PUBLICATIONS

OF THE

UNIVERSITY OF MISKOLC

Series C.

MECHANICAL ENGINEERING

VOLUME 47.

MISKOLC, 1997

HU ISSN 0237 - 6016

A kiadvány megjelenését
a Magyar Tudományos Akadémia Miskolci Akadémiai Bizottsága,
a Gépipari Tudományos Egyesület
és a Peregrinatio IV Alapítvány támogatta

Kiadja a Miskolci Egyetem
A kiadásért felelős: Dr. Patkó Gyula rektorhelyettes
Miskolc-Egyetemváros.1997
Megjelent a Miskolci Egyetemi Kiadó gondozásnban
Felelős vezető: Dr. Péter József
Műszaki szerkesztő: Balsai Pálne
Példányszám: 200
Készült Develop lemezről, az MSZ 5601-59 és 5602-55 szabványok szerint
Miskolci Egyetemi Sokszorosító Üzeme
A sokszorosításért felelős: Kovács Tiborné üzemvezető
TU '97 1070 ME
A levonat a Sokszorosítoba leadva. '97. november 13.

Publ. Univ. of Miskolc, Series C. Mechanical Engineering, Vol. 47. (1997) pp. 111-122

EXPERT SYSTEMS AND ARTIFICIAL NEURAL NETWORKS IN STRUCTURAL OPTIMIZATION

Károly JÁRMAI*

Summary

Artificial Intelligence (AI) techniques are the best utilized in identifying and evaluating design alternatives and relevant constraints while leaving the important design decisions to the human. Expert systems and artificial neural networks are used in structural optimization. We show the benefits of these systems in the optimum design of main girders of overhead travelling cranes and stiffened plates. At the example the double crane girders are welded and stiffened box ones, with one trolley on them. We have used the British Standard for the structural analysis.

1 Introduction

Expert systems and artificial neural networks are new and promising fields in structural optimisation. Neural network's ability to perform computations is based on the hope that people can reproduce some of the flexibility and power of the human brain by artificial means.

2 Expert systems

Depending on the application, an expert system can perform ten type of projects as follows: interpretation, prediction, diagnosis, design, planing, monitoring, debugging, repair, instruction, control. We have used the expert system for design.

The three basic components of an expert system are

- the knowledge base,
- the inference engine,
- the user interface.

* Professor, University of Miskolc, H-3515, Miskolc, Egyetemváros
e-mail: altjar@gold.uni-miskolc.hu, homepage: http://www.alt.uni-miskolc.hu

There are three main streams at expert systems

- rule-based expert systems can be backward or forward chaining,
- object-oriented systems,
- hybrid systems, which combine object-oriented techniques with rule-based ones[1,2,3,4].

There were some attends to connect the expert systems and structural optimization. One of them is an expert system for finding the optimum geometry of steel bridges [5].

The connection of single- and multiobjective optimization made it possible at the structural optimization to form a decision support system. At the multiobjective optimization there are several so called weighting coefficients for the designer to give the relative importance of the objective functions [6,7]. The decision support systems (DSS) and the expert systems (ES) are close together, but it is necessary to build an inference engine. The key concept in our approach is to give the user control of important design decisions.

Therefore, our approach in applying AI to engineering design is to use AI techniques for keeping track of all the design alternatives and constraints, for evaluating the performance of the proposed design by means of a numerical model, and for helping to formulate the optimization problem. The human designer evaluates the information and advices given by the computer, assesses whether significant constraints or alternatives have been overlooked, decides on alternatives, and makes relevant design decisions.

2.1 Overview the Level 5 Object [8]

LEVEL 5 OBJECT (LO5) is an object-oriented expert system development and delivery environment. It provides an interactive, windows-based user interface integrated with Production Rule Language (PRL), the development language used to create L5O knowledge bases. The PRL Syntax Section provides syntax diagrams to follow logically when writing a knowledge base. System classes are automatically built by L5O when a new knowledge base is created, thereby providing built-in logic and object tools. The developer can use system classes in their default states or customize them. In this way, the developer can control devices, files, database interactions and the inferencing and windowing environments.

