Optimization of composite panels using neural networks and genetic algorithms

W. Ruijter [a,*], R. Spallino [b], L. Warnet [a], A. de Boer [a]

[a] University of Twente, Mechanical Engineering Department, 7500 AE Enschede, The Netherlands
[b] AIRBUS Deutschland Gmbh, Kreetslag 10, D-21111 Hamburg, Germany

Abstract

The objective of this paper is to present first results of a running study on optimization of aircraft components (composite panels of a typical vertical tail plane) by using Genetic Algorithms (GA) and Neural Networks (NN). The panels considered are standardized to some extent but still there is a wide scope of discrete and continuous design variables that can be adjusted to increase performance or reduce structural weight. A NN is trained for every panel configuration using a backpropagation algorithm with data sets taken from finite element analyses spread randomly over the design space. The trained network is then used to predict the values of the constraint functions (strain and buckling multipliers). The approach is formulated in this manner to maintain maximum flexibility regarding the implementation of new variables or models and with the prospect of optimizing the assembly as a whole. Results show that in design problems with high dimensionality the approach becomes more attractive, especially when the optimization has to be run repeatedly for panels under different loading/sizing conditions. The optimization algorithm has proven to be robust though dependent on the smoothness of the network output function. A modified method that feeds back the found optima is proposed to improve accuracy of the NN and decrease preparation time.

Keywords: Genetic algorithms; Neural networks; Panel buckling; Composites; Optimization

1. Introduction

The optimal design problem of structural aircraft components received increasing attention and has been approached with various different strategies. In this context, a strategy is commonly defined as a combination of a structural model evaluation (analytical or numerical) and a numerical optimization algorithm. Among them, gradient-based techniques, as nonlinear programming [1], SLP and SQP gained a great popularity, while biological analogy-based techniques as evolution strategies or genetic algorithms [2] emerged in the last years as reliable and robust general-purpose optimization tools. In this paper, the problem of globally optimizing a structure as a set of interacting sub-components is approached and solved. Neural networks are used as a global approximate function tool for the structural evaluation of aircraft components, genetic algorithms are employed to optimize the spar panel design. This leads to a reduction of a single optimization runtime,

so that optimizing an assembly (consisting of more than 40 components) will become affordable. In the following paragraphs, a short description is given of the structural problem, the proposed strategy is described and results are presented for a few sample runs.

2. Structural mechanics

The structural mechanics problem to which the optimization algorithm is applied, consists of a shear loaded stiffened composite panel with access holes (see Fig. 1). These panels represent partial idealizations of vertical stiffeners (spars) as applied in the Vertical Tail Plane (VTP) structure of commercial aircraft. A number of design variables are considered, containing the sequence and number of stringers and holes on the panel, the offsets of holes and stringers and the number of plies in the lay-up. Modeling of the panels was done parametrically, using the Finite Element (FE) method. The panels are evaluated for their strain and buckling reserves using linear static and buckling FE analyses combined with laminate strain and buckling cri-

* Corresponding author. Tel.: +31 (53) 489-2553; Fax: +31 (53) 489-3471; E-mail: w.ruijter@student.utwente.nl
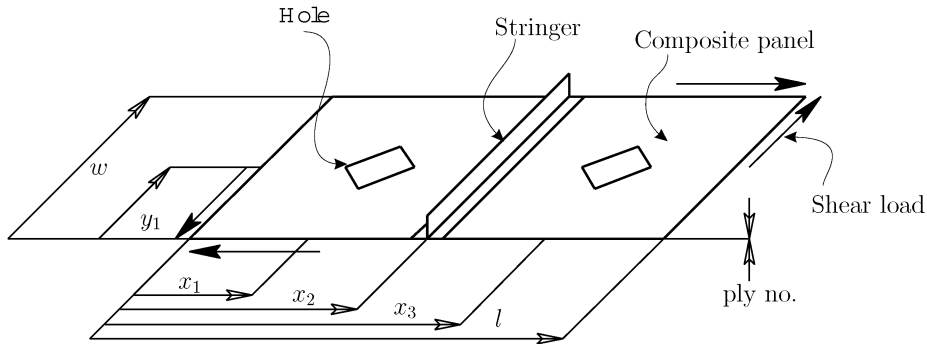
Fig. 1. Parametric panel with holes, stringer and applied loading.

teria. The FE model employs ANSYS SHELL99 elements capable of modeling shells of layered orthotropic materials for linear analysis.

