A GENETIC ALGORITHM FOR 2D SHAPE OPTIMIZATION

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

BY

WEI HANG CHEN

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

AUGUST 2008

Approval of the thesis:

A GENETIC ALGORITHM FOR 2D SHAPE OPTIMIZATION

submitted by **WEIHANG CHEN** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

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

Prof. Dr. Kemal İder
Head of Department, **Mechanical Engineering**

Prof. Dr. Süha Oral
Supervisor, **Mechanical Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Reşit Soylu
Mechanical Engineering Dept., METU

Prof. Dr. Süha Oral
Mechanical Engineering Dept., METU

Prof. Dr. Haluk Darendeliler
Mechanical Engineering Dept., METU

Prof. Dr. Suat Kadıoğlu
Mechanical Engineering Dept., METU

Assoc. Prof. Uğur Polat
Civil Engineering Dept., METU

Date: 28.08.2008

PLAGIARISM

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 : Wei Hang Chen

Signature :

ABSTRACT

A GENETIC ALGORITHM FOR 2D SHAPE OPTIMIZATION

Chen, Wei Hang
M.S., Department of Mechanical Engineering
Supervisor : Prof. Dr. Süha Oral

August 2008, 104 pages

In this study, an optimization code has been developed based on genetic algorithms associated with the finite element modeling for the shape optimization of plane stress problems.

In genetic algorithms, constraints are mostly handled by using the concept of penalty functions, which penalize infeasible solutions by reducing their fitness values in proportion to the degrees of constraint violation. In this study, An Improved GA Penalty Scheme is used. The proposed method gives information about unfeasible individual fitness as near as possible to the feasible region in the evaluation function. The objective function in this study is the area of the structure. The area is minimized considering the Von-Misses stress criteria. In order to minimize the objective function, one-point crossover with roulette-wheel selection approach is used. Optimum dimensions of four problems available in the literature have been solved by the code developed .

The algorithm is tested using several strategies such as; different initial population number, different probability of mutation and crossover. The results are compared with the ones in literature and conclusions are driven accordingly.

Keywords: Genetic Algorithm, Structural Optimization.

ÖZ

GENETİK ALGORİTMA İLE İKİ BOYUTLU DÜZLEMLERİN ŞEKİL OPTMİZASYONU

Chen, Wei Hang
Yüksek Lisans, Makina Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Süha Oral

Ağustos 2008, 104 sayfa

Bu çalışmada, Genetik Algoritma ve Sonlu Elemanlar Metodu kulanılarak iki boyutlu düzlemlerde gerilme problemlerin şekil optimizasyonu için bir bilgisayar kodu geliştirilmiştir.

Genetik Algoritmalarda, kısıtlamaların üstesinden gelebilmek için genelde ceza fonksiyonları kulanılıyor. Ceza fonksiyonları kısıtlamaların ihlal derecelerine orantılı bir şekilde, mümkün olmayan çözümlerin liyakat değerlerini düşürüyor ve onları devre dışı brakıyor.

Bu çalışmada Geliştirilimiş Bir Genetik Algoritma Ceza Şeması uygulanmıştır. Önerilen yöntem, liyakatları uygun olmayan bireyleri değerlendirme fonksiyoların belirledikleri uygunluğa göre tespit ediyor.
Bu çalışmada, amaç fonksiyonu yapının alanıdır. Alan, Von-Mises gerilme kriterine göre miniminize edilmiştir. Amaç fonksiyonunu minimize etmek için tek-nokta crossover ve roulette-wheel (çarkıfelek) yöntemleri kulanıldı. Geliştirilen bilgisayar kodu ile literatürde var olan dört problemin optimum boyutları bulundu.

Algoritma pek çok strateji kulanarak test edildi. Bunların arasında; farklı başlangıç popülasyon sayısı, ile farklı mutasyon ve crossover olasılıkları sayılabilir. Elde edilen sonuçar literatürdekilerle kıyaslanarak bir neticeye varılmıştır.

Anahtar Kelimeler: Genetik algoritma, Yapısal optimizasyon.

To my parents and to my brother...

ACKNOWLEDGMENTS

The writer wishes to express his gratitude to his supervisor Professor Dr. Süha Oral, under whose guidance this study was conducted, for his patient guidance and criticism throughout the study, for all the valuable ideas and suggestions, and for our conversations on science which I enjoyed so much.

I thank to Özkan Murat and Dr. Kutay Çelebioğlu, for their technical and moral support. I also would like to thank to Kerem Mağdenli for his financial support that makes this study possible.

I want to thank DongQing Hu who supported and encouraged me throughout my whole education. It is you who aroused my interest in engineering.

To my love Rezana Zarshat: thank you for all your patience and support, for your love and everything that makes it so worthwhile to know you.

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 1

INTRODUCTION

Shape design optimization is a challenging task that has received notice in the recent years. The purpose of this study is to develop a code for the shape optimization in plane stress problems by using genetic algorithms.

1.1 Objective and Scope of the Study

In this study a Genetic Algorithm is developed for plane stress problems. The final algorithm is tested by standard shape optimization problems and all the results obtained have been compared with those in literature.

The code is quite efficient since the analysis module is within the optimization code itself. Finally, TECPLOT is used as a post-processor in order to visualize the results.

1.2 Literature Survey

A literature survey has been done for the shape optimization and Genetic algorithms. The study started with the investigation of optimization concepts and definitions used.

Several numbers of publications are read in order to obtain the preliminary background of two dimensional shape optimization and the Genetic Algorithm. The shape design optimization, Genetic Algorithm concepts, operators and algorithm implementation are the main focus points of this survey. The publications read on optimization are as follows:

In 1997, a novel, node-based shape optimization method was developed for planar structures based on the consideration that the critical stress and displacement constraints are generally located along or near the structural boundary by X. Song, D. Baldwin [24]. They proposed to put the maximum weights on the selected boundary nodes, referred to as the design points, so that the sensitivity analysis was based on the perturbation of only these nodes.

In 2000, E. Taroco [11] wrote a paper named "Shape sensitivity analysis in linear elastic fracture mechanics". In this paper he analized the shape sensitivity of an elastic solid in equilibrium with a known load system applied over its boundary .

In 2001 P. Vinot , S. Cogan, J. Piranda [1] wrote a paper named "Shape optimization of thin-walled beam-like structures". They introduced a methodology for optimizing the shape of thin-walled structures having a beam-like dynamic behavior.

"Optimal Design Of Hole Reinforcements For Composite Structures" was written by H. Engels, W. Hansel, and W. Becker[5] in 2002. In this study the reinforcements by elliptic doublers, as well as doublers adapted to reinforcement requirements in a layerwise manner, were considered.

In 2003, Manolis Papadrakakis, Nikos D. Lagaros [4] published a paper named "Soft computing methodologies for structural optimization". They examined the efficiency of soft computing techniques in structural optimization for solving large-scale, continuous or discrete structural optimization problems.

R. Das et al. [25] in 2004 wrote a paper presenting a modified evolutionary structural optimization (ESO) algorithm for optimal design of damage tolerant structures. The proposed ESO algorithm uses fracture strength as the design objective. The formulation outline was used for shape optimization of structures and allowed for cracks to be located along the entire structural boundary.

In 2005, Sachin S. Terdalkar, Joseph J. Rencis [9] developed a graphically driven interactive stress reanalysis finite element technique for engineers to carry out manual geometric changes in a machine element during the early design stage.

In 2007, Wolfgang A. Wall et al. [23] wrote a paper on Isogeometric Structural Shape Optimization. In this paper they demonstrated a structural shape optimization framework based on the isogeometric analysis approach in order to solve an old dilemma in structural shape optimization. According to this dilema a tight link is needed between design model or geometric description and analysis model and this paper showed that isogeometric analysis offers a potential and promising way out of this dilemma.

J.T. Katsikadelis and G.C. Tsiatas [10], in 2007 published “Optimum design of structures subjected to follower forces”, where shape optimization is used to optimize the critical load of an Euler–Bernoulli cantilever beam with constant volume subjected to a tangential compressive tip load and/or a tangential compressive load arbitrarily distributed along the beam.

The in-depth GA knowledge had been gained by examining the following papers and publications;
Genetic Algorithms (GAs) were invented by John Holland and developed by him and his students and colleagues. This lead to Holland's book “Adaptation in Natural and Artificial Systems” published in 1975, Ref [30].

In 1996, Melanie Mitchell [22] wrote a book introducing the basic features of Genetic Algorithm. The book’s name was “An Introduction to Genetic Algorithm” and it gave basic information about the Genetic Algorithm. The book also includes a brief history on the development of the algorithm through the years.
P.N. Suganthan [37], in year 1999 wrote a paper named “Structural pattern recognition using genetic algorithms”. In his paper he presented a genetic algorithm (GA) based optimization procedure for the solution of structural pattern recognition

problem using the attributed relational graph representation and matching technique. The candidate solutions were represented by integer strings and the population was randomly initialized. The GA was employed to generate a monomorphic mapping. As all the mapping constraints were not enforced during the searching phase in order to speed up the search, an efficient pose clustering algorithm was used to eliminate spurious matches and to determine the presence of the model in the scene. The performance of the proposed approach to pattern recognition by subgraph isomorphism was demonstrated using line patterns and silhouette images.

Shmuel Vigdergauz [8], in year 2000 wrote a paper named "Genetic algorithm perspective to identify energy optimizing inclusions in an elastic plate". His paper described the theoretical grounds and technical features that had to be introduced for effective implementation of a regular genetic algorithm to the shape optimization problems in planar elasticity.

Mark J. Jakiela et al. [31] in 2000, published a paper named "Continuum structural topology design with genetic algorithms". They described a binary material/void design representation that was encoded in GA chromosome data structures. The main intention was to approximate a material continuum as opposed to discrete truss structures. In this study they arrived to the conclusion that the advantage of genetic algorithms is; "Since GAs only require zero'th order function evaluations, they can be applied to problems for which little is known about the nature of the design domain".

In 2000, Oğuzhan Hasançebi and Fuat Erbatur [32] published a paper named "Evaluation of crossover techniques in genetic algorithm based optimum structural design". In this paper they proposed two newly developed crossover techniques, through which a better efficiency of GAs can be obtained.

Coello and Christiansen [36], in 2000, published a paper named "Multiobjective optimization of trusses using genetic algorithms". The aim of their study was the use of the genetic algorithm (GA) as a tool to solve multiobjective optimization problems in structures.

J. Mahachi and M. Dundu [38], published a paper named “Genetic Algorithm Operators for Optimization of Pin Jointed Structures” in 2001. Their study was concerned with the structural mechanics of pin-jointed structures. They carried out an investigation in order to determine the GA operators for the optimization of pin-jointed structures and illustrated numerical examples to show the efficiency of the parameters compared to simple Genetic Algorithms and traditional mathematical programming techniques.

Kalyanmoy Deb and Surendra Gulati [39] published a paper named “Design of truss-structures for minimum weight using genetic algorithms” in 2001. The aim of their study was finding optimal cross-sectional size, topology, and configuration of 2-D and 3-D trusses with minimum weight by using real-coded genetic algorithms (GAs). Although their GA uses a fixed-length vector of design variables representing member areas and change in nodal coordinates, a simple member exclusion principle is introduced to obtain differing topologies. Practical considerations, such as inclusion of important nodes in the optimized structure, are done by using a concept of basic and non-basic nodes.

In 2001, Nanakorn and Meesomklin [26] published a paper demonstrating a new penalty scheme capable of adjusting itself during the optimization process. The penalty function they proposed was able to adjust itself during the evolution in such a way that the desired degree of penalty is always obtained.

Kahraman and Erbatur, in 2001 [35] published a paper named “A GA Approach for Simultaneous Structural Optimization”. This paper presented simultaneous layout structural optimization of trusses using genetic algorithms (GAs). The authors claimed that the inclusion of shape and topology variables in addition to size variables highly increases the complexity of the optimization problem. This is because, firstly, the number of design variables is increased and secondly, a simultaneous treatment of design variables of different types requires working on quite complicated design spaces. To solve this problem they proposed, in view of the computational efficiency of the GA, an adaptive 3-phase search approach which was used in conjunction with the GA.

In 2002, F. Cappello, A. Mancuso [6] wrote a paper on combined topology and shape optimizations using genetic algorithm. They concluded that the developed genetic algorithm gave better results at least in the topology optimisation, with respect to gradient methods. Furthermore, their results showed the advantages in refining with a shape optimisation the boundaries found by topology optimisation.

In 2003, Dimou and Koumousis [33] published a paper named “Competitive genetic algorithms with application to reliability optimal design”. In this study they aimed to calibrate the population size of the GAs by altering the resources of the system, i.e. the allocated computing time. They concluded that the evolution dynamics improve the capacity of the optimization algorithm to find optimum solutions and results in statistically better designs as compared to the standard GA with any of the fixed parameters considered.

CHAPTER 2

OPTIMIZATION

2.1 The Optimization Problem

Optimization involves finding the best solution to a problem. Mathematically, this means finding the minimum or maximum of a function of n variables, $f(x_1,\ldots,x_n)$, where n may be any integer greater than zero. The function may be unconstrained or the variables of the function might be subject to certain constraints.

The optimization problem in this study is formulated as a minimization problem. In particular a maximum of a function can be determined by a minimization method since:

$$Maximum\ f(x) = -Minimum(-f(x)) \tag{1}$$

The function *f* is referred to as the objective function whose value is the quantity to be minimized.

The design variables, $(x_1,\ldots,x_n)$, will take successive adjustment values during the optimization process. Each set of adjustments to these variables is termed iteration and in general several iterations are required before an optimum is satisfactorily approximated.

In structural optimization, there are three types of design variables. Namely;

• Size design variables

• Shape design variables

• Topology design variables

In structural optimization problems there are constraints on the values of the design variables which restrict the region of search for the minimum. A common constraint on the design variable x_i (i=1,..,n ,where n may be any integer greater than zero) is in the form of the inequality $X_{Li} \leq X_i \leq X_{Ui}$ where X_L and X_U are fixed lower and upper limits to X_i. More generally 'inequality constraints' are formulated to specify functional relationships of the design variables involved in the constraint. Most inequality constraints can be fitted into the form

$$G(x_{1,}..,x_n)<0 \tag{2}$$

For the simple case of upper and lower limits, the expression $X_{Li} \leq X_i \leq X_{Ui}$ is replaced by the two expressions

$$X_i - X_{Ui} \leq 0 \tag{3}$$

$$X_{Li} - X_i \leq 0 \tag{4}$$

The region of search in which the constraints are satisfied is termed the feasible region, while the region in which constraints are not satisfied is termed the non-feasible or infeasible region.

2.2 Categories of Optimization

Optimization algorithms can be separated into six categories. Neither of these categories are necessarily independent of each other. For example, a static optimization problem might have discrete or continuous design variables. In addition, these design variables might be constrained or unconstrained.

Figure 1 gives a chart of these six categories.

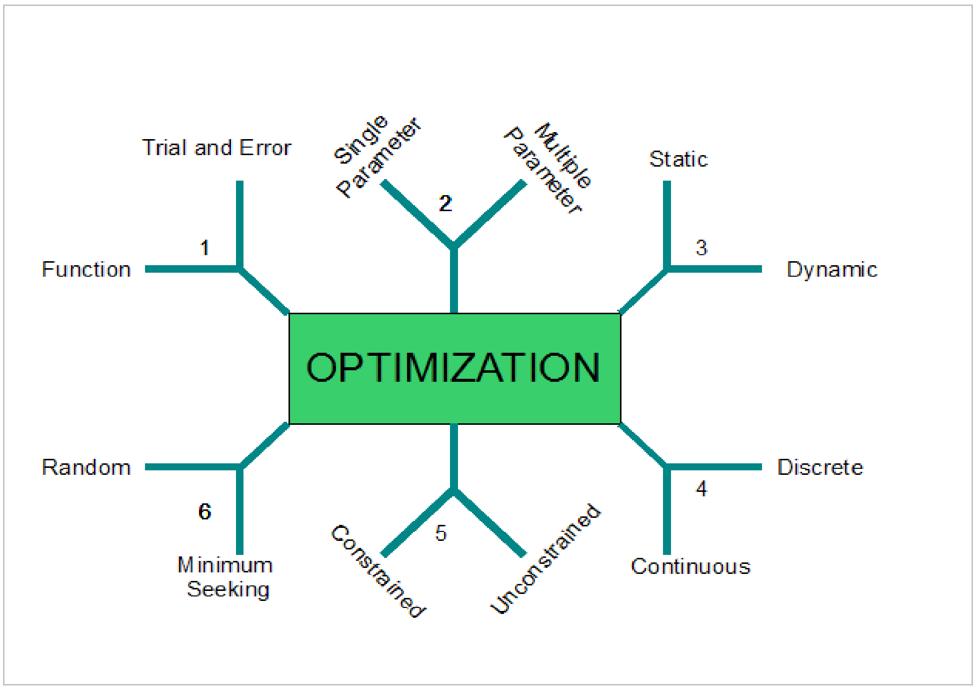

Figure 1: Categories of Optimization Algorithms Ref[20]

1. Trial and error refers to adjusting the design variables that affect the output without knowing the way that leads to the output itself.
   However, if the process is described by a mathematical formula then this method enters to the function optimization branch . While the Trial-Error is preferred by the experimentalists the mathematical function is preferred by the theoreticians.
2. The optimization would be one-dimensional if there was only one design

variable. On the other hand, multidimensional optimization is a problem that has multiple design variables. Optimization difficulty increases as the number of the design variables increase.

3. In dynamic optimization the output is a function of time, while in static optimization the output is independent of time.
   A static problem is difficult to solve for the optimum solution but adding the time factor increases the difficulty of the problem.
4. Another category is optimization by discrete or continuous design variables. While the discrete design variables are finite, continuous design variables are infinite. In discrete parameter optimization the optimum solution consists of certain combination of finite design variables. However, in order to find the minimum value of a function f(x) it is more appropriate to use continuous design variables.
5. In constrained optimization the parameter equalities and inequalities must be considered together with the fitness function. On the other hand, in unconstrained optimization the design variables can take any value. Most numerical optimization methods use unconstrained design variables.
6. Usually some optimization algorithms easily get stuck in local minimum even they tend to get the solution fast. These are called minimum seeking optimization algorithms.

   On the other hand, random optimization algorithms use probability methods to find design variables that are going to be used in optimization. They can find the global optimum, even they reach the solution more slowly.

CHAPTER 3

GENETIC ALGORITHM

3.1 An Introduction to Genetic Algorithms

"Genetic Algorithms are based on a biological metaphor: They view learning as a competition among a population of evolving candidate problem solutions. A 'fitness' function evaluates each solution to decide whether it will contribute to the next generation of solutions. Then, through operations analogous to gene transfer in sexual reproduction, the algorithm creates a new population of candidate solutions." --From *Artificial Intelligence, Structures and Strategies for Complex Problem Solving*, Fourth Edition, at page 471. Luger, George F. 2002. Harlow, England: Addison-Wesley.

3.1.1 History of Genetic Algorithms

As early as 1962, John Holland's work on adaptive systems laid the foundation for later developments; most notably, Holland was also the first to explicitly propose crossover and other recombination operators. However, the seminal work in the field of genetic algorithms came in 1975, with the publication of the book *Adaptation in Natural and Artificial Systems*. Building on earlier research and papers both by Holland himself and by colleagues at the University of Michigan, this book was the first to systematically and rigorously present the concept of adaptive digital systems using mutation, selection and crossover, simulating processes of biological evolution, as a problem-solving strategy. The book also attempted to put genetic algorithms on a firm theoretical footing by introducing the notion of schemata (Mitchell 1996, p.3; Haupt and Haupt 1998, p.147). That same year, Kenneth De Jong's important dissertation established the potential of GAs by showing that they could perform well on a wide variety of test functions, including noisy, discontinuous, and multi modal search landscapes (Goldberg 1989, p.107).

3.2 Basic Structure of Genetic Algorithm

3.2.1 Real-valued Genetic algorithms

The real-valued algorithm and the binary algorithm are two branches of a genetic algorithm. Both of them follow the same way of natural selection. One works with the real values themselves to minimize the objective function while the other represents design variables as an encoded binary string and works with the binary strings to minimize the objective function. A brief description of the real-valued algorithm is presented here, since this was also the method used while developing the genetic algorithm based code.

The real-valued genetic algorithm has the advantage of requiring less storage than the binary genetic algorithm because a single floating-point number represents the design variable instead of N_{bit} integers. As N_{bit} increases, this storage becomes significant. The other advantage is the accurate representation of the real values. It follows that the representation of the objective function is also more accurate as a result.

The flow chart in Figure 2 provides a basic structure of a real-valued genetic algorithm. This genetic algorithm is very similar to the binary genetic algorithm. The primary difference is the fact that design variables are no longer represented by bits of zeros and ones, but by real numbers over whatever range is appropriate.

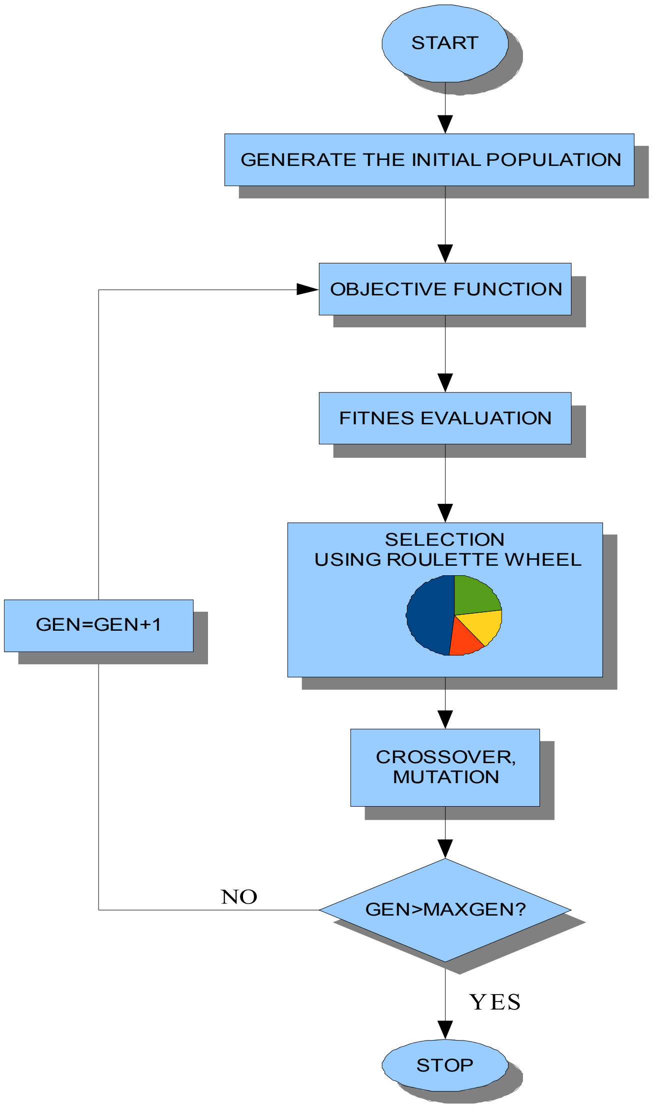

Figure 2: The Flow Chart of the Real-Valued Genetic Algorithm

The main components of the basic genetic algorithm are the creation of initial population, fitness evaluation, mating operator (selection, elitist policy, crossover and mutation).

3.2.2 Creation of Initial Population

The goal of this study is to solve some optimization problem where we search for an optimum solution in terms of the design variables of the problem. At the beginning of the process the chromosome array that is going to be optimized is defined by fitting the design variables in genetic algorithm. If the chromosome has n design variables, N_{desvar}, (an N-dimension optimization problem) given by $X_1, \ldots\ldots, X_{N_{desvar}}$ then the chromosome is written as an array with $1xN_{desvar}$ elements so that:

Chromosome = $[X_1, \ldots\ldots\ldots\ X_{N_{desvar}}]$

The design variables each are represented as floating-point numbers.

To begin the genetic algorithm, an initial population of N_{ipop} chromosomes is defined. A matrix represents the population with each row in the matrix being a $1xN_{desvar}$ array (chromosome) of continuous design variable values. Given an initial population of N_{ipop} chromosomes, the full matrix of $N_{ipop}xN_{desvar}$ random value is generated by;

$$IPOP = (hi - lo) * random(N_{ipop}, N_{desvar}) + lo \qquad (5)$$

Where;

random$\{N_{ipop}, N_{desvar}\}$=a function that generates an N_{ipop} x N_{desvar} matrix of uniform random numbers between zero and one.

hi= highest number in the design variable range
lo= lowest number in the design variable range

3.2.3 Fitness evaluation

The fitness function evaluates the quality of the chromosome as a solution to a particular problem.

Generally, an optimization problem using GAs can be expressed as:

Maximize

$F(x) = F[\, f(x)],$

$x = x_1 , x_2 ,\ldots\ldots\ldots\ldots\ldots\, x_{Ndesvar} \quad \epsilon \quad R^N,$

under constraints defined as

$g_i(x) \leq 0 \ , i = 1,\ldots\ldots\ldots,M,$

$h_i(x) \leq 0 \ , i = 1,\ldots\ldots\ldots, L,$

For structural design optimization, x is an N dimensional vector called the design vector, representing design variables of N structural components to be optimized, and f(x) is the objective function. In addition, g_i (x) and $h_i(x)$ are inequality and equality constraints, respectively. They represent constraints, which the design must satisfy, such as stress and displacements limits.

It is not possible to directly utilize GAs to solve the constrained problems. In GAs, constraints are usually handled by using the concept of penalty functions, which penalize infeasible solutions.

There are several penalty methods. These are;

• Death Penalty

• Static Penalties

• Dynamic Penalties

In this study, An Improved GA Penalty Scheme proposed by Yokota et al.[27] is used. In this method the individuals are simply evaluated by using:

$$fitness(x) = f(x) * \left\{ 1 - \frac{1}{N} \sum_{i=1}^{N} P(i) \right\} \tag{6}$$

Where P(i) is the penalty function.

Considered a optimal design problem subjected to allowable stress constrain as:

$$G(i) = \sigma(i) \leq b(i) \quad (\text{i=1,...,N})$$

then; P(i)=0 if $G(i) \leq b(i)$

otherwise;

$$P(i) = \frac{G(i) - b(i)}{b(i)}$$

Where N indicates the number of constrain.

