Struct Multidisc Optim 22, 57–64

Structural application of a shape optimization method based on a genetic algorithm*

S.Y. Woon, O.M. Querin and G.P. Steven

Abstract This paper presents a structural application of a shape optimization method based on a Genetic Algorithm (GA). The method produces a sequence of fixed-distance step-wise movements of the boundary nodes of a finite element model to derive optimal shapes from an arbitrary initial design space. The GA is used to find the optimal or near-optimal combination of boundary nodes to be moved for a given step movement. The GA uses both basic and advanced operators. For illustrative purposes, the method has been applied to structural shape-optimization. The shape-optimization methodology presented allows local optimization, where only crucial parts of a structure are optimized as well as global shape-optimization which involves finding the optimal shape of the structure as a whole for a given environment as described by its loading and freedom conditions. Material can be removed or added to reach the optimal shape. Two examples of structural shape optimization are included showing local and global optimization through material removal and addition.

Key words structural optimization, genetic algorithm

1 Introduction

The Genetic Algorithm (GA) method is a stochastic search method that was first derived by Holland (1975) and then extended by Goldberg (1989). A GA is based on evolution and genetics, and exploits the concept of survival of the fittest. For a given problem or design domain of significant complexity there exists a multitude of possible solutions that form a solution space. In a GA, a highly effective search of the solution space is performed by allowing a population of traditionally binary mathematical strings representing possible solutions to evolve through the basic random operators of selection, crossover and mutation. Over generations, the population evolves into an increasingly fit and convergent state. Survival of the fittest is implemented through biased selection schemes which ensure that characteristics of fit members have a significantly higher probability of propagating from one generation to another. Crossover involves swapping some corresponding allele values between two mathematical strings at some high probability, hence emulating the biological operation of gene swapping. In a GA, this is traditionally viewed as the main operator that efficiently exploits the solution space information inherent in a population of strings. Mutation is a random operator that expands the search space beyond that initially represented by the population. Simulated mutation is achieved by allowing a random, low percentage of string allele to switch values. It also prevents premature convergence by providing random variations. The resultant selection, crossover and mutation pressure drives the population towards an optimum solution.

Since its inception, GAs have proven to be robust, relatively efficient and effective in finding global optima. Its robustness stems from the use of problem-independent low level operators, the only requirements being a string representation of possible solutions (genotype) and a means of implicitly or explicitly evaluating the fitness of the resultant phenotype. Its efficiency as a search algorithm was mathematically established as an implicit part of the GA formulation by Holland (1975) and Goldberg (1989) through the theory of *implicit parallelism* and the *schema theorem*. Nondeterministic transition rules and operators as well as the multipoint search enabled by an initial population of possibilities greatly increase the ability of the GA to find global optima over more traditional search or optimization techniques.

Received October 14, 1999

S.Y. Woon[1], O.M. Querin[1] and G.P. Steven[2]

Dept. of Aeronautical Engineering, Building J07, University of Sydney, NSW 2006, Australia
e-mail: wsy@aero.usyd.edu.au, grant@aero.usyd.edu.au
[1] Current address: School of Mechanical Engineering The University of Leeds, Leeds LS2 9JT, UK
e-mail: ozz@mech-eng.leeds.ac.uk
[2] School of Engineering, University of Durham
e-mail: grant.steven@durham.ac.uk

* Extended version of a paper presented at WSCMO-3, held in Buffalo, NY, 1999

GAs have since been successfully applied in a multitude of forms to diverse applications. Some fields of application include combinatorial optimization (travelling salesman problem), Goldberg and Lingle (1985), floor-plan design, Cohoon *et al.* (1991), job-shop scheduling, Fang *et al.* (1993), Norman and Bean (1995), control systems, Odetayo and Dasgupta (1995), computational fluid dynamics, Doorly (1995), aerodynamic design, Periaux *et al.* (1995), artificial learning, Andre (1995) and database design, Cedeno and Vemuri (1997). In structural mechanics, there has been a similarly high level of interest. Schoenauer (1995) presented a GA approach to finding a partition of a given design domain into material-void subsets. Le Riche and Haftka (1997) and Le Riche *et al.* (1995) used a GA to optimize composite laminates. Chapman and Jakiela (1996) applied a GA to structural optimization with compliance and topology considerations. For structural shape optimization in particular, there has been research by Kita and Tanier (1998) and Wibowo and Besari (1998) amongst others. The scheme applied by Wibowo and Besari dealt specifically with the shape optimization of oval axially symmetric shells. Kita and Tanie used a GA to optimize the shape as well as the topology of continuum structures through B- spline functions. A GA was used to find the optimum positions of the knots of the B-spline functions and the number of internal boundaries, hence the optimum topology.

