Journal of Applied Mathematics and Physics, 2024, 12, 2885-2903
https://www.scirp.org/journal/jamp
ISSN Online: 2327-4379
ISSN Print: 2327-4352

Genetic Optimization via Diverse Crossover Intelligence

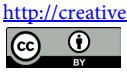
David Webb[1], Eric Sandgren[2*]

[1]AdvanSix Inc., Parsippany, New Jersey, USA
[2]Systems Engineering Department, Donaghey College of Engineering & Information Technology, University of Arkansas at Little Rock, Little Rock, Arkansas, USA
Email: david.j.webb@comcast.net, sandgreneric@gmail.com

How to cite this paper: Webb, D. and Sandgren, E. (2024) Genetic Optimization via Diverse Crossover Intelligence. *Journal of Applied Mathematics and Physics*, **12**, 2885-2903.
https://doi.org/10.4236/jamp.2024.128172

Received: July 24, 2024
Accepted: August 19, 2024
Published: August 22, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/



Open Access

Abstract

An intelligent crossover methodology within the genetic algorithm (GA) is explored within both mathematical and finite element arenas improving both design and solution convergence time. This improved intelligent crossover outperforms the traditional genetic algorithm combined with a rule-based approach utilizing domain specific knowledge developed by Webb, *et al.* [1]. The encoding of the improved crossover consists of two chromosome strings within the genetic algorithm where the first string represents the design or solution string, and the second string represents chromosome crossover string intelligence. This improved crossover methodology saves the best population members or designs evaluated from each generation and applies crossover chromosome intelligence to the best saved population members paired with globally selected parents. Enhanced features of this crossover methodology employ the random selection of the best designs from the prior generation as a potential parent coupled with alternating intelligence pairing methods. In addition to this approach, two globally selected parents possess the ability to mate utilizing crossover chromosome string intelligence maintaining the integrity of a global GA search. Overall, the final population following crossover employs both global and best generation design chromosome strings to maximize creativity while enhancing the solution search. This is a modification to a conventional GA that can be translated into GA encoding. This technique is explored initially through a Base 10 mathematical application followed by the examination of plate structural optimization considering stress and displacement constraints. Results from crossover intelligence are compared with the conventional genetic algorithm and from Webb, *et al.* [1] which illustrates the outcome of a two phase genetic optimization algorithm.

*Retired.

Keywords

Crossover, Topological Design, Structural Optimization, Genetic Optimization, Variable Material Design

1. Introduction

Solution convergence and result quality within a genetic algorithm are limited to the variety of new favorable offspring that can be generated following chromosome crossover between two parents. This limitation in offspring diversity significantly hinders solution convergence and ultimately an optimal design. Traditional methods for introducing new offspring are noted by Gen and Cheng [2] particularly point source crossover, which selects a random point within the chromosome string as the staring point for crossover of traits between parent strings. Alternative approaches to traditional crossover techniques are referenced by [3] and [4]. Inefficiencies within a traditional genetic algorithm are attributed to a heavy reliance on a random search where mutation becomes a primary contributor toward convergence. To continue the progression of the performance of the genetic algorithm, enhancing the chromosome crossover intelligence between parents is essential to desirable quality and efficiency of the solution search.

The encoding of the improved crossover methodology consists of two chromosome strings within the genetic algorithm where the first string represents the design or solution string, and the second string represents chromosome crossover string intelligence. Each population member possesses an assigned design and crossover chromosome string. This novel crossover approach saves the best population members or designs evaluated from each generation and applies crossover chromosome intelligence to the best saved population members paired with genetic algorithm selected parents. The selected parent and best generation design are mated based on the instructions presented within the encoded crossover chromosome string. In addition to this approach, two genetic algorithm selected parents possess the ability to mate utilizing crossover chromosome string instructions maintaining the integrity of a global search. When the best design or chosen parent is mated with an assigned parent, any difference within each parent's crossover string value results in the crossover of design string characteristics and the crossover string chromosome binary value. If no difference is present between crossover strings, then a chromosome string characteristic exchange does not occur within either the design or crossover instruction string. Enhanced features of this crossover methodology employ the random selection of the best two designs from the prior generation as a potential parent coupled with alternating intelligence pairing methods during even and odd generations. Odd generations employ the best design from the prior generation to be mated with a genetically selected parent while even generations utilize parents selected by the genetic algorithm within the current population. The final population

following crossover offers a diverse population of chromosome strings obtained through global and local refinement of the best generational design chromosome strings improving solution quality and convergence time. This intelligent crossover methodology is demonstrated through a comparison approach within a mathematical example and a plate design as referenced by Webb *et al.* [1], however this novel concept can be applied to an array of optimization problems for further solution enhancement while reducing computation time.

2. Background

Base 10 Mathematical Optimization

A mathematical example is explored illustrating the performance of both a conventional genetic algorithm with the incorporation of crossover intelligence. This mathematical concept involves the basic calculation of a seven digit Base 10 result rendering an optimal solution of 9,999,999. The aim of this genetic algorithm search is to locate the final solution possessing minimal convergence time. The intent of this example is to provide a lead in exercise focused on the improved methodology prior to revisiting a structural optimization example leveraging commercial finite element analysis as a solving tool.

Plate Structural Optimization

As taken directly from Webb, *et al.* [1], the plate optimization procedure considered here operates through the assignment of different material properties throughout a predefined meshed design region. In order to simplify the concept, only two different material property values are introduced. For a plate design, the logical material property to consider is the modulus of elasticity. The first material property value represents the intended design material such as that for steel or aluminum. The second material property value corresponds to an extremely weak material which adds little to the structural integrity of the design. The utilization of a weak material allows for extreme topological change to occur without requiring a re-mesh for the elements and eliminates the problem of generating a singular stiffness matrix. The stiffness matrix for the finite element analysis of the design is assembled with material property information provided by the genetic encoding on an element by element basis. The final topology of the design may be identified by simply removing the elements formed of the weak material. This is not intended to provide a final, highly detailed design, but rather a topology which may be refined in order to generate such a design.