In this penalty scheme,

"The measure P(i) adopted above indicates how much the left-hand side of the expression corresponding to the constraint condition exceeds the right-hand side. The Improved GA Penalty Scheme gives information about unfeasible individual fitness as near as possible to the feasible region in the evaluation function [27]"

3.2.4 Mating pool operator

a. Selection

Selection is the way to decide which chromosomes in the initial population are fit enough to survive and possibly reproduce offspring in the next generation.
The selection method in a GA is designed to use fitness to guide the evolution of chromosomes by selective pressure. Those with higher fitness should have a greater chance of selection than those with lower fitness, thus creating a selective pressure towards more highly fit solutions. This process only retains the best fitness members of the population for the next iteration of the algorithm. The rest dies off. This process of natural selection must occur at each iteration step of the algorithm to allow the population of chromosomes to evolve over the generations to the fittest members as defined by the objective function. There are many different selection schemes. In this study, the most common selection scheme, roulette wheel selection, is used.
Roulette wheel approach is one of the fitness-proportional selections and it can select a new population with respect to the probability distribution based on fitness values.
In roulette wheel selection, all designs in the population must be ranked from highest to lowest according to the value of each chromosome's fitness.
Then the roulette wheel approach can be constructed with the following steps:

Step 1: Calculate the total fitness for the population

$$SUM = \sum_{k=1}^{popsize} fitness(k) \tag{7}$$

Step 2: Calculate selection probability, P_k, for each individual's *fitness (k)*

$$P_k = \frac{fitness(k)}{sum}, \qquad k = 1,2 \ldots popsize \tag{8}$$

Step 3: Calculate cumulative probability q_k for each individual's *fitness(k)*

$$q_k = \sum_{j=1}^{k} p_j, \qquad k = 1,2 \ldots popsize \tag{9}$$

Step 4: Generate a random number r from the range [0, 1].

Step 5: If $r \leq q_1$, then select the first individual *fitness(1)*; otherwise, select the k^{th} individual *fitness(k)* ($2 \leq k \leq popSize$) such that

$$q_{k-1} < r \leq q_k .$$

b. Elitist policy

The elitist policy simply carries forward the fittest individual from the previous generation into the next, in order not to lose their good properties as the generation evolves.

c. Crossover

The function of crossover is to cause chromosomes created during reproduction to differ from those of their parents.

Crossover is an extremely important part of genetic algorithm. If the crossover operator was deleted from a genetic algorithm the result is no longer a genetic algorithm.

Many different approaches have been tried as crossover in real-valued genetic algorithms. The simplest methods choose one or two points in the chromosome to be marked as the crossover points. Then the design variables between these points are merely swapped between the two parents.

For example, consider two parents as;

crossing point at k^{th} position

parents

$$\boldsymbol{x} = [x_1, x_2, \ldots, x_k, x_{k+1}, x_{k+2}, \ldots, x_n]$$

$$\boldsymbol{y} = [y_1, y_2, \ldots, y_k, y_{k+1}, y_{k+2}, \ldots, y_n]$$

Figure 3: Example of One Point Cut Crossover

First the crossover point is randomly selected, then the design variables in between are exchanged:

offspring

$$\boldsymbol{x}' = [x_1, x_2, \ldots, x_k, y_{k+1}, y_{k+2}, \ldots, y_n]$$

$$\boldsymbol{y}' = [y_1, y_2, \ldots, y_k, x_{k+1}, x_{k+2}, \ldots, x_n]$$

Figure 4: Offspring After One Point Cut Crossover

d. Mutation

The function of mutation is to avoid the problem of overly fast convergence, and force the routine to explore other areas of the population by randomly introducing changes in some of the design variables.

If the mutation operator was deleted from a genetic algorithms the result could end up in a local rather than a global optimum.

The mutation operator randomly alters each gene with a small probability .

If the mutation rate is high, too many good design variables are mutated, and the algorithm stalls.

A mutation example is shown below;

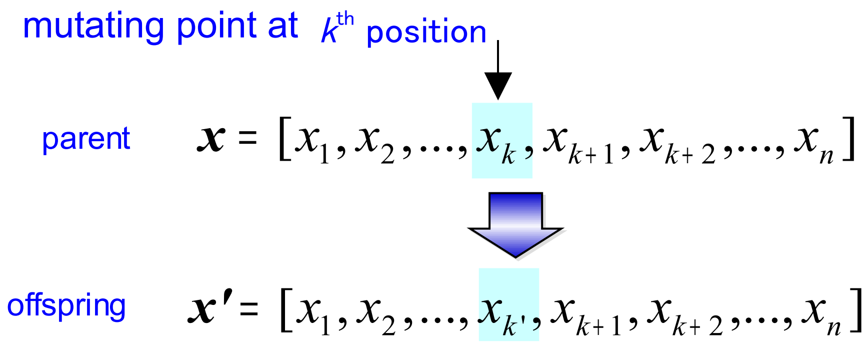

Figure 5: Example of Mutation

3.2.5 Convergence criteria

The convergence criteria used is the limit on the maximum number of generation. When the number of generations reaches to a predefined value then the optimization process stops.

3.3 The advantages of a genetic algorithm

Genetic algorithms have received considerable attention regarding their potential as an optimization technique. There are some advantages when applying genetic algorithms to optimization problems.

- Genetic algorithms do not have much mathematical requirements about the optimization problems. Due to their evolutionary nature, genetic algorithms will search for solutions regardless to the inner workings of the problem.

- Genetic algorithms can handle any kind of objective functions and any kind of constraints, *i.e.*, linear or nonlinear, defined on discrete or continuous search spaces.
- The evolution operators make genetic algorithms very effective at performing global search (in probability).
- Genetic Algorithms can deal with a large number of design variables.
- They can simultaneously search from a wide sampling of the population.
- Also, they provide a list of optimum design variables, not just a single solution.

3.4 Shape Optimization

The first two decades of numerical structural optimization were almost focused on sizing design variables. Sizing design variables can be bar cross-sectional area, plate thickness, moments of inertia, which do not require a change in the finite element model of the structure as they are changed. Shape optimization is more complex than the pure sizing optimization. Since the shapes are continuously changing in the design process. "However, for many problems shape design is more effective than sizing design. A typical example is that of a stress concentration at a hole boundary in a panel. Sizing design would increase or decrease the thickness of the panel near the hole, while shape design would change the shape of the hole boundary" Ref [29].

The present examples in this study are all structural shape optimization problems.

The optimum shape in these examples was found using the finite element method with the node coordinates as design variables.

The main difficulty encountered in this study was due to the continuously changing finite element model. It is difficult to ensure that the accuracy of the finite element analysis remain adequate throughout the design process.

CHAPTER 4

NUMERICAL EXPERIMENTS AND DISCUSSIONS

In this section the proposed framework of shape optimization is applied to four 2D example problems, under plane stress conditions. Shape optimization problems are solved and compared with the previous results in literature and papers.

4. 1 Flat Plate with Hole

The first problem to be considered is the Flat Plate with a Square Hole.

A thin 80x80 inch square flat plate with a central square hole of 12x12 inch and its finite element quarter model are shown in Figure 6. The plate is assumed to have unit thickness. A biaxial stress field is applied by specifying P_1=15000 psi and P_2=10000 psi in the x and y directions respectively. The optimization problem is to minimize the area of a plate according to the stress constraints.

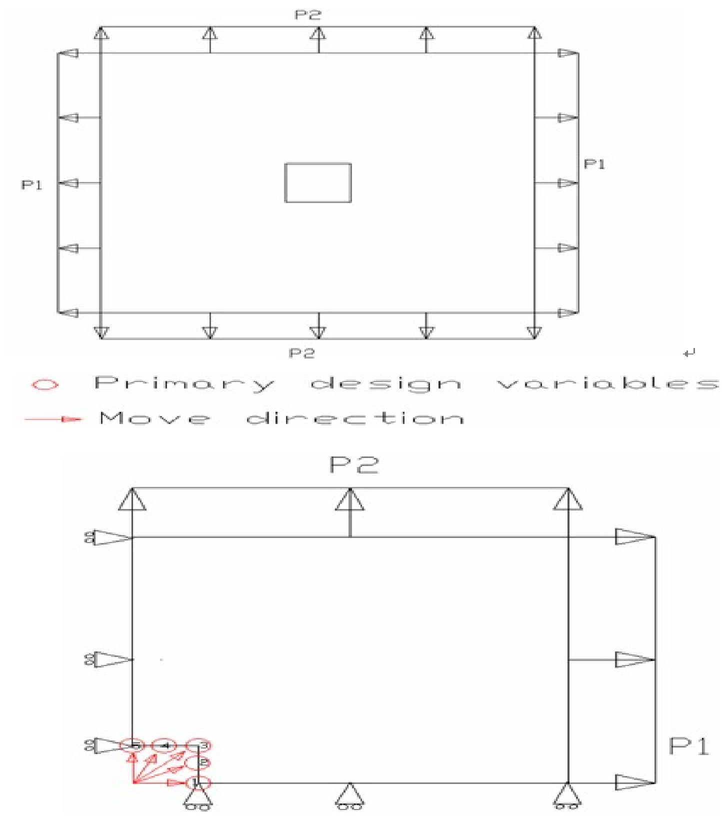

Figure 6: Plane Stress Model of a Plate

The design parameters are:

Modulus of elasticity: $E = 1\times 10^{7}$ psi

Poisson's ratio : v = 0.3

Allowable von-Mises stress: = $= 3.5\times 10^{4}$ psi

Initial area: 1564 in^2

Finite element meshing characteristics:

Number of element: 400 four-node quad elements

Number of nodes: 441 nodes

Total degree of freedom: 441 x 2 DOF/Node = 882

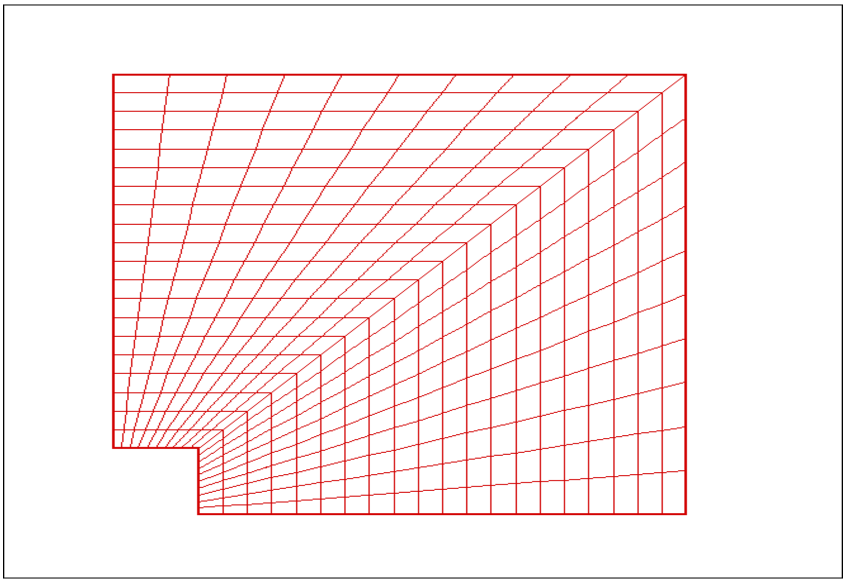

Figure 7: FEM mesh for initial design

Design variables: The coordinate of points 1, 2, 3, 4, 5

Objective: Total area of the plate

Optimization parameters

Number of design variables: 8

Step size of design variables: 0.02

Initial population: 100

Selection type: roulette wheel selection

Crossover probability: 1.00

Mutation probability: 0.06

Elitist selection: 2

The convergence is obtained at generation 99 and the final area is 1263 in^2.

Since the value of optimized area is not given in the paper, it was not possible to compare the final results with the final area, but the final shape of the plate is given and this is compared with the shape obtained using the Genetic Algorithm Code.

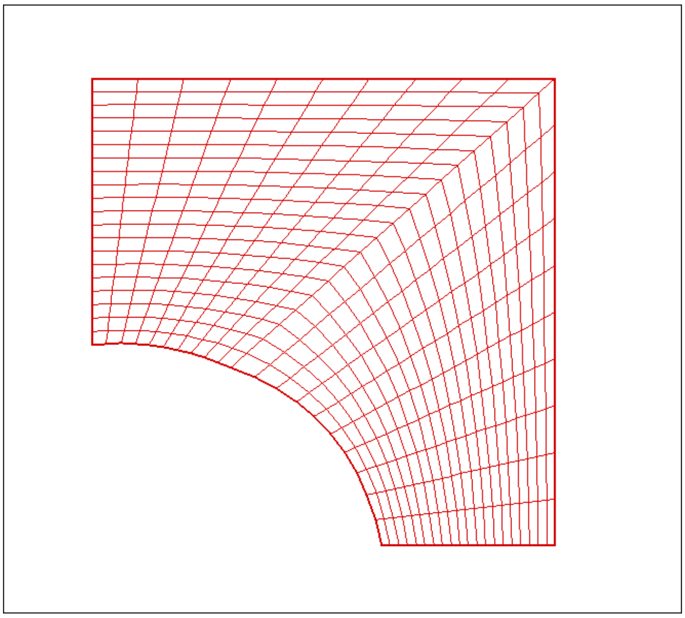

Figure 8: Final finite element mesh of the plate

Result comparison between proposed solution and the reference paper solution

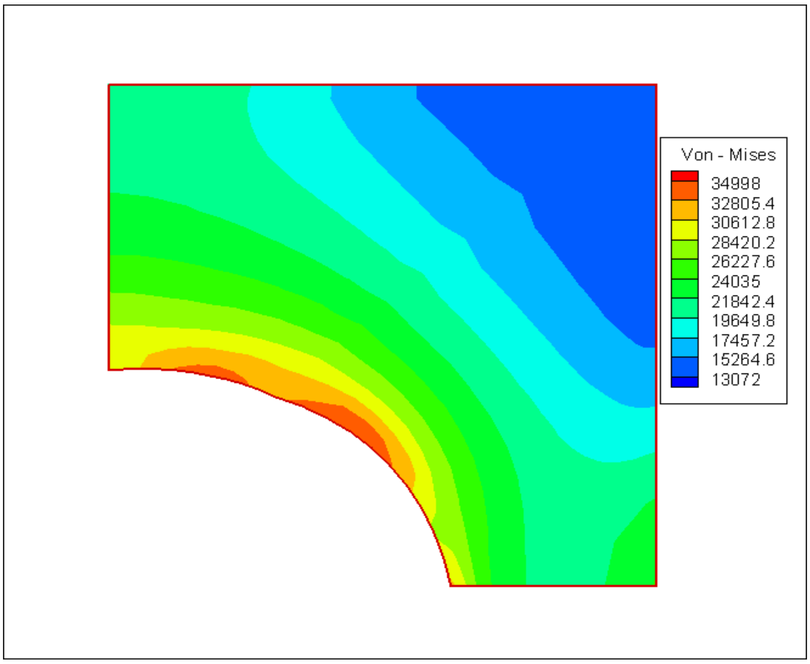

Figure 9: The contour plot of the Von-Mises stress at the final step

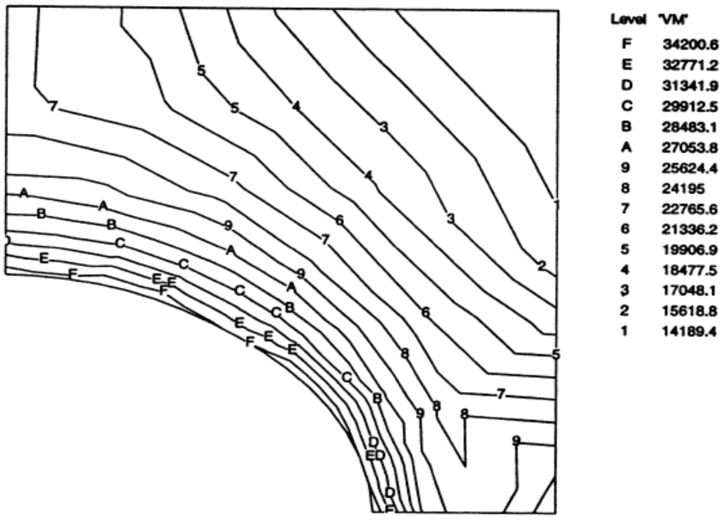

Figure 10: Final shape of the plate from Ref [24]

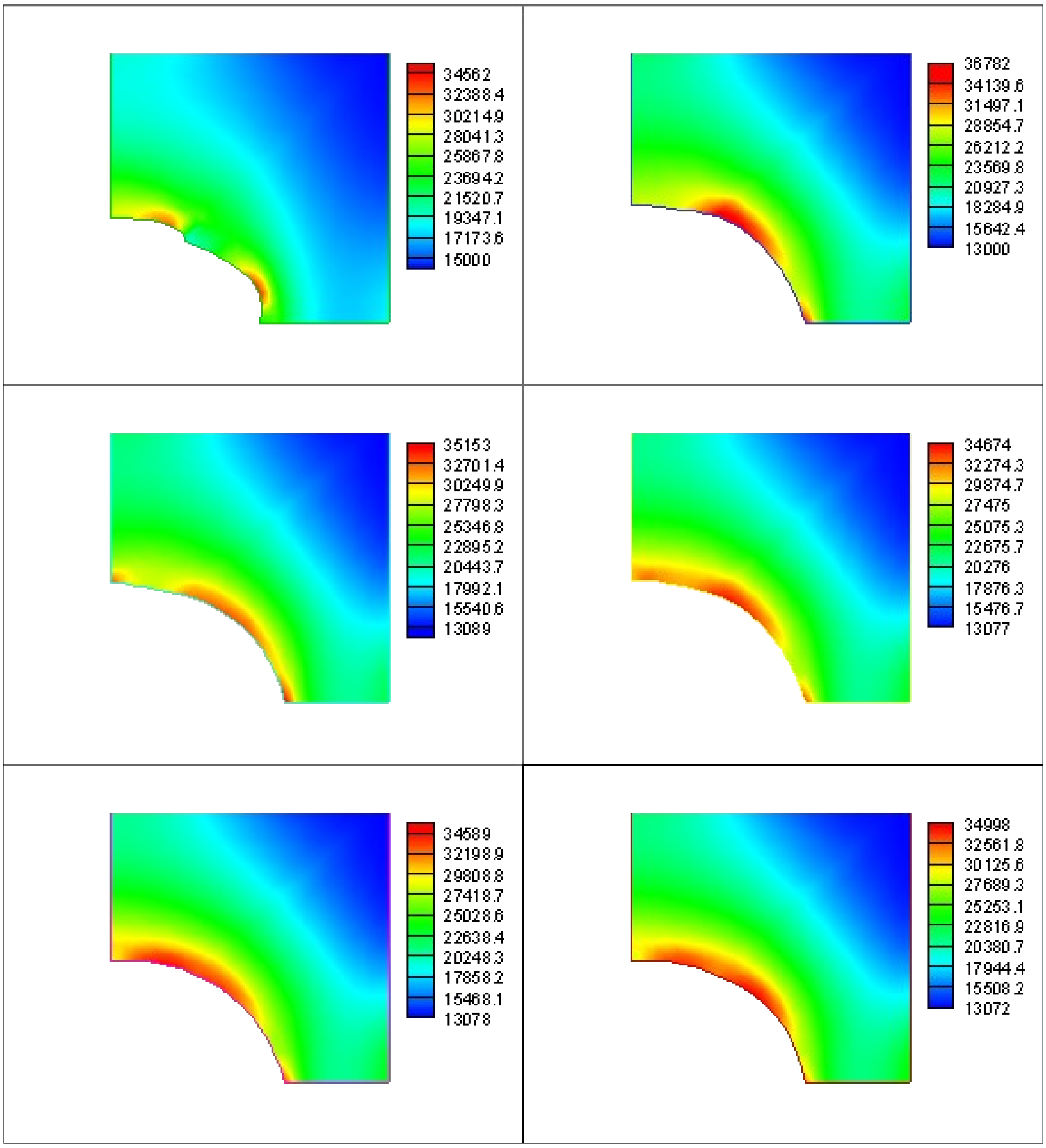

Figure 11: Generation process during shape optimization after: a) 10, b) 30, c) 50,d) 60, e) 80 and f) 99 generations

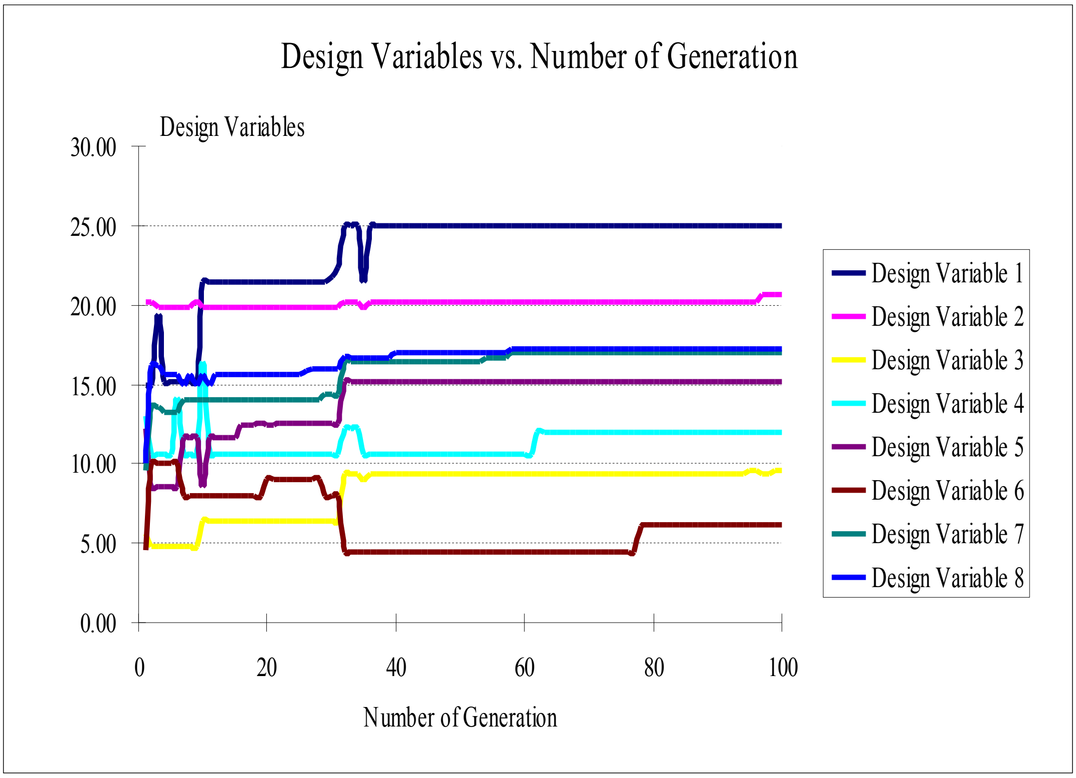

Figure 12: History of Design Variables vs. Number of Generation

To investigate the effectiveness of mutation probability of the algorithm, 7 more solutions have been performed using different mutation probabilities.

4.1.1 Run One with Mutation Probability Equal to 0.02

Optimization parameters

Number of design variables: 8

Step size of design variables: 0.02

Initial population: 100

Selection type: roulette wheel selection

Crossover probability: 1.00

Mutation probability: 0.02

Elitist selection: 2

The convergence is obtained at generation 96 and the final area is 1330.919 in^2 .

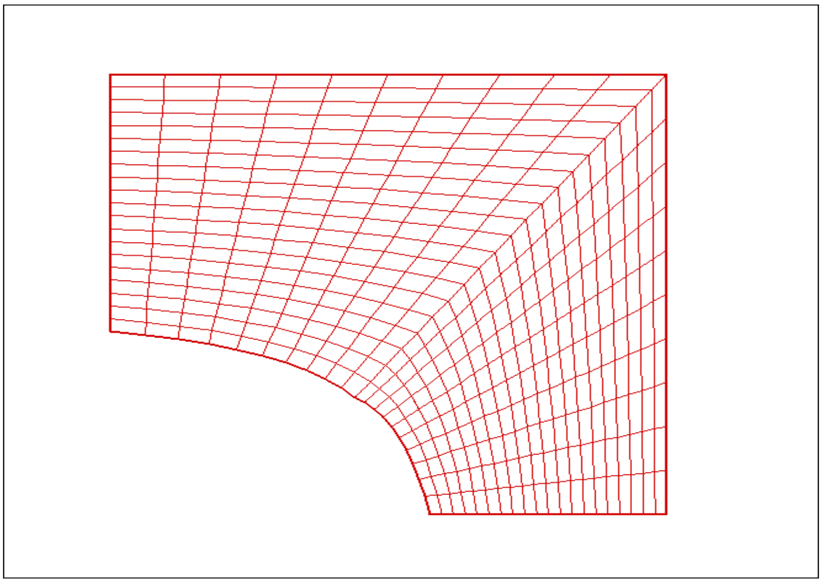

Figure 13: Final finite element mesh of the plate

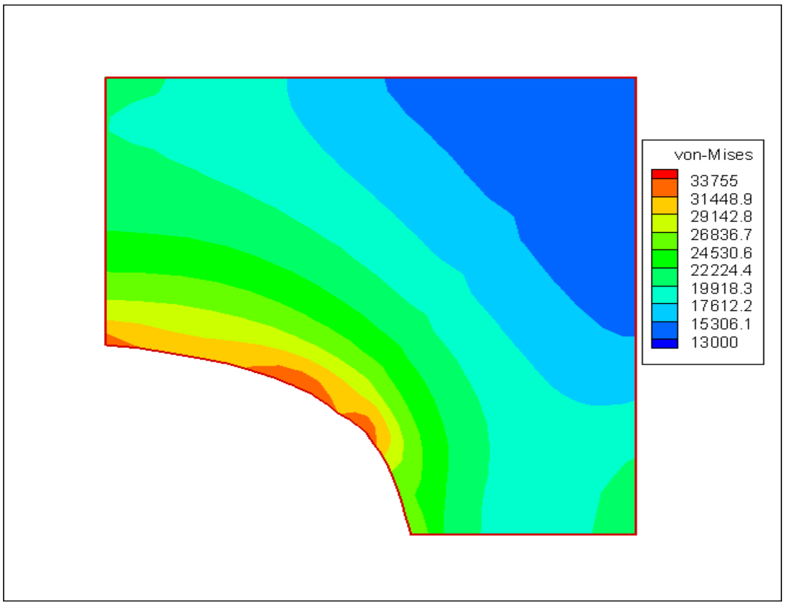

Figure 14: Contour plot of the Von-Mises stress at the final step with mutation probability=0.02

4.1.2 Run Two with Mutation Probability Equal to 0.03

Optimization parameters

Number of design variables: 8

Step size of design variables: 0.02

Initial population: 100

Selection type: roulette wheel selection

Crossover probability: 1.00

Mutation probability: 0.03

Elitist selection: 2

The convergence is obtained at generation 80 and the final area is 1287.975 in^{2}.