This paper details an alternate coding of the shape optimization problem that allows it to be amenable to the classic binary valued GA using the actual coordinates of the boundary nodes as the design variables. This coding allows direct control over the extent of the local area of a shape to be optimized as well as global shape optimization through addition or removal of material.

2 Shape optimization using a genetic algorithm

2.1 Binary string representation

An optimal shape may be arrived at by a series of stepwise movements of its boundary nodes. For example, in Fig. 1 a hypothetical optimum shape is arrived at in a series of steps of equal magnitude. Figure 1 also shows that there exists more than one sequence of equal sized steps that will lead to the optimum shape. There is only one "optimal sequence" that requires the least number of steps. Figure 1 also shows that while there may be more than one possible sequence, there conversely exist sequences that will not lead to an optimum shape. For a given step movement, there exist fit and unfit boundary node movement combinations hence the combination of nodes that are to be moved is important. The degree of fit of a given combination is determined by the fitness it imparts to the structure as a whole.

While a sequence of steps consisting of unfit combinations will not lead to an optimum shape, Fig. 1 demonstrates that it is not necessary to find the fittest possible combination at every step. A sequence of the fittest combinations defines the optimal sequence but a sequence of fit combinations also leads to an optimum shape, albeit with more steps.

For any single step movement, a range of possible combinations can be defined, all of which have an attached fitness. As such, every step movement is amenable to a binary GA scheme where a population of strings represents possible combinations of boundary nodes which are to be moved a given step. Each locus of a given string corresponds to a boundary node that is to be optimized. The number of boundary nodes to be included in the optimization process is hence equivalent to the length of the string. Each locus has binary alleles. A binary value of "0" represents no movement and a binary value of "1" represents a step-wise movement of a particular boundary node. The series of ones and zeros hence represents one possible form of a step movement arising from one possible combination of boundary nodes. A population represents a set of possible forms for a given step. The design variables are the coordinates of the boundary nodes but they are not directly coded in the GA string. The genotype, as expressed by a string, decodes to a set of movement information for the boundary nodes. This information is used to create the phenotype (shape representation of the string) by modifying the boundary node coordinates accordingly.

A sequence of fit combinations, as opposed to the fittest combinations, will suffice to create a highly effective structural shape. In GA terms, this means that a sequence of locally optimal or near-optimal combinations in unison will suffice to create the global shape optimum. The notion of a series of local optima is especially suited to a form of the Micro-GA operator, as explained in Sect. 2.2.

2.2 Genetic algorithm operators

2.2.1 Basic operators

A tournament selection scheme was used between two randomly selected strings. Random selection of strings to include in the selection process was facilitated by a random number generator. The fitter of the two strings survives to become a parent in crossover. Tournament selection is performed twice for every crossover in order to find a suitable mating pair (1)

do $k = 1$ to 2
  if $\text{fitness}_i > \text{fitness}_j$
  $\text{parent}(k) = \text{string}_i$
 else

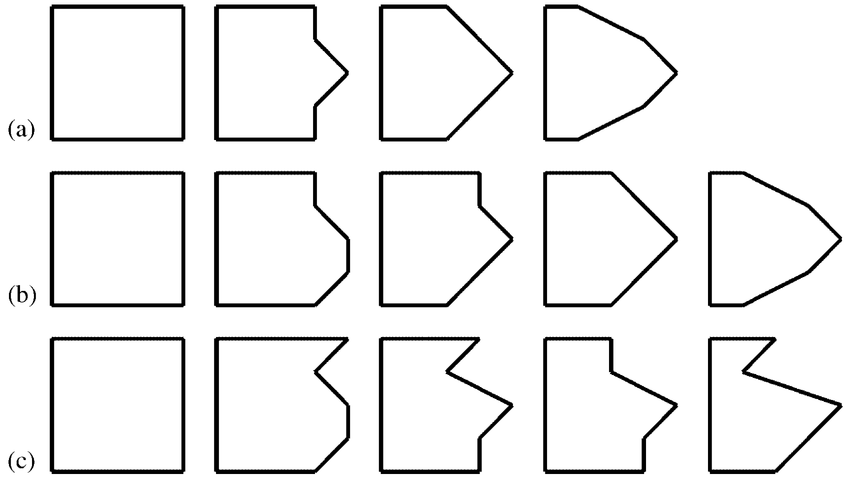

Fig. 1 Three possible sequences of boundary node combinations. (a) Optimal sequence consisting of the fittest possible combinations – optimum achieved. (b) Nonoptimal sequence consisting of fit combinations – optimum achieved. (c) Nonoptimal sequence consisting of unfit combinations – optimum not achieved

```
    parent(k) = string_j
  end if
end do                                        (1)
```