The plate design process investigated incorporates a genetic algorithm programmed in Microsoft Visual Basic [5], Microsoft Excel Visual Basic for Applications (VBA) [6], and CAEFEM [7], a finite element solver which performs a linear static analysis for each design topology considered. The plate design mesh is constructed using Femap [8], a three dimensional CAD package, which generates a neutral file that the finite element solver can read and evaluate. This neutral file [9] documents the geometry as well as all of the material properties and other parameters of each plate design investigated. The genetic algorithm gener-

ates designs through the alteration of the material property of each plate element within the selected neutral file. The alteration of material properties can generate a realm of diverse topological designs, where each design is processed by altering the neutral file and passing it on to the finite element solver. The element stresses and the nodal deflections are computed by the finite element solver and subsequently, this information is utilized to evaluate the objective function and constraints which are defined by the optimization formulation. The specific form of objective function in this application was chosen to be the minimization of the total volume of a plate based design subject to predefined loading and constraint conditions. Stress and displacement limits served as structural constraint bounds.

A major issue of concern in the assignment of an alterable material property for each element in the defined mesh is the total number of elements considered. As the number of elements increases, the complexity as well as the time required for solution by the optimization algorithm increases significantly. On the other hand, the number of elements required to define the fundamental design topology will generally be considerably below that required for a detailed design analysis. The fundamental question which must be addressed is whether a rule based genetic algorithm is capable of solving a realistic structural design problem with a reasonable amount of computational effort. The process is inherently parallel in nature which could lead to significant reduction in computational time but not in computational effort. Extensions to problems involving three dimensional solid objects are straightforward, using the same design optimization strategy with an expanded genetic encoding.

3. Base 10 Problem Formulation

Base 10 problem formulation consists of an objective function to maximize the mathematical value associated with an optimal solution of 9,999,999. Two methods for calculating base 10 results were explored which consisted of traditional genetic algorithm encoding for global parent selection and crossover of a single chromosome string per parent. The chromosome string value ranges from 0 to 9 and possesses 7 string values. Parent selection is conducted and traditional point crossover of chromosome string values between Parents occurs resulting in two base 10 offspring.

Base 10 crossover intelligence utilizes two chromosome strings where the first string represents a mathematical 7 digit base 10 representation string and the second string represents crossover intelligence to further enhance the solution and convergence timing.

An example of genetic algorithm intelligent crossover is outlined below per the 7 digit Base 10 problem formulation which results in offspring of both solution and corresponding crossover strings.

Sample parent one solution string {6, 7, 8, 4, 4, 1, 5} mathematically represents 6,784,415 where the total string length correlates to the optimal 7 digit Base 10 solution.

Sample crossover string {2, 2, 1, 2, 1, 1, 2} correlates to parent one sample solution string, where each chromosome value possesses the ability to represent value of 1 or 2 for crossover instruction. The sample string length corresponds to the total length of the parent one solution string which provides chromosome crossover instruction for each string member.

Sample parent two solution string {6, 8, 3, 9, 3, 6, 7} mathematically represents 6,839,367 and is the remaining parent selected to mate with the parent one solution string.

Sample crossover string {1, 1, 1, 1, 2, 2, 2} correlates to parent two sample solution string, where each chromosome value possesses the ability to represent value of 1 or 2 for crossover instruction. The sample string length corresponds to the total length of the parent two solution string which provides chromosome crossover instruction for each string member.

Differences between crossover strings result in the crossover of both solution and crossover string characteristics.

Below are solution and crossover strings prior to intelligent crossover operations.

{6, 7, 8, 4, 4, 1, 5} {2, 2, 1, 2, 1, 1, 2}

{6, 8, 3, 9, 3, 6, 7} {1, 1, 1, 1, 2, 2, 2}

Following intelligent crossover, below are offspring solution and crossover strings.

{6, 8, 8, 9, 3, 6, 5} {1, 1, 1, 1, 2, 2, 2}

{6, 7, 3, 4, 4, 1, 7} {2, 2, 1, 2, 1, 1, 2}

Key benefits include the selection of parent solution and crossover strings from alternating generations where parent selection is either from the best solutions from the prior generation or conventionally selected parent strings for mating.

4. Two Phase Plate Optimization Problem

The aim of this initiative is to examine plate design through the maximization of volume as depicted in Equation (1) when subjected to stress and displacement constraints within Equation (2) utilizing a two phased optimization methodology. As taken directly from Webb, *et al.* [1], two separate genetic optimization formulations are introduced. The first represents a traditional encoding, where each element in the defined structural region has its own binary variable in the encoding string. This formulation is developed in this section. The development of the rule based extension follows in the next section. A one hundred forty element plate design example is investigated which utilizes a conventional GA phase one search and a phase two rule based approach. Phase two incorporates domain specific knowledge for the refinement of the phase one solution. This two phase approach is compared to an intelligent crossover methodology, where both optimization strategies are subjected to identical stress and displacement constraint conditions as outlined within this section. Additionally, the plate de-

signs were also restricted to the same spatial design. The dimensions of the spatial design volume consisted of eighteen cm in the x direction, thirteen cm in the y direction, and a thickness of 0.5 cm in the z direction. Directional forces consisting of 67 newtons in the x, y and z directions were applied at coordinate locations (15, 3.25, 0.5) and (12, 9.75, 0.5) where the units are in cm and the origin is at the lower left corner of the bottom surface of the specified plate design volume. The use of multidirectional forces allows for an optimal yet robust topological design to be acquired. The concept of robust design optimization has been effectively demonstrated by Sandgren and Cameron [10] for truss structures as well as an automotive inner body panel. Mesh size for plate structural optimization consisted of one hundred forty elements. The maximum allowable stress constraint was set at 140 MPa and a maximum allowable displacement constraint was defined to be 0.635 cm. The objective function used to evaluate each design is based on the amount of volume removed from the original mesh. This objective function value is equal to the total element volume of weak material present in each design as specified by the genetic algorithm encoding.

The formulation of the objective function and constraints for the plate design examples considered is as follows:

$$\text{Maximize volume} = \sum\left(\text{Volume}_i\right) * \left(\text{Encoding value}_i\right), \tag{1}$$

$i = 1, \ldots,$ number of elements

subject to

$$g_i\left(x\right) = \left(\frac{\text{Constraint}_{\text{limit}} - \text{Constraint}_{\max}\left(\text{Calculated}\right)}{\text{Constraint}_{\text{limit}}}\right) \geq 0 \tag{2}$$

where $i = 1\ldots$ number of constraints.

$g_1\left(x\right)$ correlates to stress and $g_2\left(x\right)$ displacement.

The encoding value for each element has a value of zero if the element is assigned to the design material and one if it is assigned to the weak design material. This way, the summation of volume to be maximized in Equation (1) consists of the total volume of weak material elements, or alternatively, the volume of material removed. In the constraint equations, S_{limit} and D_{limit} are the maximum allowable stress and displacement values while $S_{\max}$ (calculated) and $D_{\max}$ (calculated) are the maximum computed values for the design being analyzed over all of the elements considered. The formulation of Equation (2) generates ratios, which allow for normalization of constraint violation in the disparate magnitudes of the displacement and stress values as well as to allow for a consistent graphical representation over the diverse set of designs within each genetic population. Both constraint equations produce positive values for any design which does not exceed the design limits.