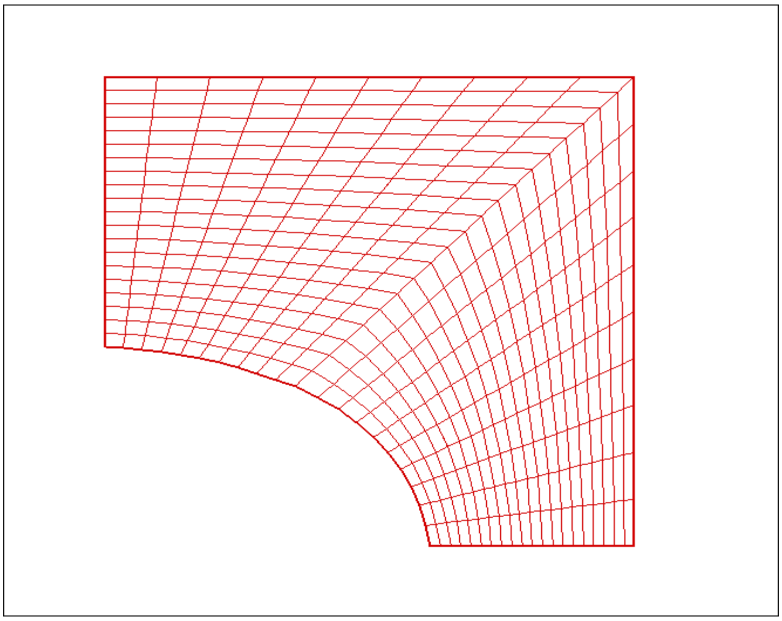

Figure 15: Final finite element mesh of the plate

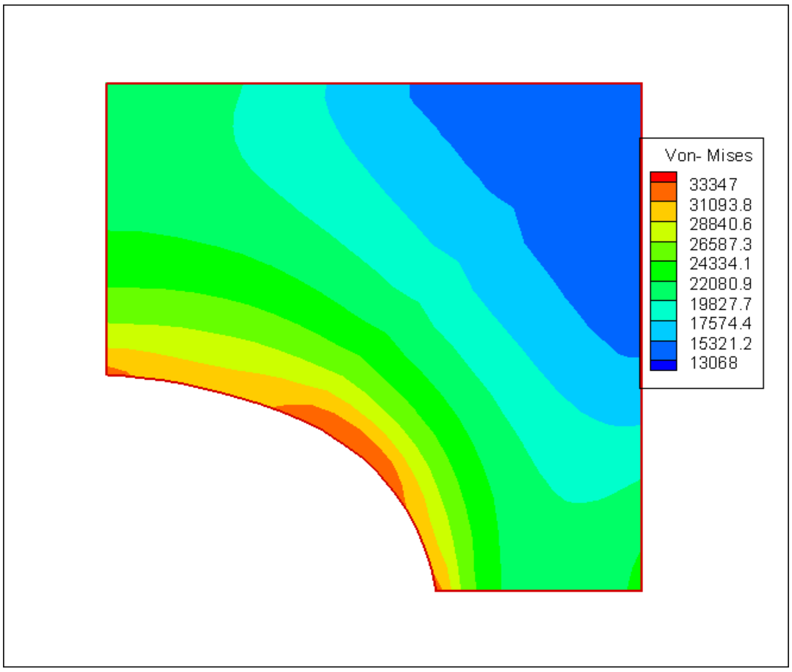

Figure 16: Contour plot of the Von-Mises stress at the final step with mutation probability=0.03

4.1.3 Run Three with Mutation Probability Equal to 0.05

Optimization parameters

Number of design variables: 8

Step size of design variables: 0.02

Initial population: 100

Selection type: roulette wheel selection

Crossover probability: 1.00

Mutation probability: 0.05

Elitist selection: 2

The convergence is obtained at generation 99 and the final area is 1329 in^2.

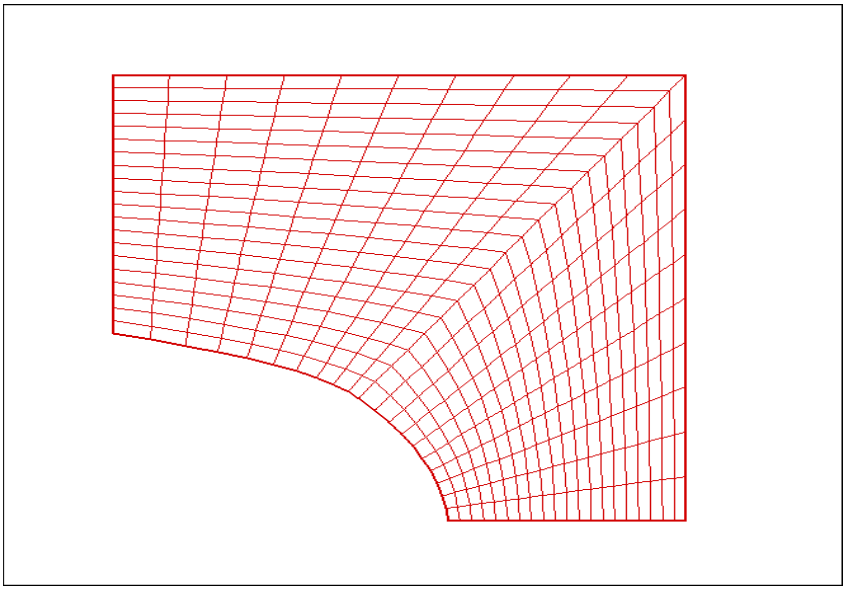

Figure 17: Final finite element mesh of the plate

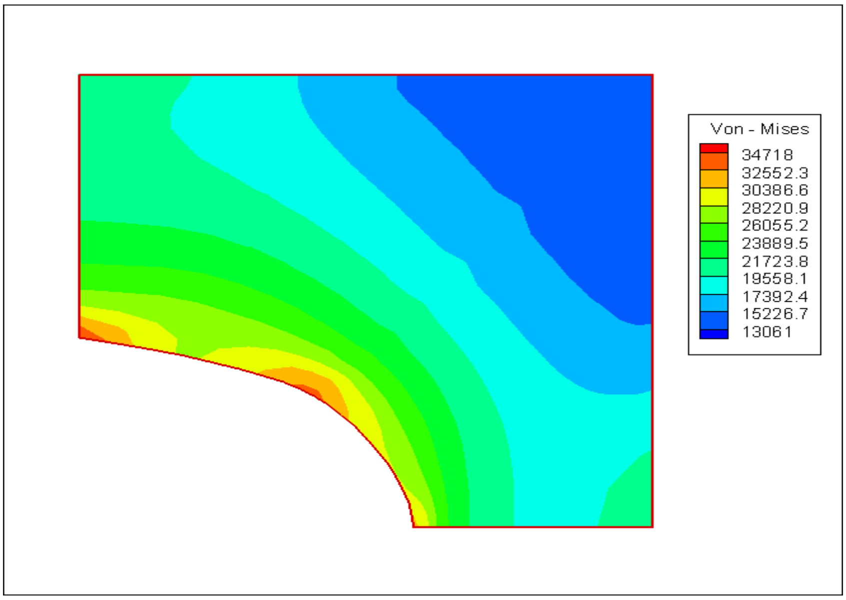

Figure 18: Contour plot of the Von-Mises stress at the final step with mutation probability=0.05

4.1.4 Run Four with Mutation Probability Equal to 0.07

Optimization parameters

Number of design variables: 8

Step size of design variables: 0.02

Initial population: 100

Selection type: roulette wheel selection

Crossover probability: 1.00

Mutation probability: 0.07

Elitist selection: 2

The convergence is obtained at generation 81 and the final area is 1281 in^{2}.

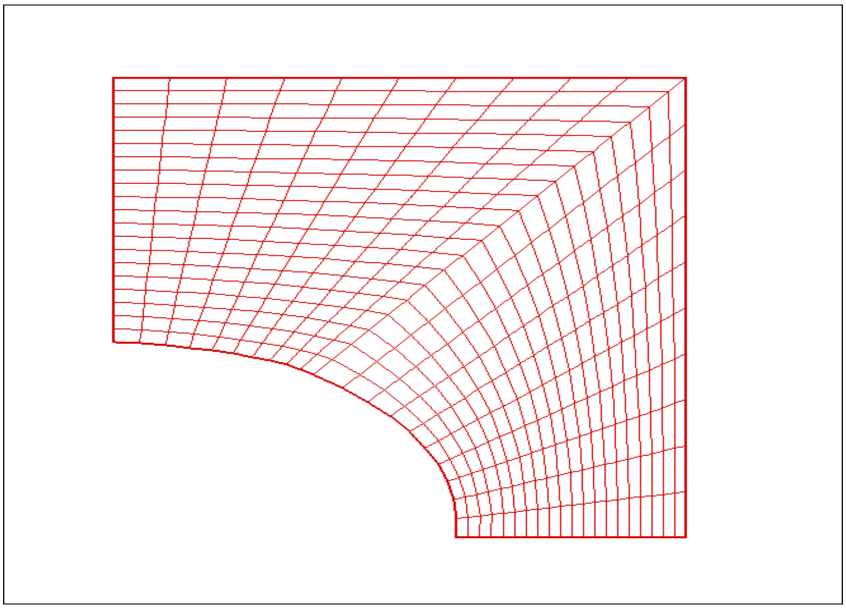

Figure 19: Final finite element mesh of the plate

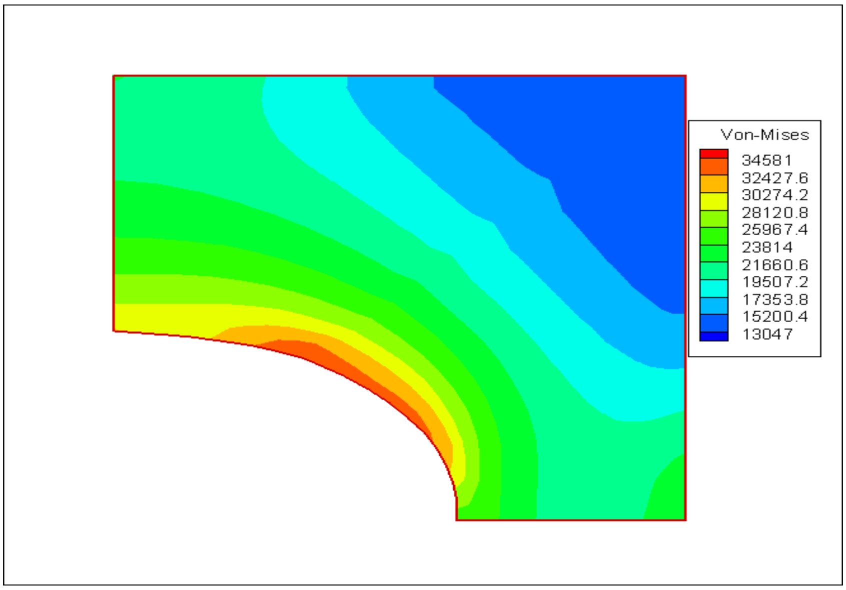

Figure 20: Contour plot of the Von-Mises stress at the final step with mutation probability=0.07

4.1.5 Run Five with Mutation Probability Equal to 0.08

Optimization parameters

Number of design variables: 8

Step size of design variables: 0.02

Initial population: 100

Selection type: roulette wheel selection

Crossover probability: 1.00

Mutation probability: 0.08

Elitist selection: 2

The convergence is obtained at generation 92 and the final area is 1298 in^2.

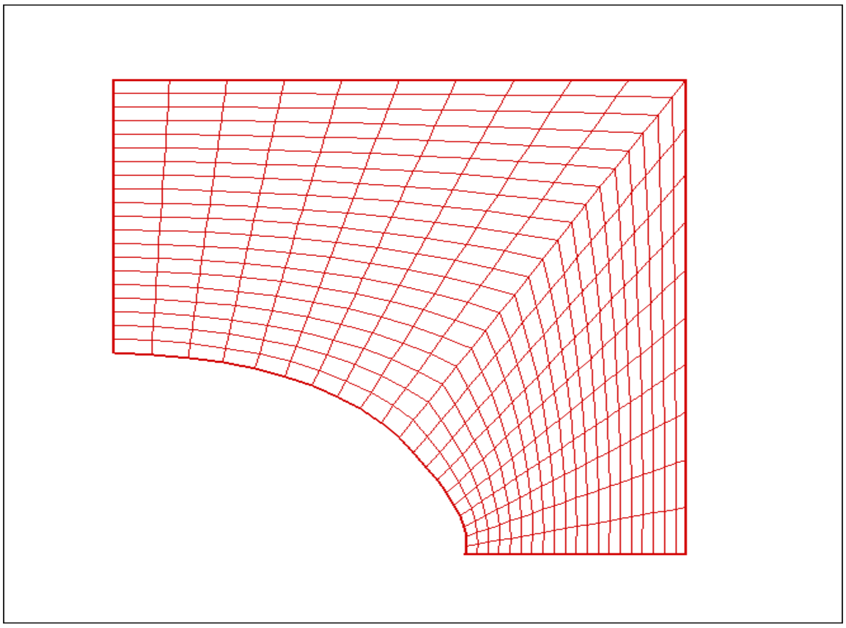

Figure 21: Final finite element mesh of the plate

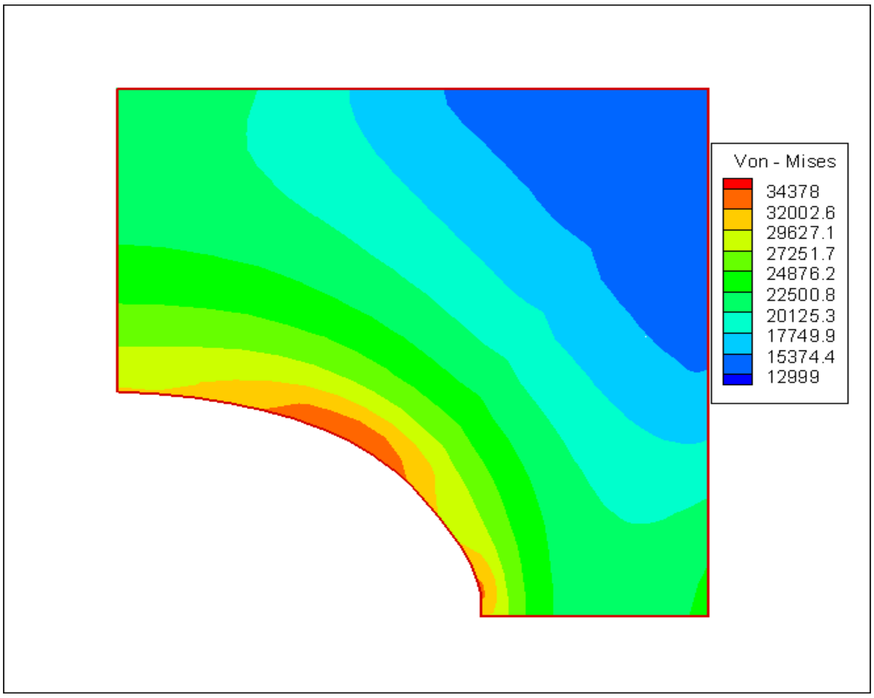

Figure 22: Contour plot of the Von-Mises stress at the final step with mutation probability=0.08

4.1.6 Run Six with Mutation Probability Equal to 0.09

<u>*Optimization parameters*</u>

Number of design variables: 8

Step size of design variables: 0.02

Initial population: 100

Selection type: roulette wheel selection

Crossover probability: 1.00

Mutation probability: 0.09

Elitist selection: 2

The convergence is obtained at generation 91 and the final area is 1303 in^2.

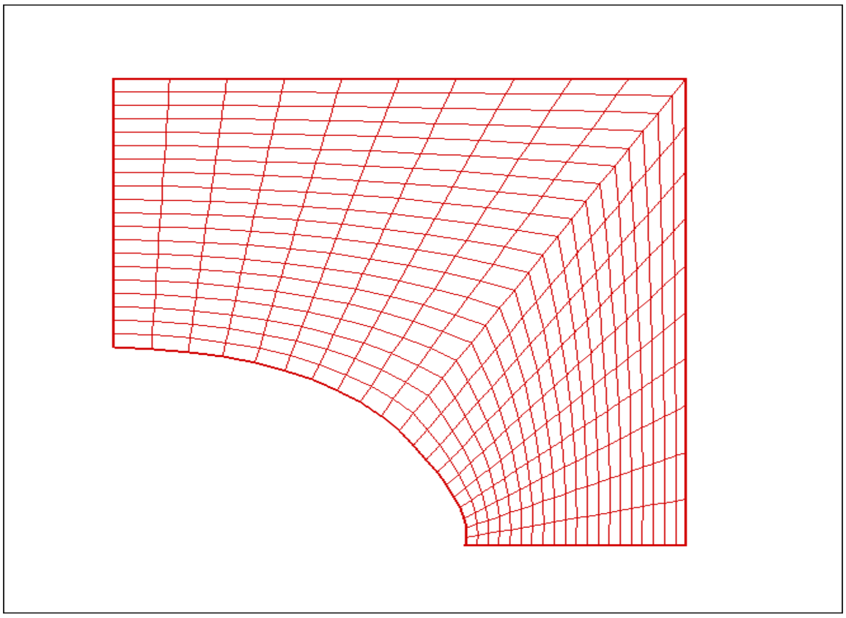

Figure 23: Final finite element mesh of the plate

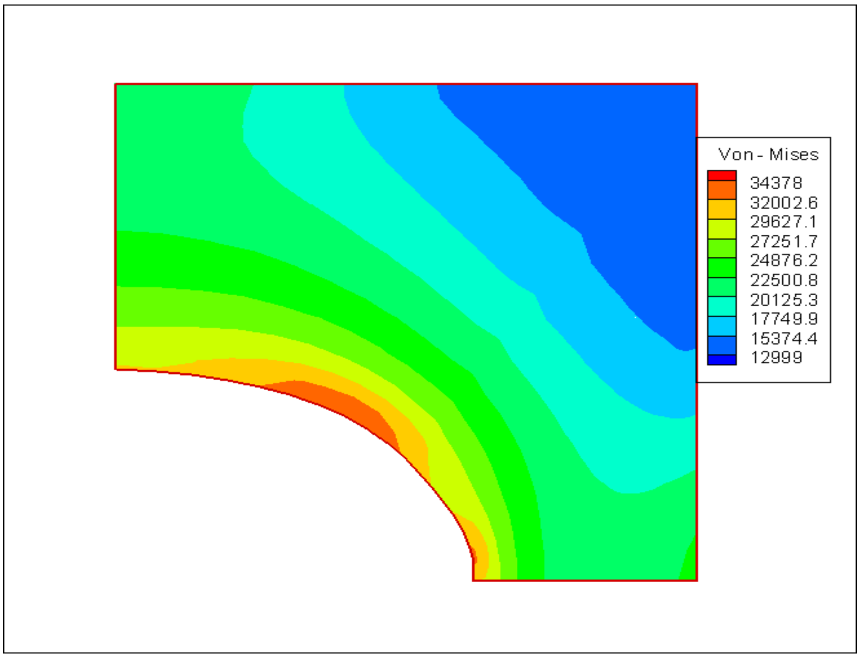

Figure 24: Contour plot of the Von-Mises stress at the final step with mutation probability=0.09

4.1.7 Run Seven with Mutation Probability Equal to 0.10

Optimization parameters

Number of design variables: 8

Step size of design variables: 0.02

Initial population: 100

Selection type: roulette wheel selection

Crossover probability: 1.00

Mutation probability: 0.10

Elitist selection: 2

The convergence is obtained at generation 96 and the final area is 1302 in^2.

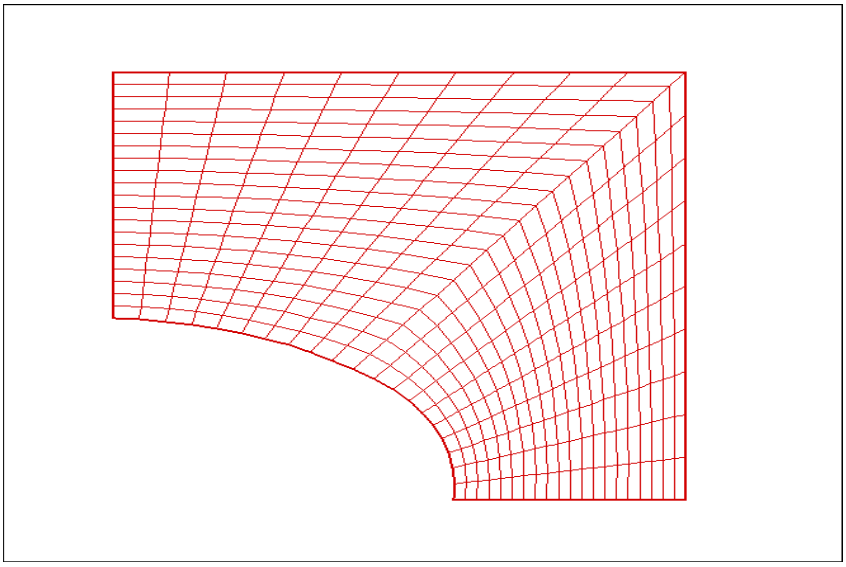

Figure 25: Final finite element mesh of the plate

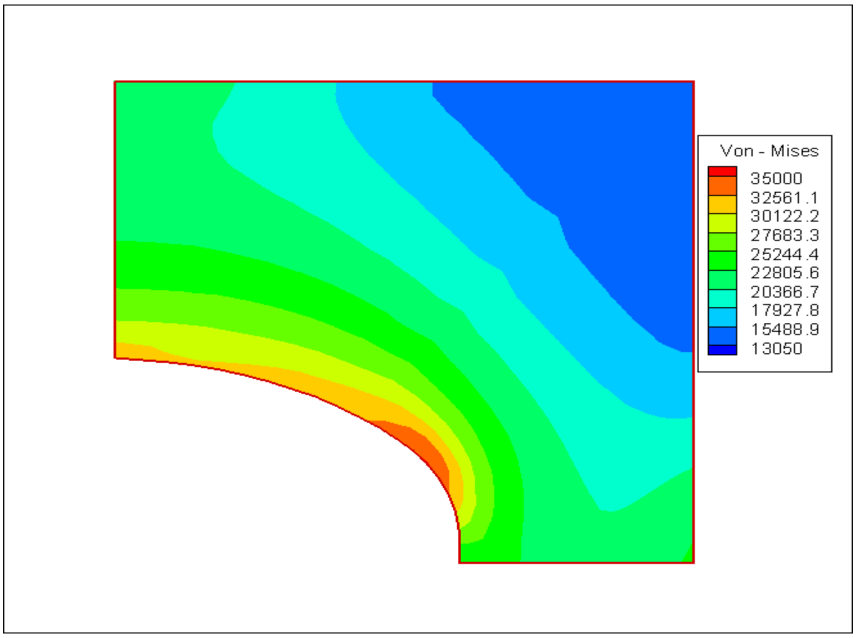

Figure 26: Contour plot of the Von-Mises stress at the final step with mutation probability=0.1

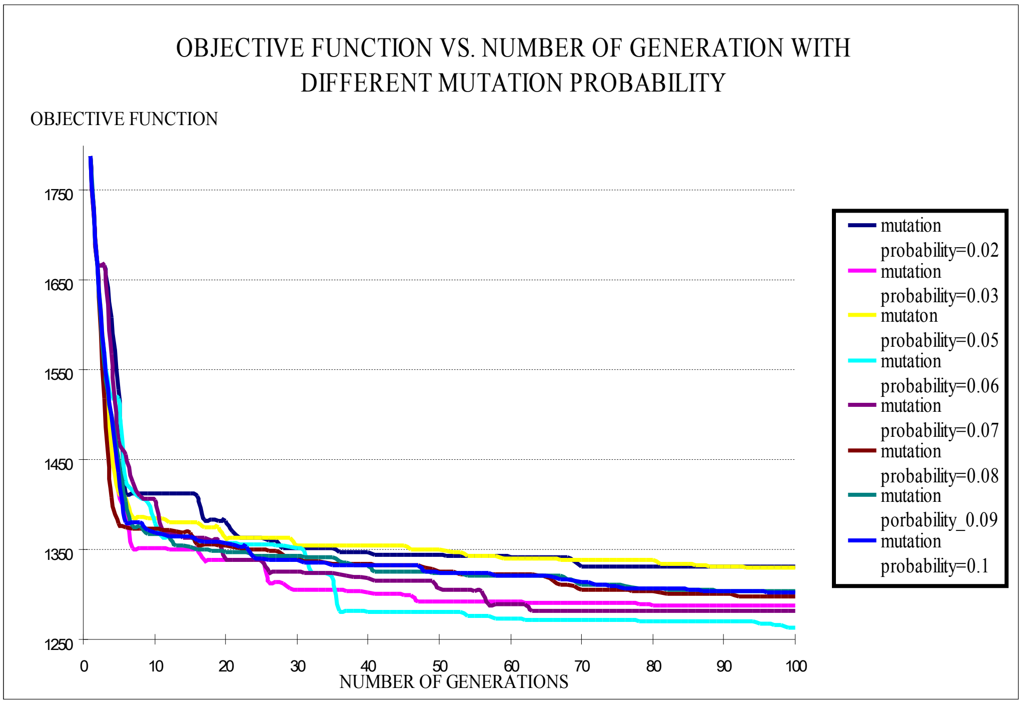

Figure 27: Objective function vs. number of generation with different mutation probability

4.1.8 Discussion

In this first sample problem, the shape of a plate with a square hole has been optimized. Only quarter of the plate was modeled using the symmetrical boundary conditions.

The objective is to minimize the area of the plate under stress constraint conditions.

Kristensen and Madsen [28] concluded that the analytical solution for the optimum shape of the hole in an infinite plate under a biaxial loading condition should be an ellipse with the axis ratio b/a equal to the ratio of biaxial stresses, that is, $b/a = p_1/p_2$, where a and b represent the axes of the ellipse.
In this sample problem the radio of p_1/p_2 is 1.5.

Figure 11 shows the process of the numerical optimization. After 30 generations the smoothed hole shape is showed in Figure 11(b). The changes in the hole shape during the next 99 generations are shown in Figure 11(c) – Figure 11(f). From Figure 12 the final result of the major and minor axes ratio was 1.453, which is only a 3.13% error compared with the analytical result. This shows that the proposed code with the reference paper were in consistency for the final shape of the plate.

For this example to investigate the effectiveness of mutation probability of the algorithm, 7 more runs have been performed using different mutation probabilities. Figure 27 shows that the best fitness result is obtained when the mutation probability is equal to 0.06, for a mutation probability ranging from 0.02 to 0.10.

In order to compare the results with the literature ones British Units are used in this sample problem.

4.2 Shoulder Fillet in Flat Plate under Tension

The second problem to be considered is The Shoulder Fillet in a Plate under Tension.

Figure 28 shows the geometries used for the initial design. Figure 29 shows the FEM mesh for the initial design. The plate is assumed to have unit thickness. The displacement U_x of the left vertical face is restrained and displacement U_y is restrained for the bottom horizontal face. The stress applied at the end is kept as 1N/mm^2. The optimization problem is to minimize the area of a plate according to the stress constraints.

