

Tracing Engineering Evolution with Evolutionary Algorithms

Tino Stanković, Kalman Žiha and Dorian Marjanović
*University of Zagreb/Faculty of Mechanical Engineering and Naval Architecture
Croatia*

1. Introduction

The mankind achieved an astonishing technological development through centuries of innovation, creation and continuous improvement. The history of engineering is the inherent component of the civilization. Moreover, outstandingly important lessons for further development can be studied in the history of engineering. The investigations of the recent complex engineering knowledge, experience, analytical and computational tools may serve to explain the technical progress and facilitate the future development. For this purpose this chapter will present how the evolutionary algorithms can simulate the developing complexity of engineering reasoning that in reverse can back-trace the primitive origins of modern technical products. The chapter will resume the evolutionary algorithms as well as the evolutionary optimization and design processes based on innovative and creative activities with the aim to define their potentialities in discovering the evolution of engineering products. More so when adding to the whole process the touch of randomness introduced in form of mutation operator the algorithm gains the property to converge to the global optimum within multi-modal search space. Both of these processes crossover and mutation have been present in the natural evolution for eons of time. From an algorithmic perspective crossover and mutation enable adaptation of the population of feasible solutions to the imposed environment conditions of the search spaces. The chapter will concentrate on the multi-objective optimization problems taking for example the NSGA-II algorithm (Deb, 2001). The result will be obtained as the population of optimal solutions distributed along the Pareto frontier. Using constraint domination condition and constrained tournament selection operator the evolution of the object under consideration will be explained. Normally engineering relies on the design process that is for technical purposes modelled as a set of cyclic activities put in a logical order to guide the procedure until the desired technical aim is reached. The design process is comprehensible as a shortcut to a satisfying product that is also in clear correlation with the formulation of an algorithm, particularly with evolutionary algorithm. The evolutionary methods affect design process and teach about process itself. They stimulate innovation and creativity during the human efforts to design and apply processes which are all in reality an attempt to produce an unbiased human performed heuristic search. The evolutionary design relies on the normal contemporary progressive engineering reasoning aspired with achievement of highly efficient products providing appropriate safety levels by employing genetic algorithms. The chapter will consider how the reverse process to the technical progress can reconstruct the origins of contemporary products using evolutionary

algorithms on engineering models in two manners. Chapter will present two examples. The first case study in this chapter investigates the evolution of a truss structure possibly in the future up to the very important engineering entity – the wheel. The second case study back-traces the development of the stiffened shells in aerospace and shipbuilding industry to their primitive origins in boats originally made of carved-out log.

2. Evolutionary Algorithms

C. Darwin wrote that evolution begins with the inheritance of the gene variations. Inspired by the natural evolution in the mid 20th century the field of evolutionary computation emerged to its dawn and new class of algorithms was born. By the principle of the survival of the fittest, solution or set of solutions to given problem evolve in time using fitness - objective function that is, as an evolutionary guide. During the search new solutions are generated by reusing and mixing together pieces of the past solutions. Like with the living organisms – information in digital computer comprehensive manner was being exchanged between the most feasible solutions. Important class of evolutionary algorithms, genetic algorithms resembled natural evolution in the most (Yokey, 2005). Information exchange between solutions was done by exchanging binary number strings by crossover operators. Information chunk exchange was described in well known Building Block Hypothesis (Goldberg, 1989). Goldberg's attempt of proving the convergence of heuristic genetic algorithm is in line with the mid 20th century genetics theories. It is known that information exchange during forming of amino acids is also linear and digital like in computers and it is build from chunks of information (Yokey, 2005). Since they were not calculus based application of such genetic algorithms was soon to be recognized as they were applied as general optimization problem solvers. Range of applications included combinatorics (scheduling, TSP problem, close packing problems), various engineering optimization problems (single or multy-objective optimization), neural network trainers etc.