Material properties of each plate element were designated to be either the intended design material, AISI 4340 steel, or a substantially weaker material with a modulus of elasticity of 2 MPa. Additionally, Poisson's Ratio was assigned a value of 0.32 for all elements in each plate design investigated. The modulus for the

weak material may be defined as any reasonably small value as long as it is significantly less than that of the selected design material. The main function of this material property is to prevent the formulation of a singular stiffness matrix within the finite element algorithm and to avoid the necessity of re-meshing the design region. The assumption is made that low stress magnitudes in plate elements assigned the weak material property are deemed to be practically nonexistent. Steel material properties were represented by the encoding value of 0, while weak material properties were assigned the encoding value of 1. These assigned numeric values correspond to defined material properties generated within Femap which are encoded within the genetic algorithm for material property manipulation of each element. A sample string is shown below for a ten element plate.

Sample string {0, 1, 1, 0, 0, 0, 1, 0, 0, 1}

For this string, elements one, four, five, six, eight and nine are formed from steel, while elements two, three, seven and ten are formed from the weak material. The assignment of order in the encoding string is directly related to the element number in the design mesh. Since the design mesh remains constant throughout the optimization process, the encoding strategy is valid throughout the search.

Based upon the structure of the sample string, an initial set of designs is generated randomly and the genetic algorithm methodology begins. Both crossover and mutation operations are applied by the genetic algorithm, and ultimately new generations of offspring or designs are formed. The method of crossover involves the selection of two parent designs where material properties or chromosomes within each chosen design string are interchanged at a random position, forming two new offspring designs. Parent designs are selected randomly, based on their overall fitness value, which considers the value of the objective function as well as any constraint violation. A single numeric value for each design is formed through the use of an exterior penalty function. Using the penalty function allows parent design strings which meet or exceed predefined constraint values to be assigned an overall fitness value solely based upon the objective function value. Parent designs which fail to meet the constraint criteria, are reduced in value by a penalty function.

A typical penalty function is represented by the equation

$$P(x) = f(x) - R\sum_i \{g_i(x)\}^2 \tag{3}$$

for $i = 1, \ldots,$ number of constraints

where

$\{g_i(x)\} = 0$; for $g_i(x) \geq 0$

$\{g_i(x)\} = g_i(x)$; for $g_i(x) < 0$.

In this equation, $f(x)$ represents the objective function value at the design point x, which specifies how much material was removed from the spatial design region. The penalty factor, R is a numeric value which continually increases from an initial value by a user specified amount as the optimization proceeds. This allows an early exploration phase where some constraint violation is al-

lowed, but builds a penalty over time so that later designs are pulled toward the feasible design region. Lastly, the array $g_i(x)$ represents the constraint value for each of the defined constraints. Additional background on penalty function theory and operation is provided by Gen and Cheng [2]. The penalty function or specific fitness value for each population member determines which members of the population are best suited for producing new design offspring, resulting in the term parent designs. An in depth discussion of parent selection as well as the genetic optimization process is given by Goldberg [11] and Davis [12].

Once the crossover of traits between parent designs has been completed, stresses and displacements for each of the offspring designs are calculated and evaluated in the stress and displacement constraint functions. Element stresses and nodal displacements are immediately available once the finite element analysis algorithm, CAEFEM, has completed the analysis. Now the evolutionary process has commenced. This evolutionary process begins with the selection of new parent designs from the current generation. Once all parent designs have bred new offspring, one generation has been completed. The total number of generations of offspring produced is a user specified variable, which is application specific. The effectiveness of a specified set of input parameter values may be observed through feedback from the graphical user interface.

5. Plate Optimization Problem Formulation Phase Two

In this example, a two phase optimization approach was explored utilizing a one hundred forty element plate for comparison purposes. This two phase methodology is a novel artificial intelligence approach as referenced by Webb, *et al.* [1]. In phase two, domain specific knowledge was injected into a genetic algorithm to further drive convergence. This level of artificial intelligence drives input in the form of how many rules were utilized toward the formulation of an optimized solution.

6. Plate Optimization Problem Utilizing Intelligent Crossover

Plate optimization is further explored through the implementation of intelligent crossover to further improve the diversity of offspring population members to drive objective function value convergence per defined constraint criteria while accelerating convergence time. This plate example explores the optimization of the one hundred and forty element plate as previously constructed by Webb *et al.* [1]. A comparison of results will be explored for the one hundred forty element plate mesh between the two phase optimization approach and intelligent crossover.

An example of genetic algorithm intelligent crossover for plate optimization is outlined below which results in the offspring of both design and corresponding crossover strings.

Sample design parent one string {0, 1, 1, 0, 0, 0, 1} represents two possible material properties each plate element can possess. The length of this string is

proportional to the amount of plate elements derived from a finite element mesh. A value of 0 represents a viable design material and the value of 1 represents a substantially weaker material that can be interpreted as non-existent.

Sample crossover string {2, 2, 1, 2, 1, 1, 2} correlates to parent one sample solution string, where each chromosome value possesses the ability to represent value of 1 or 2 for crossover instruction. The sample string length corresponds to the total length of the parent one design string which provides chromosome crossover instruction for each string member.

Sample design parent two string {0, 1, 1, 1, 0, 1, 1} represents the remaining parent selected to mate with the parent one chromosome string.

Sample crossover string {1, 1, 1, 1, 2, 2, 2} correlates to parent two sample solution string, where each chromosome value possesses the ability to represent value of 1 or 2 for crossover instruction. The sample string length corresponds to the total length of the parent one design string which provides chromosome crossover instruction for each string member.

Differences between crossover strings result in the crossover of both design and crossover string characteristics.

Below are design and crossover strings prior to intelligent crossover.

{0, 1, 1, 0, 0, 0, 1} {2, 2, 1, 2, 1, 1, 2}

{0, 1, 1, 1, 0, 1, 1} {1, 1, 1, 1, 2, 2, 2}

Following intelligent crossover, below are offspring design and crossover strings.

{0, 1, 1, 1, 0, 1, 1} {1, 1, 1, 1, 2, 2, 2}

{0, 1, 1, 0, 0, 0, 1} {2, 2, 1, 2, 1, 1, 2}

Key benefits include the selection of parent design and crossover strings from alternating generations where parent selection is either from the best designs from the prior generation or conventionally selected parent strings for mating.