The most remarkable tools of LO5 are:

- object oriented programming (OOP),
- relational database handling (RDB),
- computer aided software engineering (CASE), and
- graphical development system.

The most remarkable tools of LO5 for PCs are:

- Microsoft Windows,
- programming with an object-oriented language (Borland C++),
- direct connection with dBase,
- direct connection with the fourth generation FOCUS data handling system,
- EDA/SQL interface to relational and non-relational databases,
- Rdb/SQL interface to VAX RDB/VMS databases, and

- own worksheet handling system (similar to LOTUS 123).

Using LO5 there are two ways of developing programmes: they can be generated either by word processors or in the developing environment. Taking these capabilities into account, LEVEL 5 OBJECT was found suitable for development expert systems for structural engineering.

There are a great number of expert shells available such as ART (Automated Reasoning Tool, Inference Corporation), KEE (Intellicorp), Intelligence Compiler (Intelligence Ware Inc.), Symbologic Adept (Symbologic Corporation), GURU (Micro Data Base Systems), etc. They are available on APOLLO or SUN workstations or on PC-s [2].

2.2 Application of an expert system for the optimum design of the main girders of overhead travelling cranes

The aim was to develop an expert system, which is able to find the optimum sizes of the welded box girder of the crane due to different geometry, loading, steel grades and design codes.

The total number of variants is about 60000 and it can be increased if we take into account other aspects and constraints in a modular way.

The decision support system, which was connected to the expert one, contains 5 various single-objective and 7 various multiobjective optimization techniques. These techniques are able to solve nonlinear optimization problems with practical nonlinear inequality constraints. It could contain finite element procedures to compute the mechanical behaviour of the structures. The DSS system is described in [6].

2.3 Economic design of box girders of overhead travelling cranes

Objective functions

- material cost of the girder, $C_m = k_m \ \rho \ V$ [kg], where ρ is the material density, V is the volume of the girder, k_m is the specific material cost.
- labour cost contains welding cost and surface preparation cost $C_l = C_w + C_s$,
- welding cost, $C_w = k_w \ (a^2_w / \sqrt{2}) L_w \ \rho \ k_c$ [\$], where a_w is the effective size of welded joint, L_w is the length of welded joint, k_c is the difficulty factor of welding, which depends on the position of welding,
- surface preparation and painting costs, $C_s = k_s \ (2bL+2hL)$ [\$] , where b and h are width and height of the girder, k_s is the specific cost of manufacturing,
- total cost contains material and labour costs $C_t = C_m + C_l$.

Design constraints

- constraint on the static stress at midspan due to biaxial bending according to BS 2573 and 5400 [9,10] is described by

$$M_x/W_x + M_y/W_y \leq \alpha_d \, {}^*P_s \qquad (1)$$

where M_x, M_y are the bending moments, W_x, W_y are section moduli, P_s is the permissible static stress, α_d is the duty factor.

- constraint on fatigue stress is as follows

$$M_{xf}/W_x + M_y/W_y \leq P_{ft} \qquad (2)$$

where M_{xf} contains the live load multiplied by the impact factor and the spectrum factor. P_{ft} is the fatigue stress.

- local flange buckling constraint is

$\sigma_{1f}/(P_s * K_{1f}) + \{(\sigma_{bf}/(P_s * K_{bf})\}^2 \leq 1$, where $\sigma_{1f} = M_x/W_x$; $\sigma_{bf} = M_y/W_y$, the K factors depend on the slenderness of the plate

$\lambda_f = (b/t_f) * \sqrt{R_{yf} / 355}$, where R_{yf} is the yield stress of the flange plate

- local web buckling constraint is

$$\sqrt{\{(0.8\sigma_{1f} + \sigma_{bw})/(P_s K_{1w})\}^2 + \{(\sigma_{cw} / (P_s K_{2w})\}^2 + \{(0.2\sigma_{1f} / (P_s / K_{bw})\}^2 + 3(\tau_q / P_s / K_{qw})^2} \leq 1$$

where $\sigma_{bf} = \sigma_{bw}$; $\sigma_{cw} = F/(t_{1w} * a_w)$; $a_w = 50 + 2(h_r + t_f - 5)$

the K factors depend on the slenderness of the plate

$\lambda_e = (h_w/t_{w1}) \sqrt{R_{yw} / 355}$, where R_{yw} is the yield stress of the web plate.

h_r is the height of the rail.