Evolutionary algorithms own their properties and behavior to the process that they are trying to mimic in order to find solution - the natural evolution of living organisms. The solution or the set of solutions to the given problem evolves in time from the feasible solution population by the principle of the survival of the fittest – selection operator, with the fitness function acting as the evolutionary guide. The discreteness of algorithm is devised from its crossover operators, which when generating new solutions, are reusing and mixing together pieces of the past solutions making it very useful when dealing with non continuous problems. Such usage of the past knowledge described by Goldberg in the Building Block Hypothesis (Goldberg, 1989) gives to the algorithm property to converge to desired better solution to a given problem, which ultimately distinguish it from the plain random walk algorithms. More so when adding to the whole process the touch of randomness introduced in the form of mutation operator, the algorithm gains the property to avoid the pitfalls of local optima. Both of these processes, crossover and mutation, have been present in the natural evolution for eons of time. From an algorithms perspective crossover and mutation enable adaptation of the population of feasible solutions to the imposed environment conditions of the search spaces. The recent 15 years have presented a significant number of methods and tools (Goldberg, 2002) for application in engineering. The general multi-objective optimization problem is tackled by the NSGA-II algorithm (Deb, 2001) that is implemented as a dynamic-link library in C# within Microsoft .NET Framework 2.0. to provide a generic multi-objective solver for various optimization models

in engineering. NSGA-II algorithm generates populations of optimal solutions distributed along the Pareto frontier, using constraint domination condition and constrained tournament selection operator (Deb, 2001). Normally the design process is structured as a set of cyclic activities put in a logical order to control and guide the procedure until the desired aim is reached (Goldberg, 2002).

3. Evolutionary optimization

Evolutionary algorithms (EA) have been used as a general optimization tool for technical systems ranging from general single objective benchmark optimization cases, such as Golinski's problem (Golinski, 1970), to multi-objective genetic algorithms (Tan, Lee&Khor, 2002), (Deb, 2001). Evolutionary algorithms provide a solution or a set of solutions of optimization problems in discrete time using the fitness function and the simple prime mechanisms, crossover, mutation and selection. The crossover operator is in fact a great recombination machine that combines bits and pieces of past solutions into a new sequence. The binary encoded genotype in the form of binary string sequences is shuffled throughout the evolution. Applicability as a discrete problem solver of the genetic algorithm is a direct result of the crossover operator. With the introduction of a reasonably small degree of randomness in the form of a bit-flip mutation operator, the algorithm gains the property to avoid pitfalls of local optima. Both of these processes have been present in the natural evolution for. Crossover and mutation enable the adaptation of the feasible solution population to the imposed environmental conditions of the search spaces. Selection, the third operator, acts as a collective learning enforcer, which will ruthlessly guide the evolution by means of the survival of the fittest. Fitness merit is assigned after the evaluation of the objective function to each and every population member. The collective learning process and a possibility to impose the search strategy distinguish GAs from the plain random walk algorithms (Bäck&Fogel, 2000). The research of evolutionary computation-based tools for enhancing the design optimization is therefore reasonable and justified because real engineering design search spaces are often multimodal, full of discontinuities and constrained. The range of applications (Bäck&Fogel, 2000) included planning (scheduling, TSP problem, close packing problems), simulations (behavior prediction), recognition, and control (adaptation and evolvable hardware).

4. Evolutionary design

The design process is comprehended in this text as a shortcut to a satisfying product using general and personalized knowledge and experience of design modeling in order to accelerate the technical development which naturally should occur evolutionary in spacio-temporal and social circumstances. The design process can be put in correlation with the formulation of an algorithm as an iterative problem solving procedure involving a finite number of steps. One could define such a procedure as a search algorithm where the search space itself is built on lists of requirements or design variables and constraints - the problem or design task formulation, and the search for the feasible solution is being conducted by iteration, abstraction, concretization and improvement (Goldberg, 2002). All of these four processes are built in core of an evolutionary algorithm. They are iterative - searching for solution during each new generation, abstracting - a common practice in multi-objective optimization where the objectives are put in order by degree of importance and evaluated