3. Optimization

The optimization strategy evaluated, is formulated to fit the need for a shorter optimization time (that is, optimization of a single panel) while allowing all possible designs to be chosen by the algorithm. The proposed strategy for optimizing a single panel is displayed schematically in Fig. 2 and can be formulated as: *For given panel length, width, loading and required access, find the panel with lowest weight that's able to resist the load by varying the stringer and hole numbers, locations and ply number*. The trajectory displays a feedback loop of output data allowing a reduction of FE evaluations required to train the network, as proposed in the section Conclusions. Note that the scope of the full project is a nesting of this routine for all panels at different locations in a VTP assembly.

The stages indicated in Fig. 2 are reviewed briefly.

(1) A parametric finite element model is implemented (see Section 2), this model is run using an automated randomized procedure to generate datasets within the function space. The generation of this database is run parallel and the database can be applied in a knowledge-based system for other purposes. The FE model will be referred to as the "exact" function $f(X)$.

1a. A number of datasets scattered in a partial function space for a configuration, defined by a sequence of holes and stringers, is shown. These data sets are used to train the neural network. The full design space, as mentioned in Section 2, is covered by a few of these spaces. This assures that a partial design space (approximated by one NN) contains no switching type parameters, which is likely to improve fitting accuracy.

(2) A neural network is applied to fit the approximate function $\tilde{f}(X)$ through the data points. The MATLAB Neural Network toolbox was used. A neural network consists

of a series of basic function evaluations (sigmoid functions) analogous to the neurons in the human brain. The neurons in a NN are connected by synapses possessing "conductivities" or weights w_{ij}. A neural network can approximate a function when supplied with sample data from that function (1a from Fig. 2). An iterative procedure is applied to reduce the network error $MSE(w_{ij}) \equiv [\sum_L \sum_k (\tilde{f}(X, w_{ij}) - f(X))^2]/L$ where L the number of training sets and k the number of network outputs. This training procedure converges as displayed in Fig. 3. The "order" of $\tilde{f}(X)$ is mainly determined by the number of neurons in the network. NN are preferred to polynomial fitting because of implementation flexibility.

2a. The approximated function supplies an output two orders of magnitude faster than the FE model, this allows for extensive search algorithms such as genetic algorithms to be employed to search an optimal solution.

(3) A MATLAB implemented genetic algorithm (by Houck et al. [3]) is used to search for an optimal solution within the approximated function space as defined by the Neural Networks. Instead of using gradient information to find an optimal solution a GA employs a strategy analogous to biological evolution maintaining a population of solutions and running the following stages iteratively:

- Evaluate fitness function (objective and penalties for constraint violation)
- Rank solutions according to fitness
- Select solutions and recombine their parameters (crossover)
- Apply mutation

Because the GA does not use gradient information, it is less likely to get trapped in a local region of the design space, furthermore it can implement discrete and continuous parameters simultaneously. Gradient based algorithms are generally faster than GAs, but GAs are more robust in their operation as shown by Spallino et al. [4,5]. Robustness of the algorithm is considered more important as the total computational cost is dominated by the cost of FE analysis, which is done independently from the optimization. When the optimization output is fed back to the neural network as