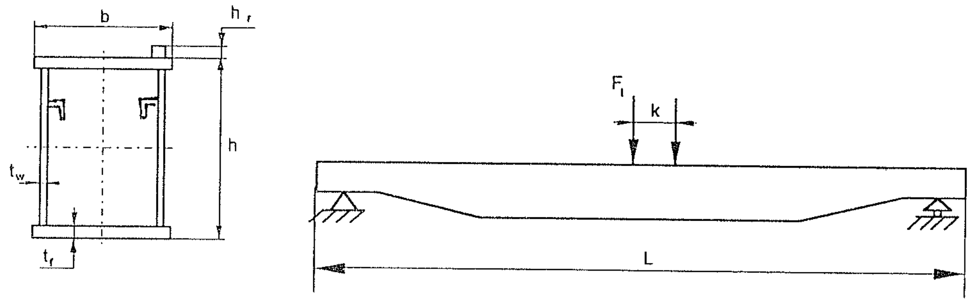

Fig. 1. Cross section of the welded main girder of overhead travelling crane

- local web buckling constraint on the secondary web is similar to the main web one, but we have to use t_{w2} instead of t_{w1} and there is no local compression, so $\sigma_{cw} = 0$.
- deflection constraint due to wheel load can be expressed as $w_{max} \leq L/800$, L/1000, where L is the span length.

2.4 Example

Main data of an example are as follows:

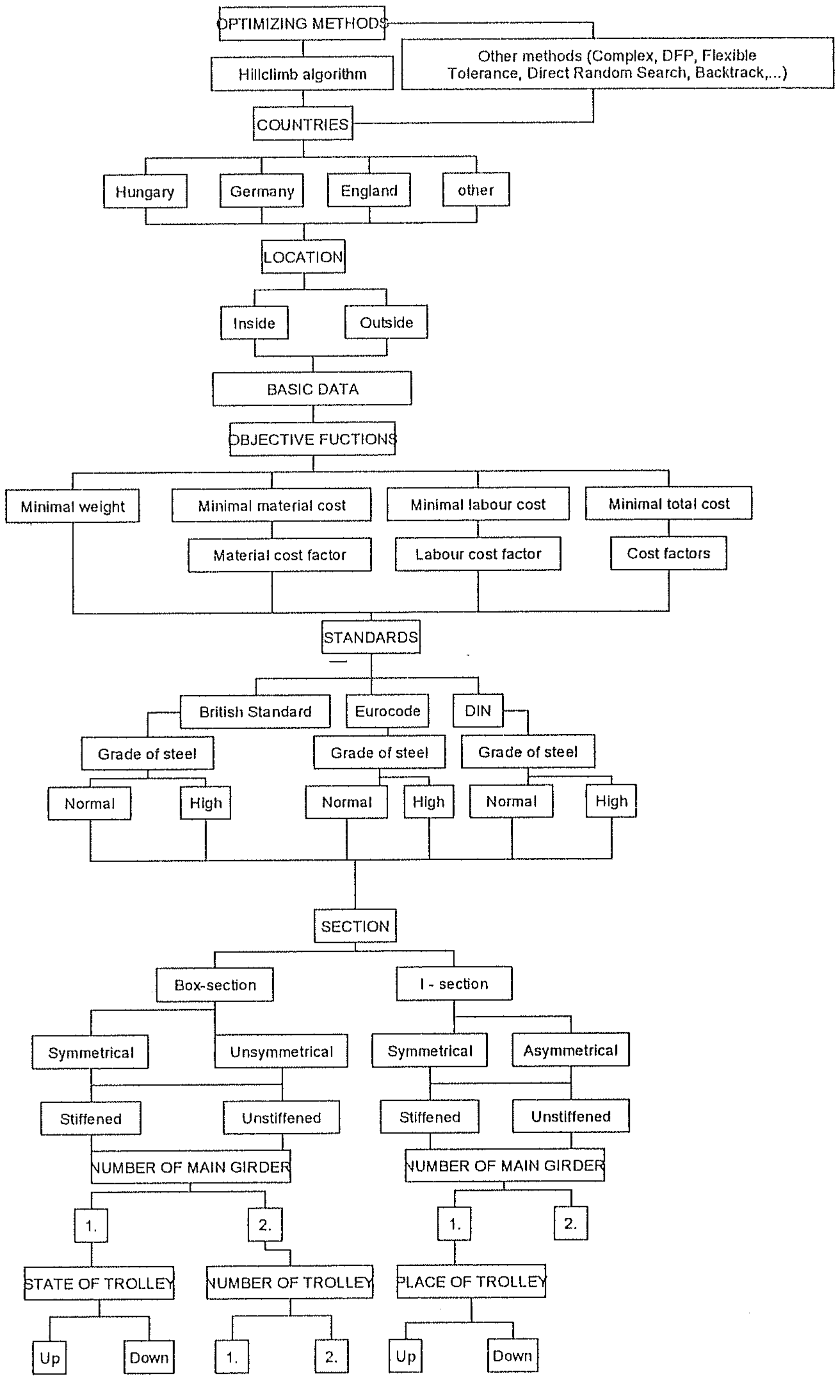

Fig. 2 Logical structure of the expert system

Hook load is $H = 240$ kN, spanlength is $L = 25$ m, mass of trolley is $G_t = 30$kN, distance between the trolley axes is $k = 2.5$ m, height of rail is $h_r = 50$ mm, mass of the rail is $p_r = 80$ kg/m, the Young modulus is $E = 2.06$ GPa, class of the crane is A7, steel grade is Fe 430, stiffeners are 120*80*8 mm angle profiles.

The program is made in MS FORTRAN on PC type computer. In the expert system one part of the rules are concerning to the selection of the crane (see Fig. 1.). The second part is concerning to the selection of optimization techniques.

The weighting factors at the multiobjective optimization system and the uncertainty parameters at the expert system for the various objective functions are the same. There ranging is from 0 to 100 percent. It means the relative importance of the objective function.

The third part of the rules are concerning to the results of the optimization, to find the smallest objective function value, where the ratio of web height and flange width and the ratio of the two web thicknesses are acceptable. The first ratio is given near to the golden ratio, the second ratio has technological reasons.

$$0.4 \leq b/h \leq 0.8 \; ; \qquad t_{w1}/t_{w2} \leq 1.5 \qquad (3)$$

The result for a crane girder is determined with box girder section, asymmetrical section, stiffeners on the webs, two main girders, one trolley and the degree of interest of total cost = 0.4 , material cost = 0.3, labour cost = 0.3. Specific costs are: material cost $k_m = 1$ [$/kg], welding cost $k_w = 10$ [$/kg], surface preparation cost $k_s = 100$ [$/m^2]. The optima are as follows:

web height	$h = 1260.$ [mm],
main web thickness	$t_{w1} = 6.$ [mm],
secondary web thickness	$t_{w2} = 5.$ [mm],
width of the flange	$b = 700.$ [mm],
thickness of the flange	$t_f = 18.$ [mm],
total cost of the structure	$C_t = 16677.04$ [$]

The discrete value ranges of the variables are as follows: for h and b the step sizes were 20 [mm], for the thicknesses step sizes were 1 [mm].

3 Artificial neural networks (Ann)

The basic processing elements of neural networks are called artificial neurons, or simply neurons. One can simply call them nodes. Neurons perform as summing and nonlinear mapping junctions. The year 1943 is often considered the initial year in the development of artificial neural systems [13]. McCulloch and Pitts outlined the first formal model of an elementary computing neuron. The model included all necessary elements to perform logic operations, and thus it could function as an arithmetic-logic computing element. Donald Hebb first proposed in 1949 a learning scheme for updating neuron's connections that we now refer to as the Hebbian learning rule. He stated that the information can be stored in connections, and postulated the learning technique that had a profound impact on future developments in this field. Hebb's learning rule made primary contributions to neural networks theory.