respectively (Bäck&Fogel, 2000), concretizing - in order not to hinder the process the objectives can be introduced at a desired point in evolution when solutions are evolved enough, improving - by evolving solutions in every generation using selection, crossover and mutation operators. The evolutionary methods may provide enhancement of design process or findings about process itself. Properties of search spaces will depend on complexity of the design aim and could be constrained, multimodal and full of discontinuities. Many applications of evolutionary algorithms in search spaces have been recognized (Bentley, 1999), (Golinski, 1970), (Tan, Lee&Khor, 2002). Various methods enhance design innovation and creativity such as Delphi method, 635 method and synectics (Pahl&Beitz, 1988), (Wood&Otto, 1999), or brainstorming that support an unbiased human search for technical solutions. The evolutionary algorithms for this purpose use the form of mutation operator which stochastically alters feasible solutions. It can be hypothesized that in order to produce innovative solutions a design process as well as the natural evolution should be performed without a bias. By using evolutionary design the designer is shaping and adjusting his designs enabling their existence in constraint bounded design space similarly to the principles recurring in natural evolution.

5. Evolution of truss structures

In the realm of genetics, natural evolution and information theory, the problem of information encoding and decoding is permanently reconsidered (Yockney, 2005). A common point of all three mentioned researched areas is how to design a specific encoding of sequences carrying information, or how to decode them properly from a noisy environment to an error-free state. Genotype encoding and its counterpart, decoding into a phenotype, present a special point of interest in the evolutionary computation community (Goldberg, 2002) and (Bentley, 1999). Notions regarding overall convergence and phenomena that occur during the information exchange between chromosomes were first tackled in Goldberg's famous "building-block hypothesis". Explored further by the same author (Goldberg, 1989), an attempt was made to evaluate the quality of binary string building blocks when used in different classes of problems. Evolutionary algorithms are considered to be robust due to both their operators and their easy customization to suite different areas of application. However, to accomplish robustness specific to encoding, decoding and phenotype representations must be created. To found out the right answers to the addressed problem, evolutionary algorithms must have a good material to work with. In the paper, a new encoding/decoding scheme will be presented. The results point out that the scheme is well suited for the structural optimization of truss structures.

The applicability of evolutionary and genetic algorithms as optimization tools is elaborated in the next case study. Then, a state-of-the-art overview of the truss structure optimization and research motivation for using advanced methods is presented. Drawbacks of each method, resulting from different problem approaches are also addressed. The pseudo-code of the proposed encoding for the 2-D continuous domain is presented.

The aim of the research is to move away from the orthodox structural continuous optimization approach, where most of topology is initially fully predefined or locally constrained (Hasançeb, 2007), (Coello&Christiasen, 2000). In such a way, the search space is reduced, thus inhibiting evolution to progress towards new uncovered solutions. Such genotypes are easily coded because one can predict nodal inter-arrangements. They do not cover the possibilities of the initial randomness of the structure shape; instead, they optimize the usual truss bound design variables (cross-sectional area, length etc.). By contrast, a different type of coding is

proposed in the topological optimum design (TOD) (Jakiela et al., 2000), (Hamda&Schoenauer, 2002) and (Kim&Weck, 2004). The structure is represented in a discrete domain, in the form of material distribution, which is a straightforward approach to the shape optimization. The genotype encodings are done through matrices, Voronoi representations (Hamda&Schoenauer, 2002), or even by using 3-D FEM building blocks. In the same manner, the phenotype representation then visually depicts the resulting structure. An interesting cantilever optimization problem has been elaborated by Kim and de Weck (Kim&Weck, 2004), who addressed the quality of search with the chromosomal length (Goldberg, 1989). They increased the domain resolution throughout the evolution course. By taking this approach, they (Kim&Weck, 2004) also addressed the design concretization defined in literature (Hubka, 1992), (Pahl&Beitz, 1988) as progression from abstract to concrete through the design process stages. Although TOD is computationally demanding, it optimizes the structure in the form of the in-domain material distribution, but the other design variables, such as the cross-sectional area, remain predefined or out of reach. A more subtle approach using the shape annealing (SA) method and shape grammars for structural optimization purposes was proposed by (Shea&Cagan, 1998). Shape grammars, as their linguistic fundament (Chomsky, 1957), provide language and in this case a design language for the structure shape manipulation. They are driven by a simple set of production IF-THEN rules. The evolution begins with the initial structural member expanding and growing slowly through the rule implementation to a complete structure. However, to search all possible truss structures that can be constructed within a search domain in order to obtain a global optimum, than the rule set must be adequate to enable the emergence of such solutions within a design language. So for generic approach, to evolve structures one should be able to evolve the rules (Gero&Louis, 1995). The genotype encoding and decoding should enable the search space to be as large and unconstrained as possible. Genotype are a collection of binary encoded nodes and the phenotype represented as truss structure is then defined as a result of the inter-nodal arrangement. Although this idea is straightforward, it has not been explored so far. The problem in the 2-D continuous domain is depicted in Fig. 1.