7. Base 10 Results

Intelligent crossover solution search scenarios versus a conventional GA crossover methodology for the 7 digit Base 10 mathematical problem are displayed in Table 1 listed below.

Table 1. Overview of population and generation sizes assigned Base 10 scenarios considered.

Base 10 Populations	Base 10 Generations
25	100
50	100
200	100
400	100

Comparison results are provided for both conventional and intelligent crossover approaches per the 7 digit Base 10 GA example. Figure 1 and Figure 2 illustrate comparison results across population sizes of 25 and 50 where Figure 1 depicts conventional GA crossover and Figure 2 portrays optimized GA intelligent crossover. Additionally, Figure 3 and Figure 4 illustrate crossover behavior for increased population sizes of 200 and 400 where Figure 3 depicts the conventional GA crossover and Figure 4 captures optimized GA intelligent crossover. A significantly improved solution convergence to the optimal value of 9,999,9999 was captured utilizing the intelligent crossover approach as illustrated in Figure 2 & Figure 4. The outcome of the conventional GA crossover possessing a population of 25 rendered only a final solution of 9,999,988 at generation 71 as illustrated in Figure 1. A converged solution for the conventional GA crossover approach occurred for a population size of 50, however convergence occurred at generation 58. As for the optimized GA crossover depicted within

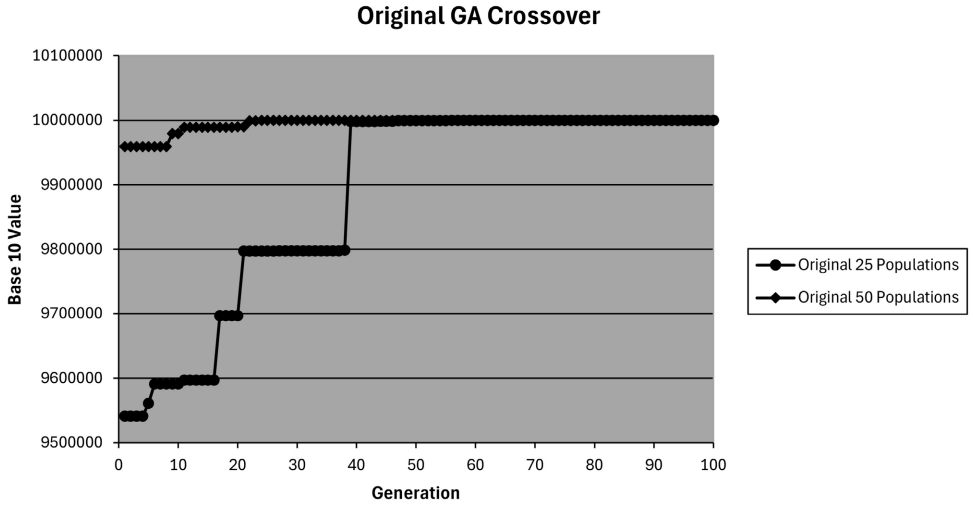


Figure 1. Conventional GA crossover (25 & 50).

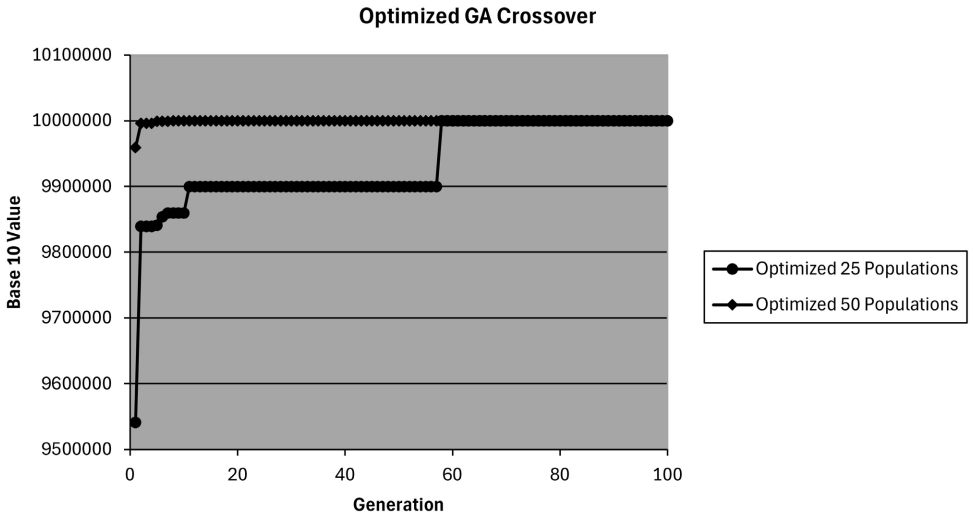


Figure 2. Optimized GA crossover (25 & 50).

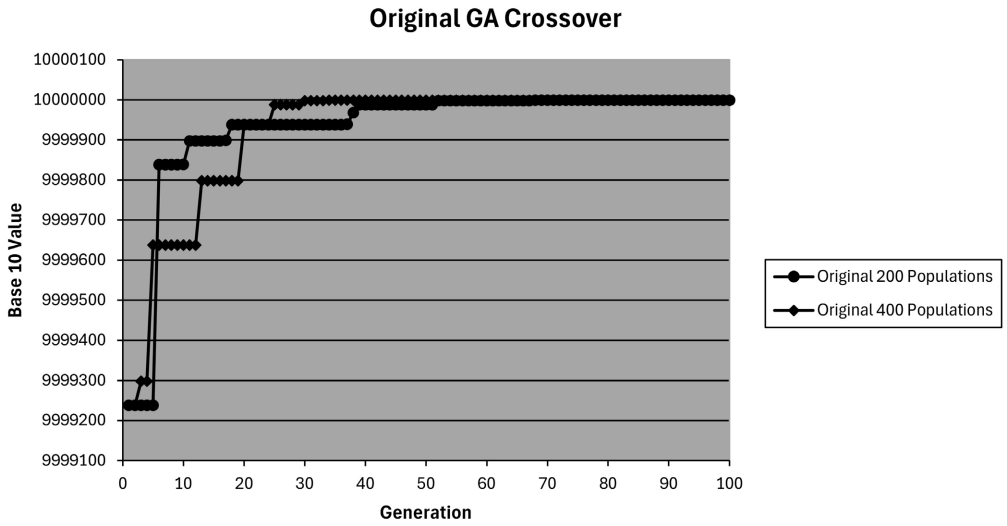


Figure 3. Conventional GA crossover (200 & 400).