During the 1950s, the first neurocomputers were built and tested. They adapted connections automatically. During this stage, the neuron-like element called a perceptron was invented by Frank Rosenblatt in 1958. It was a trainable machine capable of learning to classify certain patterns by modifying connections to the threshold elements. Neurons usually operate in parallel and are configured in regular architectures. They are often organized in layers, and feedback connections both within the layer and toward adjacent layers are allowed. Each connection strength is expressed by a numerical value called a weight, which can be modified.

One of the most promising applications of artificial neural networks is probably in the area of different classes of optimization problems [14,15,16]. The ability of analogue neuron-like networks to process simultaneously a large number of variables make it possible to find solutions for complex optimization problems in almost realtime. In fact, with the advent of analogue VLSI technologies and electrooptics it is feasible today to design programmable chips which can solve a specific optimization problem considerably faster than by using a sequential algorithm on a general purpose digital computer and sometimes even faster than an specialized hardware.

3.1 Backpropagation Neural Network

The simple feedforward networks have a layer of neurons to which the external stimuli are presented, a series of hidden layers, and a layer of neurons at which the output is available. The input neurons do not process the input stimulus, they simply serve as "fan-out" points for connections to neurons in successive layers. The presence of the hidden layer, and the nonlinear activation functions, enhance the ability of the networks to learn nonlinear relationships between the presented input and output quantities. This "learning" or "training" in feedforward nets simply requires the determination of all interconnection weights w_{ji} and bias parameters T_j in the network. Once such a trained network is established, it responds to a new input within the domain of its training by producing an estimate of the output response. Variations of the generalized delta error backpropagation algorithm have been used for this training; this scheme is essentially a special purpose steepest descent algorithm, and indeed, any optimization method can be used towards this end. The only concern would be the computational effort necessary for network training when the network size (number of independent networks parameters) increases [17]. A description of the training process is summarized here for completeness.

Consider a three-layer network the interconnection weights are first initialized randomly and the input pattern is presented to the network from which the network output is determined. This output is compared to the expected output and a sum of the squares of all errors is determined. The network parameters such as the interconnection weights and threshold levels are adjusted to minimize this error to some level of desired accuracy.

3.2 The structure of Neudesk Windows

Neural Desk version 1.1, Network development tool. NeuDesk is an interface program designed to permit simple breadboarding of neural nets. The program is "data driven" in that the user inputs the training set of the network, and the program will design a network to suit. The user can modify the network topology if he requires, and select from a range of different learning algorithms.

NeuDesk runs under MS Windows 3.x and follows the MDI standard as seen in Program Manager supplied with Windows. The Neudesk window consists of a title bar, a menu bar and a window frame. Within the window frame is the "Client Area" and in this area several child windows can coexist either opened or as Icons. The Neudesk Menu bar will change depending on which child window is active. The Windows menu enables one to arrange the child windows on the screen in a tiled or overlapped fashion, and enables one to easily move from window to window.

Child windows are of four types:

- Spreadsheet type window for data entry.
- Network display window showing the network designed, (Graph window).
- Control Window enabling control over the whole process of developing a network.
- Runtime Window taking the place of Neurun showing an error graph as it learns.

Four spreadsheet windows are created when the program is started, but there can only be one control, one graph, and one runtime window.

3.3 Design of plates stiffened on one side

3.3.1 Structure of stiffened plates

The cost function contains the material and the fabrication costs (Figure 3.). The volume of the plate is as follows

$$V = b^2 t_f + 4bht_r(\varphi + 1) \tag{4}$$

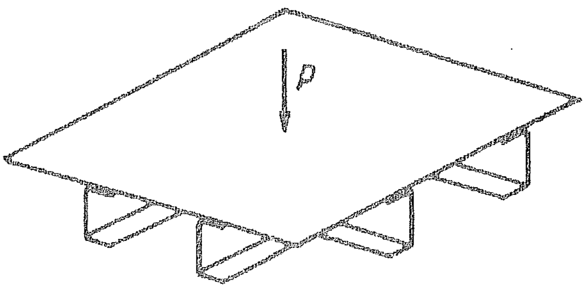

Fig. 3. Stiffened plate with simply supported ends

Fig. 3 and 4 show the general view and a node of a stiffened plate with an orthogonal grid of cold-formed channel ribs.