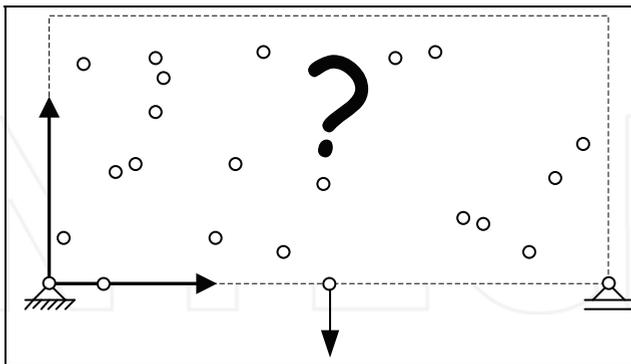


Fig. 1. Random nodal arrangement in 2-D continuous domain - predefined nodes in supports and in force node

There is no problem for the genetic operators to function properly for an ordinary binary encoded string, but getting a structure out of such genotype encoding poses a more serious difficulty.

There exist a number of connections between nodes as the ones presented in Fig. 1. In fact, the number of combination is enormous and it strongly depends on what is tried to be accomplished. How to connect 2-D collection of nodes (for a prototype planar collection of nodes see Fig. 1) with the resulting truss structure at the phenotype level is elaborated in the following chapter.

5.1 Proposed coding scheme for 2-D continuous domain

Genotype and phenotype representation coding schemes are proposed with some restrictions and are based on the following assumptions:

- Firstly, the structural model is defined as a FEM model. In each evolution turn for every new population member the system stiffness matrix is re-assembled. In the matrix, truss (rod) elements are defined as a consequence of the inter-nodal arrangements. Hence, a new structural model is to be generated according to the current system topology.
- Secondly, to use common knowledge in the design of truss structures and to avoid possible singularities of the system stiffness matrix, the system components are always arranged in triangular schemes (see Fig. 2). To clarify, it is a widely known fact that the triangle substructure presents a stiff building block common in engineering. Such a restriction narrows the search space and enhances the conversion of the algorithm by eliminating the known unfeasible and mathematically singular problems. By following the described procedure the result would not always be a structure bounded by a convex polygon.
- And finally, there exist a number of predefined nodes. These nodes have defined positions in a given 2-D domain. Predefined nodes are always supports and nodes with force vector (see Fig. 1). The total number of fixed nodes is given by $NoNx$, and the total number of free nodes is given by NoN .

5.1.1 Genotype encoding

Genotype is encoded with binary strings. Nodes are coded as shown in Table 1. Chromosome is then represented as a collection of nodes (see Table 2.). All of the genetic operators are easily applicable to such coding.

<i>Node_j</i>	
<i>x</i> coordinate - binary encoded string <i>lj</i>	<i>y</i> coordinate - binary encoded string <i>lj</i>

Table 1. Encoding of node

<i>Chromosomes</i>					
<i>Node1</i>	<i>Node2</i>	, ...,	<i>Nodej</i>	, ...,	<i>Noden</i> ($n = NoN + NoNx$)

Table 2. Chromosomes structures

Crossover is made by randomly selecting a crossover point among nodes on the chromosomal level (see Table 2), and then by selecting another crossover point on the nodal level (see Table 1). Mutation is performed easily, in a bit-flip manner, directly altering the nodal position.

