of a search healthier by comparing the found results with the best possible one. Many multiple runs were made for different population sizes and different N_{me} values to compare Elitist (EL), multiple elitist 1 (ME1) and multiple elitist 2 (ME2) selection methods. The optimum selection method and N_{me} values for the multiple elitist selection methods were tried to be identified. The optimal range for the mutation and swap operator probabilities were examined and the best population size range as a percent of the chromosome length was sought for the maximum reliability and the minimum cost.

Code versions 2 and 3 (GAver2 and GAver3) are developed for the stacking sequence optimization of a composite pressure vessel. Software which is called GACOM was created in Matlab 7 with a graphical user interface. GACOM contains both GAver2 and GAver3 codes. This software enables defining and running an optimization problem easier. The program has significant advantageous features as stated in **section 5.1**.

In **chapter 2**, the analytic formulas for the stresses and the displacements were derived and compared with the thin wall solution. It was seen that the results obtained by the derived solution are similar to the ones obtained by thin wall solution when the wall thickness of the vessel is small. Two different optimization problems were defined for the stacking sequence design of a composite pressure vessel. The difference between these two problems is the objective function evaluation. The objective of the first problem is to minimize the radial displacement of the vessel at the outer surface under the safety factor constraint. GAver2 is developed for the solution of the first problem. Several runs were made for different loading conditions. Expected results were obtained for the pure internal pressure case. For the loading cases for which the optimal result is unpredictable, stacking sequence of the best ever found design was changed a little bit to see whether the design is the optimum one in the search area or not. It was seen that the code was successful in finding the optimum design in a few number of generations when the optimum parameters for the search are used. The performance of the code was examined for different selection methods and N_{me} values in the combined loading case and the optimum selection method and N_{me} value was tried to be found.

The objective of the second problem is to minimize the weight of the vessel under the safety factor constraint. GAver3 is developed for the solution of this problem. The optimum design was searched for the same combined loading condition given in GAver2. Slight changes were made on the stacking sequence of the best ever found design to identify the change of the safety factor of the designs. It is obvious that the objective function values of the designs which have same number of plies and satisfy the safety factor constraint are equal due to equal fitness values. This means that the fitness value is dependent on the safety factor value only when the safety factor of the design is lower than the required safety factor value. Thus, the designs which are equal in weight and satisfy the safety factor constraint are not sorted according to their safety factor values. Yet, these optimal designs exist in the last generation of the search, which allows user to see alternative optimum designs. Best design was again searched for the same combined loading condition for a different safety factor criterion. It was seen that GAver3 easily locate the optimum designs satisfying the required safety factor criterion. Optimum probability values of ply addition and ply deletion operators which are used only for GAver3 were also sought in GAver3 runs.

This study can be extended more by means of the expansions involving genetic algorithm and the composite optimization problem. The genetic operators and/or the selection schemes used in GA codes can be changed to increase the efficiency of the search. As an example, two-point crossover may be used instead of one-point crossover. The effect of the new operator may be searched. Moreover, the composite optimization problem may be changed so that GAver2 and GAver3 are featured to handle more than one type of material.

REFERENCES

[1] Jones, R.M., “Mechanics of Composite Materials" Hemisphere, New York, 1975.

[2] “Boeing 777 roll out" Aviation Week and Space Technology, April 11, 1994, pp. 36-58.

[3] Soremekun, Grant A. E., “Genetic Algorithms for Composite Laminate Design and Optimization”, February 5, 1997.

[4] Chang, J.B., “Development of COPV-Related Standards” Structures, Structural Dynamics, and Materials Conference, AIAA 2007-2146.

[5] Thesken, J.C., Murthy, P. and Phoenix, L., “Composite Overwrap Pressure Vessels: Mechanics and Stress Rupture Lifing Philosophy” Structures, Structural Dynamics, and Materials Conference, AIAA 2007-2145.

[6] Greene, N., Saulsberry, R., Yoder, T., Forsyth, B., Thesken, J. and Phoenix, S.L., “Testing of Full-scale Flight-qualified Kevlar Composite Overwrapped Pressure Vessels” Structures, Structural Dynamics, and Materials Conference, AIAA 2007-2326.

[7] Foye, R., “Advanced Design for Advanced Composite Airframes” Airforce Materials Laboratory, Wright-Patterson Air Force Base, Ohio, AFML TR-69-251, 1969.