1. Parametric Finite Element Model

$n = 1$

1.a. Initial (random) function evaluations

$n = n + 1$

Additional training

Initial training set

2. Feedforward Backpropagation Neural Network

Signal

X → $\tilde{f}(X) f(X)$

Error

Improved approximated space

2.a. Mapped function by NN

3. Apply GA Optimizer to mapped space
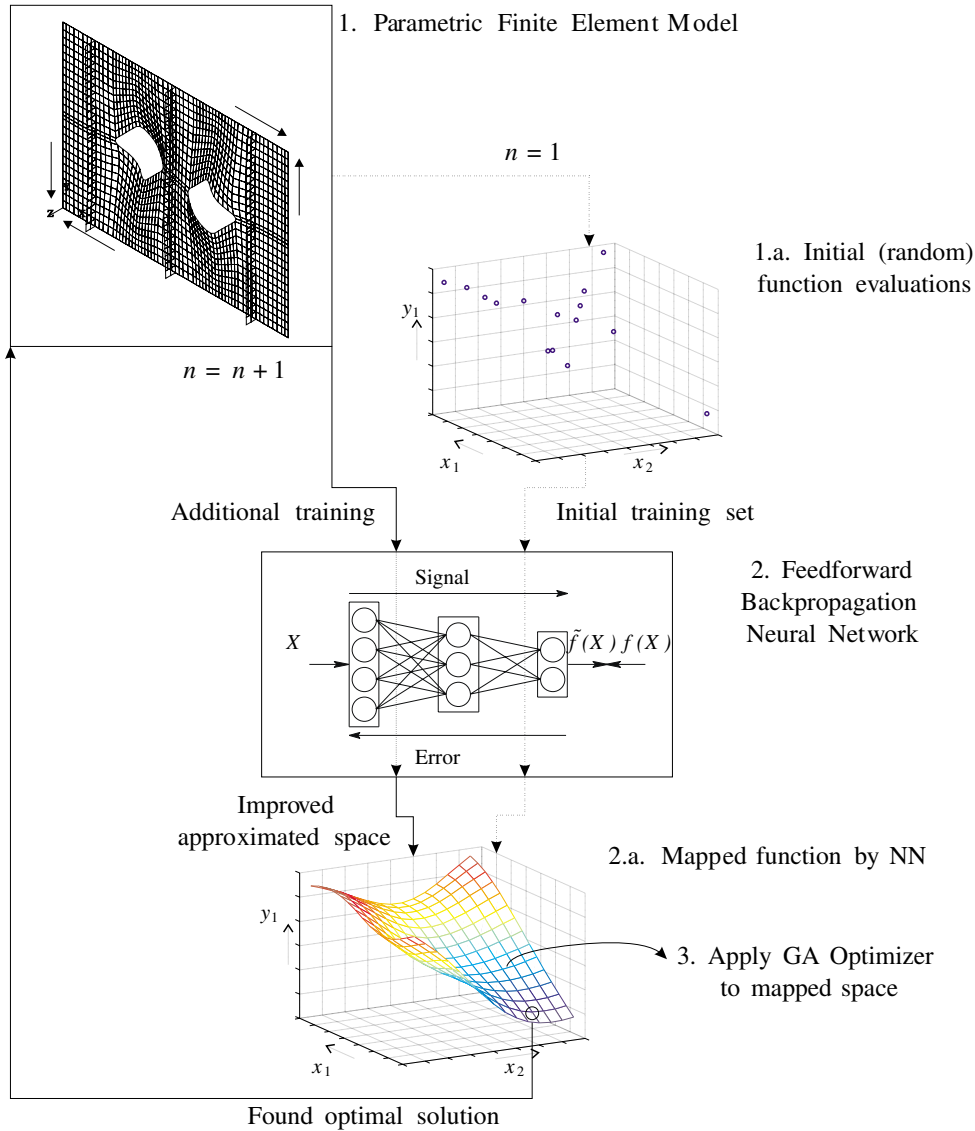
Found optimal solution

Fig. 2. Optimization strategy for one panel (displaying a design space of two inputs and one output for clarity).

shown in Fig. 2 the cost of the GA operation is expected to become even less significant.