5.1.2 Genotype decoding and phenotype representation

For every evaluation during the evolution, a genotype must be decoded into a phenotype. The algorithm for getting a phenotype, i.e. defining the rods and truss structure, based on the nodal positions is presented in the pseudo-code below:

```

1 sort nodes ascending over x, if equal compare over y
2 move to first node in chromosome  $j \leftarrow 0$ 
3 while ( $j < n - 2$ ) do
  A. if  $\text{node}(j).y \geq \text{node}(j + 1).y$ 
    for rod FEMs definition consider nodes( $j + 1, j + 2, \dots, j + k$ ) that are ordered
    ascending over y do
      break the search if  $\text{node}(j + k + 1)$ :
        a. is not ordered ascending over y
        b. is second node in ascending order satisfying  $\text{node}(j + k + 1).y > \text{node}(j).y$ 
        c.  $j + k + 1 > n$ 
    od
  B. else: do the same as in A but considering the descending order of nodes
  C. define rod FEMs between  $\text{node}(j)$  and all found nodes within A or B
  D. move to next node  $j \leftarrow j + 1$ 
4 od

```

The algorithm starts with all of the nodes being sorted ascending based on their x coordinate. From collection of nodes the first node is taken into consideration by setting counter j to zero (pseudo-code line 2). In the following *while loop* marked by number 3, the algorithm will search for possible ways to define FEM rod elements between considered node and all the nodes having greater x coordinate. Resulting structure must be triangular with no FEM elements intersections. Inside the loop two possibilities exist (marked with letters A and B); based on its position the first following node can be below or on equal height (A) or above the considered node j (B). Inside A all of the nodes will be ranked feasible for FEM definition if they do not violate the conditions inside a, b and c. Condition a takes into an account weather all of the following $j + k + 1$ nodes are in ascending order over y , b breaks the search if the node is the second one above the node j and c prevents the counter being larger the overall collection of nodes n . B takes into account situation opposite of A - the first following node being above node j thus considering the decreasing order over y . Afterwards the FEMs are defined and whole procedure is repeated for node $j + 1$.

The singular conditions that occur when two or more nodes occupy same position are regulated with general constraints. The results of inverting the order of sorting in the way that the sorting is first conducted over the y and then over the x node coordinate (pseudo-code line 1), were not explored in this paper. Besides for the reasons of common practice, the procedure runs first over x and then over y . On Fig. 2, a truss structure phenotype is depicted corresponding to the nodal arrangement already shown in Fig. 1.

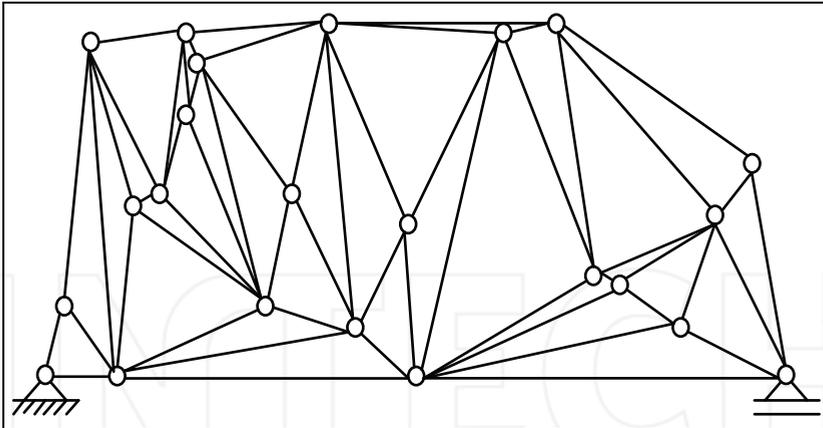


Fig. 2. Truss structure obtained from nodal positions shown in Fig. 1

5.2 Structural FEM model

The structure is modeled with FEM planar trusses with 6 degrees of freedom. It is necessary to introduce bending to trusses and implicitly convert them into beams. The result of the evolution with infinite stiffness to bending will always converge to a single horizontal rod. Such structure would have zero displacement since it cannot bend, it would be minimal in mass since it is just a horizontal line. Normal forces would also be equal to zero if the force vector is put vertically as in Fig. 3-5.

