Multi-Objective Two-Dimensional Truss Optimization by using Genetic Algorithm

Harun Alrasyid[1] and Pujo Aji[1]

Abstract—During last three decade, many mathematical programming methods have been develop for solving optimization problems. However, no single method has been found to be entirely efficient and robust for the wide range of engineering optimization problems. Most design application in civil engineering involve selecting values for a set of design variables that best describe the behavior and performance of the particular problem while satisfying the requirements and specifications imposed by codes of practice. The introduction of Genetic Algorithm (GA) into the field of structural optimization has opened new avenues for research because they have been successful applied while traditional methods have failed. GAs is efficient and broadly applicable global search procedure based on stochastic approach which relies on "survival of the fittest" strategy. GAs are search algorithms that are based on the concepts of natural selection and natural genetics. On this research Multi-objective sizing and configuration optimization of the two-dimensional truss has been conducted using a genetic algorithm. Some preliminary runs of the GA were conducted to determine the best combinations of GA parameters such as population size and probability of mutation so as to get better scaling for rest of the runs. Comparing the results from sizing and sizing– configuration optimization, can obtained a significant reduction in the weight and deflection. Sizing–configuration optimization produces lighter weight and small displacement than sizing optimization. The results were obtained by using a GA with relative ease (computationally) and these results are very competitive compared to those obtained from other methods of truss optimization.

Keywords—truss optimization, genetic algorithm, multi-objective optimization

Abstrak—*Selama tiga dekade terakhir, banyak metode pemrograman matematis telah dikembangkan untuk memecahkan masalah optimasi. Namun, ada satu metode telah ditemukan yang sepenuhnya efisien dan kuat untuk berbagai masalah teknik optimasi. Kebanyakan aplikasi dalam desain teknik sipil melibatkan pemilihan pada satu set variabel desain yang menggambarkan perilaku dan kinerja dari masalah tertentu yang memenuhi persyaratan dan spesifikasi tertentu sesuai kode kepraktisan. Pengenalan Genetic Algorithm (GA) ke dalam bidang optimasi telah membuka jalan baru untuk penelitian karena telah terbukti berhasil diterapkan ketika metode tradisional menemui kegagalan. GA lebih efisien dan luas dalam prosedur pencarian secara global yang didasarkan pada pendekatan stokastik yang bergantung pada strategi "survival of the fittest". GA merupakan algoritma pencarian yang didasarkan pada konsep seleksi alam dan genetika secara alami. Pada penelitian ini Multi-tujuan optimasi dan konfigurasi dari truss dua dimensi dilakukan dengan menggunakan algoritma genetik. Beberapa hal dilakukan GA untuk menentukan kombinasi terbaik dari parameter GA seperti ukuran populasi dan probabilitas mutasi, hal ini untuk mendapatkan skala yang lebih baik untuk sisa berjalan. Dengan membandingkan hasil dari ukuran dan ukuran-konfigurasi optimasi, dapat diperoleh dari pengurangan yang signifikan dalam berat badan dan defleksi. Ukuran-konfigurasi optimasi menghasilkan bobot yang lebih ringan dan ukuran optimasi perpindahan yang kecil Hasil dengan menggunakan GA diperoleh relatif mudah dalam hal komputasi dan hasil ini sangat kompetitif dibandingkan dengan yang diperoleh dari metode selain optimasi truss.*

Kata Kunci—*optimasi truss, algoritma genetika, optimasi multi obyek*

.

I. INTRODUCTION

During last three decade, many mathematical programming methods have been develop for solving optimization problems [1, 2]. However, no single method has been found to be entirely efficient and robust for the wide range of engineering optimization problems [3]. Most design application in civil engineering involve selecting values for a set of design variables that best describe the behavior and performance of the particular problem while satisfying the requirements and specifications imposed by codes of practice. Mathematically these design variables are discrete for most practical design problems. However most mathematical optimization applications are suited and developed for continuous design variables. In discrete optimization problems, searching for the global or local optimal solution becomes a difficult task. A few mathematical methods have been developed for solving problem in discrete optimization. These methods include complete enumeration techniques, integer programming, branch and bound algorithms, and dynamic programming. All these methods use mathematical programming techniques.

