UNIFIED FORMULATION OF J-INTEGRAL FOR COMMON CRACK TYPES USING GENETIC PROGRAMMING

Ibrahim H. GUZELBEY[a], Nihat ATMACA[b], Abdulkadir CEVIK[c]

[a] Department of Mechanical Engineering, University of Gaziantep, TURKEY,
[b] Gaziantep Vocational School of Higher Education, University of Gaziantep, TURKEY
[c] Department of Civil Engineering, University of Gaziantep, TURKEY

ABSTRACT

This study proposes Genetic Programming (GP) as a new tool for the analysis and formulation of the J-integral for the opening mode of fracture mechanics. The proposed GP formulation is based on extensive Finite Element (FE) results. A GP based J-integral formulation for the three different geometries which are commonly used in fracture mechanics has been obtained. The results of this study are very promising.

Keywords: J-integral; Displacement Extrapolation Method; Explicit Formulation, Genetic Programming

GENETİK PROGRAMLAMA KULLANILARAK YAYGIN ÇATLAK TİPLERİNİN J-İNTEGRALİNİN BİRLEŞİK FORMÜLASYONU

ÖZET

Bu çalışmada, kırılma mekaniğinde açılma moduna göre J-integralin analiz ve formülasyonu için yeni bir araç olarak Genetik Programlamadan (GP) faydalanılmıştır. Önerilen GP formülasyonu kapsamlı Sonlu Eleman (SE) sonuçlarına dayanmaktadır. Kırılma mekaniğinde yaygın kullanımı olan üç farklı geometri için GP' ya bağlı bir J-integral formülü elde edilmiştir. Çalışmanın sonuçları oldukça ümit vericidir.

Anahtar kelimeler: J-integral; Deplasman Ekstrapolasyon Yöntemi; Açık Formülasyon, Genetik Programlama

1. INTRODUCTION

The finite element method (FEM) is widely used for the evaluation of Stress Intensity Factor (SIF) for various type of crack configurations [1, 2]. Basically there are two groups of estimation methods. The first group's methods are based on point matching (or extrapolation methods) techniques with nodal displacements are widely used extrapolation techniques due to its simple applicability to various crack configurations [3, 4].

In addition, the second group's methods are based on energy-based methods like **J-Integral**, energy release and the stiffness derivative methods are also used for the determination of SIF. This group requires some special post-processing routines. Many reference books in fracture mechanics [5, 6] and commercial finite element codes (ABAQUS, ANSYS, and COSMOS) are recommend for the energy-based methods as the most efficient for computing K_I due to relatively coarse meshes. It can give satisfactorily results with these methods.

However some parameters of both groups can be expressed in terms of each other. K can be stated as the J-Integral parameter and this makes it very easy to get the K values with a coarse mesh in Linear Elastic Fracture Mechanics (LEFM). It is possible to predict very accurate J-Integral values using a suitable representation of path.

This study aims to propose a unified J-integral formulation valid for varying geometry using GP for the first time in literature. GP is a relatively new tool in engineering mechanics problems. Studies in this field are scarce. Cevik and Guzelbey have proposed a GP based formulation for the prediction of ultimate strength of metal plates in compression [7]. On the other hand, Cevik has recently proposed GP formulations for web crippling strength of cold-formed steel sheeting [8] and rotation capacity of wide flange beams [9].

J–Integral calculations have been done with an ANSYS macro. For this purpose, a Fortran subroutine has been developed for ANSYS which reads the results from a stress analysis and computes the appropriate line integral along a path through the integration points. The obtained J-integral values using ANSYS have been used for GP training and formulation. The GP results are compared with FE (ANSYS) results and are found to be accurate.

II. J–INTEGRAL METHOD

J-Integral was developed by Rice [10] and it represents the energy extracted through the crack tip singularity. The path independent J-Integral characterizes the stress strain field at the crack tip whose path is taken sufficiently far from the crack tip for the cracks to be analyzed elastically, where the singularities or the non-linear elasto-plastic behaviours are not encountered.

It has been defined a number of contour integrals that are path independent by virtue of the theorem of energy conservation. The two-dimensional form of one of these integrals can be written as:

$$J = \oint_{\Gamma} \left(w dy - t \frac{\partial u}{\partial x} d\Gamma \right) \qquad (1)$$

With

$$w = \int_{0}^{\varepsilon} \sigma_{ij} d\varepsilon_{ij} \qquad (2)$$

where; w is the strain energy density;
Γ is a closed contour followed counter-clockwise, as shown in Fig. 1;
t is the outward traction vector acting on the contour around the crack;
u is the displacement vector, and $d\Gamma$ is the element of the arc along the path Γ.
Since J-Integral is a path independent line integral [10], it can be determined from a stress analysis where σ and ε are established using finite element analysis around the contour enclosing the crack.