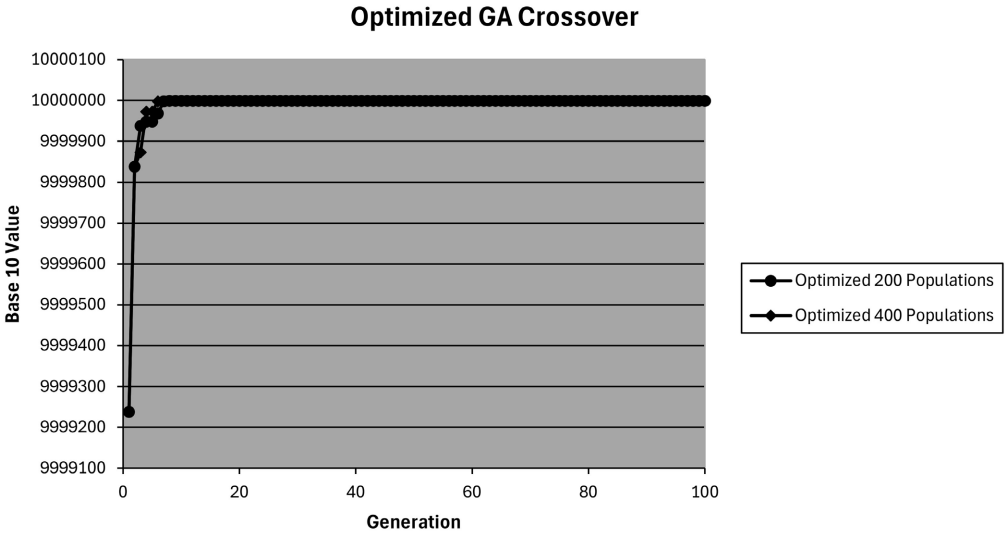


Figure 4. Optimized GA crossover (200 & 400).

Figure 2, solution convergence for the 25 and 50 population sizes occurred at generations 75 and 35 respectively which offers both an attained final converged solution with less required computation time.

Following the examination of the 7 digit Base 10 results for GA population sizes of 25 and 50, crossover behavior was further explored with increased population sizes of 200 and 400 spanning conventional GA crossover and optimized GA intelligent crossover as illustrated within Figure 3 and Figure 4. The conventional GA crossover rendered solution convergence for populations 200 and 400 within generations 68 and 34 respectively. Per an examination of the optimized GA intelligent crossover, significantly improved solution convergence to the optimal value of 9,999,999 was achieved at generation 8 for both 200 and 400 population values. Overall, the intelligent crossover method achieved the desired result of 9,999,999 for each population and generation example explored. Intel-

ligent crossover solution convergence accelerated significantly as population and generation sizes increased which enabled the creation of more diverse populations for intelligent parent and crossover selection. In comparison, the conventional GA is limited to offspring diversity and intelligent parent selection attributed to a heavy reliance on a random search where mutation becomes a primary contributor toward solution convergence. This limitation is particularly evident from the outcome of a population and generation size of 25 and 50 respectively where the desired result of 9,999,999 was unattainable.

8. Plate Optimization Results

Conventional GA Crossover via Two Phase Optimization

Specific input parameters utilized for the two phased optimization approach are documented in Table 2 which includes population size and number of generations of offspring for both phase one and phase two search processes.

Table 2. Overview of population and generation sizes.

Number of Plate Elements	Phase One Population Size	Phase One Generations	Phase Two Population Size	Phase Two Generations
140	280	140	100	40

The plate example utilized a one hundred forty element mesh to fill the designated design space with the predefined loading and constraint conditions established for all of the examples. The 0.5 cm. thick plate is fixed in all directions along the left edge, and two nodal forces were applied which consisted of sixty seven Newtons in the x, y, and z directions. A maximum stress constraint of 140 MPa and a maximum displacement constraint of 0.635 cm. were imposed as well. The phase one population size and number of generations produced were 280 and 140 respectively. The resulting plate design from the phase one search is illustrated in Figure 5. The dark shaded regions within Figure 5 represent the elements which were assigned the design material. The light regions indicate elements which were assigned the weak material and are treated as voids. The element numbering proceeds from left to right by rows, starting at the bottom row. The final objective function value for the design shown in Figure 5 was 88.59 cm^3. This represents the volume removed from the original design region.

The phase two search consisted of a population size and number of generations produced of 100 and 40 respectively. The phase two search refined the phase one design to that shown in Figure 6. This design can be seen to be a much "cleaner" design and contains no obvious design flaws other than the diagonal attachment of several elements. This problem could be resolved by adding appropriate constraints or by other means [2] [11]-[13]. This issue is not dealt with on the examples presented as only the design topology is desired and refinement of this topology is considered as a separate task. While the result pictured in Figure 5 from the phase one search contains elements that are be-

yond the point of force application, the phase two result does not contain such elements. The total volume removed for this design is improved to a total of 92.76 cm^3. Design topologies corresponding to phases one and two results are illustrated in Figure 5 and Figure 6. An examination of Figure 5 reveals that the phase one global search solution had several disassociated elements in the final design as well as elements beyond the load application points which are both indicators of the level of difficulty this size problem presents for a traditional genetic algorithm in locating a refined solution. Once again, the phase two solution remedies the design discrepancies and produces a fairly refined design.

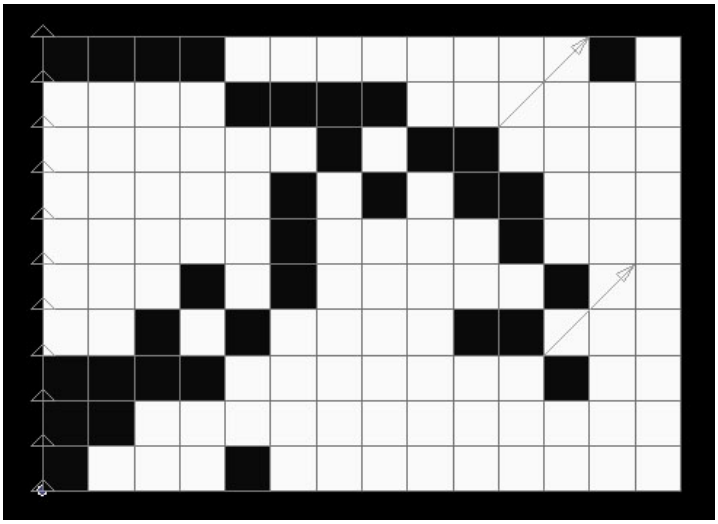


Figure 5. Phase one solution.