Respective force F and displacement δ vectors per element are given as follows:

$$\{F\} = \begin{Bmatrix} N_1 \\ Q_1 \\ M_1 \\ N_2 \\ Q_2 \\ M_2 \end{Bmatrix} \quad \{\delta\} = \begin{Bmatrix} u_1 \\ w_1 \\ \phi_1 \\ u_2 \\ w_2 \\ \phi_2 \end{Bmatrix}$$

The element stiffness matrix K is common (Zienkiewicz, 1971) and is given here by:

$$[K] = \begin{bmatrix} \frac{EA}{l} & & & & & \\ & \frac{12EI}{l^3} & \frac{6EI}{l^2} & & & \\ & & \frac{4EI}{l} & & & \\ & & & \frac{EA}{l} & & \\ & & & & \frac{12EI}{l^3} & \frac{6EI}{l^2} \\ & & & & & \frac{4EI}{l} \end{bmatrix}$$

symmetrical

The vector of the nodal displacements u of the evolved structures is calculated by solving the usual linear equation system given by:

$$\{F\} = [K]\{u\}$$

5.3 Optimization model

In this optimization case the goal is to find an optimal distribution of trusses that comprise truss structure for a given 2-D domain. The result will be a structure that is a result of its interaction with the environment - the objective function, design space conditions and constraints. For the reasons of simplicity a number of involved nodes are given by the user as a process input parameter. Since the algorithm uses fixed length chromosomes nodes cannot extinct during the course of evolution.

Next, all the trusses have the same fixed cross-section area therefore significantly reducing the search space. These parameters will be introduced as variables in the future work. Finally, the optimization problem is formulated as follows:

$$\left. \begin{array}{l} \min [m(F, A, I, E, NoN, NoNx, BC)] \\ \text{subject to:} \\ \delta \leq \delta_{\max} \\ l \geq l_{\min} \\ \sigma \leq \sigma_{\max} \\ x \in [x_{\min}, x_{\max}] \\ y \in [y_{\min}, y_{\max}] \end{array} \right\}$$

The objective function is the minimization of structure mass m .

The optimization parameters are given as follows:

- F – force vector in vertical direction,
- A, I, E – truss cross-section properties and material Young's modulus,
- NoN – number of free nodes,
- $NoNx$ – number of fixed nodes,
- BC – boundary conditions – number and type of supports at particular nodes.

Problem variables:

- x and y coordinates of each node considered,
- δ – absolute deflection vector,
- l – length of respective rod.

Constraints are defined as the maximally allowable absolute nodal deflection δ_{\max} and the minimally allowable beam length l_{\min} . The domain or design space is defined as a 2-D bounding box.

5.3.1 Implementation, control parameters and evolutionary operators

The applied genetic algorithm is a struggle genetic algorithm. Reasons for applying this particular GA lay in its elitist steady-state evolution. Only the best solution from the offspring population replaces the closest solution from the parent population if it is better. The distance between solutions is measured in the Euclidian objective space. The struggle GA manages to maintain diversity during the evolution process, thus preventing premature convergence. The control parameters of algorithm (Bäck&Fogel, 2000) are as follows:

- population size $\lambda = 60$,
- offspring population $\lambda = \lambda$,
- one point crossover probability $p_c = 1$,
- bit-flip mutation probability when static $p_m'' = 0.02$.

Mutation operator was introduced to help guide and boost the process in order to avoid local optima in the early stages of evolution. In literature, there exist a number of such mutation approaches (Goldberg, 1989), (Bentley, 1999), (Deb, 2001) and (Bäck&Fogel, 2000). This one is driven by the notions from the natural evolution, where initial mutation rates were much higher because of the imposed environmental conditions. Bit flip mutation rate is a function of its initial rate $p_m' = 0.1$ and the number of evolution iteration N . Mutation rate is simply linearly scaled over a desired number of iterations by the following formula:

$$p_m = \begin{cases} \frac{p_m'' - p_m'}{1000} N + p_m' & N < 1000 \\ p_m'' & N \geq 1000 \end{cases}$$

Constraint handling is done by measuring how potential solutions violate constraints. Two populations of feasible and unfeasible solutions were ranked accordingly; the feasible one sorted ascending over the objective function and the latter sorted in the same manner but over the violation measure. Constraint violation measure $\Omega(x^{(i)})$ (Deb, 2001) of i -th solution $x^{(i)}$ is derived as the summation of normalized violations $\omega_j(x^{(i)})$:

$$\Omega(x^{(i)}) = \sum_{j=1}^3 R_j \omega_j(x^{(i)})$$

No violations were favored so the weighting factor used is $R=1$ for all constraints. Every chromosome is a collection of n nodes. For encoding of the nodal x and y coordinates, binary strings were used. In addition to the refinement of the search, the Gray coding was applied (Bäck&Fogel, 2000).