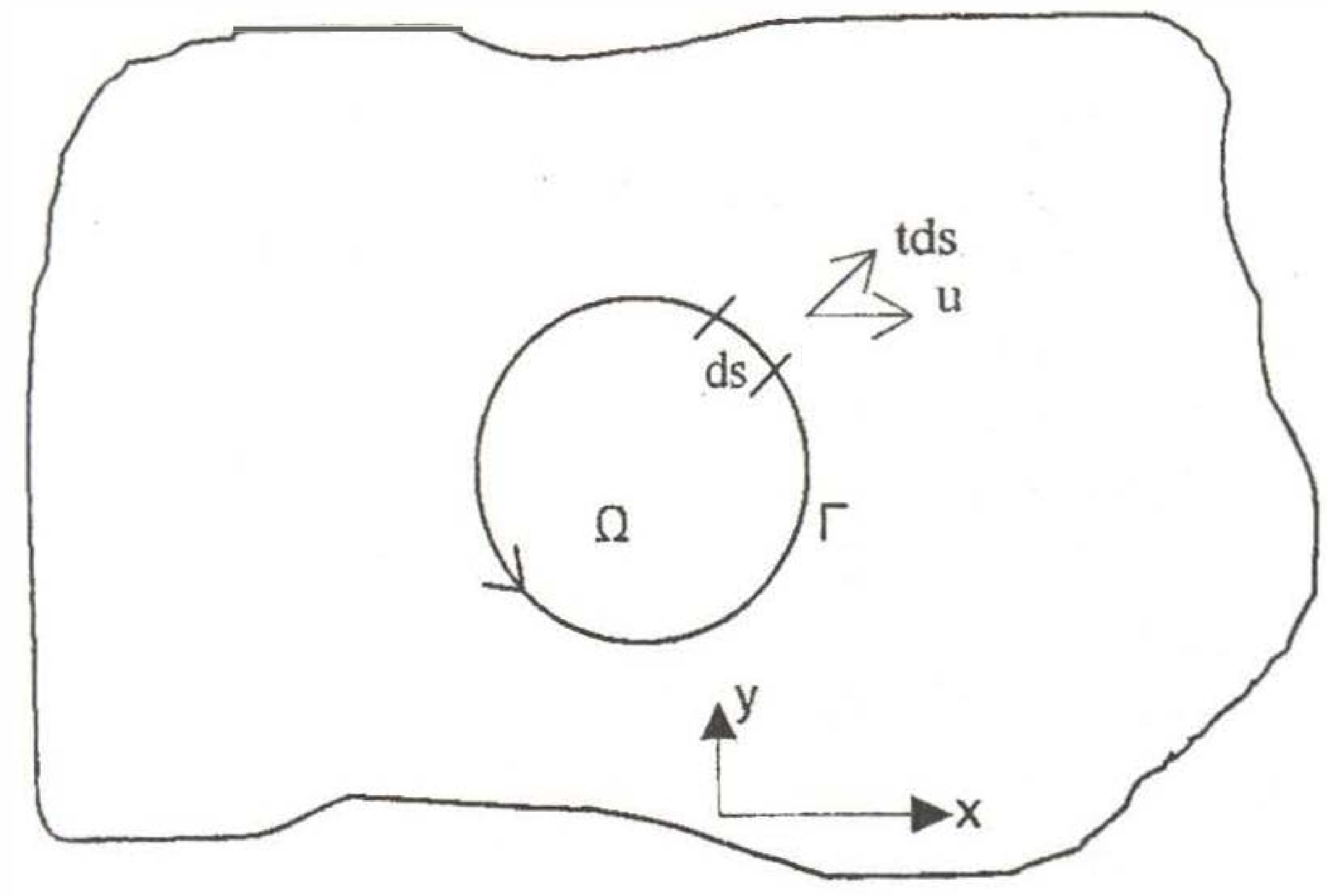


Figure 1. J Integral definition around a crack

The J-Integral can be interpreted as the potential energy difference between two identically loaded specimens having slightly different crack lengths. The main point of the J-Integral approach can be formulized as follows:

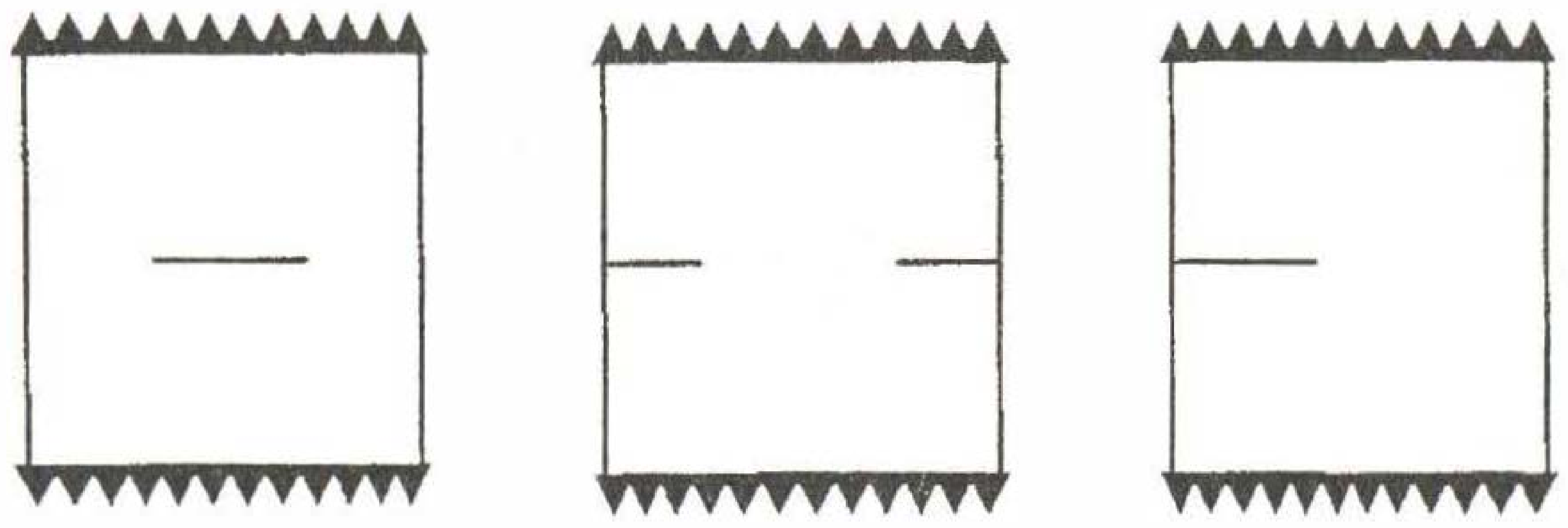
$$J = \frac{\partial U_0}{\partial a} = G = \frac{K^2}{E^{'}}$$

(3)
where;
U is the potential energy difference;
a is the crack length;
G is the strain energy release rate;
K is the elastic SIF parameter and
$E^{'} = E$ for plane stress
$E^{'} = \frac{E}{1 - \nu^2}$ for plane strain

The Eq.(3) points out that the value of J-integral obtained under elastic–plastic conditions is numerically equal to the strain energy release rate obtained under elastic conditions. This situation has been demonstrated by small fully plastic regions of elements' critical mode I J-integral J_{Ic} values and large elastic regions of elements' critical energy release rate G_{Ic} values respectively. These values must be satisfying the plane-strain conditions of LEFM.

III. MODELS FOR COMMON CRACK TYPES

Numerical analysis (FEM) has been applied to determine the J-integral of the three well known geometries: the center cracked plate, the double cracked plate and the single cracked plate. The crack geometries are given in Fig. 2.



a) TYPE b) TYPE2 c) TYPE3

Figure 2. Type of the crack geometries

All of the three models have dimensions with [20*20, 40*40, 60*60 and 80*80] mm cross sections and 1mm thickness. The J-integral values for a series of crack lengths are calculated (for a=2, 3, 4, 5 and 6 millimeters).

Rectangular eight-node isoparametric and six-node elements are used for the configurations with the following material properties and loading: E = 80000 N/mm^2, ν = 0.3 and σ =60, 80, 100 and 120 N/mm^2
An example of standard and crack tip eight-node elements can be seen in Fig.3. Plane stress analysis and three-point gaussian numerical integration has been used for the analysis.

The center and double cracked geometries are shown in Fig. 4a and 5a respectively. Quarter symmetries are used in modeling as shown in Fig. 4b and 5b.