4. Results

In presenting the results obtained from the optimization procedure a few sample panels were taken from a typical VTP design. The panels under consideration were optimized to initial data given in Fig. 3. The output of the procedure is a full parameter set of a feasible panel of certain configuration as in Fig. 3. The outputs were verified with the FE model and the deviations found for buckling multiplier λ and strain ε are given. The networks used

contained one "hidden layer" with a number of neurons h defined by $h = \delta \cdot (L-1) \cdot m/(n+m+1)$ where n is the configuration dependent number of input neurons, the number of outputs $m = 2$ and $\delta = 0.5$, as implemented by Jenkins [6]. The networks were trained using an accuracy termination of the training phase $MSE \leq 1 \cdot 10^{-5}$. A learning function employing Bayesan regularization (application of a modified network error function to smoothen the network output, see MATLAB help system [7]) and a learning rate set to 0.1 (this parameter determines the step size by which the weights of the neural network can be adjusted in the network's learning phase). The GA was run 50 generations with a population of 40 individuals in about 50 seconds, typical convergence is shown in Fig. 4. The cost

Input parameters			Network		Output		
length (mm)	width (mm)	shear (N/mm)	Neurons	Datasets	λ %	ε %	Type
800	600	200	65	314	4.44	2.3	6
800	500	130	65	314	1.4	0.8	6
900	450	100	65	314	20.2	2.4	6
600	650	170	96	393	3.9	10.7	5