3.3.2 Design constraints of a stiffened plate

(a) *Constraints on compressive stress* in the central face plate element can be formulated as

$$\sigma_{\max.1} + \sigma_{f.\max} \le \sigma_{adm} \tag{5}$$

where $$\sigma_{\max.1} = \frac{c_m p b^2}{I_x / a} y_c \tag{6}$$

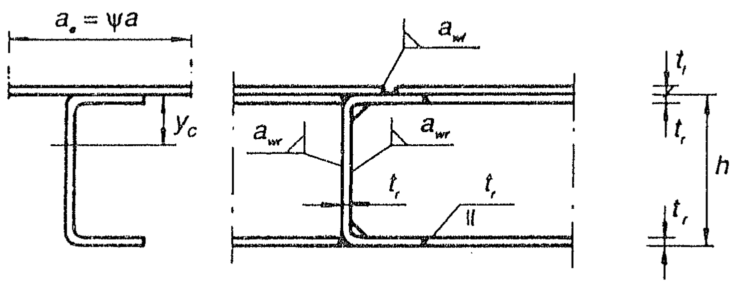

Fig. 4. Stiffener of the plate

According to Fig.4.2., the distance of the centroidal axis y_c can be calculated as

$$y_c = \frac{2h t_r}{2h t_r + a_e t_f} \cdot \frac{h}{2} = \frac{h}{2} \cdot \frac{1}{1+\alpha}; \qquad \alpha = \frac{a_e t_f}{2h t_r}; \qquad a_e = \psi a \tag{7}$$

Here simpler formula was used for ψ, proposed by Usami and Fukumoto [18, 19]

$$\psi = \frac{0.75}{\lambda_p}; \qquad \lambda_p = \frac{a}{t_f}\sqrt{\frac{\sigma_{\max.1}}{E}}; \qquad \psi \le 1. \tag{8}$$

The moment of inertia can be expressed as

$$I_x = a_e t_f y_c^2 + 2h t_r \left(\frac{h}{2} - y_c\right)^2 + \frac{h^3 t_r}{3} = \frac{h^3 t_r}{6} \cdot \frac{2+5\alpha}{1+\alpha}. \tag{9}$$

Substituting (6) into (8) we get again a quadratic equation for ψ, the solution of which is

$$\psi = \frac{5 \times 0.75^2 \varphi^3 t_f^3 h E}{12 c_m p b^4}\left(1 + \sqrt{1 + \frac{96 c_m p b^3 t_r}{25 \times 0.75^2 \varphi^3 t_f^4 E}}\right) \tag{10}$$

(b) *Constraint on shear buckling of rib webs* at the edge is the following

$$\tau = \frac{0.42 p b^2}{h t_r \varphi} \le \frac{\tau_{cr}}{\gamma_b} = \frac{5.34 \pi^2 E}{12\left(1-\nu^2\right)\gamma_b}\left(\frac{t_r}{h}\right)^2 \qquad \text{for} \qquad \frac{\tau_{cr}}{\gamma_b} \le \tau_{adm} \tag{11}$$

$$\tau = \frac{0.42 p b^2}{h t_r \varphi} \le \tau_{adm} \qquad \text{for} \qquad \frac{\tau_{cr}}{\gamma_b} > \tau_{adm} \tag{12}$$

where τ_{adm} is the admissible shear stress, γ_b is a safety factor, $\gamma_b = 1.35$,

and with the $\nu = 0.3$, the (11) can be written as

$$\tau = \frac{0.11748 p b^2 h}{E t_r \varphi} \leq 1 \tag{13}$$

(c) *Deflection constraint*

$$w_{\max} = \frac{c_w p_0 b^4}{E I_X / a} \leq w^* = c^* b. \tag{14}$$

According to Schade [20] for a torsional stiffness $H = 0$, $c_w = 0.0082$.