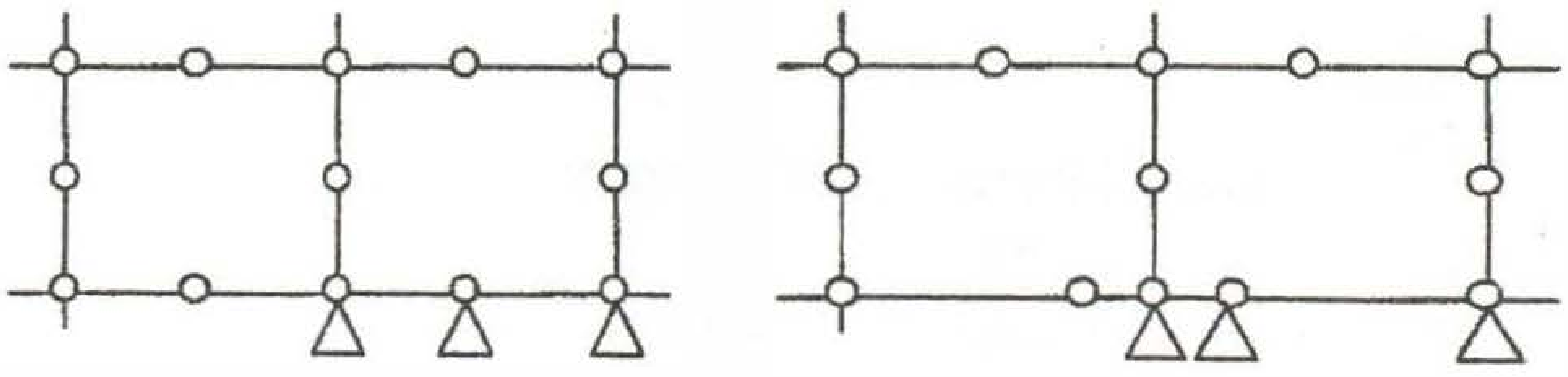
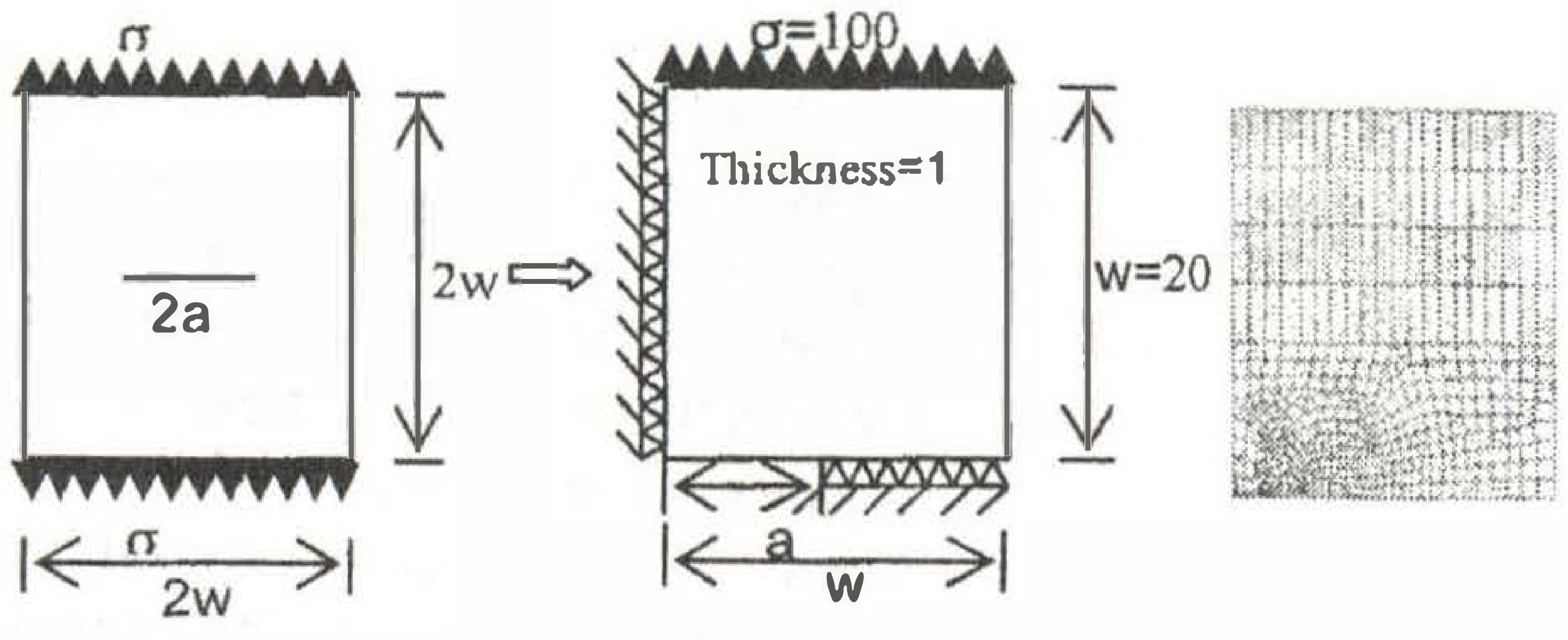