5.4 Results

The results were obtained after roughly 1000 iterations in each of the presented examples. The evolution was conducted on a PC with the AMD Athlon 64 X2 5000+ processor. For the sake of further research, object-based dynamic-link libraries were designed in C# (MS .NET Framework 2.0) to provide a generic multi-objective solver for various engineering optimization models. The results of the optimal truss structure on the following three figures (Fig. 3-5) present the course of evolution under different initial conditions. The number of free nodes is increased as the load is increased in the force node.

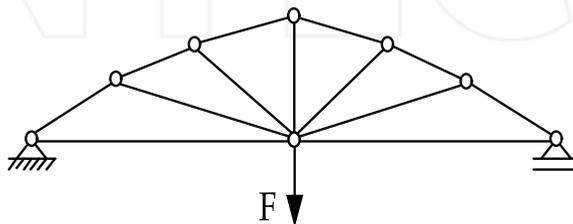


Fig. 3. Optimal structure with 5 free nodes

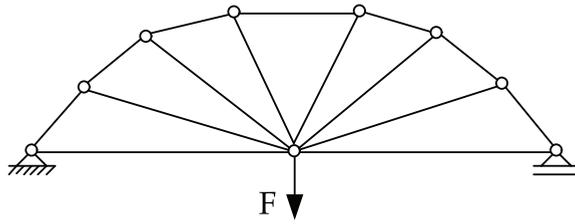


Fig. 4. Optimal structure with 6 free nodes

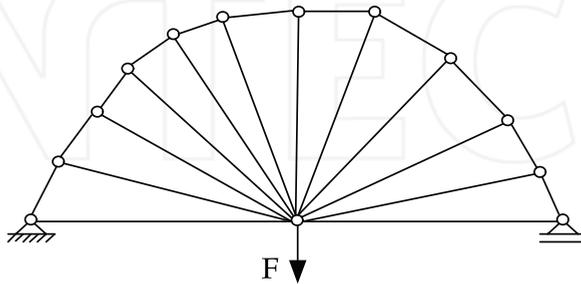


Fig. 5. Optimal structure with 10 free nodes

Of course, this is a subjective approach. There is an obvious correlation between the number of nodes, the amount of load imposed and the cross-section of the respective rod. At this point of research it was impossible to tackle all of the possible influences since the research focus was on a new type of encoding.

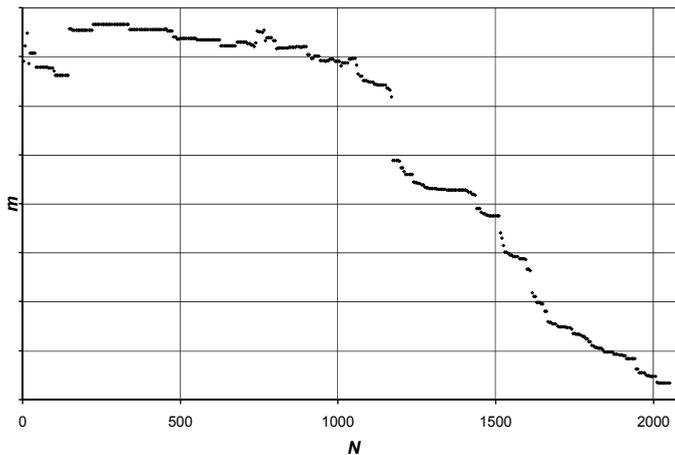


Fig. 6. Graph of the algorithm performance shown in terms of objective function m put against evolution step N

Graph of the algorithm performance shown in terms of objective function m and number of evolution steps N is presented in Fig. 6. This particular graph corresponds to the initial

condition of five free nodes. Every fifth solution is drawn in graph. Because of its stochastic nature algorithm cannot produce same performance graph in every run, however some key features remain. In Fig. 6 it is clearly visible that after the step $N=1100$ the value of the objective function stepwise diminishes. With sufficient certainty it could be said that the algorithm moved search to the feasible space. Before the $N=1100$ the aim of the algorithm was the minimization of the constraint violation expressed in equation (6) resulting in dispersion of the values of the objective function. After the step $N=2000$ the evolution slows down and no significant improvements were noted.