After two parents were selected, single point crossover was performed. This was done by selecting a randomly generated crossover point and exchanging allele values between parents in order to form two new strings. Each crossover was only set to produce one offspring, with the second offspring discarded. (Fig. 2)

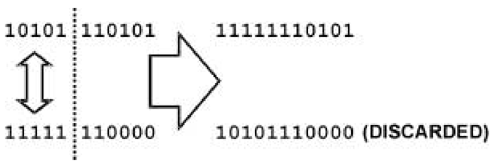

Fig. 2 Single point crossover between two parents with only the first offspring allowed to survive

The probability of crossover was set to 1 while the mutation probability was set to 0. These values were set to encourage quick convergence to an optimum, either local or global.

2.2.2
Advanced operators

The advanced operators used were elitism and micro-GA. Elitism is especially important in GA schemes that involve quick convergence, whereby the schemata of high fitness lost through sampling error may not have sufficient time to re-emerge. Elitism was used throughout to ensure the survival of fit schemata.

Micro-GA is a GA concept that was obtained from and first encountered in coding published on the Internet by Carroll (1997). Micro-GA, which may be applied in place of or in conjunction with mutation, allows an effective search using small population sizes. It works by allowing a small population to converge quickly. The degree of convergence is determined by counting the number of different bits in the population compared to the fittest individual. If the difference is less that a set percentage (e.g. 5%), then the population is deemed to have converged. If this occurs, the entire population, except the fittest string, which is reproduced through elitism, is repopulated with random strings to form a new population. This is akin to starting a new search after finding a local optimum, but still retaining the most effective schema from the previous search. This method does an effective search of the solution space using a small population size, thus reducing computational time.

2.3
String fitness

In this work two criteria were used to define optimality. The first is low deflection (or high stiffness) and the second is low weight. These two criteria were combined to define the fitness function in (2),

$$\text{fitness}_i = \frac{1}{\text{weight}_i \times \text{deflection}_i} \,. \tag{2}$$

The fitness of a given string "i" is equivalent to the inverse of the product of its resultant structural weight and its deflection. The deflection of a structure is defined as the average of the absolute value of the deflection experienced by a pre-determined number of points. If there are "j" points and if each point has a deflection of "δ" to be included in deflection consideration, the deflection function is shown in (3),

$$\text{deflection}_i = \frac{1}{j}\sum_{n=1}^{j} |\delta_n| \,. \tag{3}$$

The fitness function optimises the shape of a structure for the best balance between weight and stiffness. A structure can be made infinitely stiff if there are no material or weight constraints. Conversely, the weight of a structure can generally be reduced if minimum stiffness constraints are not imposed. The fittest string is the string representing the combination of nodal step-wise movements that decodes to a structure that has the best trade-off between weight and stiffness.

In order to define, solve and obtain structural properties of a design, the GA was coupled with a Finite Element Analysis solver, STRAND6 by G+D Computing (version 6.17 1993). The FEA solver acts as an external evaluator for a given structure.

2.4 Options

The design domain of a given structure can be explicitly defined by highlighting the relevant boundary nodes. Each node can be individually set to move in the positive and negative x and y directions. The step size can be set to any given magnitude. Termination of the optimization process can occur either through achieving a fixed number of micro-GA convergences or a pre-defined number of generations. If a line of symmetry exists, a *mirroring* function can be used. The function allows definition of a line of symmetry about which selected nodes move synchronously. This allows shorter strings to be used for symmetric structures and hence saves computational time.

2.5 Binary GA shape optimization methodology

The initial structure creates a set of base coordinates consisting of the boundary node coordinates. The structural representation, or phenotype, of a given string is defined by moving the appropriate boundary node by a small fixed amount, based on the base coordinates. The normal GA steps of fitness evaluation, selection, crossover and mutation are used to process the string population. Micro-GA is used to check for convergence. Upon convergence, the fittest string is used to generate a new set of base coordinates, based on the old set of base coordinates. When there is convergence, the population is repopulated with random strings and elitism invoked. The final set of base coordinates represents the final optimized set of boundary coordinates that can be used to define the optimized shape of the structure (Fig. 3).