Figure 3. Standard and crack tip eight-node isoparametric elements

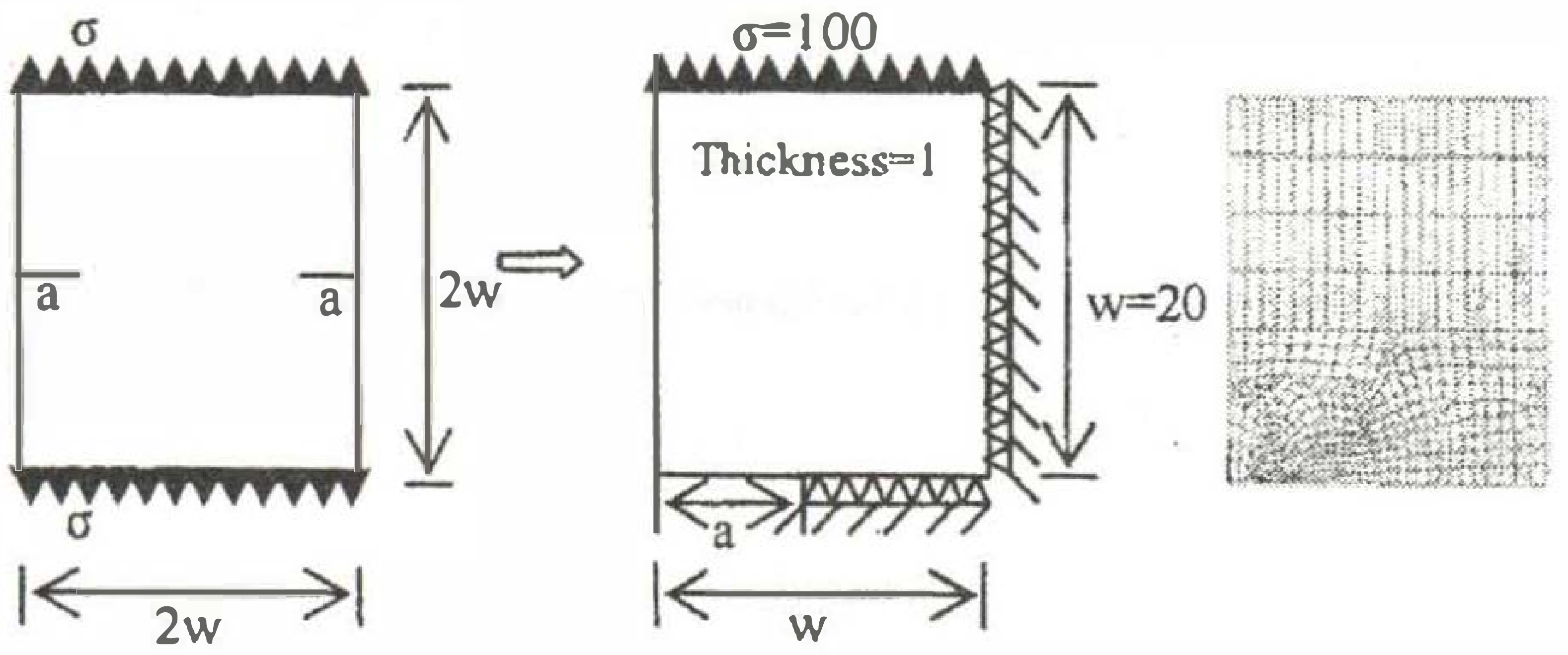


a) Center crack

b) Quarter model

c) FE Mesh for a=2mm, 1510 nodes & 487 elements

Figure 4. Center cracked model geometry



a) Double crack

b) Quarter model

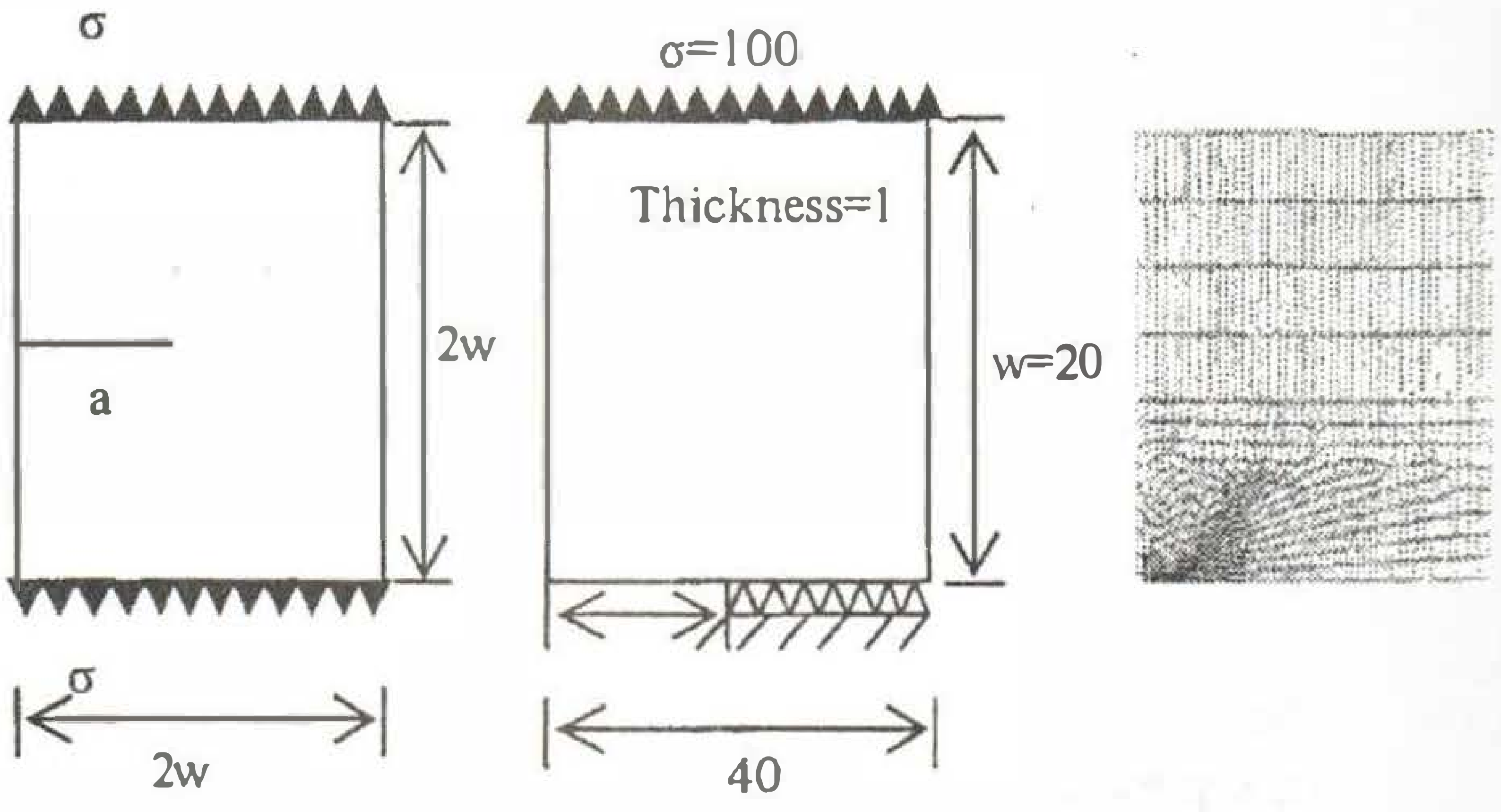
c) FE Mesh for a=2mm, 1510 nodes & 487 elements

Figure 5. Double cracked model geometry

FE fine mesh configuration of the center and double cracked cases are shown in Fig. 4c and 5c for 2mm crack length. In FE mesh configurations for 3mm crack length, 1477 nodes and 472 elements, for 4mm crack length, 1516 nodes and 485 elements, for 5mm crack length, 1530 nodes and 489 elements, and for 6mm crack length, 1470 nodes and 469 elements have been used in both cases.

Single cracked case's geometry and its half symmetry for modeling are shown in Fig 6a and Fig.6b.

FE fine mesh configuration of the single cracked case is shown in Fig. 6c for 2mm crack length. In FE mesh configurations for 3mm crack length, 3384 nodes and 1093 elements, for 4mm crack length, 3465 nodes and 1124 elements, for 5mm crack length, 3247 nodes and 1048 elements, and for 6mm crack length, 3253 nodes and 1052 elements have been used.



a) Single Crack

for

b) Half Model

c) FE Mesh

a=2, 3357 nodes & 1082 elements

Figure 6. Single cracked model geometry

The mesh used in the analysis consists of triangular elements for crack region and 8-node quadrilateral elements for the remaining regions (i.e.PLANE82 type element in ANSYS).

The main disadvantage of the most numerical analysis is the time-consumption. The calculation time is directly proportional with the number of nodes, elements and loading conditions.

An explicit formulation of J-integral using GP will decrease the computation time of the certain geometries. A GP program is developed for this purpose.

IV. OVERVIEW OF GENETIC PROGRAMMING

Genetic algorithm (GA) is an optimization and search technique based on the principles of genetics and natural selection. A GA allows a population composed of many individuals to evolve under specified selection rules to a state that maximizes the "fitness" (i.e., minimizes the cost function). The method was developed by John Holland [11] (1975) and finally popularized by one of his students, David Goldberg [12], solved a difficult problem involving the control of gas-pipeline transmission for his dissertation [13]. The fitness of each individual in a genetic algorithm is the measure the individual has been adapted to the problem that is solved employing this individual. It means that fitness is the measure of optimality of the solution offered, as represented by an individual from the genetic algorithm. The basis of genetic algorithms is the selection of individuals in accordance with their fitness; thus, fitness is obviously a critical criterion for optimization [14].

Genetic programming (GP) is an extension to Genetic Algorithms proposed by Koza [15]. Koza defines GP as a domain-independent problem-solving approach in which computer programs are evolved to solve, or approximately solve, problems based on the Darwinian principle of reproduction and survival of the fittest and analogs of naturally occurring genetic operations such as crossover (sexual recombination) and mutation. GP reproduces computer programs to solve problems by executing the following steps (Fig. 7.):

1) Generate an initial population of random compositions of the functions and terminals of the problem (computer programs).
2) Execute each program in the population and assign it a fitness value according to how well it solves the problem.
3) Create a new population of computer programs.
i) Copy the best existing programs (Reproduction)
ii) Create new computer programs by mutation.
iii) Create new computer programs by crossover (sexual reproduction).
iv) Select an architecture-altering operation from the programs stored so far.
4) The best computer program that appeared in any generation, the best-so-far solution, is designated as the result of genetic programming [15].