The introduction of Genetic Algorithm (GA) into the field of structural optimization has opened new avenues for research because they have been successful applied while traditional methods have failed. GAs is efficient and broadly applicable global search procedure based on stochastic approach which relies on "survival of the fittest" strategy [4]. GAs as search algorithms that are based on the concepts of natural selection and natural genetics. GAs differ from traditional optimization methods in the following aspects: (1) GAs work with a coding set of variables and not with the variables themselves; (2) GAs operate on population of potential solutions rather than improve a single solution; (3) GAs

Harun Alrasyid and Pujo Aji Are with Department of Civil Engineering, FTSP, Institut Teknologi Sepuluh Nopember, Surabaya, 60111, Indonesia. E-mail: harun_s45@yahoo.com, harun@ce.its.ac.id.

use objective function information without any gradient information; (4) GAs use a transition scheme that is probabilistic, whereas traditional methods use gradient information [3].

A. Introduce of Genetic Algorithm

GAs utilized a strategy that models the mechanism of genetic evolution [5]. The core characteristics of GAs are based on the principles of survival of the fittest and adaptation. The advantages of applying GAs to the optimized design of structures include discrete design variables, open format for constraint statement, and multiple load cases. GAs do not require an explicit relationship between the objective function and the constraints. Instead, the value of the objective function for a set of design variables is adjusted to reflect any violation of the constraint.

GAs operate on a population of design variable sets, with each design variable set defining a potential solution is called a string. Each string is made up of a series of characters, typically binary numbers, repre-senting the values of the discrete design variables for a particular solution. The fitness of each string is measurement of performance of the design variables as defined by the objective function and the constraint.

GAs basically consist of a series of three processes: coding and decoding design variables into strings, evaluating the fitness of each solution string, and applying genetic operators to generate the next generation of solution strings. The fitness of each string is evaluated by performing some type system analysis to compute a value of objective function. If the solution violates constraints the value of the objective function is penalized.

Most genetic algorithms are variations of simple genetic algorithm (SGA) proposed by Goldberg [5]. Goldberg's SGA consists of three basic genetic operators: reproduction, crossover, and mutation. The reproduction operation in the SGA is the basic engine of Darwinian natural selection and survival of the fittest. The crossover operation creates variations in the solution population by producing new solution strings that consistof parts taken from selected parent solution strings. The mutation operation introduces random chage in the solution population. In a GA, the mutation operation can be beneficial in reintroducing diversity in a population.

The objective of the reproduction process is to allow the information stored in strings with good fitness values to survive into the next generation. Typically, each string in the population is assigned a probability of being selected as parent string based on the string's fitness. However, reproduction does not change the features of parent strings. The next generation of solution strings is developed from selected pairs of parent's strings and the application of other explorative operators such as crossover and mutation.

Crossover is a procedure wherein a selected parent string is broken into segments and some of these segments are exchange with corresponding segments of another parent string. The one-point crossover implemented in GAs breaks each string of a selected parent string set into two segments and interchanges the second segment to create two new strings.

Mutation is usually used as an insurance policy [5]. Mutation allows for the possibility that non-existing features from both parent strings may be created and passed to their children. Without an operator of this type, some possibly important regions of the search space may never be explored

B. Structural Optimization

The optimization of truss structures can be classified into three categories depending on what component of the structure is used as design variable : sizing configuration and topological optimization. In sizing optimization of trusses the cross-sectional areas of the members are the design variables and the coordinates of the nodes and the connectivity between various members are fixed. This can be made more practically useful by restricting the member areas to pre-specified discrete values. In configuration optimization the design variables are the nodal coordinates, and in topological optimization the number of nodes and the connectivity between nodes are the design variables. These optimization problems have been discussed separately however the most efficient design will be obtained by considering all three simultaneously. In general, multilevel optimization methods are used in which topological optimization is first performed keeping the configuration and member sizes fixed. When an optimal topology is found, configuration and/or sizing optimization is performed on the topology found in the previous step. As mentioned earlier this method will not provide the most optimal solution as all the three problems are not linearly separable. As a result, traditional methods of optimization have failed and the use of other tools such GAs is gaining popularity in the field of structural optimization.