The procedure for the method is as follows.

1. Create an initial structural design.
2. Select the nodes on the boundary whose coordinates will be used as design parameters to be optimized.
3. Specify the movement direction of each node.
4. Specify the nodes whose displacements are to be used in the fitness function.
5. If necessary, define the line of symmetry and the mirrored nodes.
6. Save the boundary node coordinates as the initial set of base coordinates.
7. Create a random population
8. Decode one of the chromosomes of the population to boundary movement information and modify the boundary nodes selected in step 2 to form the structural phenotype.
9. Using the FEA solver, calculate the displacement of the nodes selected in 4.
10. Determine the chromosomal fitness as defined by (2) and (3).
11. Repeat steps 8 through 10 until all members of the population have been evaluated.
12. Store the string finesses for later post-processing and identify the fittest string.
13. Select any two chromosomes from the population through tournament selection and perform crossover to create one offspring. Repeat until a new population of offspring has been created.
14. Perform micro-GA. Check the binary form of all chromosomes against the fittest chromosome. If the bits differ by less that 5%, the population is deemed to have converged. If so, create a new set of base coordinates based on the phenotype of the fittest chromosome. Create a new random parent population.
15. If a set number of micro-GA convergences have occurred, go to step 17, otherwise perform elitism by checking if the fittest chromosome has been reproduced and if not, randomly replace one offspring with the fittest string as determined in step 12.
16. If a set number of generations have been run, go on to step 17, otherwise return to step 8.
17. Replace the coordinates of the nodes in the finite element structure with that defined by the movement vector of the fittest chromosome of the last population. This final set of base coordinates defines the final optimized shape.

3 Examples

For the initial tests conducted in this research, two-dimensional plane stress elasticity problems were studied.

3.1 Simple spanner head

The objective of this example was to apply the method to a simplified two-dimensional spanner design, specif-

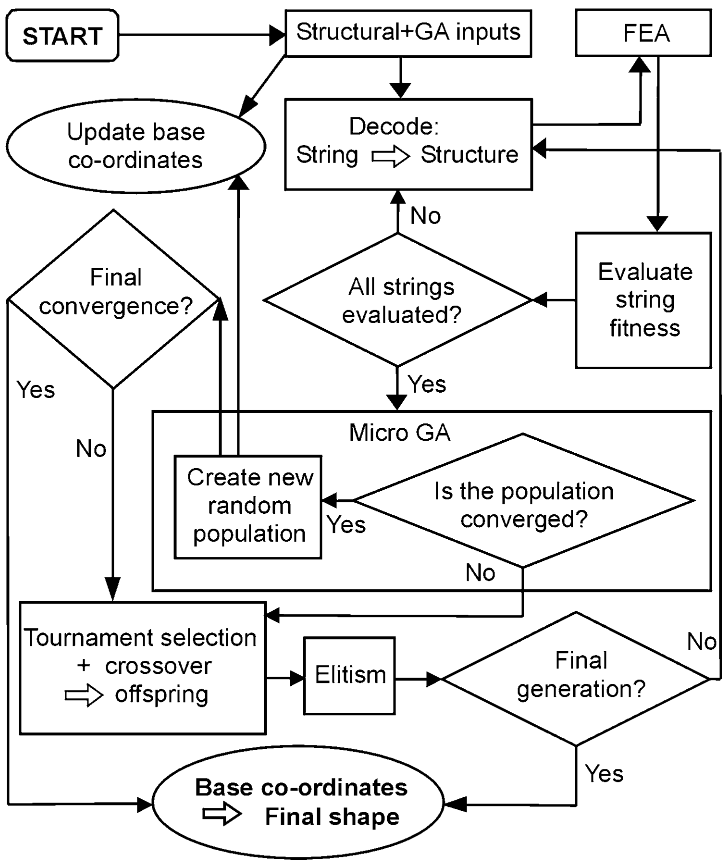

Fig. 3 Schematic representation of the binary GA shape optimization methodology

ically to optimize the shape of its head. The initial design domain consisted of a 429 mm by 150 mm plate with a rectangular notch 99 mm by 30 mm in size. The finite-element mesh consisted of elements of two sizes, 16.5 mm by 15 mm and 33 mm by 15 mm. The smaller elements covered the notched end with a 12 by 10 grid while the larger elements were used in the remaining portion of the block in a 7 by 10 grid. The applied load consisted of a distributed load of magnitude 100 N/mm. The freedom conditions were set around the notch to approximate a spanner locking a nut as shown in Fig. 4. The material properties were set as mild steel (Young's modulus 210 GPa and Poisson's ratio 0.3). The thickness was arbitrarily set to one.