(d) *Size constraints* are the thickness limitations defined as follows

$$t_f \geq t_0 \qquad \text{and} \qquad t_r \geq t_0. \tag{15}$$

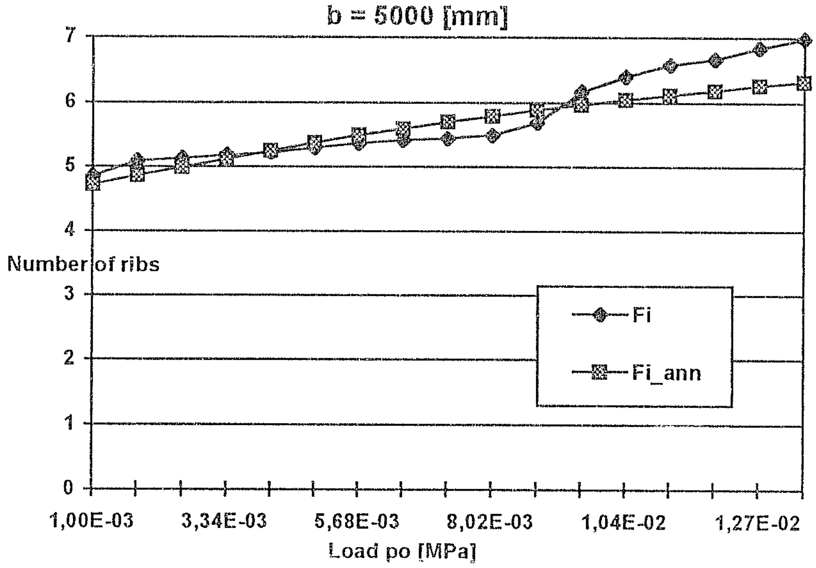

Fig. 5. Number of ribs in the function of the load with optimization (Fi) and with the neural network (Fi_ann)

3.3.3 Minimum cost design of a stiffened plate [18]

The number of structural elements is now

$$\kappa = \varphi^2 + \varphi(\varphi - 1) + \varphi + 3 = 2\varphi^2 + 3 \tag{16}$$

The manufacturing times T_1 and T_2 are computed in the following way

$$T_1 = 3\sqrt{\rho V (2\varphi^2 + 3)} = 265.80068\sqrt{V(2\varphi^2 + 3)} \tag{17}$$

$$T_2 = 2 \times 10^{-3} b a_{wf}^{1.5} \varphi + 1.6 \times 10^{-3} h(\varphi^2 + 1)(2a_{wr}^{1.5} + t_r^{1.5}) \tag{18}$$

The objective function is the following

$$\frac{C}{k_m} = \rho V + \frac{k_f}{k_m}(T_1 + T_2) \tag{19}$$

Fig. 5 shows the comparison of the results of the traditional (SUMT) method (Fi) and the results given by the NeuralDesk (Fi_ann), the number of ribs in the function of the load p_0.

4 Conclusions

Both applications show, that expert systems and Neural networks are useful in engineering calculations [22].

LEVEL 5 OBJECT asks for the unknowns during the computation, it knows what to ask for, more easy to jump from one level to another on the rules' tree and the optimum computation part is build into the expert shell.

Using NeuNet, we can optimize, but we can be quite sure that neural networks will not replace conventional computers. Basic reasons preclude neurocomputers replacing conventional ones for most tasks. Conventional computers are now very inexpensive. They are extremely fast and accurate for executing mathematical subroutines, text processing, computer-aided design, data processing and transfer, and many other tasks. In general, conventional digital computers outperform any other computing devices in numerical calculations.

The best hope for the widespread use of artificial neural systems, or neurocomputing, is in computationally intensive areas that are not successfully attacked by conventional computers. It seems that the areas requiring human-like inference and perception of speech and vision are the most likely candidates for applications. If these succeed, there will be more applications in real-time control of complex systems and other applications that we cannot yet anticipate.

Neural networks are also expected to be widely applied in expert systems and in a variety of signal processors. At present, such systems are available as aids for medical diagnosis, financial services, stock price prediction, solar flare forecasting, radar pulse identification, and other applications. As most researchers agree, future artificial neural systems are not going to replace computational and artificial intelligence simulations on conventional computers either. Rather, they will offer a complementary technology. The ultimate goal may be to exploit both technologies under the same roof, while presenting a single, flexible interface to the user [14,17].