1. Formulation of Structural Optimization

The most popular optimization criterion in structural design is cost. Typically, cost is a function of the total structural weight. Other factors that may be involved in estimating the cost of a structure include maintenance (related to the total surface area of a steel structure) and connection costs. An objective function in terms of the properties of both the structures as whole and individual structural members can be expressed as:

$$F = f(p_m, p_c, p_s) \qquad (1)$$

where F is Objective Function, p_m is material properties, p_c is connection characteristics, p_s are structural characteristics. The general form of structural optimization can be expressed as minimize

$$F = f(p_m, p_c, p_s) \qquad (2)$$

$$g_1 \geq 0,\ g_2 \geq 0,.... g_n \geq 0 \qquad (3)$$

where g_1, g_2, g_3 and g_n are the constraint functions minimize

For example, when the structural weight is the only term in the objective function and is subjected to stress, displacement and fabrication constraints. The optimization problem can be expressed as:

minimize

$$F = \sum_e^n \rho_e L_e A(\eta_e) \qquad (4)$$

subject to

$$F = \begin{cases} s^l \le s \le s^u \\ d^l \le d \le d^u \\ A^l \le A \le A^u \end{cases} \qquad (5)$$

where ρ_e and L_e are material density and the length element e. $A(\eta_e)$ is index of the cross-sectional area of the element e dan n is the total number of elements. The vector *s, d, A* contains value of stress, displacement, and cross - sectional area. The superscripts refer to the prescribed lower and upper boundaries of each constraint.

2. Penalty Function

To evaluate the performance of fitness of particular solution string, the string's characters are decoded into values of the design variables. Using these design variables an analysis is performed and a value is computed for the objective function. If any constraints are violated a penalty is applied to the objective function with the value of the objective function. In general there are two possibilities to combine objective function with penalty function. The first is by adding a penalty function with the objective function in equation form as follow [5]

$$eval_{(x)} = f_{(x)} + p_{(x)} \qquad (6)$$

Where x is the chromosome $f_{(x)}$ is the objective function and $p_{(x)}$ is penalty function. The second form of evaluation function is to multiply the penalty function with the objective function. So that the evaluation form are as follows [5]

$$eval_{(x)} = f_{(x)} p_{(x)} \qquad (7)$$

Most of the penalty function depends on the parameters contained in each problems. So the penalty function tends to change according to the existing problems. On structural optimization penalty function is commonly composed of control the stress of bar and control the deformation of each node that exists. Forms of penalty functions can be shaped

$$h_{(x)} = C_1 \sum_i^n g_{1i} + C_2 \sum_i^n g_{2i} \qquad (8)$$

where,

C_1, C_2 = Pinalty Coefisient
g_{1i} = The function of cross-sectional analysis of each elements
If $\sigma_{allowable} > \sigma_i$ than $g_{1i} = 0$
$\sigma_{allowable} < \sigma$ than $g_{1i} = \sigma - \sigma_{allowable}$
g_{2i} = The Function of displacement of analysis of each nodal
If $\Delta_{allowable} > \Delta$ than $g_{2i} = 0$
$\Delta_{allowable} < \Delta$ than $g_{2i} = \Delta - \Delta_{allowable}$
n = number of elements

II. METHOD

In the first case as shown in the Figure 1, the structure will be optimized weight and nodal deformation of truss elements with 15 elements. All elements have a density of 7850 kg/m³, Modulus of elasticity of 200 MPa and the cross-sectional are the profile provided angle section on the market. A vertically downforce of 15 KN is given to the vertical direction at node 5. Node 1 and 6 are the supports, of which node 1 is roller (having only horizontal reaction) and node 6 is a pin support (having horizontal and vertical reaction) . There are three type of grouping of element A1, A2, A3. The stress of steel material used is f_y = 240 MPa and f_u = 370 MPa.