[8] Waddoups, M.E., “Structural Airframe Application of Advanced Composite Materials – Analytical Methods” Airforce Materials Laboratory, Wright-Patterson Air Force Base, Ohio, AFML TR-69-101, 1969.

[9] Schmit, L.A., and Farshi, B., “Optimum Laminate Design for Strength and Stiffness” International Journal for Numerical Methods in Engineering, 1973, Vol. 7, No. 4, pp. 519-536.

[10] Schmit, L.A., and Farshi, B., “Optimum Design of Laminated Fiber Composite Plates” International Journal for Numerical Methods in Engineering, 1977, Vol. 11, No. 4, pp. 623-640.

[11] Pedersen, P., “On Sensitivity Analysis and Optimal Design of Specially Orthotropic Laminates”, Engineering Optimization, 11, 1987, pp. 305-316.

[12] Haftka, R.T., and Walsh, J.L., “Stacking Sequence Optimization for Buckling of Laminated Plates by Integer Programming” AIAA Journal, Vol. 30, No. 3, 1992, pp. 814-819.

[13] Todoroki, A., and Hirano, Y., “Stacking Sequence Optimization of Composite Delta Wing to Improve Flutter Limit Using Fractal Branch and Bound Method” JSME Internatinal Journal, Series A, 2005, Vol. 48, No. 2.

[14] Todoroki, A., and Matsuzaki, R., “Stacking sequence optimization using fractal branch and bound method for unsymmetrical laminates” Composite Structures, 78 (2007), pp. 537-550.

[15] Todoroki, A., Hirano, Y., and Hiroshi, H., “Multi-Objective Stacking Sequence Optimization of Composite Wing Structure Using FBBM” AIAA 2007 Conference and Exhibit, 7-10 May 2007, Rohnert Park, California.

[16] Le Riche, R., “Optimization of Composite Structures by Genetic Algorithm” Ph.D. Dissertation, Virginia Polytechnic Institute and State University, 1994.

[17] Mesquita, L., and Kamat, M.P., “Optimization of Stiffened Laminated Composite Plate with Frequency Constraints”, Engineering Optimization, 11, 1987, pp. 77-88.

[18] Callahan, K. J. and Weeks, G. E., “Optimum design of composite laminates using genetic algorithm” Composites Engineering, 1992, 2(3), pp. 149–160.

[19] Nagendra, S., Haftka, R. T. and Gürdal, Z., “Stacking sequence optimization of simply supported laminates with stability and strain constraints” AIAA Journal, 1992, Vol. 30, No. 8, pp. 2132–2137.

[20] Nagendra, S., Haftka, R.T. and Gurdal, Z., "Design of blade stiffened Composite Panel by a Genetic Algorithm.” AIAA-93-1584-CP, Proc. 34th Structures, Structural Dynamics and Material Conf, La Jolla, California, April 19-21, 1993, pp. 2481-2436.

[21] Nagendra, S., Haftka, R. T. and Gürdal, Z., “A genetic algorithm based design procedure for stiffened composite panels under stability and strain constraints” In Proceedings of the the 10th DOD/ NASA/ FAA Conference on Fibrous Composites in Structural Design, Hilton Head, 1993.

[22] Le Riche, R. and Haftka, R. T. “Optimization of laminate stacking sequence for buckling load maximization by genetic algorithm” AIAA Journal, 1993, 31(5), pp. 951–956.

[23] Harrison, P. N., Le Riche, R. and Haftka, R. T., “Design of stiffened panels by genetic algorithm and response surface approximation” In Proceedings of the 36th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 1995, pp. 58–68, New Orleans, LA. AIAA Paper 95-1163-CP

[24] Nagendra, S., Jestin, D., Gürdal, Z., Haftka, R. T., and Watson, L. T., “Improved genetic algorithm for the design of stiffened composite panels” Computers & Structures, 1996, 58(3), pp. 543–555.

[25] Malott, B., Averill, R. C., Goodman, E. D., Ding, Y., and Punch, W. F., “Use of genetic algorithms for optimal design of laminated composite sandwich panels with bending-twisting coupling” In Proceedings of the 37th AIAA/ ASME/ASCE/ AHS/ ASC Structures, Structural Dynamics, and Materials Conference, 1996, pages 1874–1881, Salt Lake City, UT. AIAA Paper 1996-26801

[26] Gantovnik, V.B., Gürdal, Z. and Watson, L.T., “A genetic algorithm with memory for optimal design of laminated sandwich composite panels” Composite Structures, 58 (2002), pp. 513-520.