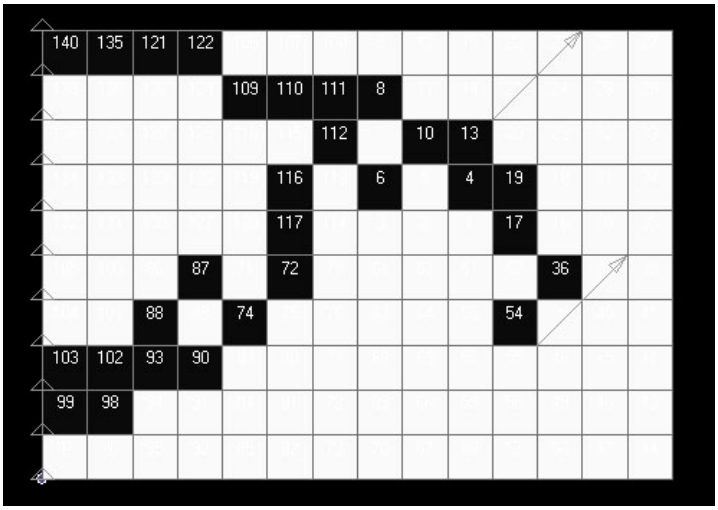


Figure 6. Phase two solution.

The behavior of the genetic algorithm is presented graphically for the phase one solution in Figure 7 and Figure 8. An objective function value of 88.59 cm^3 was located after approximately one hundred twenty generations of phase one operation as illustrated in Figure 7. The constraint value versus generation

graph displayed in Figure 8 reveals that the final phase one design satisfied both the stress and displacement constraint criterion.

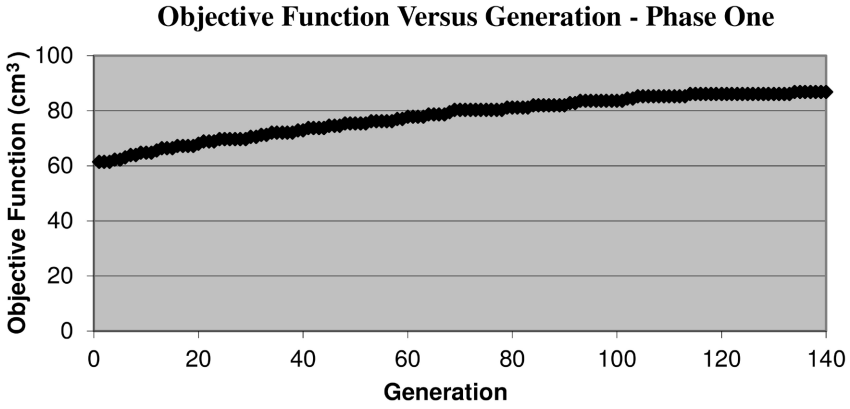


Figure 7. Objective function versus generation.

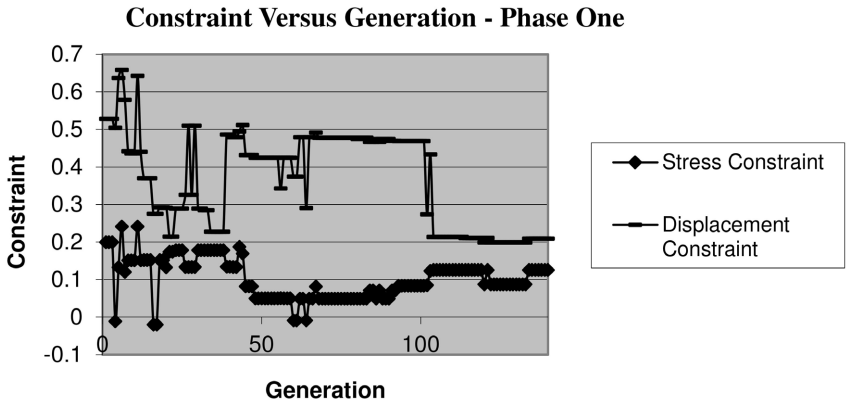


Figure 8. Constraint versus generation.

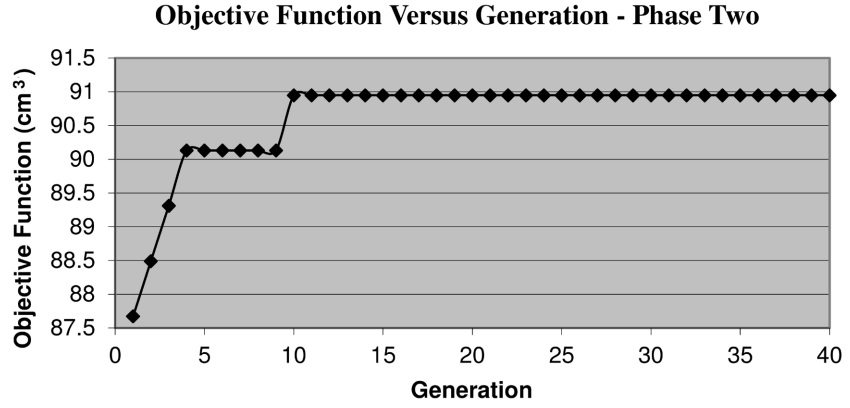


Figure 9. Objective function versus generation.

Graphical representations of the phase two search process are illustrated within Figure 9 and Figure 10. Ten generations were required to reach the final design solution. The objective function value was increased from 88.59 cm^3 to 92.76 cm^3 while maintaining acceptable stress and displacement design criterion.

Both displacement and stress constraints are shown to be satisfied throughout the phase two process in Figure 10.

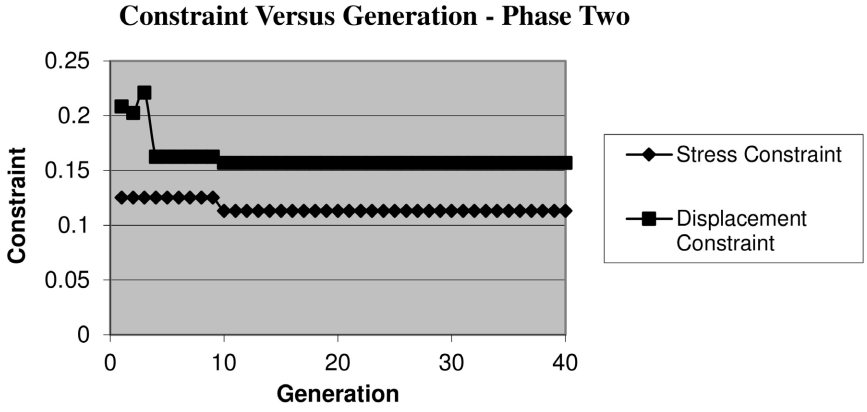


Figure 10. Constraint versus generation.