Initially, only the upper surface boundary was used as the design domain. Each chromosome or string had a length of twenty bits, corresponding to the twenty boundary nodes on the upper surface. These nodes were set to move in the negative y-direction only. The deflection of interest was the deflection of the nodes along the length of application of the load. A population size of 18 was used and the magnitude of each step change in geometry was set to 0.5 mm, or 0.33% of the original height of the beam. When the material in the top most layer of

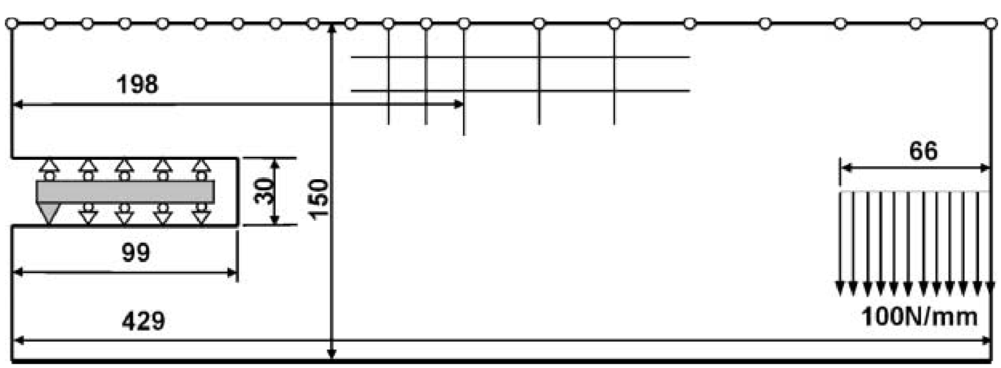

Fig. 4 Initial design for a simple spanner, boundary nodes to be moved highlighted. All units in mm unless specified otherwise

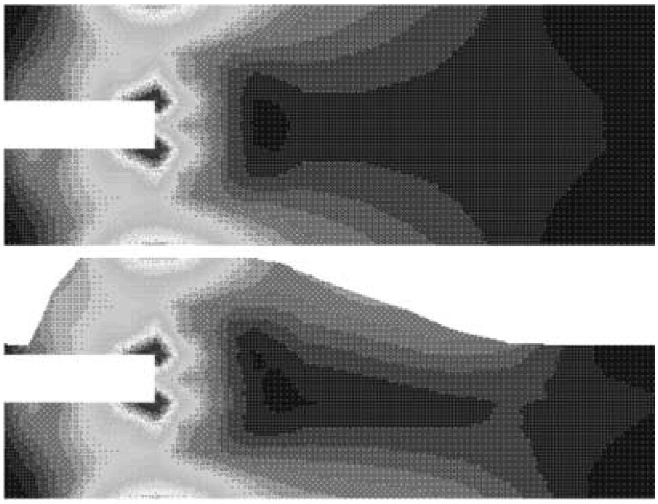

Fig. 5 Final shape of upper half of spanner after 807 generations. Contours show von Mises stress distribution. The figure above shows initial shape and stress distribution

elements was sufficiently reduced such that it began to encroach into the next layer of elements, the structure was manually re-meshed in order to allow optimization into the next layer of elements. The simple spanner example was allowed to run a total of 807 generations.

The resultant structural domain, after a final re-mesh is given in Fig. 5. The final fitness was increased by 21.7% ($2.7628 \times 10^{-6}\,\mathrm{kg}^{-1}\,\mathrm{mm}^{-1}$ to $3.3622 \times 10^{-6}\,\mathrm{kg}^{-1}\,\mathrm{mm}^{-1}$) at the nominal cost of a 9.9% increase in the value of the maximum von-Mises stress (1.946 GPa to 2.138 Gpa). The higher fitness was achieved in conjunction with a reduction in volume from $60\,522\,\mathrm{mm}^3$ to $49\,921\,\mathrm{mm}^3$ (17.5%) without a reduction in stiffness. The average absolute deflection of the nodes along the application of the load remained constant at 6.0 mm per node. There were a total of 109 step movements of magnitude 0.5 mm. The gain in fitness was largely due to the GA based method's ability to identify and remove inefficiently used, low stressed material which did not contribute significantly to its stiffness (see Fig. 5).