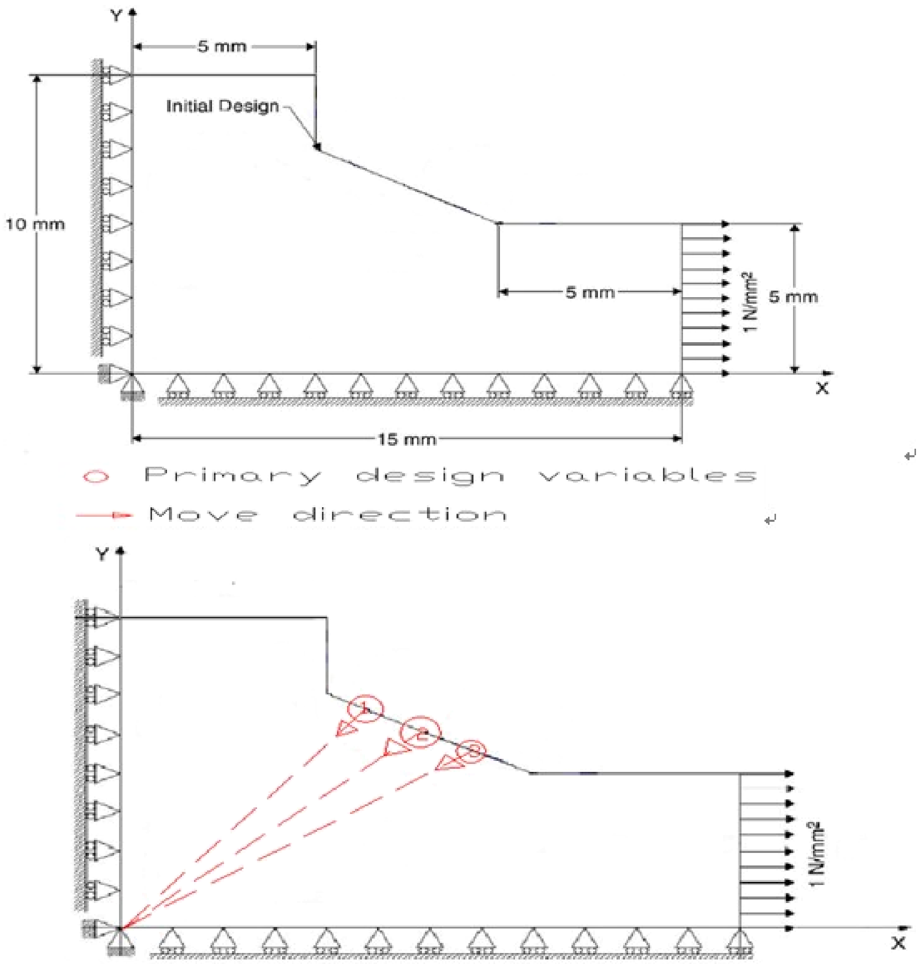

Figure 28: Plane Stress Model of a Fillet

<u>*Design parameters are:*</u>

Modulus of elasticity : E = 72400 N/mm^2 (Aluminum alloy)

Poisson's ratio : v = 0.33

Allowable Von-Mises stress : = 1.19 N/mm^2

Initial area: 106.25 mm^2

Finite element meshing characteristics:

Number of element: 200 four-node quad elements

Number of nodes: 234 nodes

Total degree of freedom: 234 x 2 DOF/Node = 468

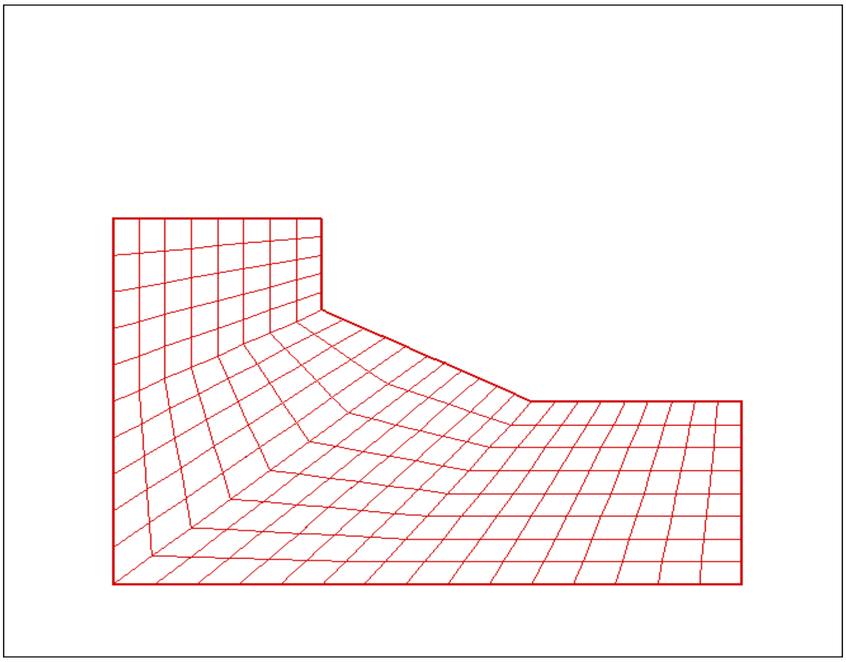

Figure 29: FEM mesh for initial design

__Design variables:__ The coordinate of points 1, 2, 3

__Objective:__ Total area of the plate

__Optimization Parameters__

Number of design variables: 6

Step size of design variables: 0.02

Initial population: 100

Selection type: roulette wheel selection

Crossover probability: 1.00

Mutation probability: 0.01

Elitist selection: 2

The convergence is obtained at generation 57 and the final area is 98.65723 m^2.

Since the value of optimized area is not given in the paper, it was not possible to compare the final results with the final area, but the final shape of the plate is given and this is compared with the shape obtained using the Genetic Algorithm Code.

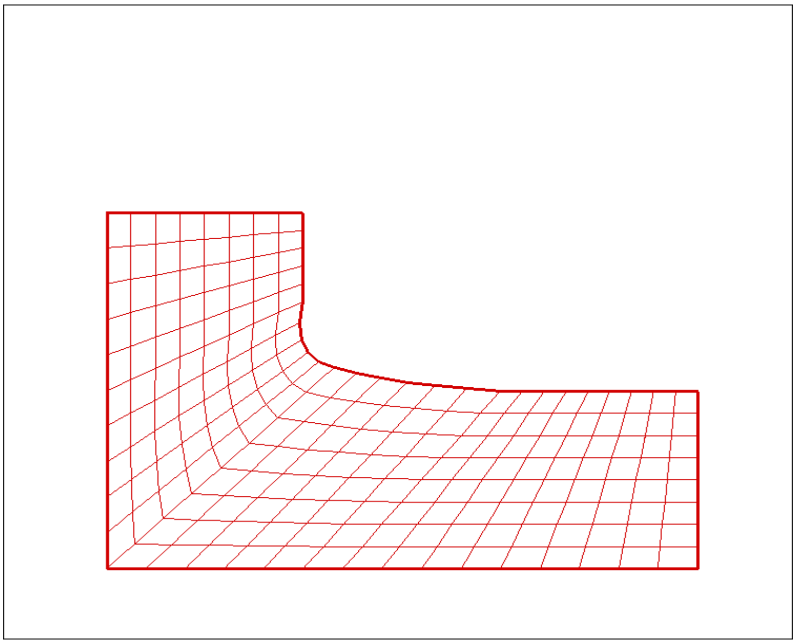

Figure 30: Final finite element mesh of the plate

Result comparison between proposed solution and reference paper

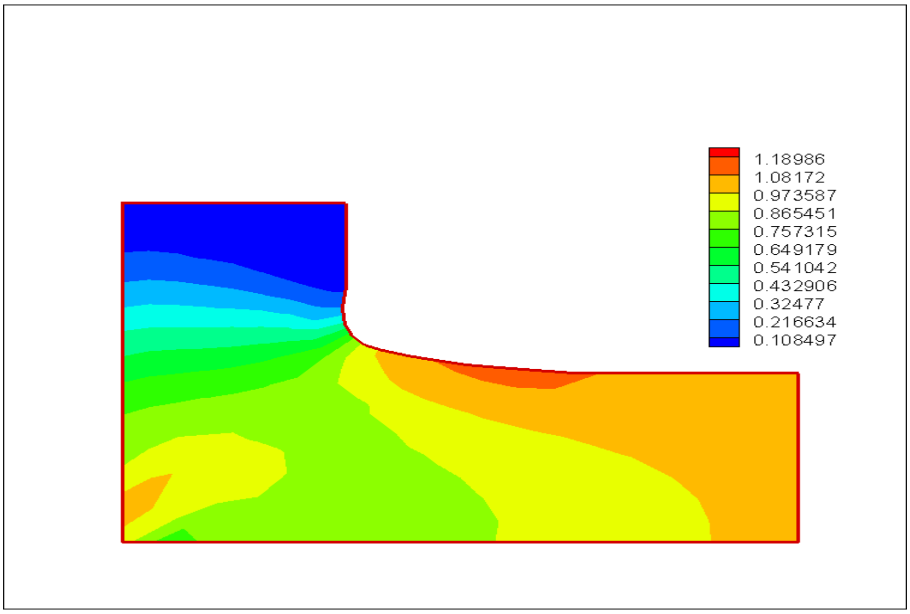

Figure 31: The contour plot of the Von-Mises stress at the final step

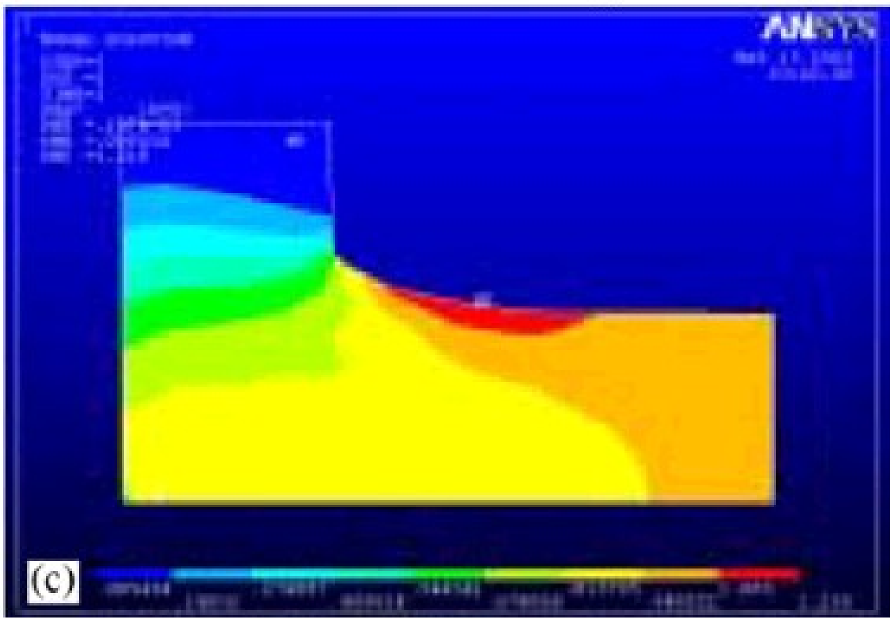

Figure 32: Final shape of the plate from Ref [9]

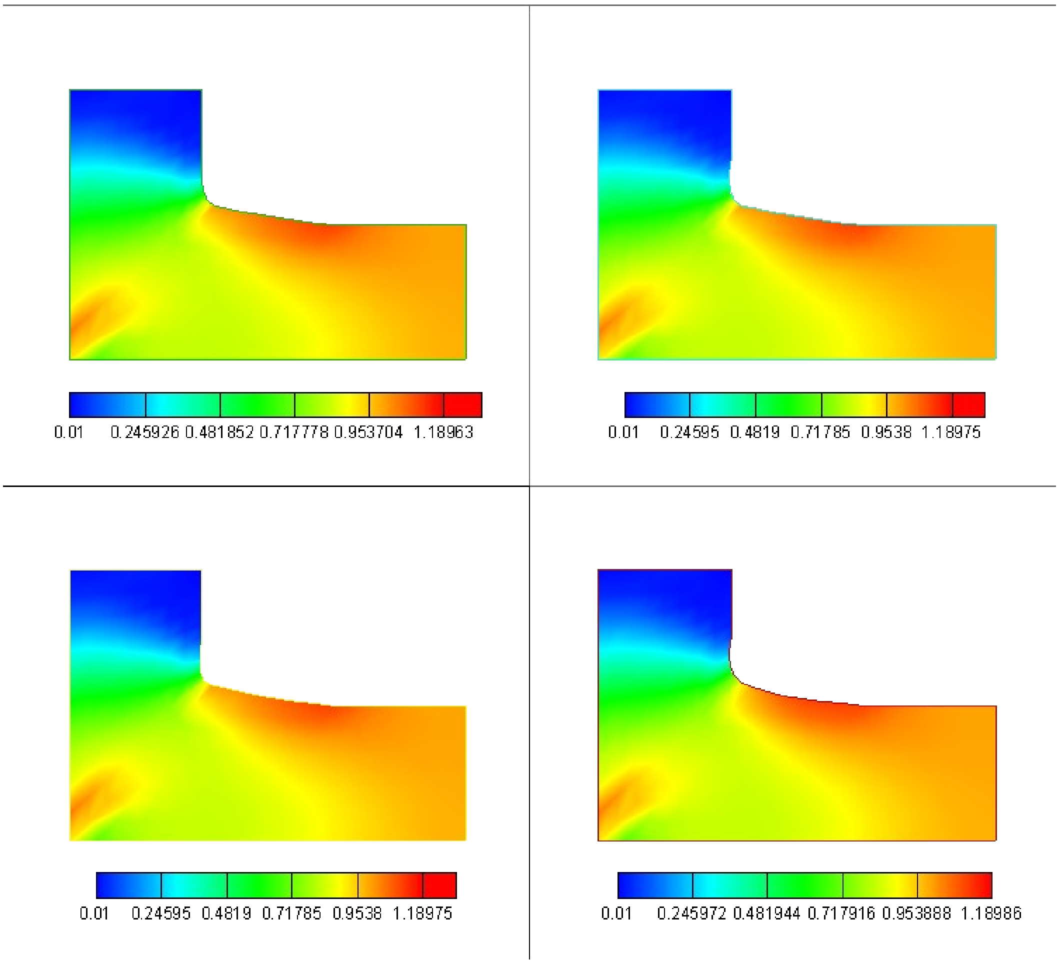

Figure 33: Generation process during shape optimization after; a) 10, b) 20, c) 30 and d) 57 generations

To investigate the stability of the algorithm, 5 more runs have been performed using the same optimization design variables.

4.2.1 First run

The convergence is obtained at generation 33 and the final area is 98.33775 m^2.

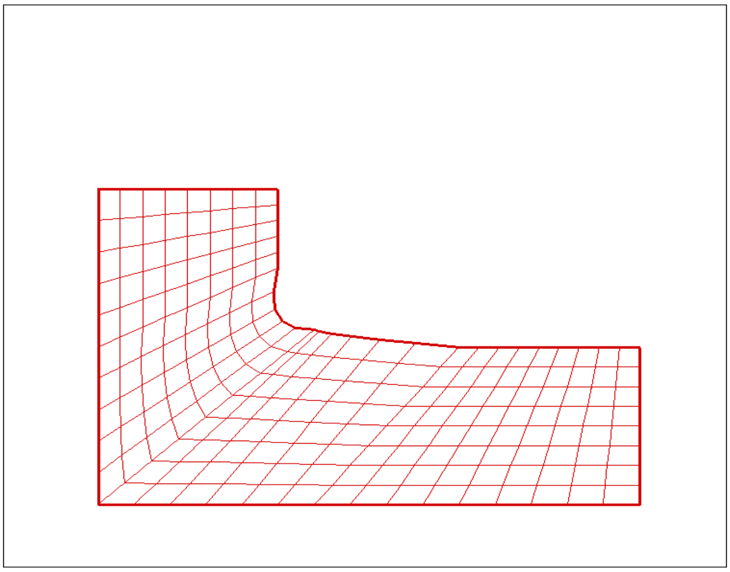

Figure 34: Final finite element mesh of the plate

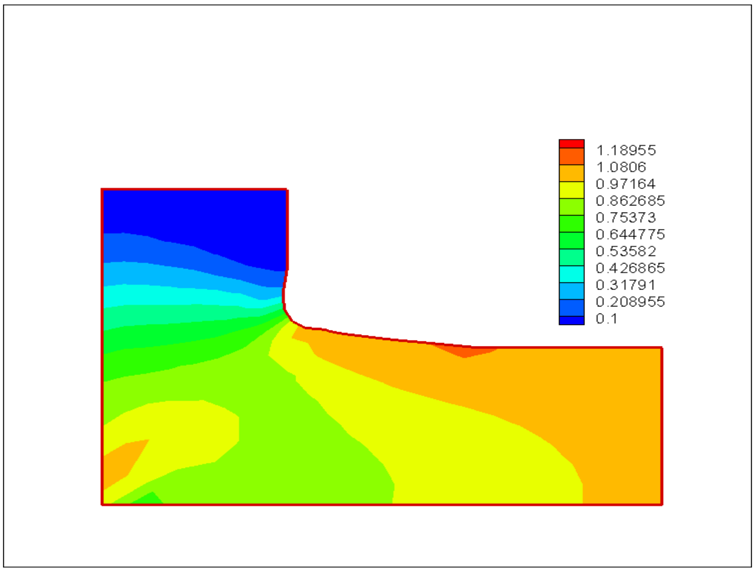

Figure 35: Contour plot of the Von-Mises stress at the final step after the first run.

4.2.2 Second run

The convergence is obtained at generation 36 and the final area is 98.64266 m^2.

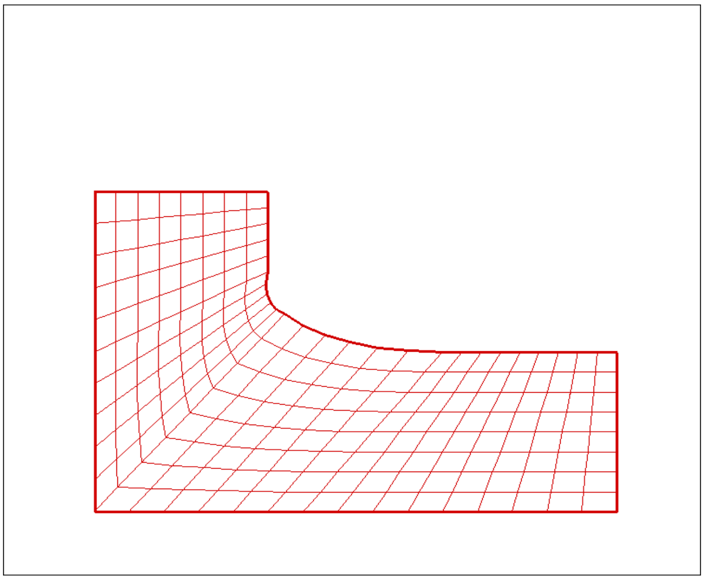

Figure 36: Final finite element mesh of the plate

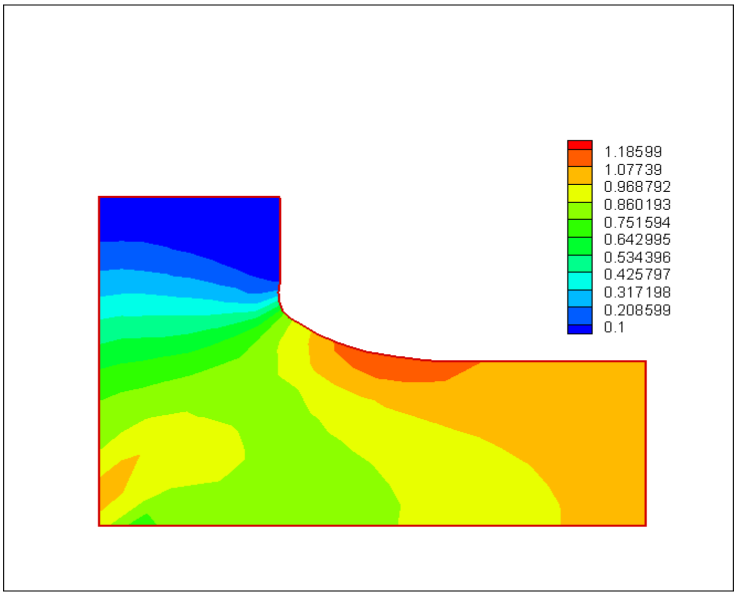

Figure 37: Contour plot of the Von-Mises stress at the final step after the second run

4.2.3 Third run

The convergence is obtained at generation 71 and the final area is 98.52057 m^2.

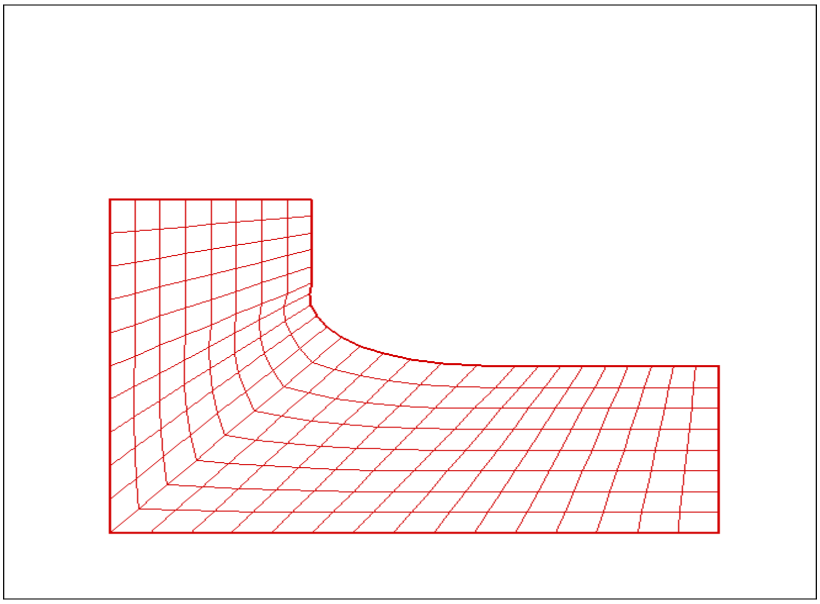

Figure 38: Final finite element mesh of the plate

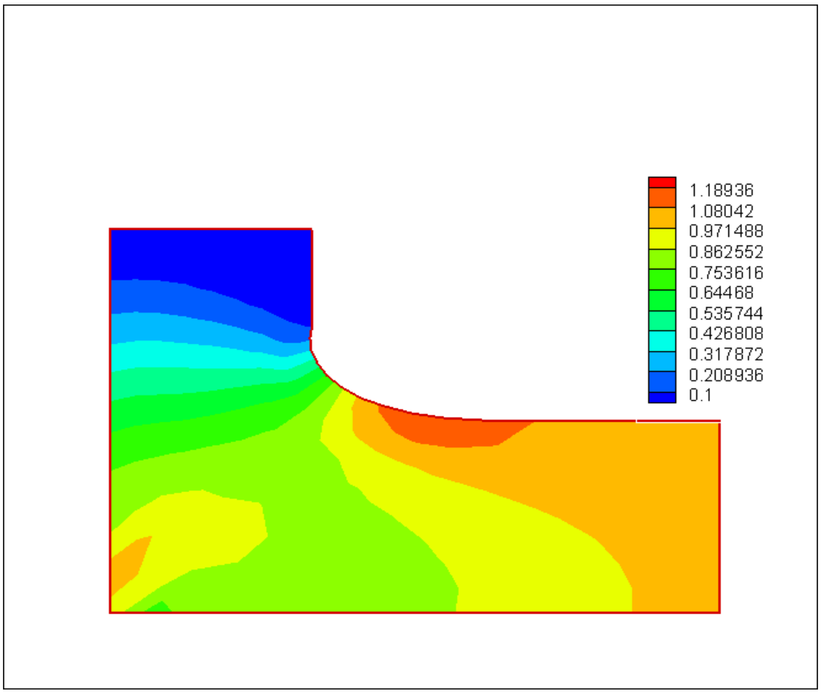

Figure 39: Contour plot of the Von-Mises stress at the final step after the third run.

4.2.4 Fourth run

The convergence is obtained at generation 14 and the final area is 98.56158 m^2.

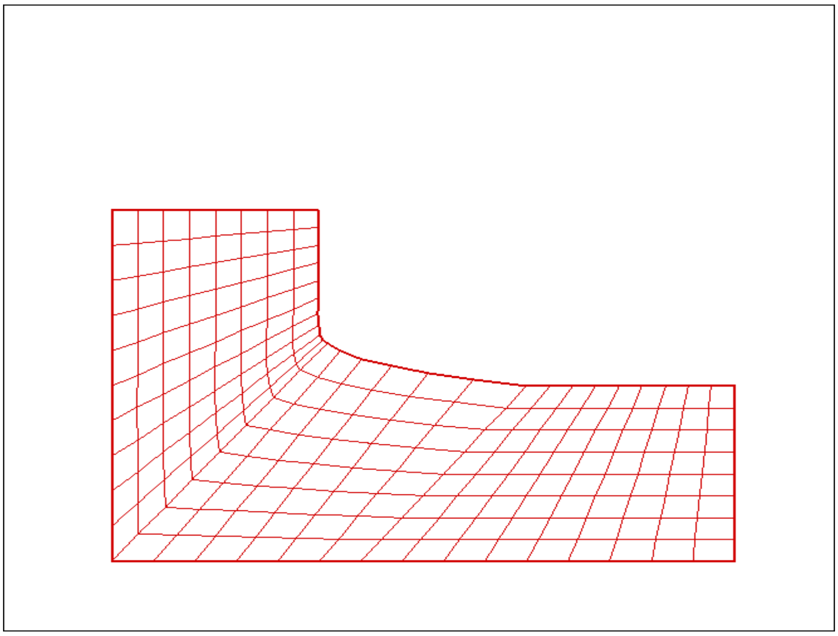

Figure 40: Final finite element mesh of the plate

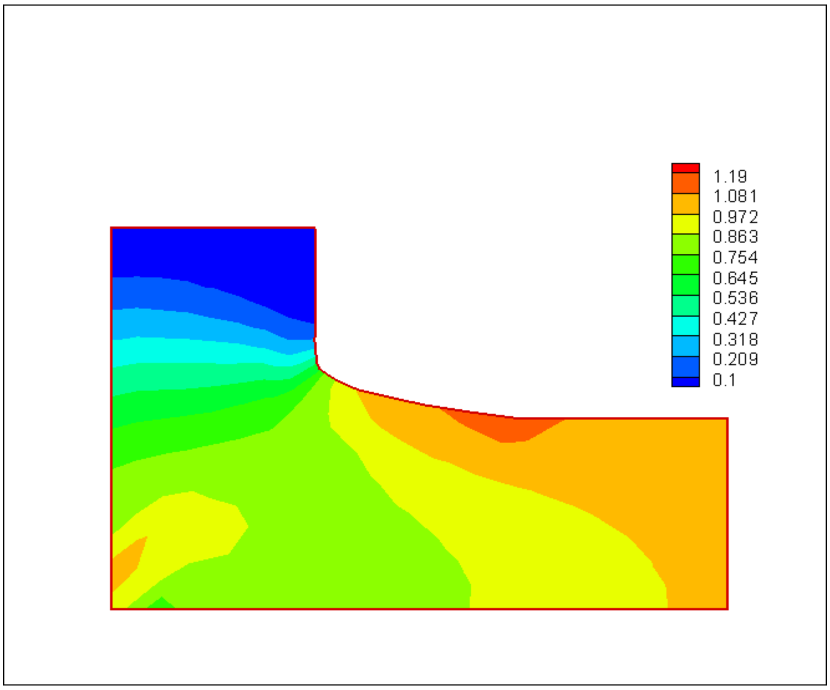

Figure 41: Contour plot of the Von-Mises stress at the final step after the fourth run

4.2.5 Fifth run

The convergence is obtained at generation 66 and the final area is 98.47289 m^2.