A. Fitness Function

There are two objectives in the current optimization problem, minimizing the total weight of the structure and the deflection at node 5 (which can be seen will always have the highest vertical deflection). The design variables considered are the member cross-sectional areas and the nodal coordinates of the free nodes. The constraints here is that the axial stresses in no member should exceed the allowable stress. Since there are two objectives a weighting method is used to transform the simple GA to a multi-objective solver where a composite fitness function of the weighted sum of the objectives is assigned to each individual of the population as shown below :

$$f_{(x)} = \sum_i^n \rho A_i L_i + 100 g_1 + 100 g_2 \qquad (8)$$

where

ρ_i and L_i = material density and the length element i.
A_i = the cross-sectional area of the element i
N = the total number of elements
g_1 = the sum of the penalty function of each element of the tensile elements and compression elements

The equation for tensile element

$$\sum_i^n slenderness_i + \sum_i^n tensionyield_i + \sum_i^n tensionultimate_i$$

if $\lambda_{max} < 240$ than slenderness$_i = 0$

$\lambda_{max} > 240$ than slenderness$_i$ = 1

if $P_u \le \phi A_g F_y$ than tensionyield$_i = 0$

$P_u > \phi A_g F_y$ than tensionyield$_i$ = 1

if $P_u \le \phi A_e F_u$ than tensionultimate$_i = 0$

$P_u > \phi A_e F_u$ than tensionultimate$_i$ = 1

• Compression Section

$$\sum_i^n compressionstress_i + \sum_i^n slenderness_i + \sum_i^n section_i$$

if $\dfrac{b}{t} < \dfrac{250}{\sqrt{f_y}}$ than section$_i = 0$

$\dfrac{b}{t} > \dfrac{250}{\sqrt{f_y}}$ than section$_i$ = 1

if $\lambda_{max} < 200$ than slenderness$_i = 0$

$\lambda_{max} > 200$ than slenderness$_i$ = 1

if $P_u \le \phi \dfrac{A_g f_y}{\omega}$ than compressionstress$_i = P_u > \phi \dfrac{A_g f_y}{\omega}$

than compressionstress$_i$ = 1

g_2 : the sum of displacement control function of each nodal

$$\sum_i^n deflection_i$$

if $\Delta_{allowable} > \Delta$ than deflection$_i = 0$

$\Delta_{allowable} < \Delta$ than deflection$_i$ = 1

B. Program and Logic

The logic of the program flows as shown below:
1. Generate population of strings.
2. Map string elements of each string of the population to member area or nodal coordinate. This will be input to the truss solver.
3. Run truss solver to obtain maximum vertical deflection, total weight and maximum stress.
4. Calculate the fitness from the maximum deflection and total weight.
5. Perform tournament selection (size 2) without replacement.
6. Perform two-point crossover without replacement and simultaneous mutation.
7. Repeat step 2 to 6 till termination criteria is satisfied.

C. Parameter Setting

In order to get a competent GA that scales well, we have to get a proper combination of the GA parameters i.e. the population size N, probability of mutation p_m, and probability of crossover p_c. In this study, only population size and probability of mutation were varied to examine their effect on the rate of convergence. Population size of the GA should be large enough so that a significant amount of the parts that form the optimal solution are accounted for somewhere in the population. If population size is too small there will be genetic drift and the GA will converge to a non-optimal solution. But if the population size is too large the computation time will be compromised. Figure 2 shows the convergence of the GA for population sizes of 10, 20, 100, 500 and 1000. The other parameters were kept constant for these computations, t = 50 p_c = 0.8 and p_m = 0.07. For remaining computations, a population size of 300 was decided upon observing the results of Figure 2.Another set of experiments were conducted for the following p_m values: 0, 0.001, 0.01, 0.05, 0.07 and 0.1 (N = 100, p_c = 0.8, t = 100) and the results are shown in Figure 3. From Figure 3 it can be seen that when no mutation is introduced initially there is high convergence but after the initial convergence there is no search for better the solutions. Those individuals in the population are the same and crossover does not produce anything new. Thus, there is a need for diversity which caused about by mutation. However the amount of mutation should be carefully chosen, though responsible for maintaining the local diversity, is also an inherently disruptive force [3] as shown in Figure 3 for p_m =0.1. From Figure 3. It can be seen that p_c value between 0.01 and 0.07 provides a balance between maintaining diversity and causing disruption in the population and for remaining computations a p_m value of 0.07 was chosen.