Note the flat region between the two smooth curves on the upper boundary. Although the boundary within this region was also to be optimized, no movement occurred. A study of the stress distribution revealed that this was because it is a highly stressed region. In order to get a continuous curve on the upper surface, the height of the initial design domain will have to be larger. The lip at the mouth of the spanner is the residual material from the optimization of the upper surface that was not allowed to go to zero.

3.2 Simple spanner head with mirroring function

The structure obtained after 807 generations (Fig. 5) was used as the basis of a new optimization run. The purpose of the additional run was to fine tune the general shape obtained from the previous run. As such, the step size was halved to 0.25 mm or 0.17% the original height. Before the optimization was carried out, the lip was manually removed and the top half was reflected about the initial line of symmetry to obtain a symmetrical structure. Using the mirroring function, both the upper and lower boundaries were included in the new design domain while retaining the same chromosome length, hence saving computational time. After 391 generations, no further nodal movement occurred, indicating a global optimum had been reached for the given structure with respect to the fitness function and its environment. The initial design domain and stress distribution as well as the final resultant structural domain, after a final re-mesh, is shown in Fig. 6. Fitness increased by a further 1.8% and a further 1.9% of material was removed without any decrease in stiffness.

The results in Fig. 6 demonstrate the method's ability to produce smooth boundaries and optimize detail as well as perform shape optimization on a global scale.

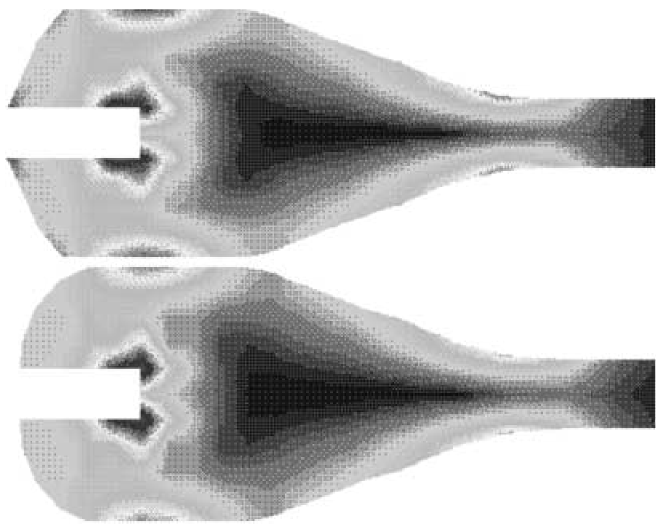

Fig. 6 Simple spanner head fine-tuned using a smaller step size and the mirroring function. Figure above shows initial shape and von Mises stress distribution before fine-tuning; figure below shows the final optimal shape

3.3 Flange webbing

The objective of this example was to demonstrate local optimization by "growing" a two-dimensional optimally shaped webbing between the two arms of an inverted L shaped flange. The initial design domain consisted of a flange 110 mm in height and length. The FE mesh of the horizontal arm consisted of 40×2 elements of 2.5 mm by 5 mm. The vertical arm was meshed with 4×2 elements of 25 mm by 5 mm. Both arms had a width of 10 mm. The intersection of the two arms was meshed with 2×2 elements of 5 mm by 5 mm. The applied load consisted of a distributed force of 100 N/mm applied on the upper surface of the horizontal arm. The flange was fully supported at 5 points distributed evenly along the vertical arm (Fig. 7) to approximate a flange bolted at 5 places to a given surface. The material properties used were the same as those used for the example in Sect. 3.1.

A thin strip of boundary nodes covering the inner lower half of the horizontal arm was to be optimized by allowing it to grow away from the surface. The length of each chromosome was of size 20, and each locus corresponded to a single node that was to be optimized. These nodes were set to move in the negative y-direction. The deflection of interest was the deflection of the nodes along the length of application of the load. A population size of 18 was used and the magnitude of each step change in geometry was initially set to 1 mm. The webbing was periodically manually re-meshed as it grew. After 342 generations, the step movement magnitude was reduced to 0.5 mm for a further 408 generations. For the fine tuning process, the magnitude was reduced to 0.25 mm and the structure allowed to evolve for a further 425 generations after which no further movement occurred.