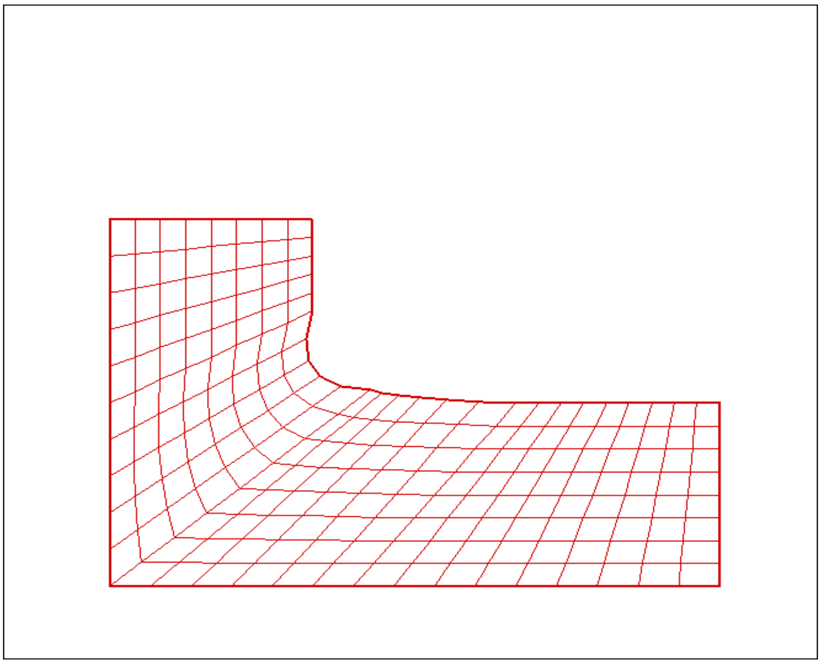

Figure 42: Final finite element mesh of the plate

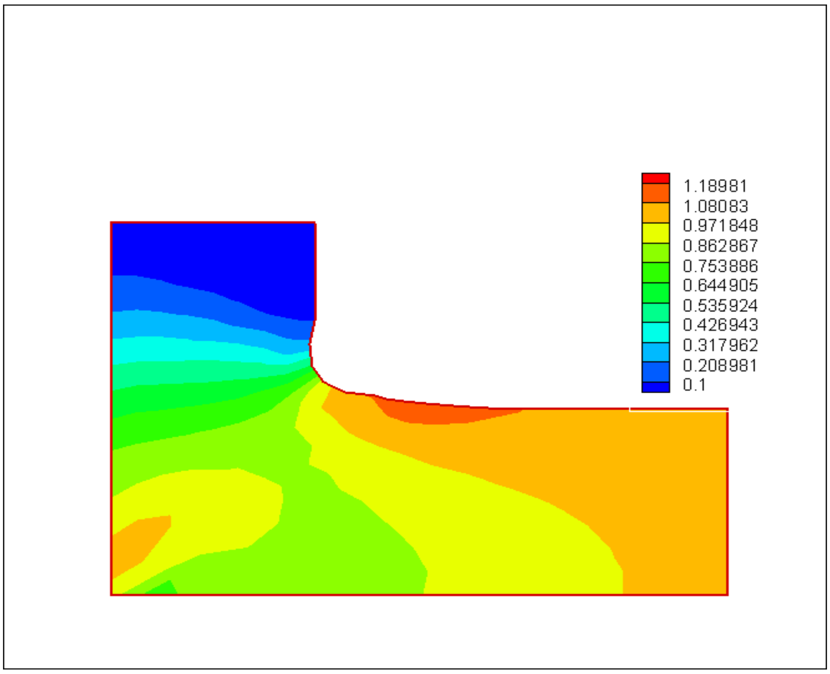

Figure 43: Contour plot of the Von-Mises stress at the final step after the fifth run.

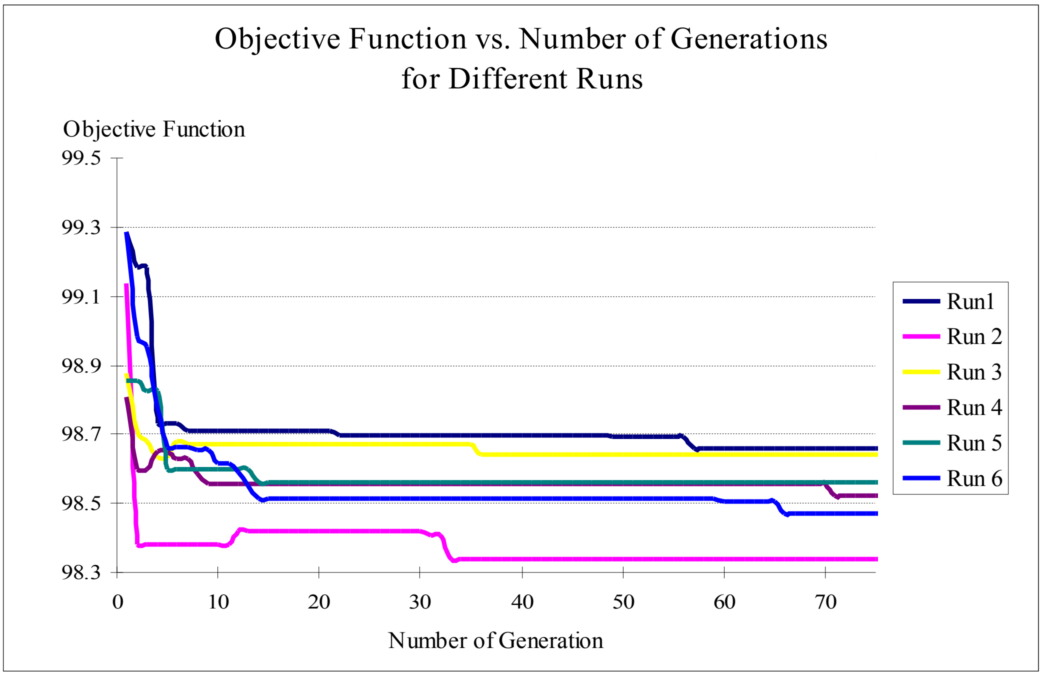

Figure 44: Objective function vs. number of generation for different runs

4.2.6 Discussion

In this sample, the shape of a fillet was optimized. The objective is to minimize the area of the plate under stress constraint conditions. Optimum solutions for the first run can be observed in the objective function history in Figure 44. In Figure 33 it can be seen that at generation 57 the objective function attained a final area of 98.65723m^2. The shape corresponding to the 57th generation was considered as a near optimal geometry which has a close shape with the reference paper [9] Figure 32.

The changes in the fillet shape are shown through Figure 33(a) – (d).

For this example to investigate the stability of the algorithm, 5 more runs have been performed using the same optimization parameters.

From Figure 44 it can be seen that the first run gave a maximum objective value and the second run gave a minimum objective value. The value difference between these two runs is about 0.25, which shows that the proposed algorithm is quite stable.

4.3 Cantilever Beam

The third problem to be considered is The Cantilever Beam problem.

An initial rectangle design of height $h = 6$, with a predefined length of $l = 30$ is shown in Figure 45 . The plate is assumed to have unit thickness. The objective is to minimize the vertical displacement U_F at the point of load application with $F = 10$, subject to an area constraint of 70% of the maximum area of 300 and an allowable height of $1.5< h<10$.

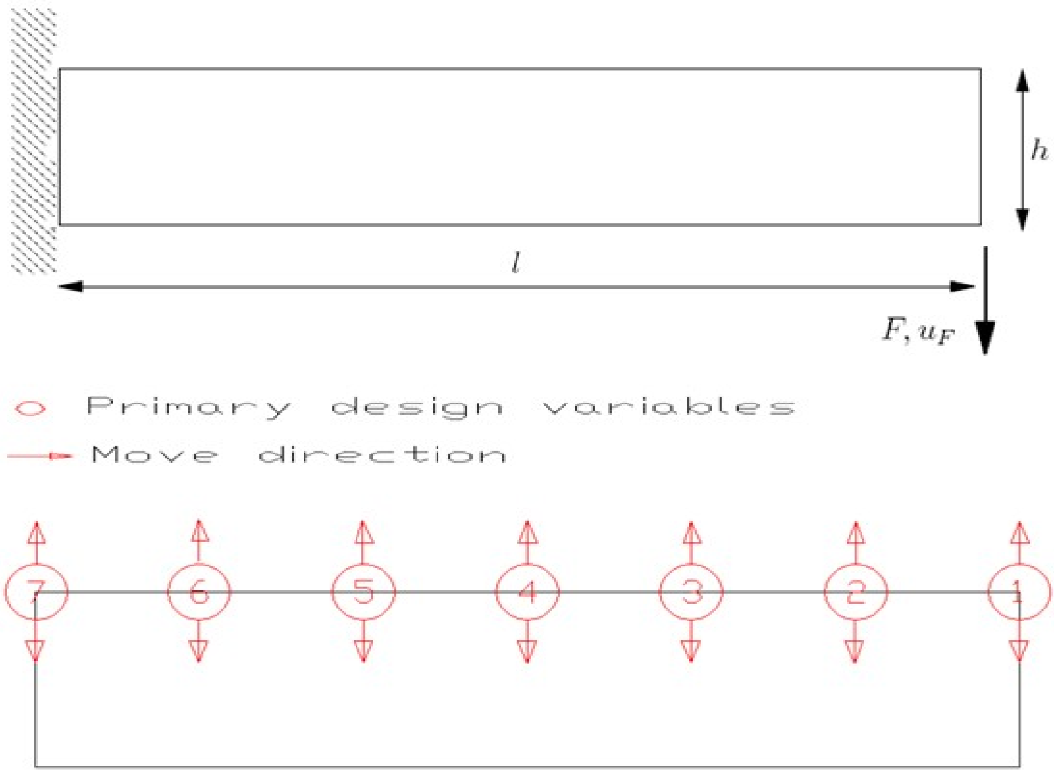

Figure 45: Plane stress model of a cantilever beam

Design parameters are:

Modulus of elasticity : $E = 210 \times 10^3$ N/mm^2

Poisson's ratio : $v = 0.3$

Allowable area : $A \leq 210$mm^2

Allowable height : $1.5 < h < 10$

Finite element meshing characteristics:

Number of element: 180 four-node quad elements

Number of nodes: 217 nodes

Total degree of freedom: 217 x 2 DOF/Node = 434

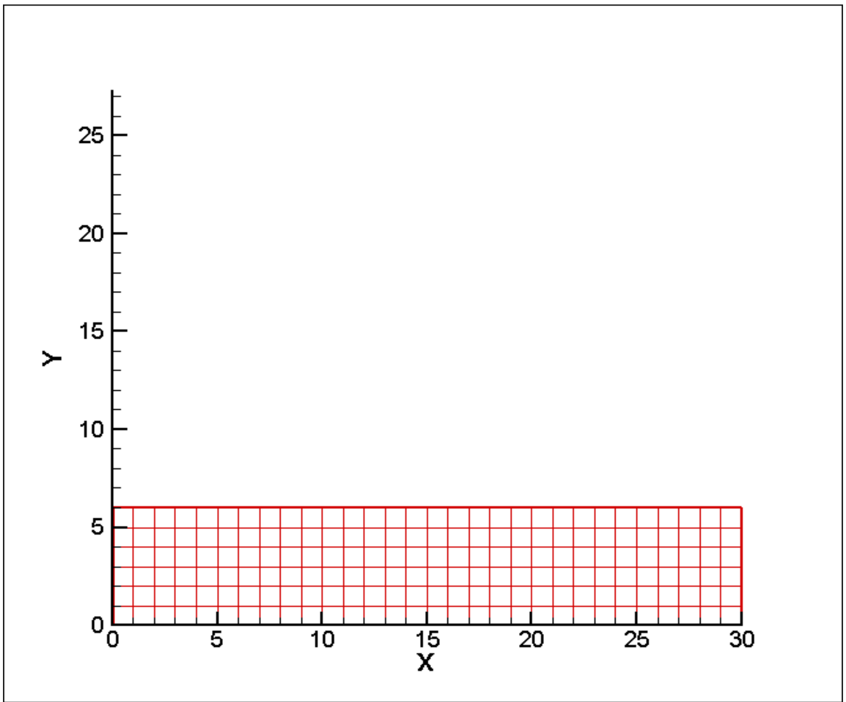

Figure 46: FEM mesh for initial design

Design variables: The coordinate of points 1, 2, 3, 4, 5, 6, 7

Objective: Vertical displacement U_F at the point of load application with $F = 10$

Optimization parameters

Number of design variables: 7

Step size of design variables: 0.05

Initial population: 100

Selection type: roulette wheel selection

Crossover probability: 0.80

Mutation probability: 0.06

Elitist selection: 2

The convergence is obtained at generation 54 and the vertical displacement at the point of load is 0.000195310175. The final area is 209.94999.

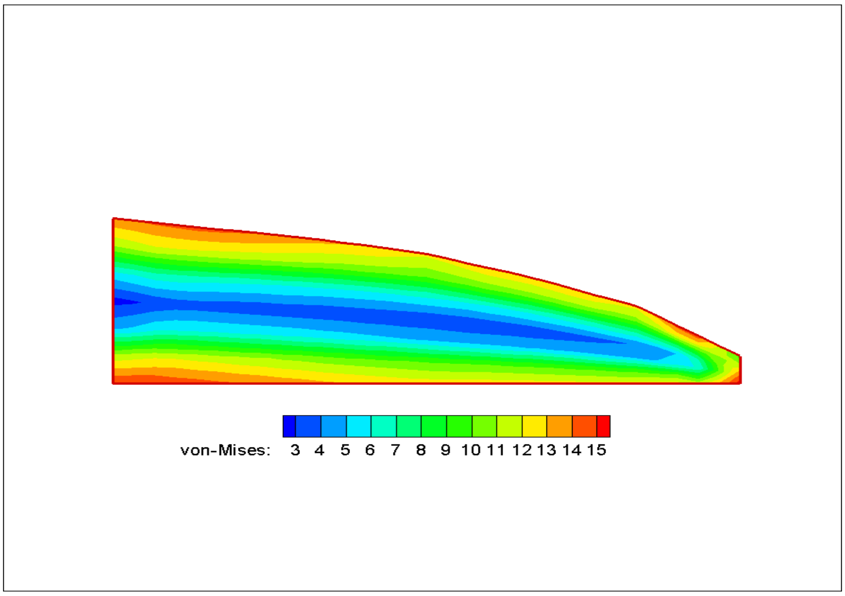

Figure 47: Contour plot of the Von-Mises stress at the final step

Result comparison between proposed solution and reference paper

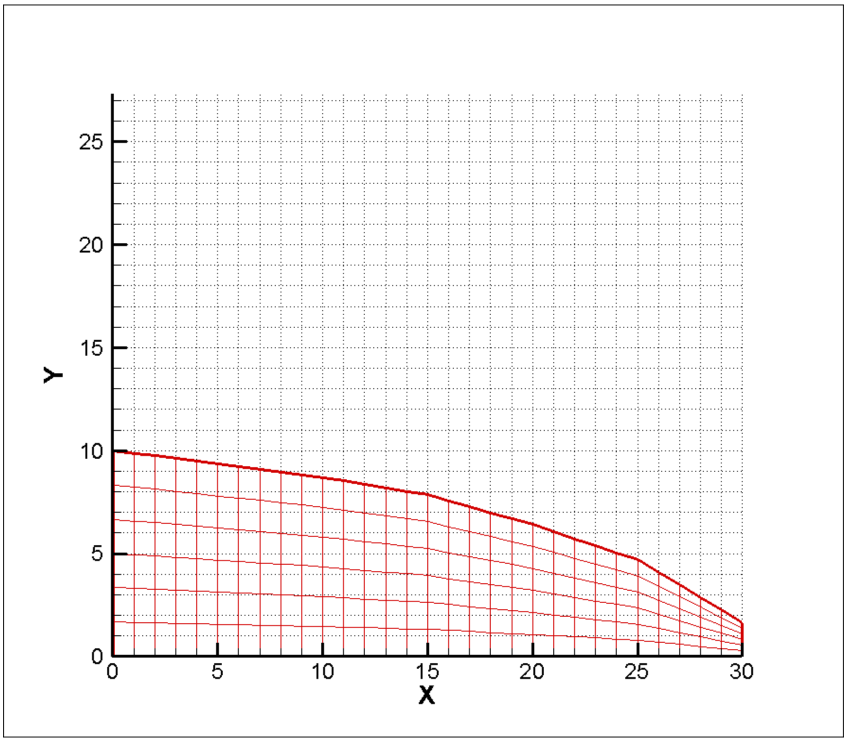

Figure 48: Final finite element mesh of the beam

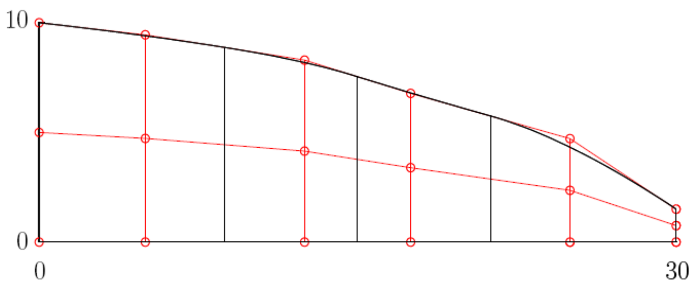

Figure 49: Final shape of the plate from Ref [23]

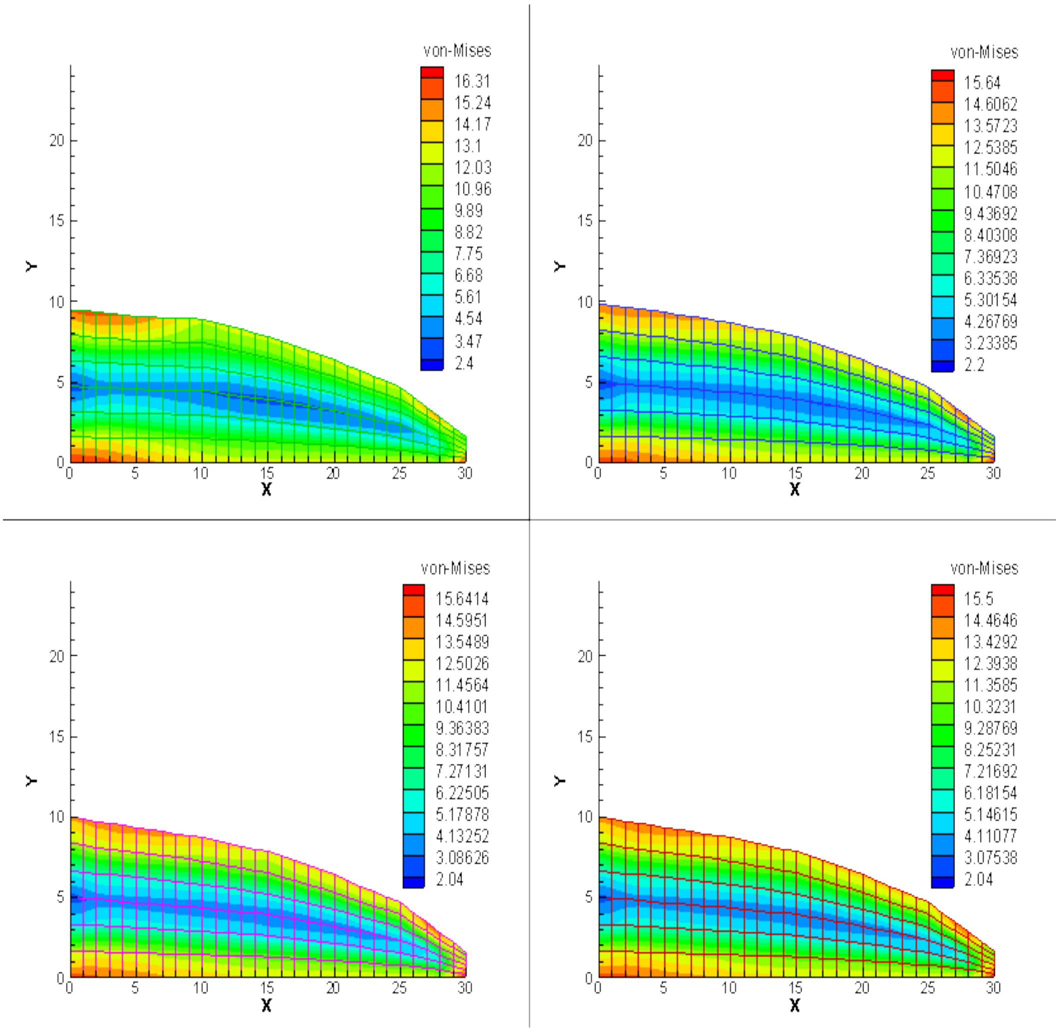

Figure 50: Generation process during shape optimization after, a) 10, b) 20, c) 30, and d) 54 generations

To investigate the effectiveness of crossover probability of the algorithm, 5 more solutions have been performed using different crossover probabilities.

4.3.1 Run One with crossover probability Equal to 1

Optimization parameters

Number of design variables: 7

Step size of design variables: 0.05

Initial population: 100

Selection type: roulette wheel selection

Crossover probability: 1.00

Mutation probability: 0.06

Elitist selection: 2

Result

The convergence is obtained at generation 30 and the vertical displacement at the point of load is 0.00019646791. The final area is 209.99995

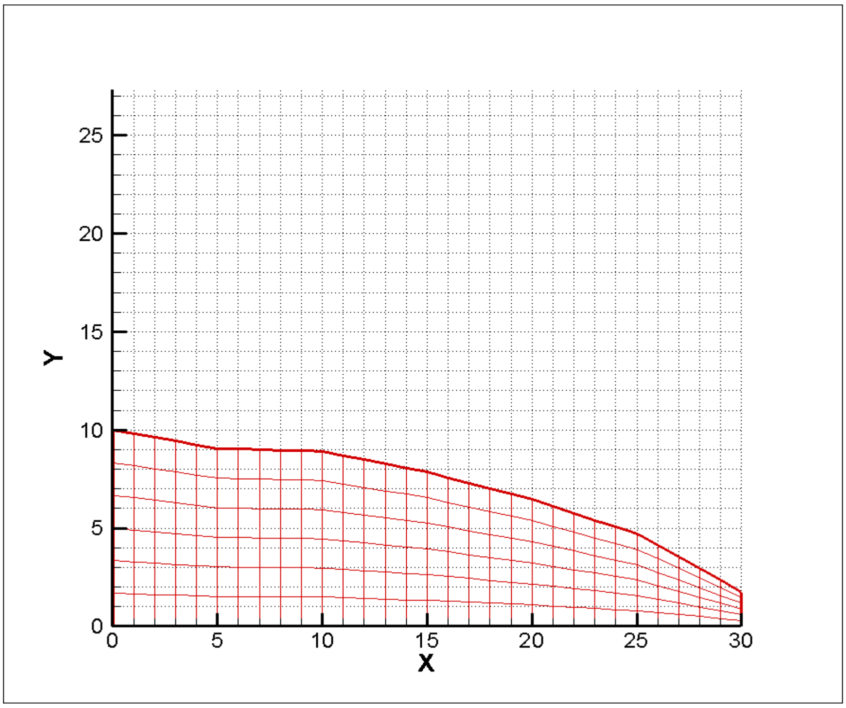

Figure 51: Final finite element mesh of the beam

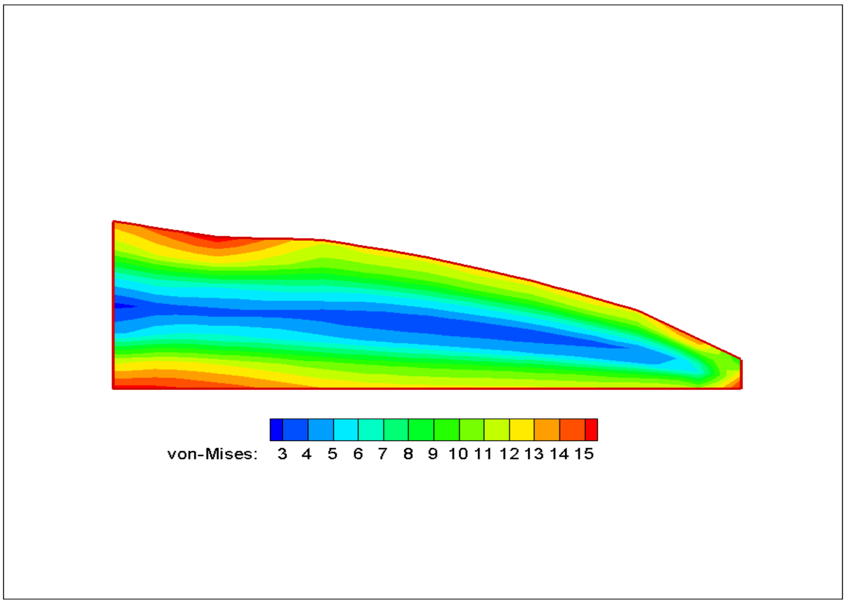

Figure 52: Contour plot of the Von-Mises stress at the final step crossover probability = 1

4.3.2 Run Two with crossover probability Equal to 0.90

Optimization parameters

Number of design variables: 7

Step size of design variables: 0.05

Initial population: 100

Selection type: roulette wheel selection

Crossover probability: 0.90

Mutation probability: 0.06

Elitist selection: 2

Result

The convergence is obtained at generation 46 and the vertical displacement at the point of load is 0.000196376626. The final area is 210.