Learning convergence

$SSE_{\text{training set}}$

10^{0}

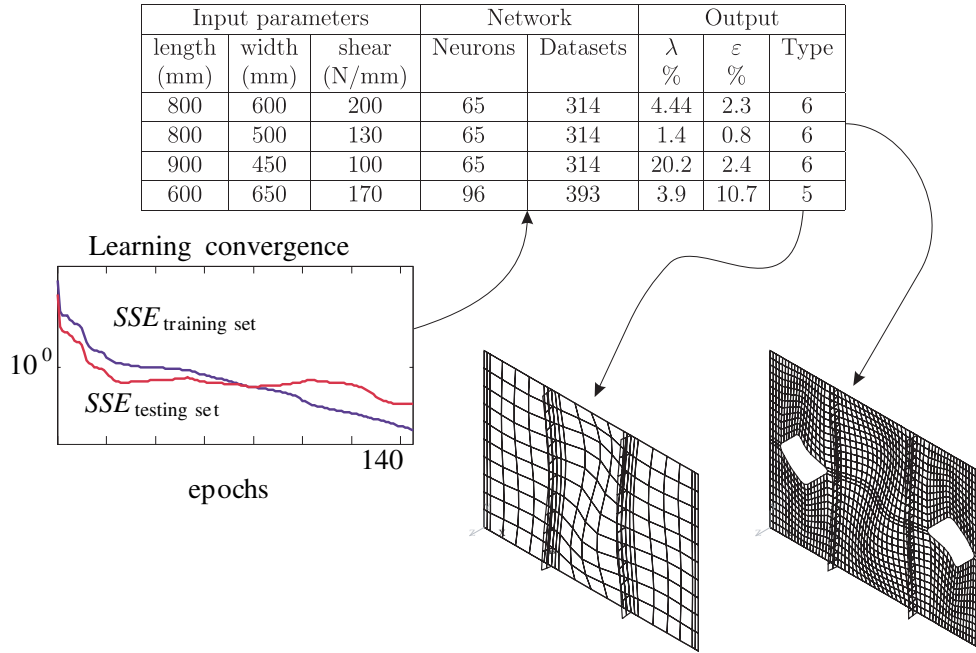$SSE_{\text{testing set}}$

140

epochs

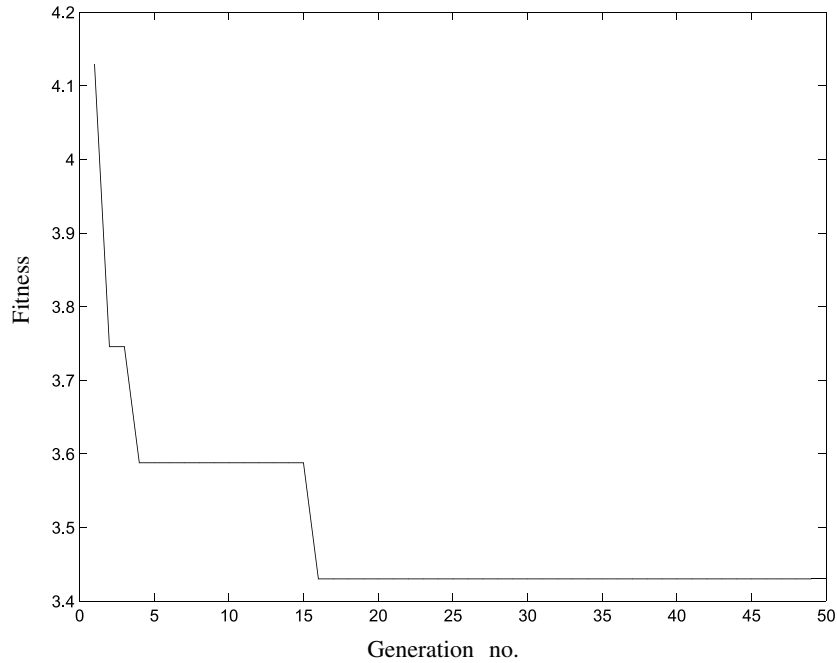Fig. 3. Output configurations from NN-GA optimization runs (continuous variable output not displayed).

Fig. 4. Typical convergence of GA.

is negligible compared to the time needed to generate the training database (4 days for $5 \cdot 10^3$ data sets spread over 10 configurations on a PC, AMD 1600 processor, 512 MB RAM) and network training (1 day for 10 networks).

5. Conclusions

The use of neural networks and genetic algorithms for the optimization of shear loaded composite panels has been

investigated. The computational effort that is related to using the investigated method is distributed different from traditional optimization. The number of FE calculations that is done in advance is relatively large which makes the approach attractive mainly to applications where panels with standard topology have to be optimized repeatedly (as is the case in airplane industry). In these applications the fact that the search of the algorithm is wide-ranged and optimization cost is low weigh heavier than preparation time. The speed of the optimization makes the procedure attractive for its intended use in a multi-level strategy. It is expected that preparation time can be lowered and accuracy bounds can be narrowed significantly by implementing feedback of FE checked optimal solutions into the NN training set.

Acknowledgement

The contributions of Jorg Entzinger and Jeroen Hol are gratefully acknowledged.

References

[1] Arendsen P. The B2000 Optimization Module: B2OPT, NLR-TP-1994-116. National Aerospace Laboratory (NLR), Amsterdam, 1994.

[2] Goldberg DE. Genetic Algorithms in Search, Optimization and Machine Learning. Reading: Addison-Wesley Publishing Co. Inc, 1989.

[3] Houck CR, Joines JA, Kay MG. A Genetic Algorithm for Function Optimization: A MATLAB Implementation. GAOTv5 Manual (Genetic Algorithm Optimization Toolbox), 1998.

[4] Spallino R, Rizzo S. Optimal design of laminated composite plates. Computer Aided Optimum Design of Structures IV, Boston: 1999.

[5] Spallino R, Rizzo S, Giambanco G. Evolution strategies for the shakedown optimal design of R.C. framed structures. European Congress on Computational Methods in Applied Science and Engineering. Barcelona, 2000.

[6] Jenkins WM. Neural Network-based Approximations for Structural Analysis. Developments in Neural Networks and Evolutionary Computing for Civil and Structural Engineering. 1995:25–35.

[7] The Mathworks: MATLAB 6.1 R12.1 Help system, 2001.