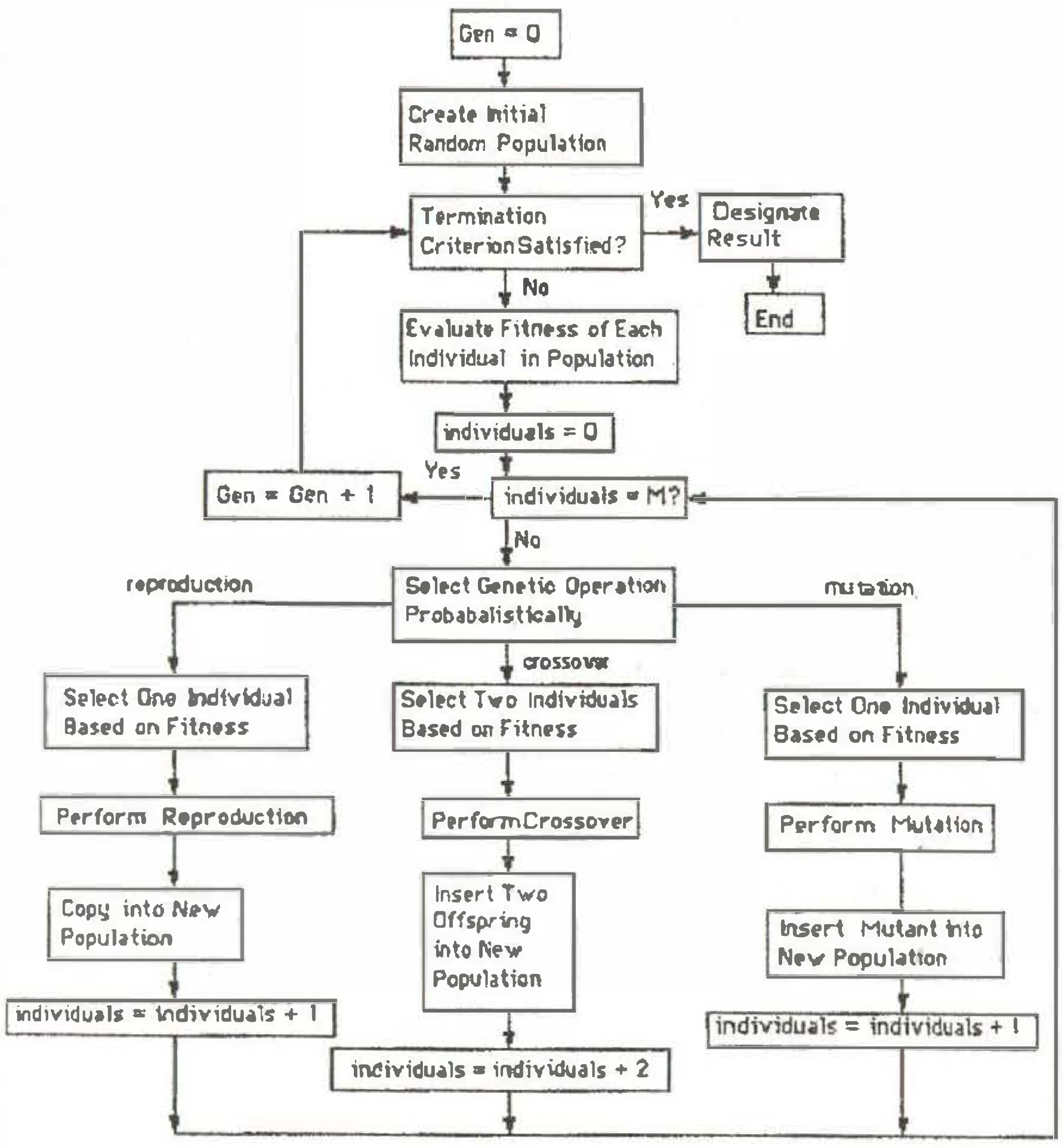

Gen = 0
Create Initial Random Population
Termination Criterion Satisfied?
Yes
Designate Result
End
No
Evaluate Fitness of Each Individual in Population
individuals = 0
Yes
Gen = Gen + 1
individuals = M?
No
reproduction
Select Genetic Operation Probabalistically
mutation
crossover
Select One Individual Based on Fitness
Select Two Individuals Based on Fitness
Select One Individual Based on Fitness
Perform Reproduction
Perform Crossover
Perform Mutation
Copy into New Population
Insert Two Offspring into New Population
Insert Mutant into New Population
individuals = individuals + 1
individuals = individuals + 2
individuals = individuals + 1

Figure 7. Genetic programming flowchart [15]

IV.1 Brief overview of GEP

Gene expression programming (GEP) software which is used in this study is an extension to GP that evolves computer programs of different sizes and shapes encoded in linear chromosomes of fixed length. The chromosomes are composed of multiple genes, each gene encoding a smaller sub-program. Furthermore, the structural and functional organization of the linear chromosomes allows the unconstrained operation of important genetic operators such as mutation, transposition, and recombination. One strength of the GEP approach is that the creation of genetic diversity is extremely simplified as genetic operators work at the chromosome level. Another strength of GEP consists of its unique, multigenic nature which allows the evolution of more complex programs composed of several sub-programs. As a result GEP surpasses the old GP system in 100-10,000 times. [16-18]. **APS 3.0** [19], a GEP software developed by Candida Ferreira is used in this study.

The fundamental difference between GA, GP and GEP is due to the nature of the individuals: in GAs the individuals are linear strings of fixed length (chromosomes); in GP the individuals are nonlinear entities of different sizes and shapes (parse trees); and in GEP the individuals are encoded as linear strings of fixed length (the genome or chromosomes) which are afterwards expressed as nonlinear entities of different sizes and shapes (i.e., simple diagram representations or expression trees). Thus the two main parameters GEP are the chromosomes and expression trees (ETs). The process of information decoding (from the chromosomes to the ETs) is called translation which is based on a set of rules. The genetic code is very simple where there exist one-to-one relationships between the symbols of the chromosome and the functions or terminals they represent. The rules which are also very simple determine the spatial organization of the functions and terminals in the ETs and the type of interaction between sub-ETs. [16-17-18]

That's why two languages are utilized in GEP: the language of the genes and the language of ETs. A significant advantage of GEP is that it enables to infer exactly the phenotype given the sequence of a gene, and *vice versa* which is termed as *Karva* language. Consider, for example, the algebraic expression

$(d4* \sqrt{(d3-d0+d1*d4}\ ))$- $d4$ can be represented by a diagram (Fig 8) which is the expression tree:

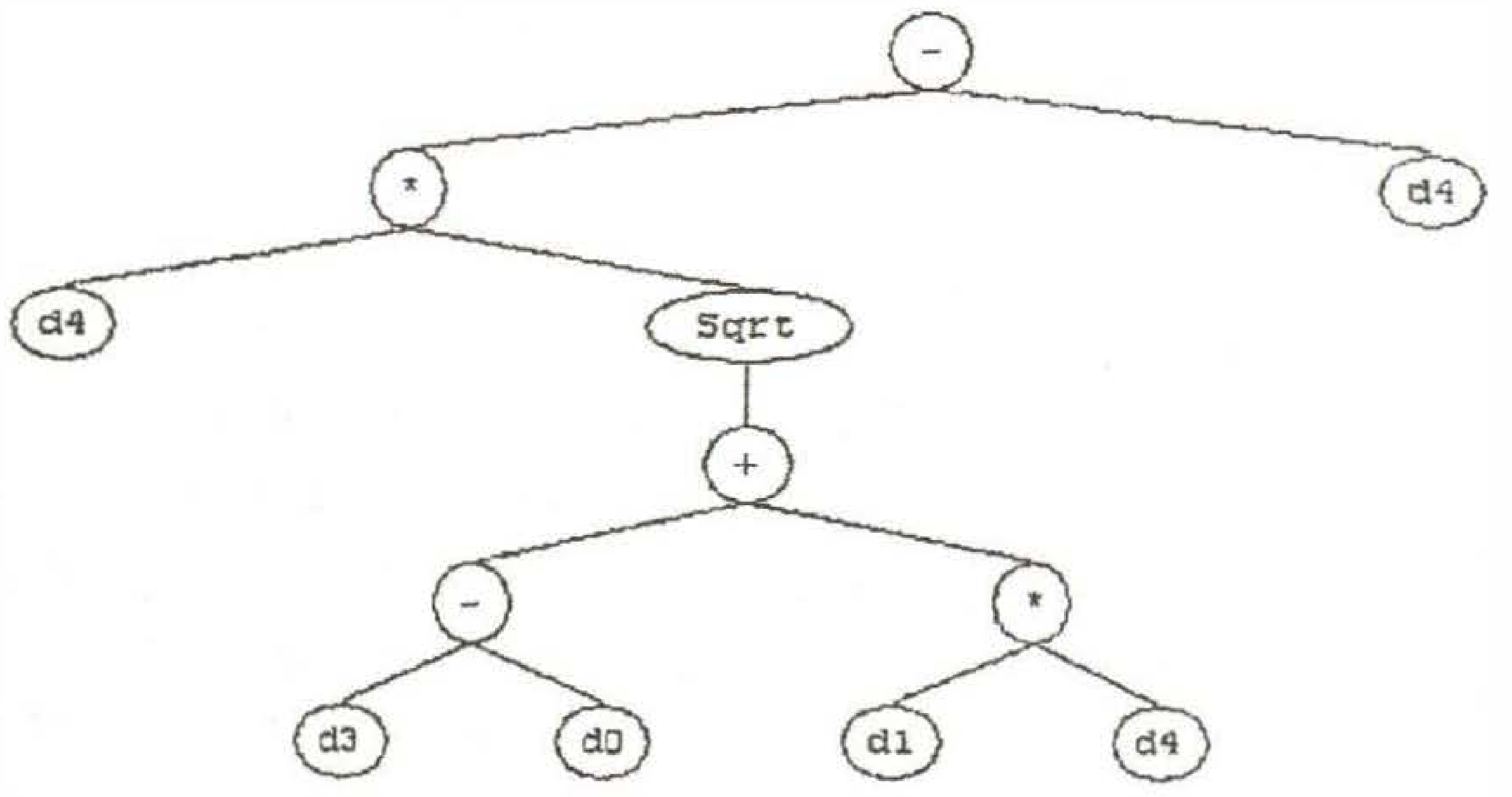


Figure 8 Expression tree (ET)

IV.2 Solving a Simple Problem with GEP

For each problem, the type of linking function, as well as the number of genes and the length of each gene, are *a priori* chosen for each problem. While attempting to solve a problem, one can always start by using a single-gene chromosome and then proceed by increasing the length of the head. If it becomes very large, one can increase the number of genes and obviously choose a function to link the sub-ETs. One can start with addition for algebraic expressions or OR for Boolean expressions, but in some cases another linking function might be more appropriate (like multiplication or IF, for instance). The idea, of course, is to find a good solution, and GEP provides the means of finding one very efficiently. [17]

As an illustrative example consider the following case where the objective is to show how GEP can be used to model complex realities with high accuracy. So, suppose one is given a sampling of the numerical values from the curve (remember, however, that in real-world problems the function is obviously unknown):

$$y = 3a^2 + 2a + 1 \qquad (4)$$

over 10 randomly chosen points in the real interval [-10, +10] and the aim is to find a function fitting those values within a certain error. In this case, a sample of data in the form of 10 pairs (a_i, y_i) is given where a_i is the value of the independent variable in the given interval and y_i is the respective value of the dependent variable (a_i values: -4.2605, -2.0437, -9.8317, -8.6491, 0.7328, -3.6101, 2.7429, -1.8999, -4.8852, 7.3998; the corresponding y_i values can be easily evaluated). These 10 pairs are the fitness cases (the input) that will be used as the adaptation environment. The fitness of a particular program will depend on how well it performs in this environment [17].

There are five major steps in preparing to use gene expression programming. The first is to choose the fitness function. For this problem one could measure the fitness f_i of an individual program i by the following expression:

$$f_i = \sum_{j=1}^{C_t} (M - |C_{(i,j)} - T_j|) \qquad (5)$$

where M is the range of selection, $C_{(i,j)}$ the value returned by the individual chromosome i for fitness case j (out of C_t fitness cases) and T_j is the target value for fitness case j. If, for all j, $|C_{(i,j)} - T_j|$ (the precision) less than or equal to 0.01, then the precision is equal to zero, and $f_i = f_{\max} = C_t{*}M$. For this problem, use an $M = 100$ and, therefore, $f_{\max} = 1000$. The advantage of this kind of fitness function is that the system can find the optimal solution for itself. However there are other fitness functions available which can be appropriate for different problem types [17].

The second step is choosing the set of terminals T and the set of functions F to create the chromosomes. In this problem, the terminal set consists obviously of the independent variable, i.e., $T = \{a\}$. The choice of the appropriate function set is not so obvious, but a good guess can always be done in order to include all the necessary functions. In this case, to make things simple, use the four basic arithmetic operators. Thus, $F = \{+, -, *, /\}$. It should be noted that there many other functions that can be used.

The third step is to choose the chromosomal architecture, i.e., the length of the head and the number of genes.

The fourth major step in preparing to use gene expression programming is to choose the linking function. In this case we will link the sub-ETs by addition. Other linking functions are also available such as subtraction, multiplication and division.

And finally, the fifth step is to choose the set of genetic operators that cause variation and their rates. In this case one can use a combination of all genetic operators (mutation at p_m = 0.051; IS and RIS transposition at rates of 0.1 and three transposes of length 1, 2, and 3; one-point and two-point recombination at rates of 0.3; gene transposition and gene recombination both at rates of 0.1).

To solve this problem, lets choose an evolutionary time of 50 generations and a small population of 20 individuals in order to simplify the analysis of the evolutionary process and not fill this text with pages of encoded individuals. However, one of the advantages of GEP is that it is capable of solving relatively complex problems using small population sizes and, thanks to the compact Karva notation; it is possible to fully analyze the evolutionary history of a run. A perfect solution can be found in generation 3 which has the maximum value 1000 of fitness. The sub-ETs codified by each gene are given in Fig. 9. Note that it corresponds exactly to the same test function given above in Eqn. (4) [17].
Thus expressions for each corresponding Sub-ET can be given as follows:

$$y = (a^2 + a) + (a + 1) + (2a^2) = 3a^2 + 2a + 1 \qquad (6)$$

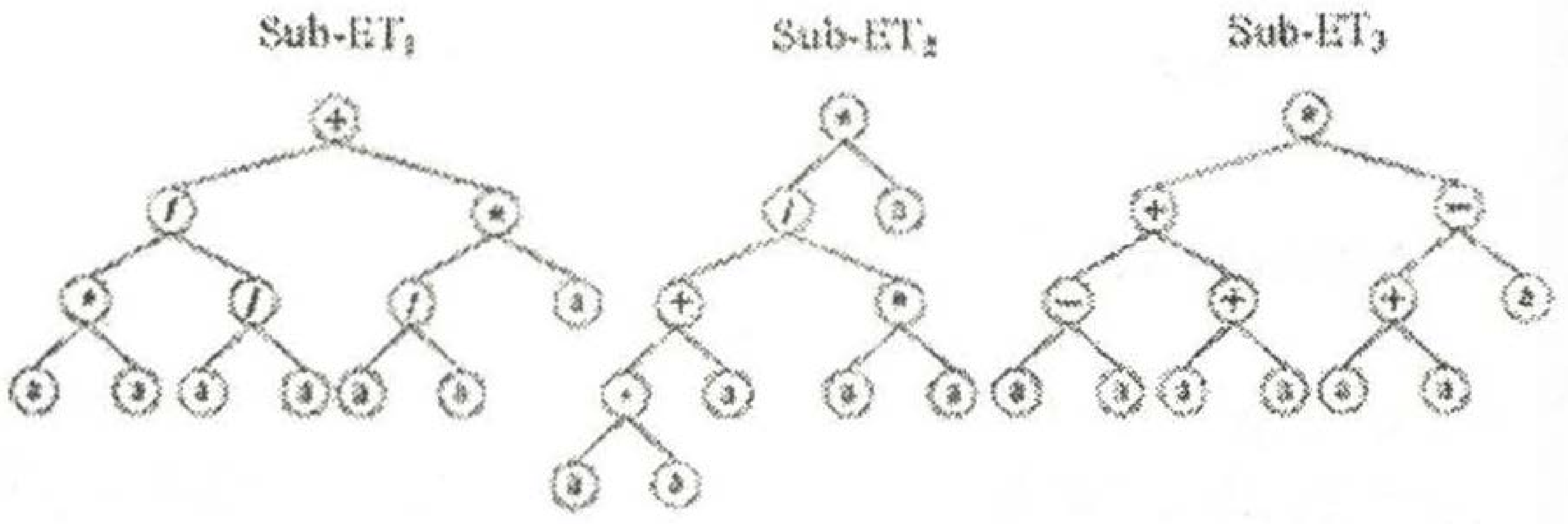


Figure 9. ET for the problem of Eq. (4)

V. NUMERICAL APPLICATION

The main purpose in this study is to predict and formulate J-integral values for varying geometries using GP based on extensive FE (ANSYS) results. FE results are divided into train and test sets where patterns in test set are randomly selected among the experimental database. The FE train and test and their randomly selected experimental patterns are not shown in the study. The training patterns for GP formulation have been obtained using ANSYS FE software package. A wide range of variables are chosen to represent a general model for NN with a data set of 167 training patterns and 25 testing patterns. The statistical parameters and performance of training and test sets for the J-integral are given in Table 1 and Fig.11. It has been seen that the errors are quite satisfactory for each case for test set and training sets.

Explicit formulation of J-integral is obtained as a function of stress, crack width, plate width and crack from Fig.10 which is the expression tree of GP formulation given as follows (in MATLAB CODE):

```
J = ((d(2)+(d(0)-G1C0))/((d(2)*G1C11)-(d(2)*d(1))))*
(exp((((d(3)^3)-d(2))/(d(2)+d(2))))*d(1)) *
(d(1)+ln(((G3C16+d(0))/d(3))));
```

Where constants are
G1C0 = 52.35; G1C11 = 50.15; G3C16 = -57.79;
It should be noted that parameters in the formulation stand for the following:
d(0)= σ
d(1)= a
d(2)= w
d(3)= Crack Type
After putting the corresponding values, the final equation becomes:

$$J=(\frac{w+\sigma-52.35}{50.15w-a*w})(a*e^{\frac{Type^3-w}{2w}})(a+\ln(\frac{\sigma-57.79}{Type})) \quad (7)$$

Table 1 Statistical parameters of the GP Model used for J-integral

	Training set	Test set
MAPE (%) (Mean absolute % Error)	36.2	43.5
Mean (Test/ FE)	1.22	1.30
R (%)	0.961	0.96
COV	0.41	0.49