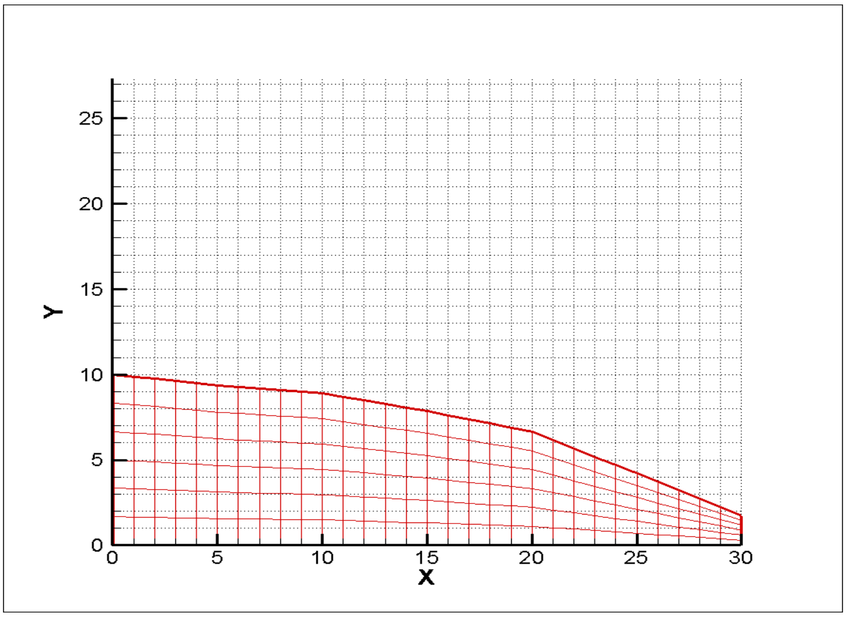

Figure 53: Final finite element mesh of the beam

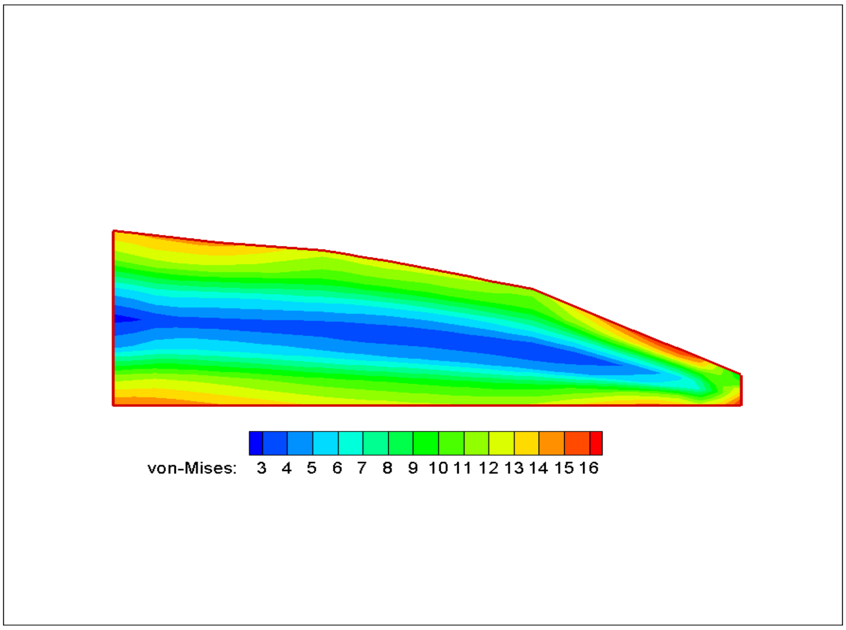

Figure 54: Contour plot of the Von-Mises stress at the final step crossover probability = 0.90

4.3.3 Run Three with crossover probability Equal to 0.7

<u>*Optimization parameters*</u>

Number of design variables: 7

Step size of design variables: 0.05

Initial population: 100

Selection type: roulette wheel selection

Crossover probability: 0.70

Mutation probability: 0.06

Elitist selection: 2

<u>Result</u>

The convergence is obtained at generation 43 and the vertical displacement at the point of load is 0.000196896377. The final area is 209.95001.

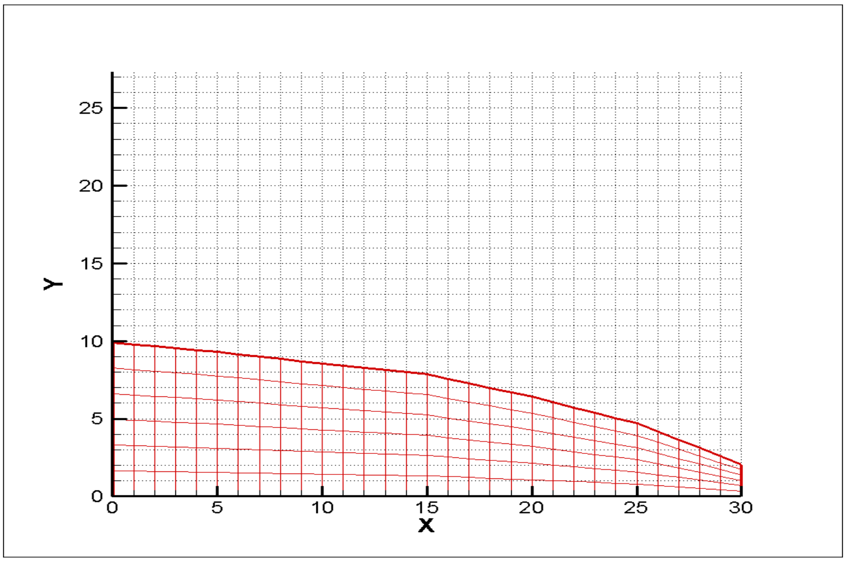

Figure 55: Final finite element mesh of the beam

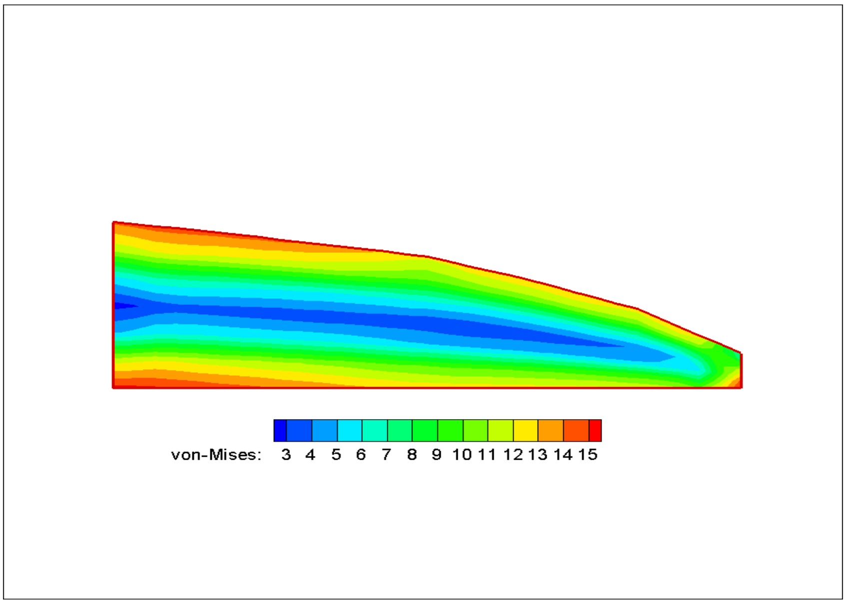

Figure 56: Contour plot of the Von-Mises stress at the final step with crossover probability = 0.70

4.3.4 Run Four with crossover probability Equal to 0.60

Optimization parameters

Number of design variables: 7

Step size of design variables: 0.05

Initial population: 100

Selection type: roulette wheel selection

Crossover probability: 0.60

Mutation probability: 0.06

Elitist selection: 2

Result

The convergence is obtained at generation 37 and the vertical displacement at the point of load is 0.00019608961. The final area is 209.89987.

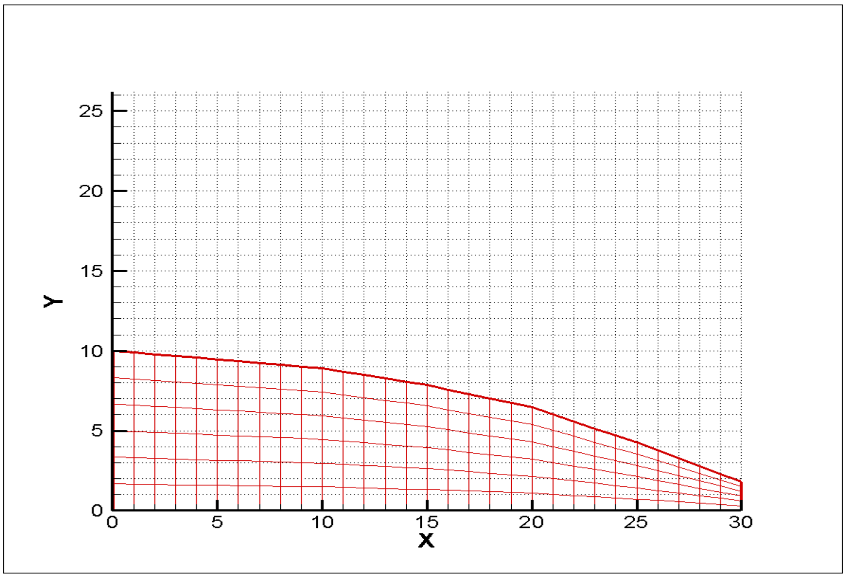

Figure 57: Final finite element mesh of the beam

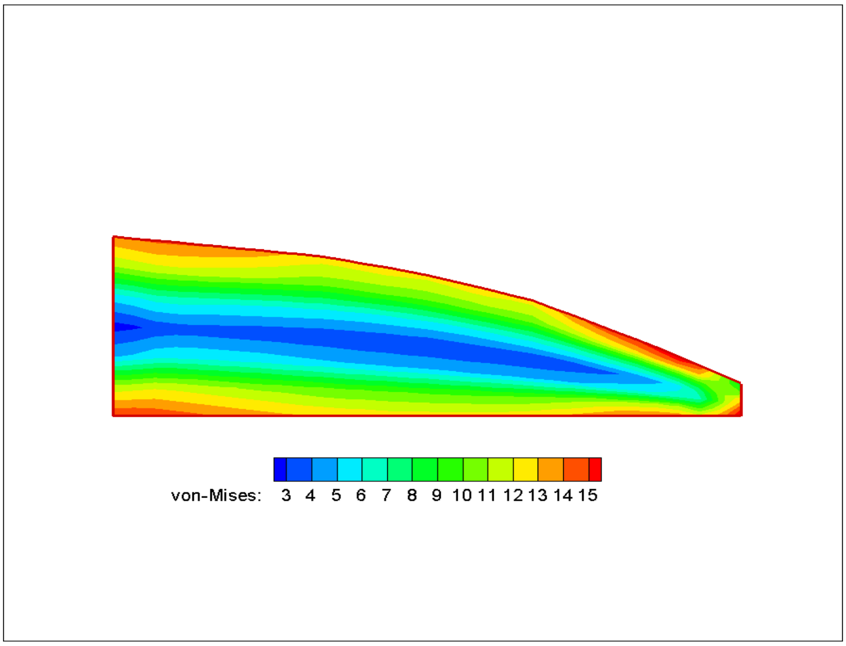

Figure 58: Contour plot of the Von-Mises stress at the final step with crossover probability = 0.60

4.3.5 Run Five with crossover probability Equal to 0.50

<u>*Optimization parameters*</u>

Number of design variables: 7

Step size of design variables: 0.05

Initial population: 100

Selection type: roulette wheel selection

Crossover probability: 0.50

Mutation probability: 0.06

Elitist selection: 2

<u>Result</u>

The convergence is obtained at generation 23 and the vertical displacement at the point of load is 0.00019738335. The final area is 209.99993.

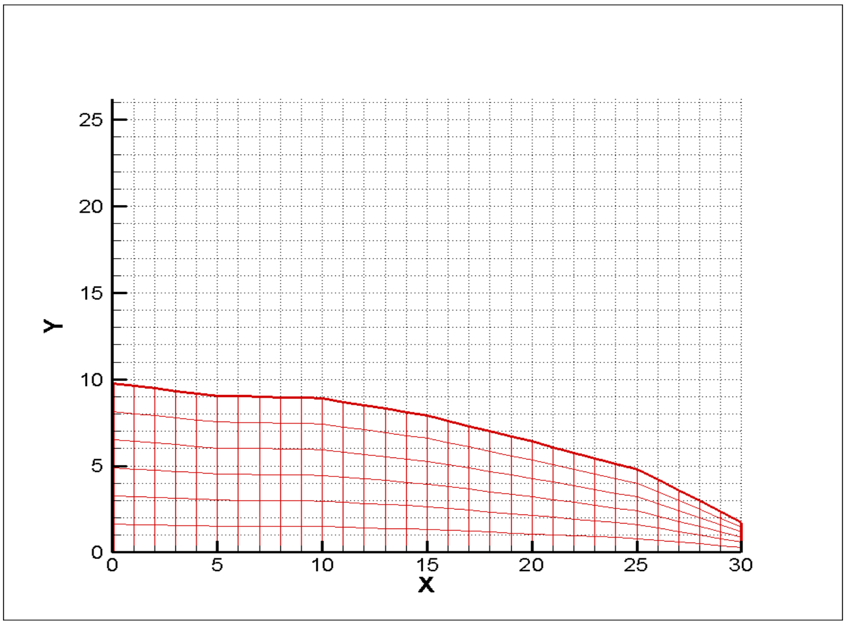

Figure 59: Final finite element mesh of the beam

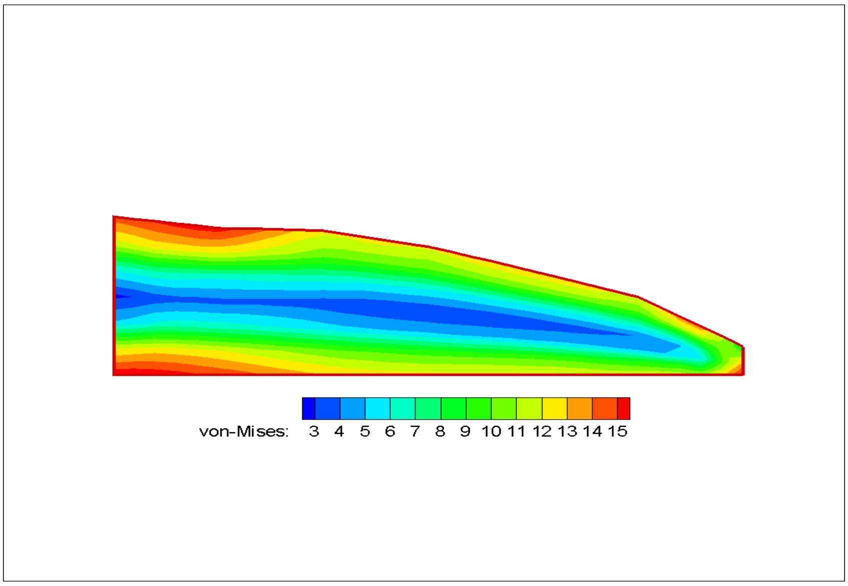

Figure 60: Contour plot of the Von-Mises stress at the final step with crossover probability = 0.50

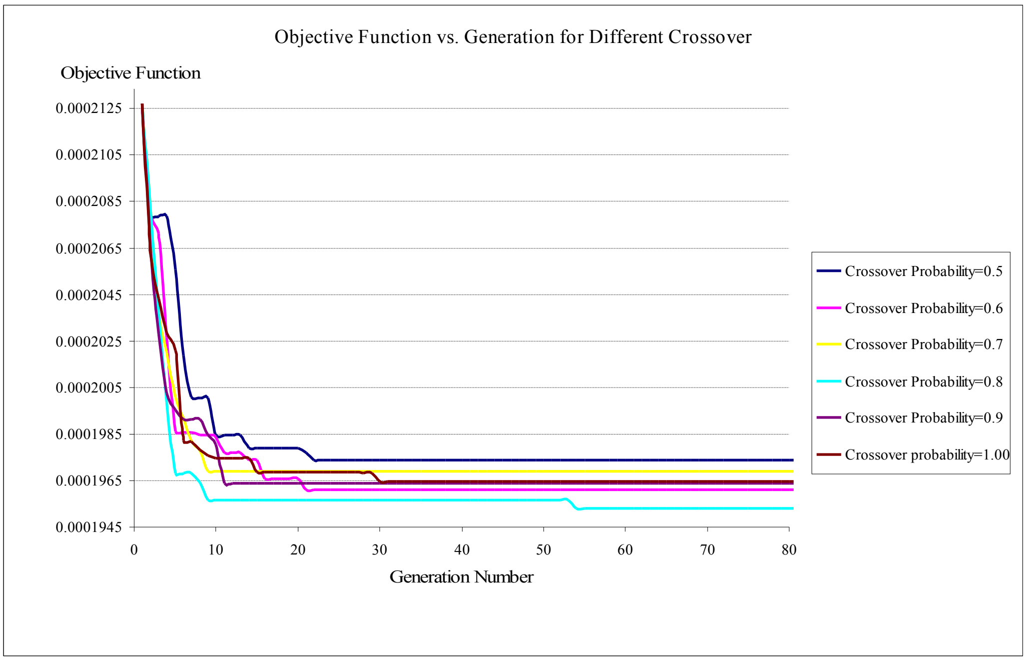

Figure 61: Objective function vs. number of generation for different crossover probabilities

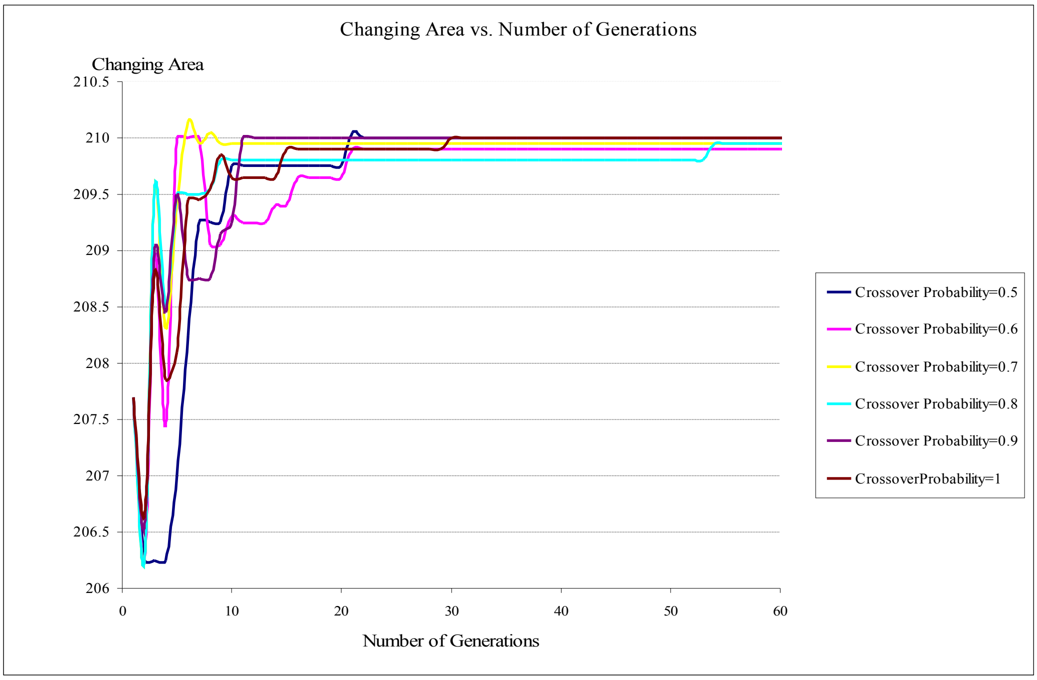

Figure 62: Changing area vs. number of generation for different crossover probabilities

4.3.6 Discussion

A simple and classical example of shape optimization is the cantilever beam. The objective here was to minimize the vertical displacement U_F at the point of load application. The optimization variables are the vertical positions of the upper row of control points, bound between 1.5 and 10. The generation process as well as the iteration history are depicted in Figure 50 and Figure 61 respectively. Convergence is gained at 54th generation and the optimum design has given similar shape (Figure 48) with Ref[23].

For this example to investigate the effectiveness of crossover probability of the algorithm, 5 more runs have been performed using different crossover probabilities. Best fitness result is obtained when the crossover probability is equal to 0.8, with the crossover probability ranging from 0.5 to 1.00.

4.4 The Bracket Problem

The fourth problem to be considered is The Bracket problem.

This problem was motivated by an example used in Ref [24]

The initial bracket and model is depicted in Figure 63.The bracket has a height of

h =508 mm, length of l = 2540 mm and radius of r = 127 mm. The Bracket is assumed to have unit thickness. A stress field is applied by specifying P=0.00689476 MPa in the y direction. The optimization problem is to minimize the area of the plate according to stress constraints.

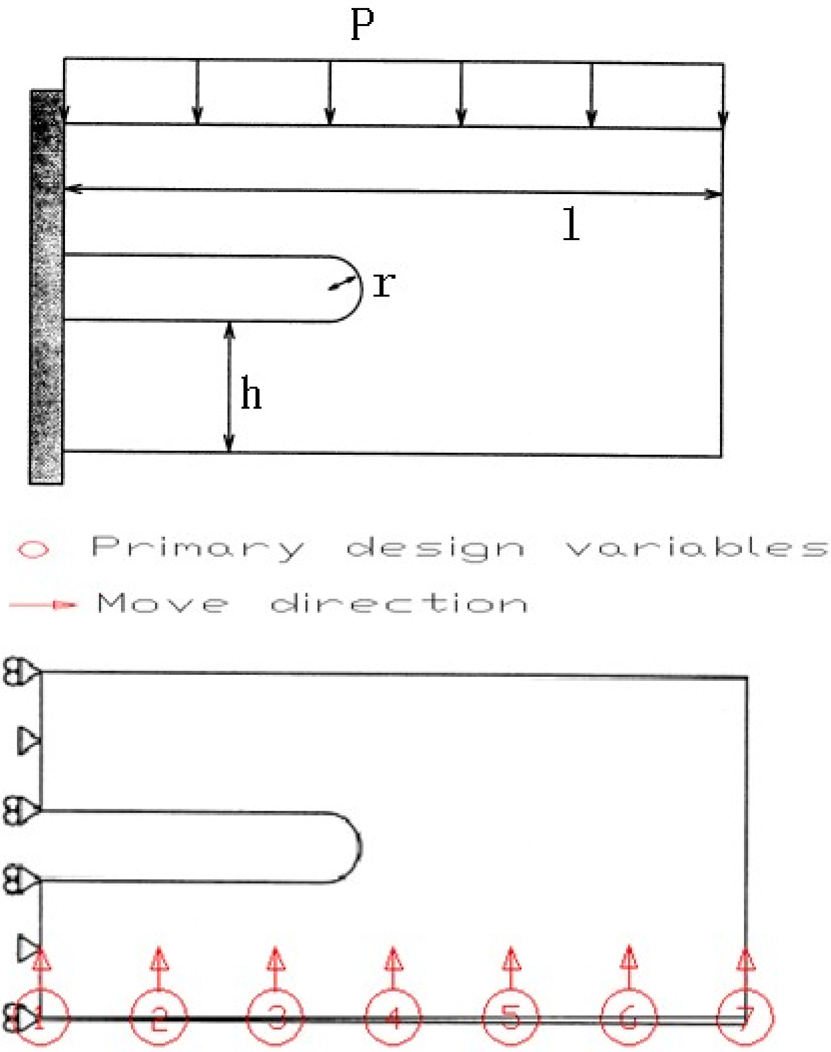

Figure 63: Plane stress model of a bracket

Seven design variables are used along the bottom boundary in this example. Geometric constraints on the design points are imposed such that their maximum allowed y-coordinates were at least 635 mm inch below the top boundary. This type of constraint is necessary since the stresses in the tip region are low.

Design parameters are:

Modulus of elasticity : E = 6894.757 kPa

Poisson's ratio : v = 0.3

Allowable Von-Mises stress = 0.137895146 MPa

Finite element meshing characteristics:

Number of element: 312 four-node quad elements

Number of nodes: 365 nodes

Total degree of freedom: 365 x 2 DOF/Node = 730

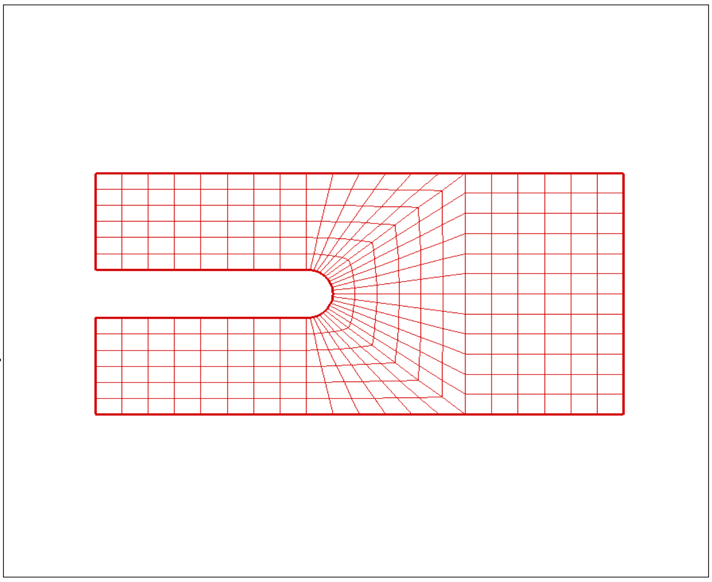

Figure 64: FEM mesh for initial design

Optimization parameters:

Number of design variables: 7

Step size of design variables: 0.2

Initial population: 140

Selection type: roulette wheel selection

Crossover probability: 1.00

Mutation probability: 0.08

Elitist selection: 2

The convergence is obtained at generation 81 and the final area is 2158705 mm^2.

Since the value of optimized area is not given in the paper, it was not possible to compare the final results with the final area, but the final shape of the plate is given and this is compared with the shape obtained using the Genetic Algorithm Code.