III. RESULTS AND DISCUSSION

After getting objective function for optimization of cantilever truss then the next step is to perform the optimization with GA parameters such as the follows, the number of population is 50, the number of generations is 300, value of crossover probability is 0.8, mutation probability is 0.07. The above parameters are used for both optimization of cantilever truss. From the analysis for sizing optimization the weight of structure is 61.71 kg and the maximum deformation is 5.9 mm this result shows in Table 1 and the convergence of sizing optimization shows on Figure. 4. While for the combined sizing and configuration optimization can be shown as follows. From the analysis for sizing and configuration optimization the weight of structure 55.37 kg and the maximum deformation 3.7 mm as shown in table 2 and the convergence of sizing and configuration optimization shows on Figure 5. The optimum shape and the nodals coordinate of truss can be seen on Figure 6.

IV. CONCLUSION

Multi-objective sizing and configuration optimization of the truss shown in Figure 1 has been conducted using a genetic algorithm. Some preliminary runs of the GA were conducted to determine the best combinations of GA parameters such as population size and probability of mutation so as to get better scaling for rest of the runs. Comparing the results from sizing and sizing–configuration optimization, we find there is a significant reduction in the weight and deflection. Sizing–configuration optimization produces lighter weight and small displacement than sizing optimization. The results were obtained using a GA with relative ease (computationally) and these results are very competitive compared to those obtained from other methods of truss optimization. Although mathematically obtained under optimum conditions, but if applied in the construction becomes less prevalent, especially from the aesthetic aspects.

REFERENCES

[1] R.H. Gallagher, and O.C. Zienkiewicz, "Optimum structural design: Theory and applications", John Wiley and Sons, 1973.
[2] F.S. Hillier and G.J. Lieberman, "Introduction to mathematical programming", McGraw–Hill Publishing Company, 1990.
[3] J.H. Holland, "Adaptation in natural and artificial systems", Ann Arbor : The University of Michigan Press, 1975.
[4] S. Rajeev, and C.S. Krishanmoorthy, "Discrete optimization of structures using genetic algorithms", ASCE Journal of Structural Enginering, Vol. 118, No. 5, pp. 1233-1250, 1992.
[5] D.E. Goldberg, "Genetic algorithms in search, optimization, and machine learning", Addison Wesley, 1989.
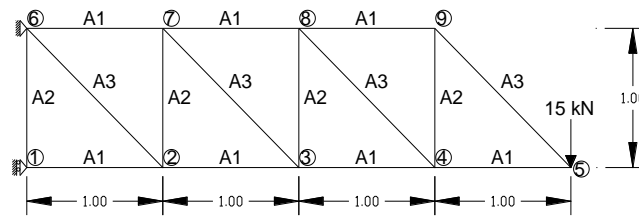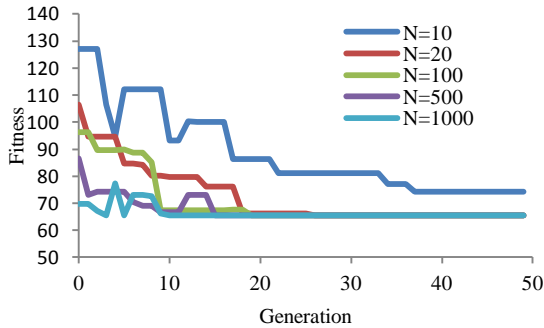
Figure 1. Truss model with 15 elements