The resultant flange with a half-armed optimized web is shown in Fig. 8. The total run was 1175 generations with 29 convergences for the 1 mm-movement steps, 42 convergences for the 0.5 mm steps and 56 convergences for the 0.25 mm steps. The final fitness was $1.1942 \times 10^{-4}\,\mathrm{kg}^{-1}\,\mathrm{mm}^{-1}$, an increase of 494% from the initial fitness of $0.2010 \times 10^{-4}\,\mathrm{kg}^{-1}\,\mathrm{mm}^{-1}$. This increase in fitness was obtained at the expense of an increase in volume of 77.2% ($2124.4\,\mathrm{mm}^3$ to $3763.7\,\mathrm{m}^3$). Nodal deflections along the length of application of the load decreased by 90.5%, indicating that the webbing was effective in substantially improving the stiffness of the upper arm. The maximum von Mises stress was decreased by 78.2% from 32.33 GPa to 7.06 GPa. The optimization method was able to add material to effectively increase the stiffness and overall fitness of the structure without an excessive increase in weight. As a result, the stress concentration was also reduced.

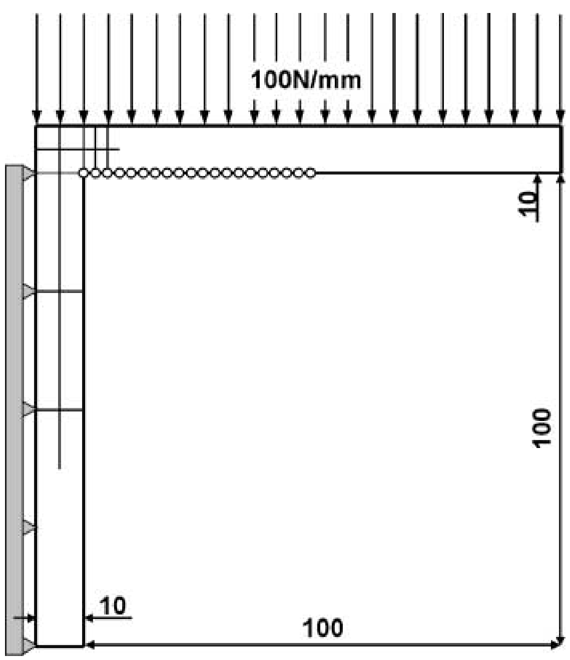

Fig. 7 Initial design for a flange, boundary nodes to be moved highlighted. All units in mm unless otherwise specified

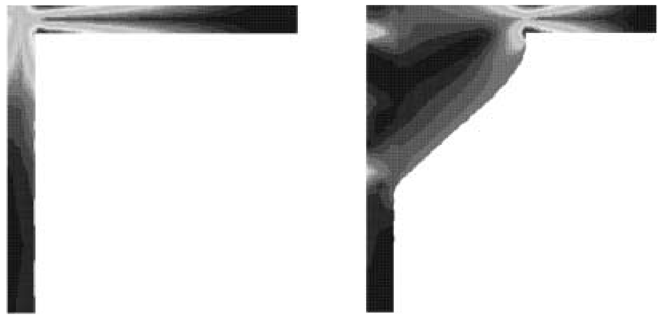

Fig. 8 The figure on the right shows the final shape of the flange with webbing that was allowed to grow until no further movement was encountered. Contours show von-Mises stress distribution. The figure on the left shows the initial shape and stress distribution

4 Concluding remarks

The research presented here shows the validity and effectiveness of the alternate binary valued Genetic Algorithm coding for structural shape optimization. Initial structural applications demonstrate the ability of the developed method to produce optimal shapes, based on stiffness and weight criteria, through a sequencing of fit steps based on sets of locally optimal combinations of boundary nodes. The method is shown to be able to identify and remove material that do not significantly contribute to stiffness as well as add material to regions in order to effectively increase the structure's fitness with minimal cost to weight. The research also underscores the ability of this method to design both optimal local details as well as global shapes. Fine-tuning of shapes through reduced step sizes produces smooth boundaries and gradual transitions and presents a useful tool for controlling the optimization fidelity of a given surface. Variations of the method that reduce the computational cost are currently being researched. At present, the mirroring function allows optimization of symmetrical structures at half the expected computational time.