In the long term, we could expect that artificial neural systems will be used in applications involving vision, speech, decision-making, and reasoning, but also as signal processors such as filters, detectors, and quality control systems. Applications are expected especially for processing large amounts of data. Also, neural networks may offer solutions for cases in which a processing algorithm or analytical solutions are hard to find, hidden, or nonexistent.

Acknowledgement

The research work was supported by the Hungarian Scientific Research Found grants OTKA 22846 and OTKA 19003. I would like to acknowledge the calculation work of my student Zoltán Lengyel.

References

1 Gero,J.S. (ed): *Expert systems in computer-aided design*. Elsevier Applied Science Publishers, 1987.

2 Harmon,P., Sawyer,B.: *Creating expert systems for business and industry*. John Wiley and Sons Inc. 1990.

3 Dym,C.L.,Levitt,R.E.: *Knowledge based systems in engineering*. Mc Graw-Hill Inc. 1991.

4 Garrett,J.H.: Knowledge-based expert systems: past, present and future. IABSE Surveys S-45/90, Zürich, 1990. 21-40.

5 Balasubramayan,K.: A knowledge based expert system for optimum design of bridge trusses. University Microfilms International, Dissertation Information Service, Ann Arbor, Michigan, 1990. No. 8812223.

6 Jármai,K.: Single- and multicriterion optimization as a tool of decision support systems. Computers in Industry, Elsevier Applied Science Publishers, 1989. **11**, (3) 249-266.

7 Jármai,K.: Decision support system on IBM PC for design of economic crane girders. Thin-walled Structures, Elsevier Applied Science Publishers, 1990. **10**, 143-159.

8 Level 5 Object, Reference Guide, FOCUS Integrated Data and Knowledge-Based Systems, Information Builders, 1250 Broadway, New York, 1990.

9 BS 2573: *Rules for the design of cranes. Part 1:* Specification for classification, stress, calculations and design criteria for structures. British Standard Institution, London 1983.

10 BS 5400: *Steel, concrete and composite bridges. Part 3*: Code of practice for design of steel bridges. British Standard Institution, London 1983.

11 Fausett, Lauren: *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*, 1994. A Paramount Communication Company

12 Ghaboussi, J., Garrett, J.H., Jr., and Wu, X.: Knowledge-Based Modelling of Material Behavior with Neural Networks, Journal of Engineering Mechanics, 1991. **117** (1), 132-153.

13 Hassoun, Mohamad H. 1985. Fundamentals of Artificial Neural Networks, The MIT Press Cambridge, Massachusetts, London, England

14 Hajela, P. and Berke, L.: Neurobiological Computational Models in Structural Analysis and Design, Computers and Structures, 1991. **41**, (4), 657-667.

15 Berke, L., and Hajela, P.: Application of Artificial Neural Networks in Structural Mechanics, Structural Optimization, 1992. **3**, (1).

16 Hajela, P. and Berke, L., Neural Network Based Decomposition In Optimal Structural Synthesis, Computing Systems in Engineering, 1991. **2**, (516) 473-481.

17 Brown, D.A., Murthy, RL.N., and Berke, L.: Computational Simulation of Composite Ply micromechanics Using Artificial Neural Networks, Microcomputers in Civil Engineering, 1991. **6**, 87-97.

18 Farkas,J.,Jármai,K.: The effect of low cost technologies in the design of welded structures. Welding in the World, Pergamon Press, 1994. **34**, 123-130.

19 Usami,T.& Fukumoto,Y. Local and overall buckling of welded box columns. *J.Struct.Div. Proc. ASCE* 1982. **108**, 525-541.

20 Schade,H.A., Design curves for cross-stiffened plating under uniform bending load. *Trans. Soc. Naval Archit. Marine Eng.* 1941. **49** 154-182.

21 Eurocode 3 Design of steel structures Part I: General rules and rules for buildings. 1990.

22 Farkas,J.,Jármai,K.: Analysis and optimum design of metal structures. Balkema Publishers, Rotterdam, Brookfield, 1997, 347. ISBN 90 5410 669 7.