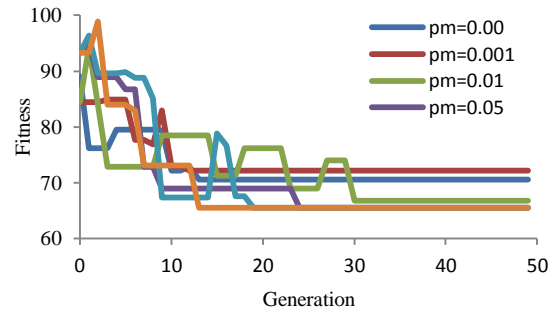Figure 2. Plot showing convergence of the GA for different population size

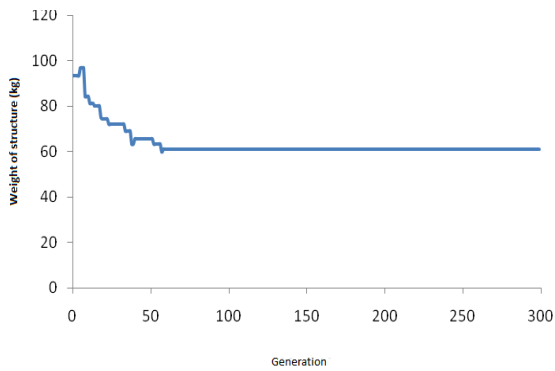Figure 3. Plot showing convergence of the GA for different p_m (probability mutation)

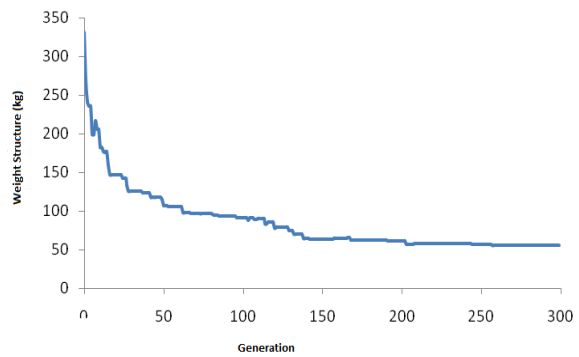Figure 4. Sizing optimization convergence with genetic algorithm

Figure 5. Sizing and configuration optimization convergence with genetic algorithm
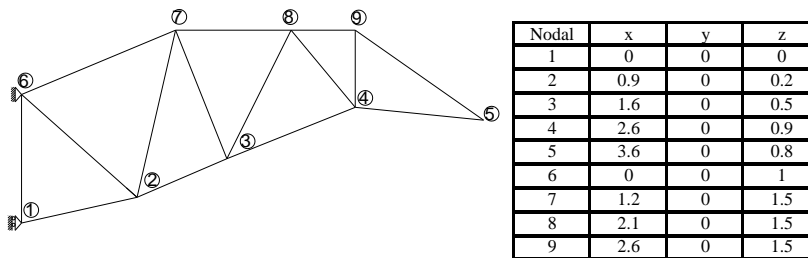
Nodal	x	y	z
1	0	0	0
2	0.9	0	0.2
3	1.6	0	0.5
4	2.6	0	0.9
5	3.6	0	0.8
6	0	0	1
7	1.2	0	1.5
8	2.1	0	1.5
9	2.6	0	1.5

Figure 6. Nodal coordinate and shape of truss after combined size and configuration optimization

TABLE 1.
THE RESULT AFTER SIZING OPTIMIZATION

Group Sectional Area (mm²)			Weight	Maximum Deformation
A1	A2	A3	Kg	mm
L 55x55x6	L 40x40x4	L 40x40x5	61.71	5.90
631	308	379		

TABLE 2.
THE RESULT AFTER SIZING AND CONFIGURATION OPTIMIZATION

Group Sectional Area (mm²)			Weight	Maximum Deformation
A1	A2	A3	Kg	mm
L 55x55x6	L 40x40x4	L 40x40x5	55.37	3.70
631	379	430		