References

Andre, D. 1995: The evolution of agents that build mental models and create simple plans using genetic programming. In: Eshelman L.J. (ed.) *Proc. 6-th Int. Conf. on Genetic Algorithms*, pp. 248–255. San Francisco, CA: Morgan Kaufmann Publishers

Carroll, D. 1997: FORTRAN GA driver. University of Illinois at Urbana-Champaign, Internet source: http://www.staff.uicc /˜carroll/ga.html

Cedeno, W.; Vemuri V.R. 1997: Database design with genetic algorithms. In: Dasgupta, D.; Michalewicz, Z. (eds.) *Evolutionary algorithms in engineering applications*, pp. 189–206. Berlin, Heidelberg, New York: Springer

Chapman, C.D.; Jakiela, M.J. 1996: Genetic algorithm based structural design with compliance and topology simplification considerations. *J. of Mech. Des.* **118**, 89–98

Cohoon, J.; Hedge, S.; Martin, N. 1991: Distributed genetic algorithms for the floor-plan design problem. *IEEE Trans. Comp. Aided Des.* **10**, 483–491

Doorly, D 1995: Parallel genetic algorithms for optimization in CFD. In: Winter, G; Periaux, J.; Galan, M.; Cuesta, P. (eds.) *Genetic algorithms in engineering and computer science*, pp. 251–270. West Sussex, England: Wiley

Fang, H; Ross, P; Corne, D 1993: A promising genetic algorithm approach to job-shop scheduling, rescheduling and open shop scheduling problems. In: Forrest, S. (ed.) *Proc. 5-th Int. Conf. on Genetic Algorithms*, pp. 375–382. San Mateo, CA: Morgan Kaufmann Publisher

Goldberg, D.E. 1989: Genetic algorithms in search, optimization and machine learning. Addison-Wesley Publishing Company

Goldberg, D.E.; Lingle, R. 1985: Alleles, loci and the travelling salesman problem. In: Grefenstette, J. (ed.) *Proc. 1-st Int. Conf. on Genetic Algorithms*, pp. 154–159. Hillsdale, NJ: Lawrence Erlbaum Associates

G+D Computing 1993: *STRAND6 Finite Element Analysis System, Reference Manual and User Guide*, 1993. Sydney: G+D Computing

Holland, J.H. 1975: *Adaptation in natural and artificial systems*. London: MIT Press

Kita, E.; Tanie, H. 1998: GA-based topology optimization of continuum structures. In: Steven, G.P.; Querin, O.M.; Guan, H.; Xie, Y.M. (eds.) *Structural optimisation* (Proc. Australasian Conf. on Structural Optimisation), pp. 87–94. Victoria: Oxbridge Press

Le Riche, R.G.; Haftka, R.T. 1997: Evolutionary optimisation of composite structures. In: Dasgupta, D.; Michalewicz, Z.; (eds.) *Evolutionary algorithms in engineering applications*, pp. 87–102. Berlin, Heidelberg, New York: Springer

Le Riche, R.G.; Knopf-Lenoir, C.; Haftka, R.T. 1995: A segregated genetic algorithm for constrained structural optimisation. In: Eshelman L. J.; (ed.) *Proc. 6-th Int. Conf. on Genetic Algorithms*, pp. 558–565. San Francisco, CA: Morgan Kaufmann Publishers

Norman, B.; Bean, J 1995: Random keys genetic algorithm for job-shop scheduling. *Technical report*, University of Michigan, Ann-Arbor

Odetayo, M.O.; Dasgupta, D 1995: Controlling a dynamic physical system using genetic-based learning methods. In: Chambers, Lance (ed.) *Practical handbook of genetic algorithms*, pp. 173–196. Boca Raton Florida: CRC Press

Periaux, J.; Sefrioui, M.; Stoufflet, B.; Mantel, B.; Laporte, E. 1995: Parallel genetic algorithms for optimization in CFD. In: Winter, G; Periaux, J.; Galan, M.; Cuesta, P. (eds.) *Genetic algorithms in engineering and computer science*, pp. 371–396. West Sussex: Wiley

Schoenauer, M. 1995: Shape representation for evolutionary optimization and identification in structural mechanics. In: Winter, G; Periaux, J.; Galan, M.; Cuesta, P. (eds.) *Genetic algorithms in engineering and computer science*, pp. 371–396. West Sussex: Wiley

Wibowo, F.X.N.; Besari, M.S. 1998: Genetic algorithms in shape optimisation of oval axially symmetrical shells. In: Steven, G.P.; Querin, O.M.; Guan, H.; Xie, Y.M. (eds.) *Structural optimisation* (Proc. Australasian Conf. on Structural Optimisation), pp. 103–111. Victoria: Oxbridge Press