[27] Lin, C.C. and Lee, Y.J., “Stacking sequence optimization of laminated composite structures using genetic algorithm with local improvement” Composite Structures, 63 (2004), pp. 339-345.

[28] Walker, M. and Smith, R.E., “A technique for the multiobjective optimization of laminated composite structures using genetic algorithms and finite element analysis” Composite Structures, 62 (2003), pp. 123-128.

[29] Todoroki, A. and Haftka, R.T., “Stacking sequence optimization by a genetic algorithm with a new recessive gene like repair strategy” Composites, Part B 29B (1998), pp. 277-285.

[30] Liu, B., Haftka, R.T., Akgün, M.A. and Todoroki, A., “Permutation genetic algorithm for staking sequence design of composite laminates” Computer Methods in Applied Mechanics and Engineering, 186 (2000), pp. 357-372.

[31] Park, J.H., Hwang, J.H., Lee, C.S. and Hwang, W., “Stacking sequence design of composite laminates for maximum strength using genetic algorithms” Composite Structures, 52 (2001), pp. 217-231.

[32] Soremekun, G., Gürdal, Z., Haftka, R.T. and Watson, L.T., “Composite laminate design optimization by genetic algorithm with generalized elitist selection” Computers & Structures, 79 (2001), pp. 131-143.

[33] Soremekun, G., Gürdal, Z., Kassapoglou, C. and Toni, D., “Stacking sequence blending of multiple composite laminates using genetic algorithms” Composite Structures, 56 (2002), pp. 53-62.

[34] H.J. Bremermann, “Optimization through evolution and recombination,” in M.C. Yovits, G.TI Jacobi and G.D. Goldman (Eds), Self-organizing systems, Spartan Books, Washington, D.C., 1962., pp. 93-106.

[35] Rechenberg, I., "Cybernetic Solution Path of an Experimental Problem," Royal Aircraft Establishment, Library Translation 1122, Farnborough, England, UK, 1965.

[36] Holland, J.H., “Adaptation in Natural and Artificial Systems”, University of Michigan Press, Ann Arbor, MI, 1995.

[37] Goldberg, D.E., and Samtani, T.A., “Engineering Optimization by a Genetic Algorithm”. Proc 9th Conf. Electronic Computation, ASCE. 1987, pp 471-482.

[38] Hajela, P., “ Genetic Search-An Approach to the Nonconvex Optimization Problem.” AIAA Journal, 1990, 26(7), pp. 1205-1210.

[39] Furuya, H., and Haftka, R.T., “ Genetic Algorithms For Placing Actuators On Space Structures. Proc. Fifth Int. Conf. Genetic Algorithms. University of Illinois at Uabana-Champaign, 17-21 July 1993, pp. 536-542. Morgan Kaufmann, Fairfax.

[40] Powell, D., and Skolnick, M.M., “ Using Genetic Algorithms in Engineering Design Optimization with Nonlinear Constraints”, Proc. Fifth Int. Conf. Genetic Algorithms, University of Illinois of Urbana-Champaign, 17-21 July, 1993, pp. 424-431, Morgan Kaufmann, Fairfax.

[41] Schoenauer, M and Xanthakis, S.,”Constrained GA Optimization”, Proc. Fifth Int. Conf. Genetic algorithms, University of Illinois at Urbana-Champaign, 17-21, July, 1993, pp. 573-580, Morgan Kaufmann,Fairfax.

[42] Watabe, H. and Okino, N., “A Study on Genetic Shape Design”. Proc. Fifth Int. Conf. Genetic Algorithms, University of Illumes at Urbana-Champaign, 17-21, July, 1993, pp. 445-450, Morgan Raufmann, Fairfax.

[43] Katırcı, N., “Design of Fiber Reinforced Composite Pressure Vessels” M.S.Thesis, Middle East Technical University, Turkey, 1998.

[44] Lekhnitskii, S. G., “Theory of Elasticity of an Anisotropic Body”, Mir Publishers, Moscow, 1981.

[45] Tsai, W. Stephen, “Composites Design”, Think Composites, USA, 1985.