Intelligent GA Crossover

Crossover intelligence rendered the following results for the one hundred forty element plate utilizing Microsoft Excel Visual Basic for Applications (VBA) [6] where genetic parameters consisted of a population size of 100 with 100 generations. Figure 11 below illustrates the final intelligent crossover design and Figure 12 provides a comparative depiction of the final two-phase optimization approach derived by Webb, *et al.* [1] incorporating domain specific knowledge. Overall, the final intelligent crossover design utilized less material to satisfy both stress and displacement constraints resulting in an objective function of 96.68 cm^3 in comparison to the final two-phase optimization approach possessing an objective function value of 92.76 cm^3. Key benefits from intelligent crossover yield an improved objective function and corresponding design requiring fewer population and generation sizes rendering a reduction in computation time required for solution convergence. As a result of intelligent crossover, generation 86 yielded the most optimal design as reflected in Figure 13 while satisfying stress and displacement constraints as illustrated in Figure 14. Conversely, the best conventional GA phase one objective function was located at generation 120 resulting in a lower value of 88.59 cm^3 in comparison to intelligent crossover. Additional generations were required within phase two as depicted in Figure 9 to achieve an improved phase one conventional outcome of 92.76 cm^3, however the optimal phase two objective function value also remained lower in comparison to the intelligent crossover approach. Intelligent crossover results produced an improved objective function value which further maximized plate element volume in comparison to the two-phase optimization approach, and derived an alternative design as represented in Figure 11. Overall, the ability to leverage intelligent crossover to produce diverse global and local collective offspring enabled the GA to locate the most optimal design and corresponding objective function that meets prescribed constraints within an efficient computational timeframe.

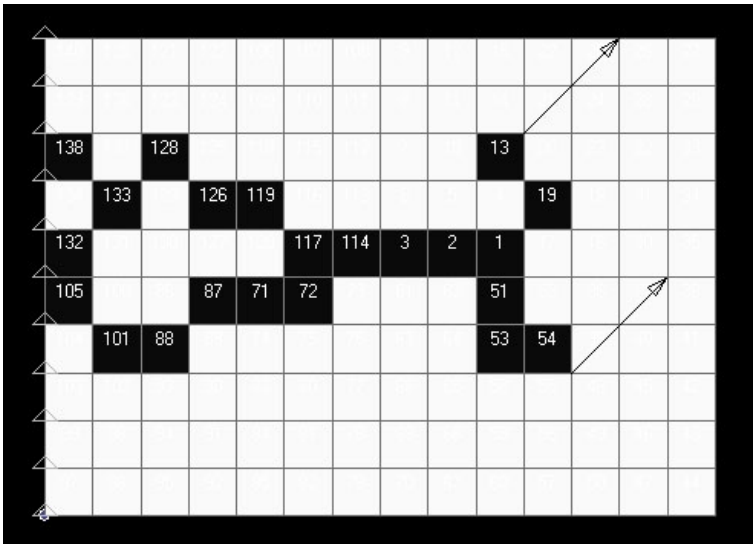


Figure 11. Intelligent crossover design.

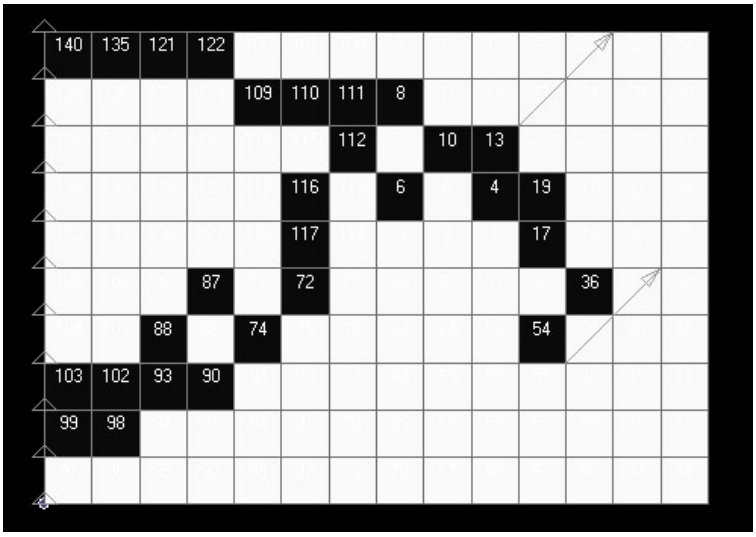


Figure 12. Phase two rule-based design.

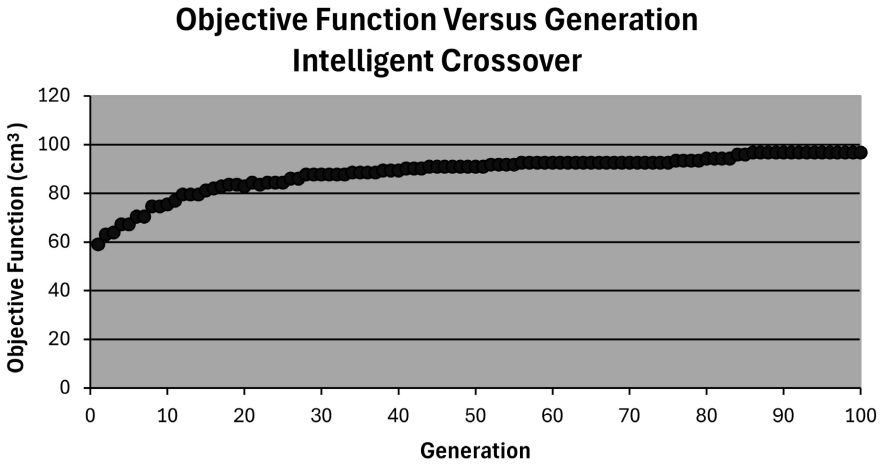


Figure 13. Objective function versus generation.