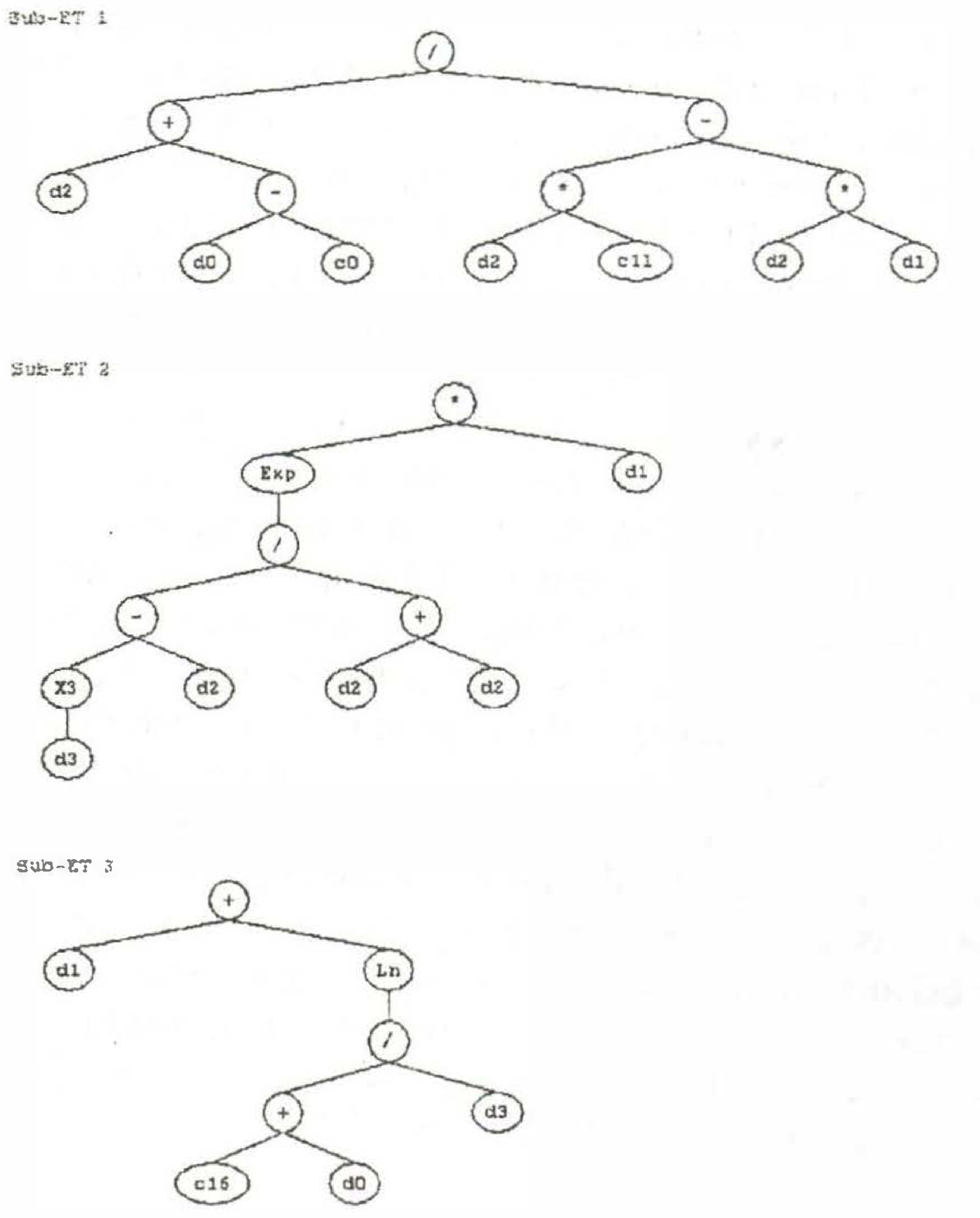


Figure 10. Expression tree (ET) of the GP formulation

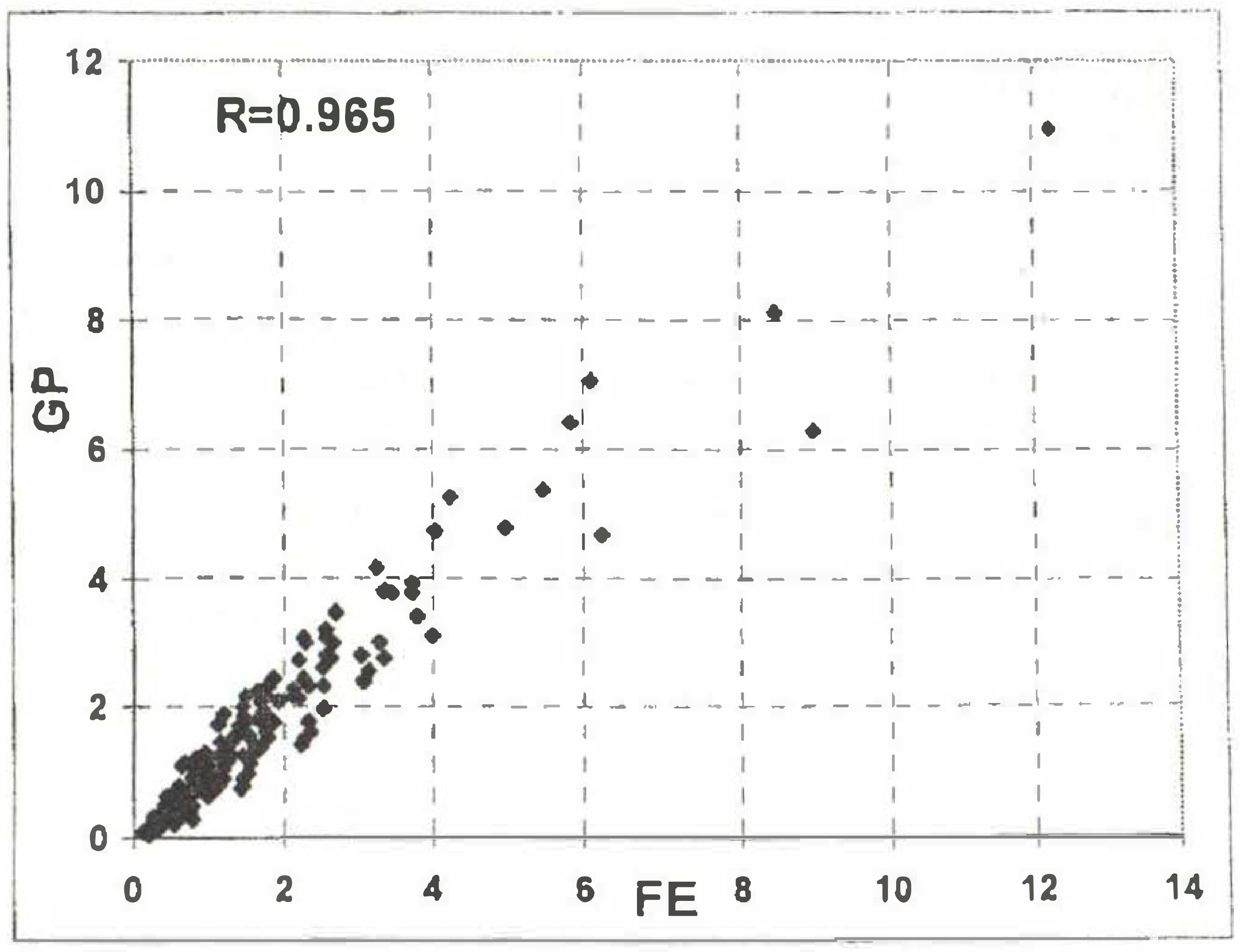


Figure 11 Performance of GP results vs. FE results

VI. CONCLUSIONS

This study proposes a novel unified formulation for the calculation of J-integral value using GP. The GP formulation is based on FE results for 3 different types of geometry namely as single, double and center crack cases. The data obtained by FE for these 3 cases were combined together and formed the unified database for the training set of the GP model. The GP results are compared with FE results and are found to be quite accurate. Thus parametric studies are later performed by the use of the proposed GP formulation to investigate the effect of varying parameters on the J-integral value. The obtained GP formulation is shown to be valid for common three cases of crack. Parametric studies are also performed to prove the generalization capability of the explicit formulation obtained by GP and the effects of each varying parameter on J-integral value is comprehensively investigated with corresponding response surface in 3D form. As a result, the proposed GP formulation is quite accurate, fast and practical for use compared to design codes and existing models. It should be noted that empirical formulations in fracture mechanics are mostly based on predefined functions where regression analysis of these functions are later performed. However in the case of GP approach there is no predefined function to be considered i.e. GP creates randomly formed functions and selects the one that best fits the experimental results. Moreover there is no restriction in the complexity and structure of the randomly formed functions as well. The outcomes of the study are very promising as it may open a new era for the accurate and effective explicit formulation of many fracture mechanics problems using GP.

REFERENCES

[1]. Parks DM, A stiffness derivative finite element technique for determination of crack tip stress intensity factors. International Journal of Fracture 10: 487-502, 1974.

[2]. Rao BN, Rahman S, An efficient meshless method for fracture analysis of cracks, Comput. Mech. 26:398 – 408, 2000.

[3]. Guinea GV, Planas J, Elices M, K-I evaluation by the displacement extrapolation technique Engineering Fracture Mechanics 66 (3): 243-255, 2000.

[4]. İbrahim H. Guzelbey, Nihat Atmaca And Bahattin Kanber, Comparison of displacement and stress extrapolation techniques for stress intensity factor CMES^1-04, Proceedings of the First Cappadocia International Mechanical Engineering Symposium, 14-16 July, Cappadocia, Turkey, 684-691, 2004.

[5]. Tada H, PC Paris, GR Irwin, The stress analysis of cracks handbook. ASME Press, 2000.

[6]. Kanninen MF, Popelar CH, Advanced fracture mechanics. Newyork, Oxford University Press, 1985.

[7]. Abdulkadir Cevik and Ibrahim H. Guzelbey, A soft computing based approach for the prediction of ultimate strength of metal plates in compression. Engineering Structures, In Press, 2006.

[8]. Cevik, A, A new formulation for web crippling strength of cold-formed steel sheeting using genetic programming. Journal of Constructional Steel Research doi:10.1016/j.jcsr.2006.08.012, 2006.

[9]. Cevik, A, Genetic programming based formulation of rotation capacity of wide flange beams. Journal Of Constructional Steel Research, Accepted For Publication, 2006.

[10]. JR Rice, A path independent integral and the appropriate analysis of strain concentration by notches and cracks, Journal of Applied Mechanics 379-386, 1968.

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

[12]. Goldberg DE, Genetic algorithms in search, optimization, and machine leaming reading, MA: Addison-Wesley, 1989.

[13]. Haupt RL, Haupt SE, Practical genetic algorithms, 2nd Edition, Wiley – Interscience, 2004.

[14]. Chambers L, The practical handbook of genetic algorithms applications. 2nd Ed. Chapman & Hall/CRC, 2001.

[15]. Koza JR, Genetic programming: On the programming of computers by means of natural selection. Cambridge, MA: MIT Press, 1992.

[16]. Ferreira C, Gene expression programming: Mathematical modelling by an artificial intelligence, 2002.

[17]. Ferreira C, Gene expression programming in problem solving invited tutorial of the 6th online world conference on soft computing in industrial applications. September 10-24, 2001.

[18]. Ferreira C, Gene expression programming: A new adaptive algorithm for solving problems. Complex Systems, Vol. 13, issue 2: 87-129, 2001.

[19]. http://www.gepsoft.com/