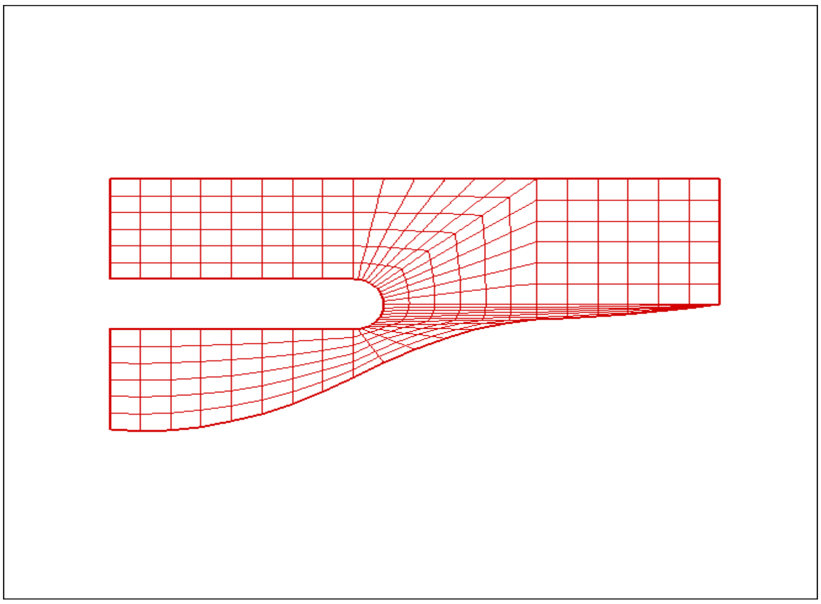

Figure 65: Final finite element mesh of the bracket

Result comparison between proposed solution and the reference paper

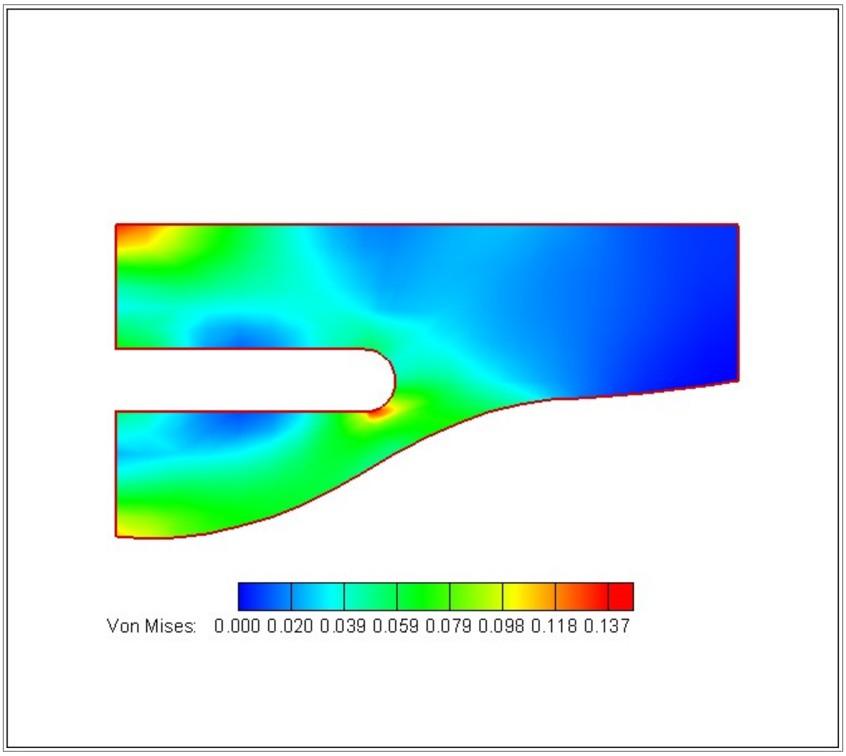

Figure 66: The contour plot of the Von-Mises stress at the final step

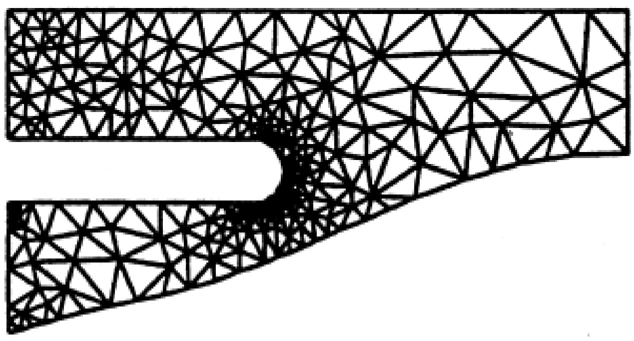

Figure 67: Final shape of the plate from Ref [24]

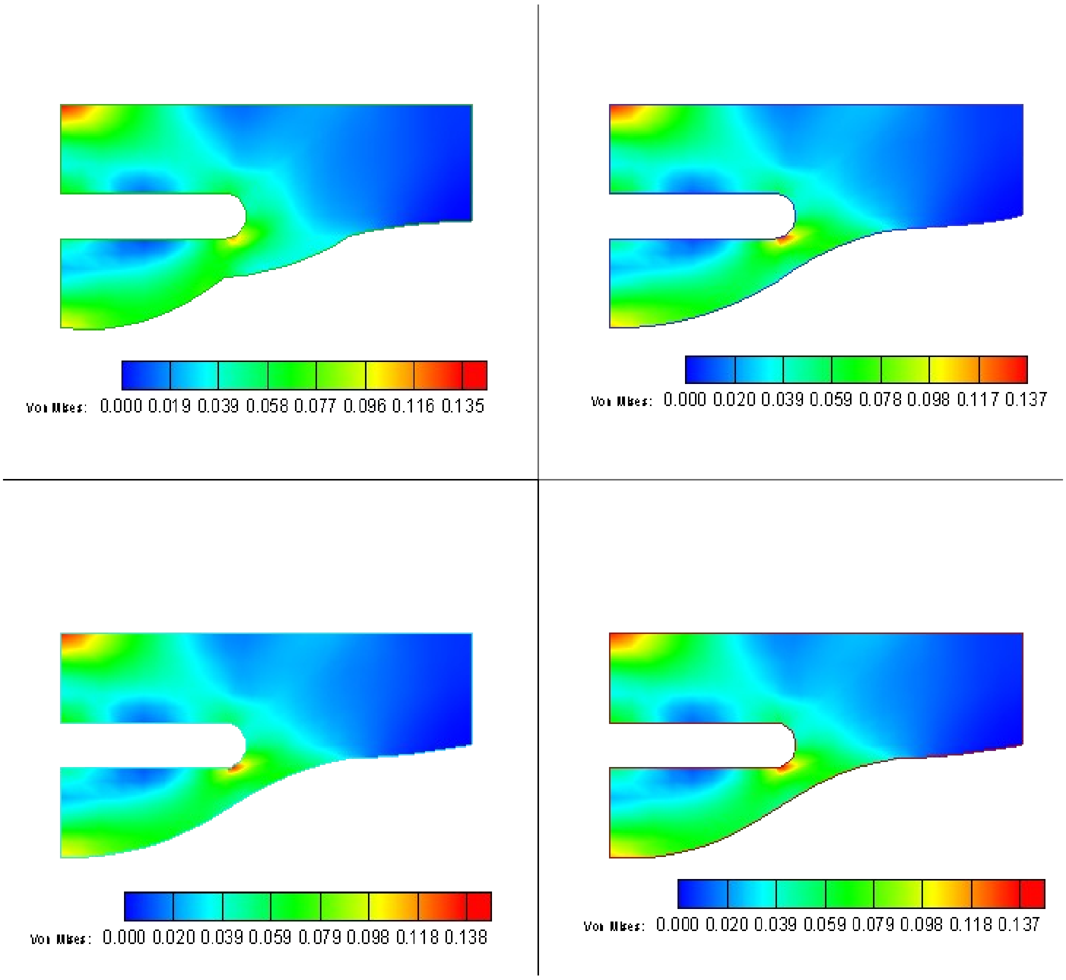

Figure 68: Generation process during shape optimization after; a) 10, b) 40, c) 60 and d) 81 generations

To investigate the effectiveness of population of the algorithm, 7 more solutions have been performed using different number of initial populations.

4.4.1 Run One with number of population Equal to 50

Optimization parameters

Number of design variables: 7

Step size of design variables: 0.2

Initial population: 50

Selection type: roulette wheel selection

Crossover probability: 1.00

Mutation probability: 0.08

Elitist selection: 2

Result

The convergence is obtained at generation 72 and the final area is 2157415 mm^2.

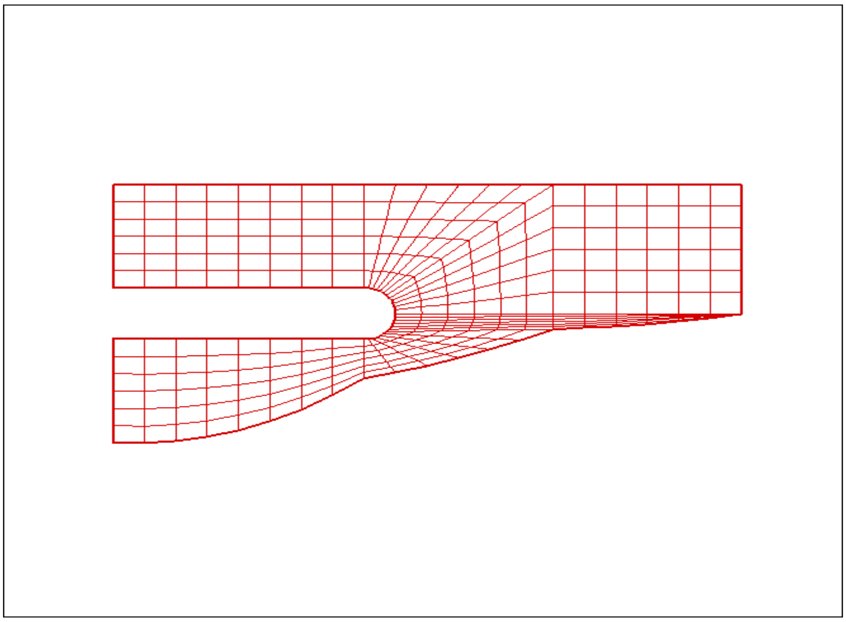

Figure 69: Final finite element mesh of the bracket

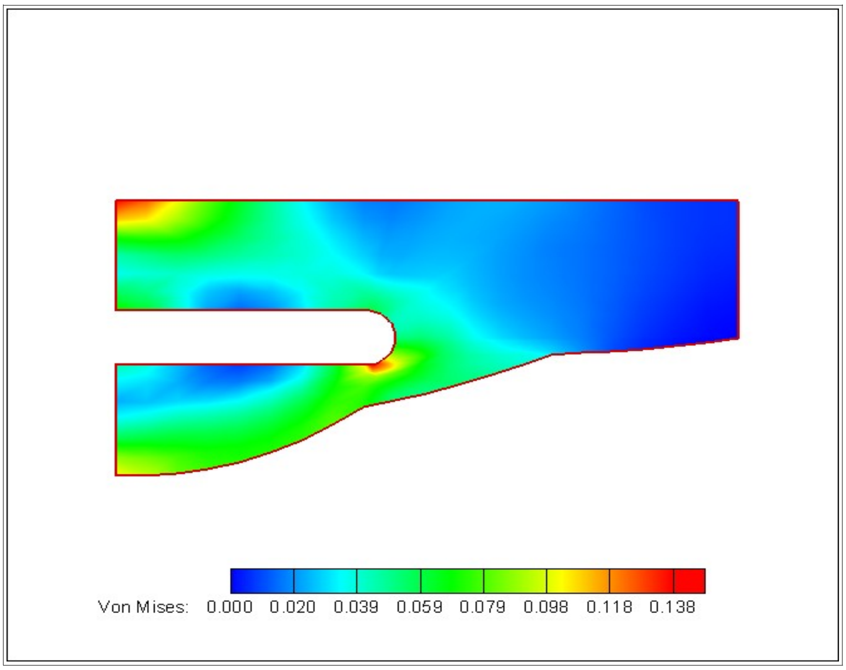

Figure 70: Contour plot of the Von-Mises stress at the final step with number of population = 50

4.4.2 Run Two with number of population Equal to 80

Optimization parameters

Number of design variables: 7

Step size of design variables: 0.2

Initial population: 80

Selection type: roulette wheel selection

Crossover probability: 1.00

Mutation probability: 0.08

Elitist selection: 2

Result

The convergence is obtained at generation 37 and the final area is 2154834 mm^2.

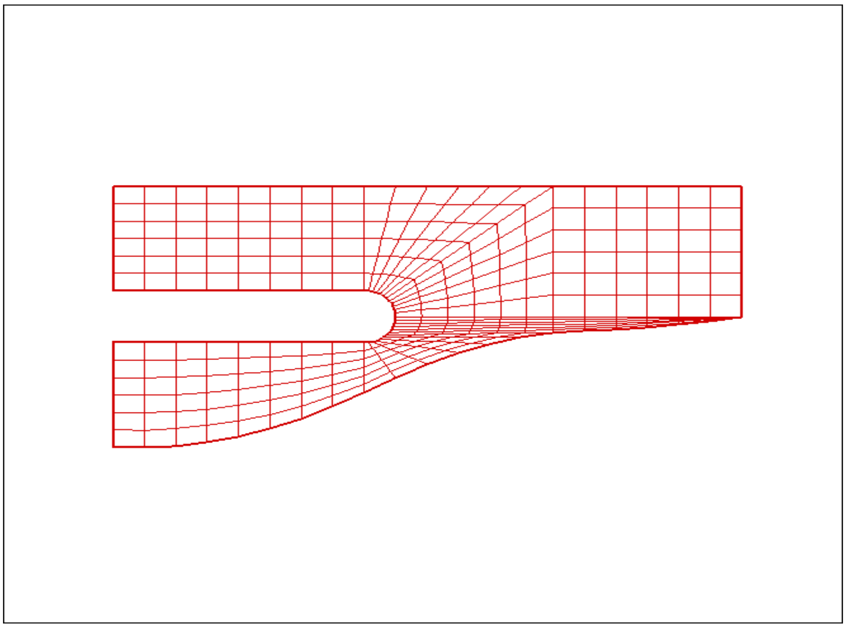

Figure 71: Final finite element mesh of the bracket

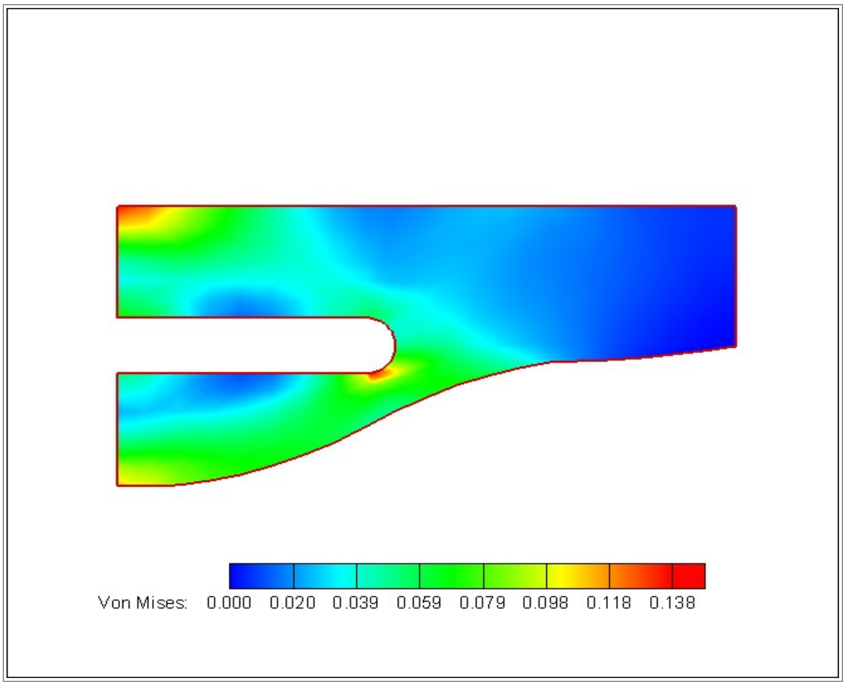

Figure 72: Contour plot of the Von-Mises stress at the final step with number of population = 80

4.4.3 Run Three with number of population Equal to 100

Optimization parameters

Number of design variables: 7

Step size of design variables: 0.2

Initial population:100

Selection type: roulette wheel selection

Crossover probability: 1.00

Mutation probability: 0.08

Elitist selection: 2

Result

The convergence is obtained at generation 55 and the final area is 2148383 mm^2.

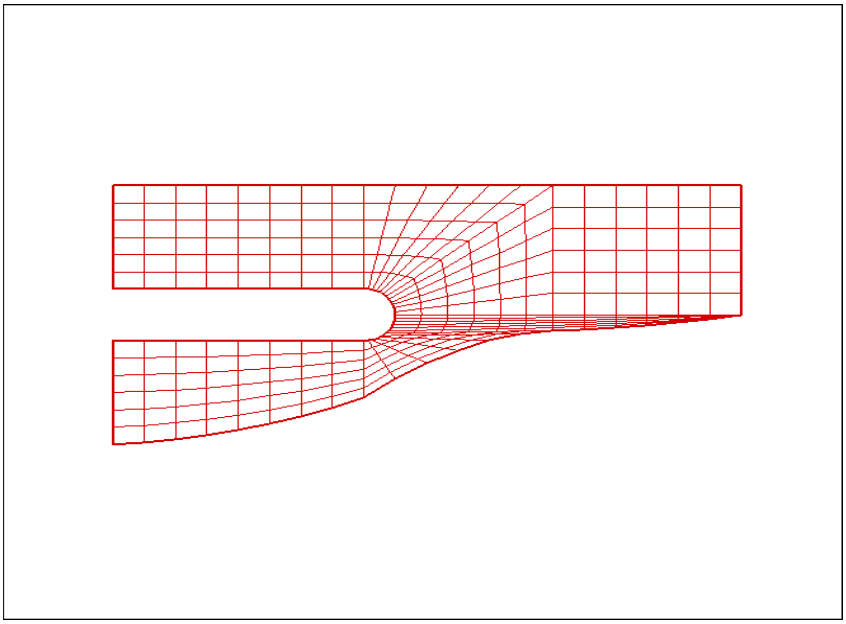

Figure 73: Final finite element mesh of the bracket

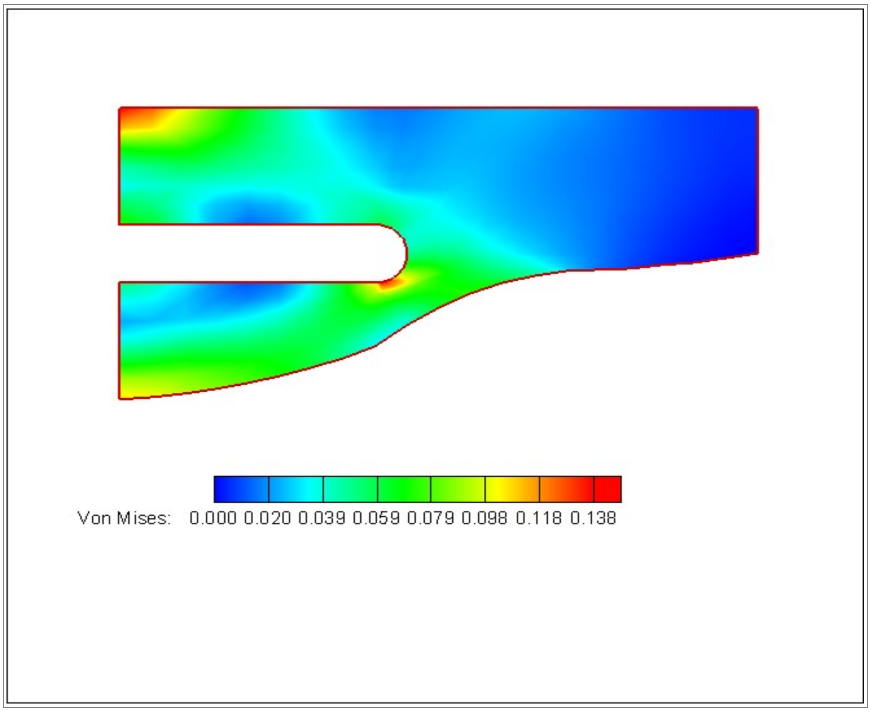

Figure 74: Contour plot of the Von-Mises stress at the final step with number of population = 100

4.4.4 Run Four with number of population Equal to 180

Optimization parameters

Number of design variables: 7

Step size of design variables: 0.2

Initial population:180

Selection type: roulette wheel selection

Crossover probability: 1.00

Mutation probability: 0.08

Elitist selection: 2

Result

The convergence is obtained at generation 15 and the final area is 2149673mm^2.

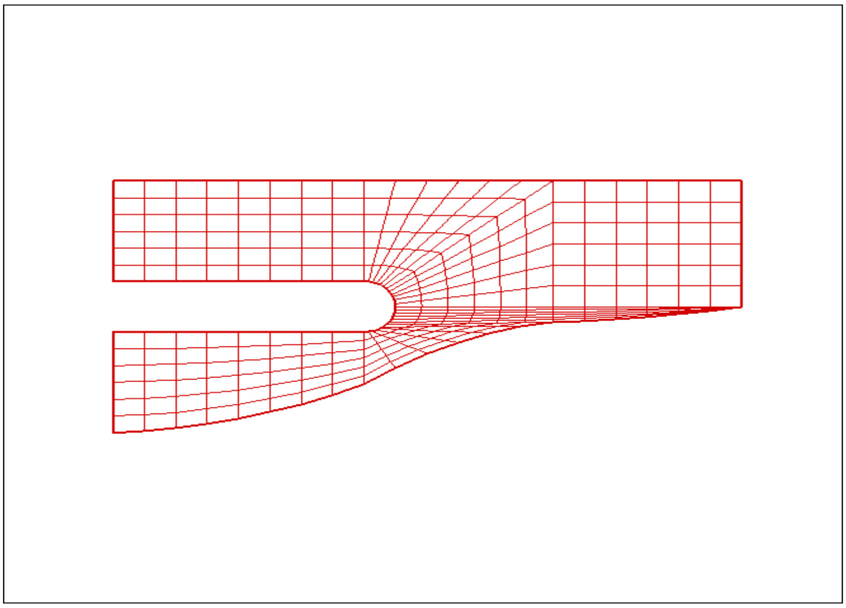

Figure 75: Final finite element mesh of the bracket

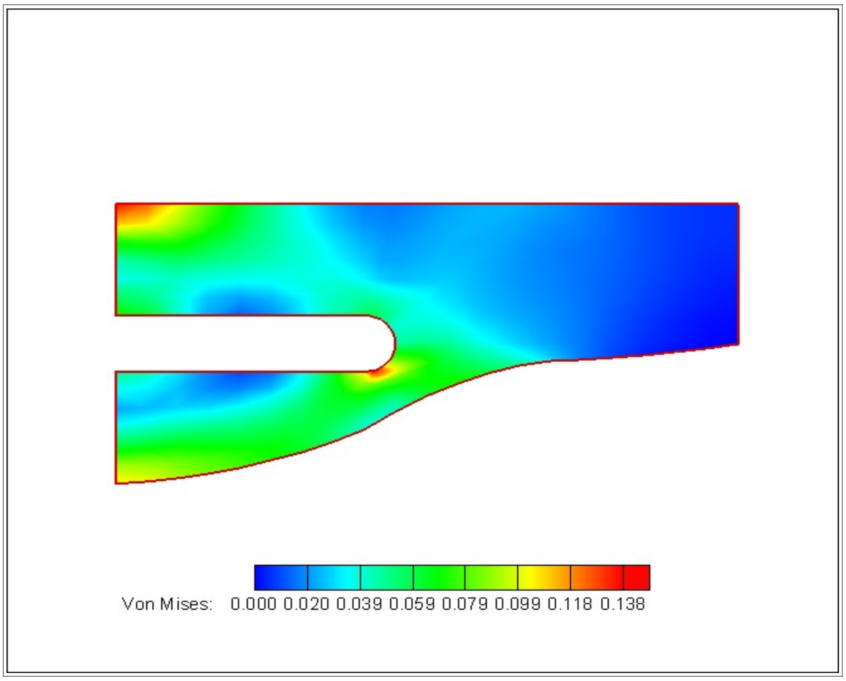

Figure 76: Contour plot of the Von-Mises stress at the final step with number of population = 180

4.4.5 Run Five with number of population Equal to 220

Optimization parameters

Number of design variables: 7

Step size of design variables: 0.2

Initial population:220

Selection type: roulette wheel selection

Crossover probability: 1.00

Mutation probability: 0.08

Elitist selection: 2

Result

The convergence is obtained at generation 11 and the final area is 2159351 mm^2.

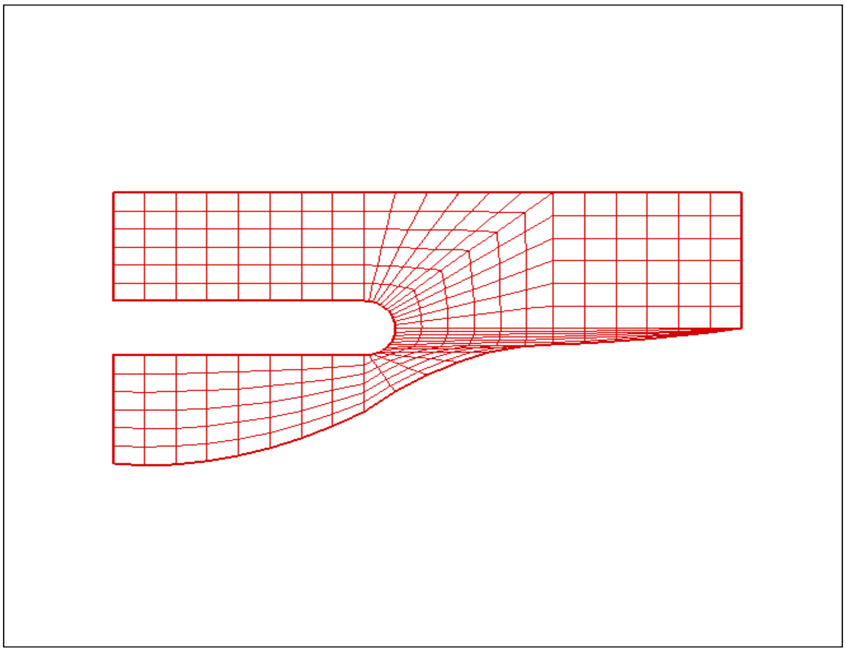

Figure 77: Final finite element mesh of the bracket

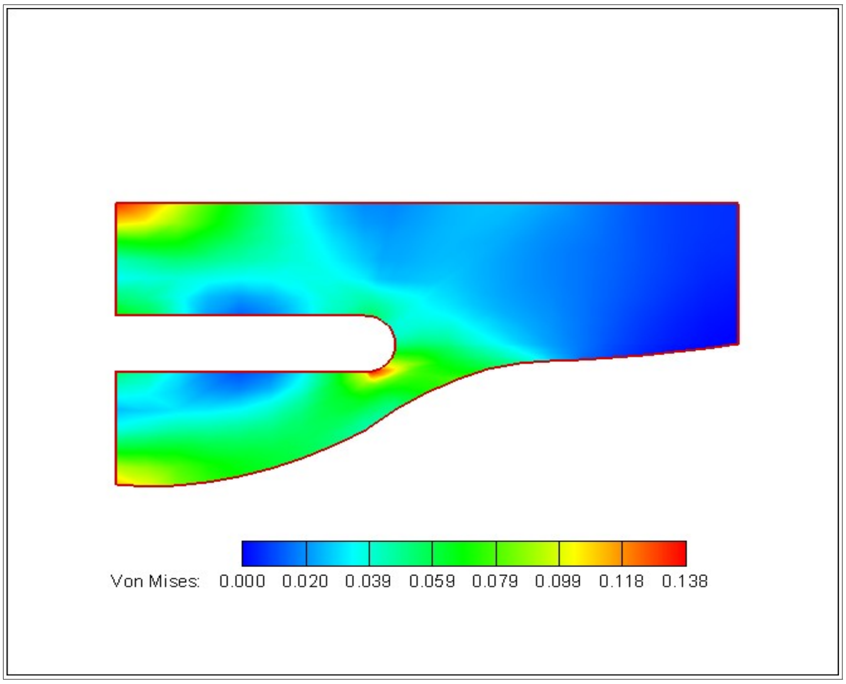

Figure 78: Contour plot of the Von-Mises stress at the final step with number of population = 220

4.4.6 Run Six with number of population Equal to 260

Optimization parameters

Number of design variables: 7

Step size of design variables: 0.2

Initial population:260

Selection type: roulette wheel selection

Crossover probability: 1.00

Mutation probability: 0.08

Elitist selection: 2

Result

The convergence is obtained at generation 15 and the final area is 2154189 mm^2.