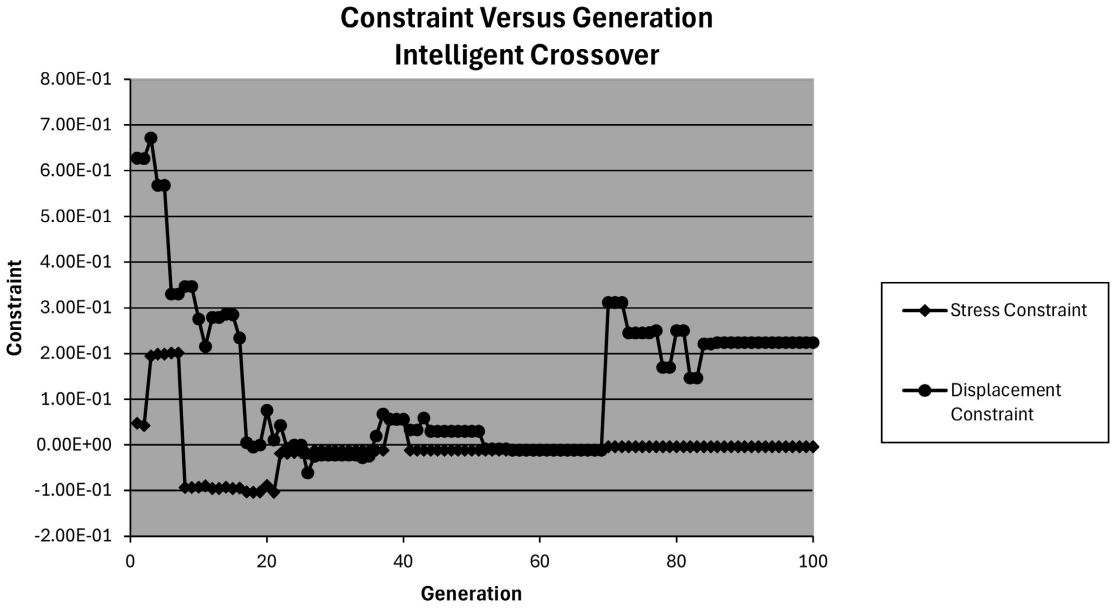


Figure 14. Constraint versus generation.

9. Extensions and Further Insight

The incorporation of intelligent crossover improves results unattainable by the two-phase optimization approach. Phase one utilizes the conventional genetic algorithm followed by a second phase incorporating domain specific knowledge to further optimize the outcome of phase one. The rule-based approach within phase two possesses limitations in comparison to the intelligent crossover method as the formulation of rules solely depends on user knowledge to enhance the phase one solution. Additionally, a significant limitation that inhibits the two-phase approach is diversity of the population members within the conventional GA. As generations progress within a conventional GA, population diversity becomes primarily driven by mutation rendering a sub-optimal solution for phase two to improve. The intelligent crossover provides instruction via a secondary chromosome string creating collective diverse populations driven globally and locally. Additionally, this secondary chromosome string experiences intelligent crossover enabling offspring to evolve and produce diverse populations to further enhance the objective function while satisfying associated constraints.

10. Summary and Conclusions

The conventional GA is a diverse optimization approach used to solve an array of problems [14]. The intelligent crossover concept has proven to be a novel approach to improving diversity within GA offspring and can be encoded into a conventional GA. Intelligent crossover rendered results unattainable by the conventional GA as reflected in the 7 digit Base 10 mathematical simulation. Additionally, this concept outperformed the two phase GA optimization approach. This two phase approach consisted of a conventional GA and a rule based methodology as reflected in plate design performance criteria outlined by Webb

et al. [1]. This plate design performance criteria includes the derived objective function value along with corresponding population and generation sizes utilized for the two phase simulation results. The diverse offspring generated from intelligent crossover resulted in the creation of an alternative plate design in comparison to conventional GA encoding while satisfying prescribed constraints and minimizing computation time. Additionally, intelligent crossover offers alternatives in parent selection which leverages artificial intelligence for the opportunity to choose the best saved parent designs or solution strings from the prior generation. Lastly, intelligent crossover occurs within both the design or solution string as well as the crossover string. This concept allows for evolving design strings and corresponding crossover decision points to ensure the continued progression of diverse offspring within the GA solution search. Overall, the intelligent crossover methodology can be incorporated into an array of combinatorial optimization problems particularly conventional GA attempted examples where alternative designs can be explored which possess historically extensive computational time.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Webb, D., Liu, Q., Alobaidi, W. and Sandgren, E. (2017) Topological Design via a Rule Based Genetic Optimization Algorithm. *American Journal of Computational Mathematics*, **7**, 291-320. https://doi.org/10.4236/ajcm.2017.73023

[2] Gen, M. and Cheng, R.W. (1997) Genetic Algorithms & Engineering Design. John Wiley & Sons.

[3] Malik, A. (2019) A Study of Genetic Algorithm and Crossover Techniques. *International Journal of Computer Science and Mobile Computing*, **8**, 335-344.

[4] Zainuddin, F.A. and Abd Samad, M.F. (2020) A Review of Crossover Methods and Problem Representation of Genetic Algorithm in Recent Engineering Applications. *International Journal of Advanced Science and Technology*, **29**, 759-769.

[5] Microsoft Corporation (1998) Microsoft Visual Basic Version 6.0.

[6] Microsoft Corporation (2022) Microsoft Excel Visual Basic for Applications.

[7] Concurrent Analysis Corporation (1998) Getting Started with CAEFEM Version 4.0.

[8] (1998) Power Solver FEMAP 5.0 CAEFEM 4.0, CD Version.

[9] (1996) Femap Neutral File Format. Enterprise Software Products Inc.

[10] Sandgren, E. and Cameron, T.M. (2002) Robust Design Optimization of Structures through Consideration of Variation. *Computers & Structures*, **80**, 1605-1613. https://doi.org/10.1016/s0045-7949(02)00160-8

[11] Goldberg, D.E. (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley.

[12] Davis, L. (1991) Handbook of Genetic Algorithms. Van Nostrand.

[13] Li, Q., Steven, G.P. and Xie, Y.M. (2001) A Simple Checkerboard Suppression Algorithm for Evolutionary Structural Optimization. *Structural and Multidisciplinary Optimization*, **22**, 230-239. https://doi.org/10.1007/s001580100140

[14] Najjarpour, M. and Jalalifar, H. (2018) Optimization of Fairhurst-Cook Model for 2-D Wing Cracks Using Ant Colony Optimization (ACO), Particle Swarm Intelligence (PSO), and Genetic Algorithm (GA). *Journal of Applied Mathematics and Physics*, **6**, 1581-1595. https://doi.org/10.4236/jamp.2018.68134