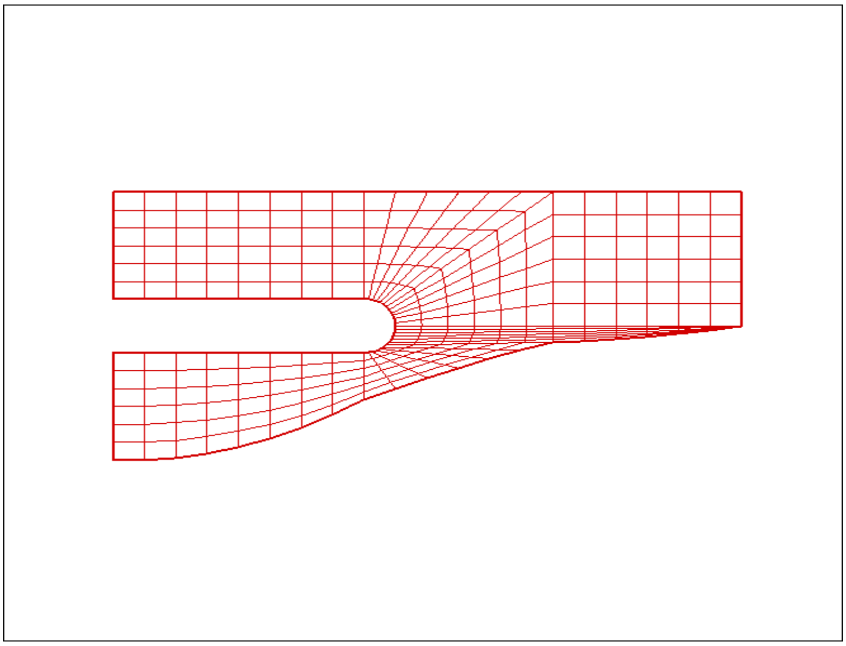

Figure 79: Final finite element mesh of the bracket

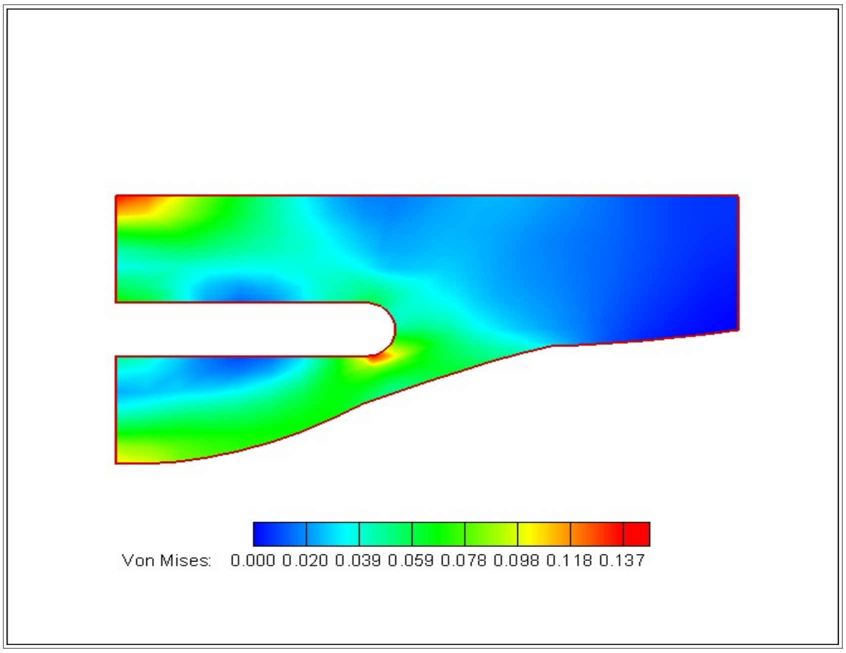

Figure 80: Contour plot of the Von-Mises stress at the final step with number of population = 260

4.4.7 Run Seven with number of population Equal to 300

Optimization parameters

Number of design variables: 7

Step size of design variables: 0.2

Initial population:300

Selection type: roulette wheel selection

Crossover probability: 1.00

Mutation probability: 0.08

Elitist selection: 2

Result

The convergence is obtained at generation 30 and the final area is 2148383 mm^2.

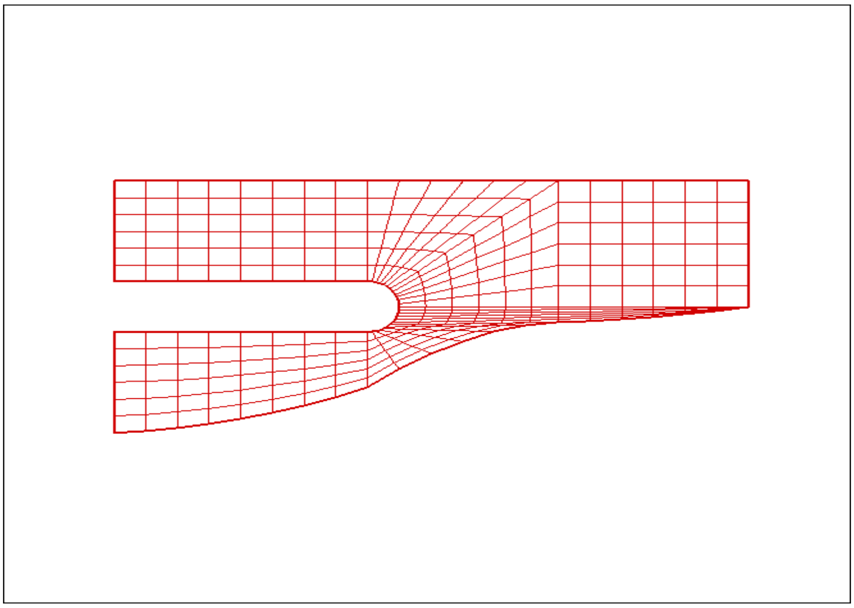

Figure 81: Final finite element mesh of the bracket

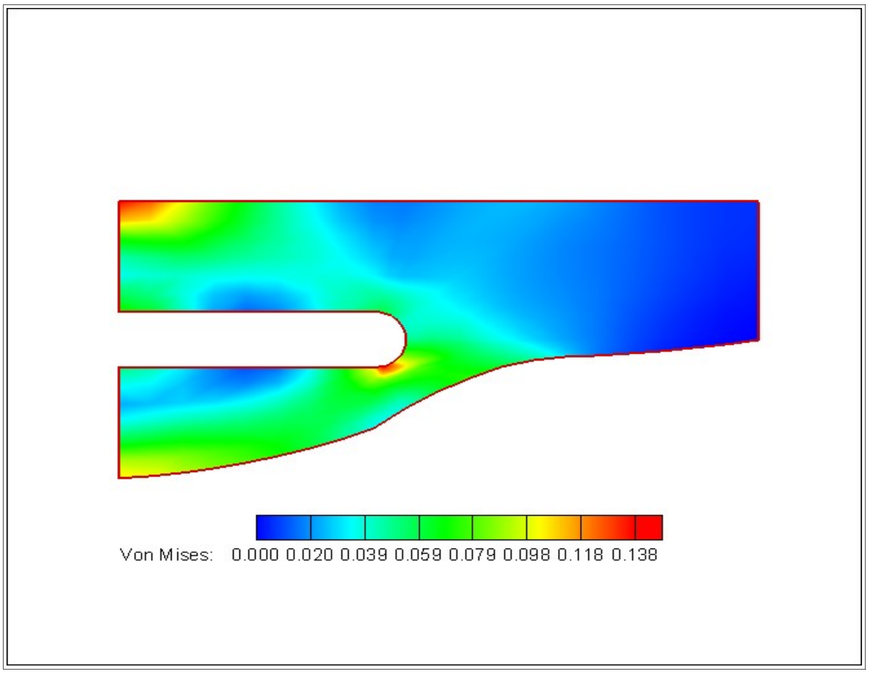

Figure 82: Contour plot of the Von-Mises stress at the final step with number of population = 300

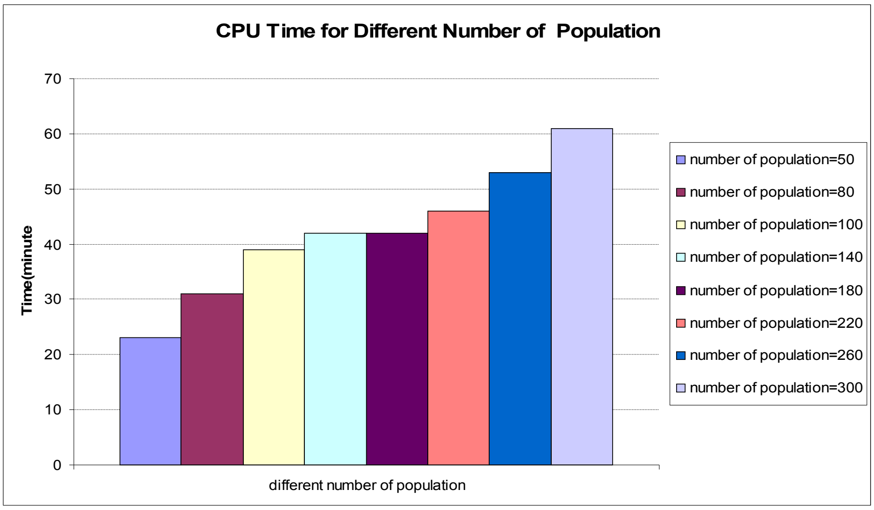

Figure 83:Comparing the CPU time using a computer with INTEL PENTIUM 1.73. GHz CPU for the bracket example with different population.

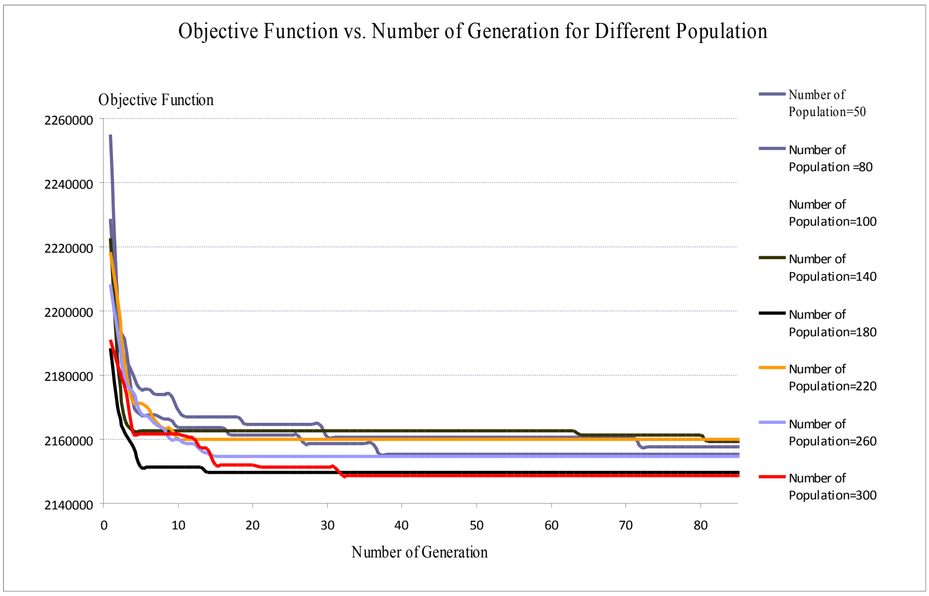

Figure 84: Objective function vs. number of generation with different number of population

4.4.8 Discussion

The objective of the bracket example is to minimize the area under Von-Mises stress constraint. Seven design points are used along the bottom boundary in this example.
Geometric constraints on the design points are imposed such that their maximum allowed y-coordinates were at least 635 mm below the top boundary. The numerical results are shown in Figure 68 for four different generations, representing the shapes at generation steps ; 10, 40, 60 and 81 respectively.
To investigate the effectiveness of different initial population of the algorithm, 7 more runs have been performed.
As expected the larger the population number, the faster the convergence is reached and the better the fitness result is (Figure 84).

CHAPTER 5

CONCLUSION

A shape optimization of two-dimensional problems is presented in this work. The proposed procedure is based on the Genetic algorithm and was implemented using the Finite Element Method (FEM).

For the first sample it may concluded that when roulette-wheel selection and elitism are used in the optimization process, the algorithm may converge to a local optimum very quickly, and it may get stuck at that point. So in order to jump from local optimum, the mutation probability was increased.

The second sample showed the Genetic algorithm is a stochastic search algorithm. This means that there is randomness in the optimization process. So each run of the optimization process is unique and also giving a stable result.

Crossover plays a very important role in genetic algorithm. It is regarded as a critical accelerator of the search process when the Genetic algorithm is running. The third example shows that a high crossover probability does not neccessarily give the best fitness result and a low crossover probability might not converge as fast as expected. However "crossover probabilities" continue to be subject of research.

It is also observed that, the initial population is crucial in the optimization process.

Since, the individuals in the population lead the process , increasing the size of the population creates high probability of convergence but it also causes an increase in running time Figure 83.

The proposed method employs the finite element method (FEM) with fixed mesh topology for estimating the fitness function values. However, the FEM meshes are distorted by the successive shape modification causing a reduce in computational accuracy. To overcome this difficulty, automatic adaptive meshing during the optimization process is recommended.

Improving the current code using multiple point cut crossover probability and instead of roulette wheel selection, tournament selection method might be a future work.

APPENDIX I

I. The Flow Chart of the Code

A genetic algorithm code written in Fortran language is developed.

All the examples demonstrated had been solved by using the developed code .

Figure 85 shows the structure of the inputs and outputs of the code written in a simple form.

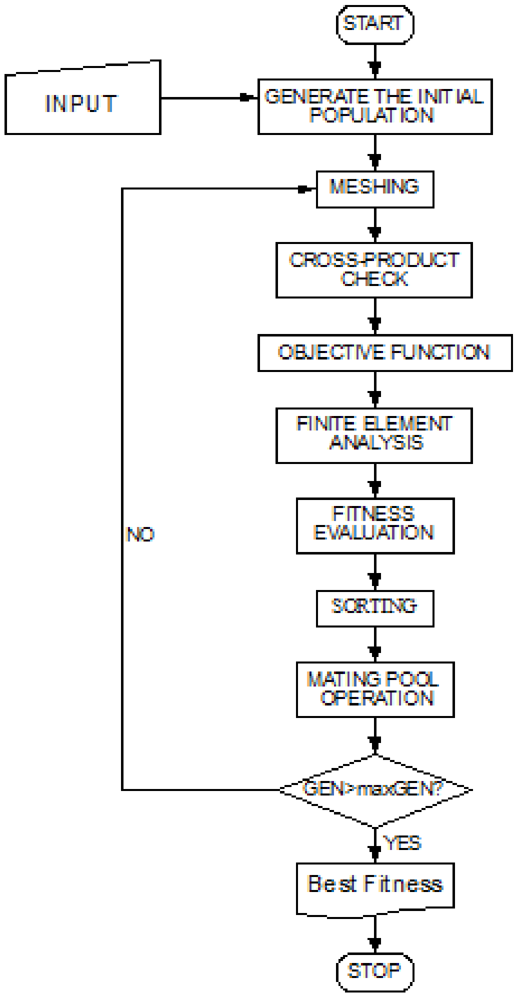

Figure 85: Flow chart of the developed code.

Input: This operator is used for input file reading. The input file contains detailed information about optimization design variables namely; “The size of the population”, “Number of design variables”, “Probability of cross over”, “Probability of mutation”, “Maximum number of generations”, “Step size for the search space”, “Continuous range for each design variable”.

Generate the Initial Population: This operator is used to form the population randomly.

Meshing: Since an external mesh generator is inefficient, a fixed topology mesh solver with four-node, iso-parametric elements is imposed. The data of design variables for each population are read in this procedure and information of connectivity and node coordinates are outputted.

Cross-Product Check: As mentioned in the previous chapter the optimum shape in these examples was found using the finite element method with the node coordinates as design variables. Thus, due to the random creation of population some irregular mesh shapes with four-node, iso-parametric elements may occur.

To avoid such problem, a cross product check method, suggested by Prof. Dr. Süha Oral is introduced.

A regular four-node iso-parametric element which has connectivity of 1-2-3-4 is shown in Figure 86.

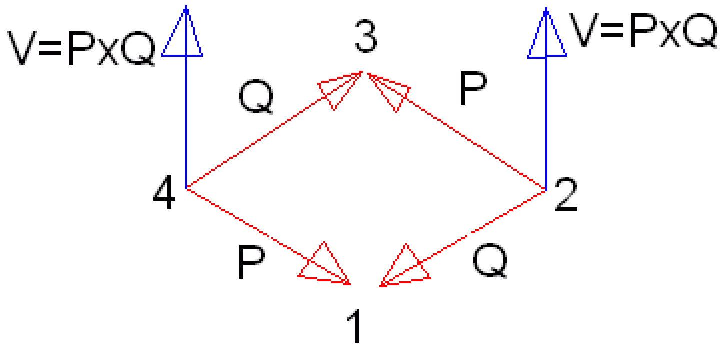

Figure 86: Regular four-node element

According to the cross-product check method if both directions of cross product vectors, V, are upward as shown in Figure 86, then the four-node-element is considered to have a regular shape. Otherwise it is irregular as in Figure 87.

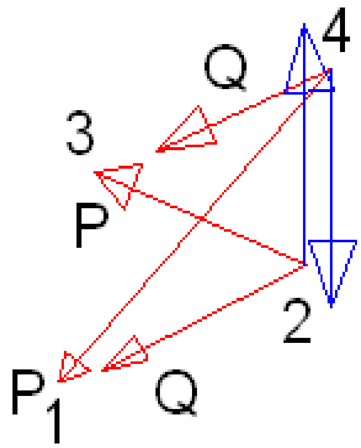

Figure 87: Irregular four node element

The Objective Function: The value of objective function is defined in this procedure.

Finite Element Analysis: The data of connectivity and node coordinates obtained previously are read in this step. The Von-Mises stress and displacement result is calculated.

Fitness Evaluation: By using this operator, the fitness of all individuals in the population is evaluated.

Sorting: The population is ranked from the best to the worst according to the fitness value.

Mating Pool Operation: This operation includes roulette-wheel selection, crossover and mutation.

REFERENCES

[1] P. Vinot , S. Cogan, J. Piranda "Shape optimization of thin-walled beam-like structures", *Thin-Walled Structures 39 (2001) 611–630*

[2] J. R. Banerjee "Explicit frequency equation and mode shapes of a cantilever beam coupled in bending and torsion", J*ournal of Sound and Vibration* (1999) 224(2), 267-281

[3] Umut Topal, Ümit Uzman "Maximization of buckling load of laminated composite plates with central circular holes using MFD method",*Struct. Multidisc. Optim. (2008) 35:131–139 DOI 10.1007/s00158-007-0119-1*

[4] Manolis Papadrakakis, Nikos D. Lagaros "Soft computing methodologies for structural optimization" *Institute of Structural Analysis & Seismic Research, National Technical University of Athens, Zografou Campus, Greece, 2003*

[5] H. Engels, W. Hansel, and W. Becker "Optimal Design of Hole Reinforcements for Composite Structures",M*echanics of Composite Materials, vol. 38, no. 5, 2002*

[6] F. Cappello, A. Mancuso, "A genetic algorithm for combined topology and shape optimizations" *Computer-Aided Design 35 (2003) 761–769*

[7] P. Pedersen, "Examples of density, orientation, and shape-optimal 2D-design for stiffness and/or strength with orthotropic materials" *Struct. Multidisc Optim. 26, 37–49 (2004)DOI 10.1007/s00158-003-0295-6*

[8] Shmuel Vigdergauz,"Genetic algorithm perspective to identify energy optimizing inclusions in an elastic plate", *International Journal of Solids and Structures 38 (2001), 6851-6867*

[9] Sachin S. Terdalkar, Joseph J. Rencis, "Graphically driven interactive finite element stress reanalysis for machine elements in the early design stage", *Finite Elements in Analysis and Design 42 (2006) 884 – 899*

[10]J.T. Katsikadelis, G.C. Tsiatas, "Optimum design of structures subjected to follower forces",*International Journal of Mechanical Sciences 49 (2007) 1204–1212*

[11] E. Taroco, "Shape sensitivity analysis in linear elastic fracture mechanics",*Comput. Methods Appl. Mech. Engrg. 188 (2000) 697-712*

[12] Zhixue Wu, “On the optimization problem of fillets and holes in plates with curvature constraints”,*Struct. Multidisc Optim. DOI 10.1007/s00158-007-0139-x*

[13] E.P. Hadjigeorgiou, G.E. Stavroulakis, C.V. Massalas, “Shape control and damage identification of beams using piezoelectric actuation and genetic optimization”, *International Journal of Engineering Science 44 (2006) 409–421*

[14]Jun Yan, Gengdong Cheng, Ling Liu, Shutian Liu, “Concurrent material and structural optimization of hollow plate with truss-like material”, *Struct. Multidisc Optim. (2008) 35:153–163, DOI 10.1007/s00158-007-0124-4*

[15] J. Parvizian & R. T. Fenner, “Shape optimisation by the boundary element method: a comparison between mathematical programming and normal movement approaches”, *Department of Mechanical Engineering, Imperial College, South Kensington, London SW7 2BX, UK*

[16] Helio J.C. Barbosa, Afonso C.C. Lemonge, “A new adaptive penalty scheme for genetic algorithms” , *Information Sciences 156 (2003) 215–251*

[17] Eisuke Kita & Hisashi Tanie, “Shape optimization of continuum structures by genetic algorithm and boundary element method”, *Department of Mechano-lnformatics & Systems, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-01, Japan,1997*

[18] Elisaveta G. Shopova , Natasha G. Vaklieva-Bancheva, “BASIC—A genetic algorithm for engineering problems solution”, *Computers and Chemical Engineering 30 (2006) 1293–1309*

[19] Ali R. Yıldız, Nursel Öztürk, Necmettin Kaya, Ferruh Öztürk, “Hybrid multi-objective shape design optimization using Taguchi’s method and genetic algorithm”, *Struct. Multidisc Optim. (2007) 34:317–332, DOI 10.1007/s00158-006-0079-x*

[20] Randy L. Haupt, Sue Ellen Haupt, “Practical Genetic Algorithms”, *John Wiley & Sons INC. , 1998*

[21] Tirupathi R. Chandrupatla, Ashok D. Belegundu,“Introduction to Finite Elements in Engineering”, *Third Edition, Prentice-Hall, 2002*

[22] Mitchell Melanie, “An Introduction to Genetic Algorithms” , *A Bradford Book The MIT Press, 1999*

[23]Wolfgang A. Wall, Moritz A. Frenzel, Christian Cyron, “Isogeometric Structural Shape Optimization” , *To appear in Comput. Methods Appl. Mech. Engrg.*

[24] X. Song, J.D. Baldwin, “A novel node-based structural shape optimization algorithm” *Computers and Structures 70 (1999) 569-581.*

[25] R. Das, R. Jones, Y.M. Xie, “Design of structures for optimal static strength using ESO”, *Engineering Failure Analysis 12 (2005) 61–80*

[26] Nanakorn P., Meesomklin K., “An Adaptive Penalty Function in Genetic Algorithms for Structural Design Optimization”, *Computers and Structures 79 2527-2539, 2001*

[27] Takao Yokota, Shozo Wada, Takeaki Taguchi, Mitsuo Gen “Optimal weight design problem of elastic structure by GA”, *Computers & Industrial Engineering 53(2007), pp 299-305*

[28] Kristensen ES, Madsen NF. “On the optimal shape of fillets in plates subjected to multiple in-plane loading cases.” *International Journal for Numerical Methods in Engineering 1976;10:1007-19.*

[29] Raphael T. Hafka, Ramana V. Grandhi, “Structural Shape Optimization-A Survey”,
Computer Methods in Applied Mechanics and Engineering 57 (1986) 91-106

[30]John Holland, “Adaptation in Natural and Artificial Systems” *published in 1975.*

[31] Mark J. Jakiela , Colin Chapman, James Duda, Adenike Adewuya, Kazuhiro Saitou “Continuum structural topology design with genetic algorithms”, *Comput. Methods Appl. Mech. Engrg. 186 (2000) 339-356.*

[32] Oğuzhan Hasançebi, Fuat Erbatur “Evaluation of crossover techniques in genetic algorithm based
optimum structural design”, *Computers and Structures 78 (2000) 435-448.*

[33] C.K. Dimou, V.K. Koumousis, “Competitive genetic algorithms with application to reliability optimal design”, *Advances in Engineering Software 34 (2003) 773–785.*

[34]Chung Hae Park, Woo Il Lee, Woo Suck Han, Alain Vautrin, “Weight minimization of composite laminated plates with multiple constraints”,*Composites Science and Technology 63 (2003) 1015–1026.*

[35]M. Kahraman and F. Erbatur, “A GA approach for simultaneous structural optimization”, *Structural Engineering, Mechanics and Computation (Vol. 2)*,*2001.*

[36] C.A. Coello, A.D. Christiansen, “Multiobjective optimization of trusses using genetic algorithms”, *Computers and Structures 75 (2000) 647-660.*

[37] P.N. Suganthan “Structural pattern recognition using genetic algorithms” , *Pattern Recognition 35 (2002) 1883–1893*.

[38] J. Mahachi and M. Dundu “Genetic Algorithm Operators for Optimisation of Pin Jointed Structures”, Structural Engineering, Mechanics and Computation (Vol. 2), 2001.

[39] Kalyanmoy Deb, Surendra Gulati “Design of truss-structures for minimum weight using genetic algorithms”, *Finite Elements in Analysis and Design 37 (2001) 447-465*