6. Evolution of stiffened panels

The simplified ship hull structure in this case study, see for example a traditional boat in Fig. 7, is modeled as a transversely framed shell of isotropic material under lateral outer pressure p , and longitudinal in-plane stress σ_L (Hughes, 1972), Fig. 8, also considering the state of the art rules and regulations of classification societies based on experience of shipbuilding and shipping (CRS, 2006) (DNV, 1978) that evolved during a long period of development of theory and practice of shipbuilding. The material properties are the elastic modulus E , the Poisson's ratio ν , the allowable normal σ_a and shear τ_a stresses in shell and in framing (CRS, 2006).

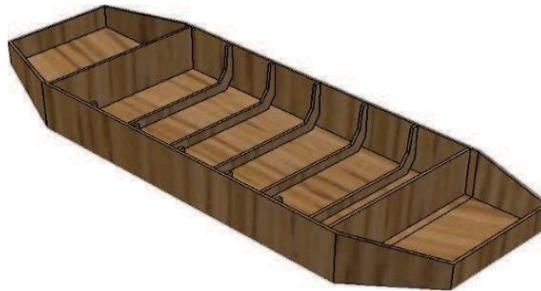


Fig. 7. Boat hull structure

The small deflection elastic plate bending theory (Hughes, 1972) defines the maximal local stress under lateral pressure p in the middle of the longer edge ℓ in the direction of the shorter edge s in the plating of thickness t clamped at stiffeners, Fig. 8. Using the semi-empirical plate side aspect ratio (Hughes, 1972) $k_s = 1 - 0,4 \left(\frac{s}{\ell} \right)^2$, the stress in the shell under lateral load p can be assessed as:

$$\sigma_p = 0,5 \cdot p \cdot k_s \cdot \left(\frac{s}{t} \right)^2 \quad (1)$$

The simple elastic beam bending theory (Hughes, 1972) defines the normal stresses in frames, Fig. 8:

$$\sigma_f = p \cdot k_m \cdot \frac{s \ell^2}{W_{f,e}} \quad (2)$$

The end connection factor for clamped frame ends is $k_m=1/12$. The elastic section modulus $W_{f,e}$ of a single frame accounts for the width of the effective plate flange. The shear stress at supporting ends of the frame web (Hughes, 1972), taking the correction factor $c_w=3/2$ for rectangular cross sectional area A_f of a flat bar (CRS, 2006) (DNV, 1978) is as shown:

$$\tau_f = \frac{c_w \cdot p \cdot s \cdot \ell}{2A_f} \quad (3)$$

The orthotropic plate elastic bending theory (Hughes, 1972) defines the stresses in the edges of the longer side in the direction of the shorter edge, Fig. 7, of the whole transversely stiffened plate as:

$$\sigma_s = K \cdot p \cdot \frac{s \cdot \ell^2 \cdot e}{I_f} \quad (4)$$

In (4), I_f is the frame moment of inertia including effective plating width and e is the distance from the neutral axes to the plating. From Shade's diagrams (Hughes, 1972) is $K=0.0916$ for the edges of the longer side in the direction of the shorter edge and $K=0.0627$ for the edges of the shorter side.

The critical buckling stress of plating under in-plane compression of plates between frames (Hughes, 1972), (CRS, 2006), (DNV, 1978) using the term $\sigma_{p,e} = \frac{\pi^2 E}{12(1-\nu^2)}$ is:

$$\sigma_{p,c} = \sigma_{p,e} \cdot \left(\frac{t_p}{s} \right)^2 \cdot k_p \cdot k_\sigma \quad (5)$$

For transversely stiffened panels is $k_p = \left[1 + \left(\frac{s}{\ell} \right)^2 \right]^2$ and for longitudinally stiffened panels is $k_p = 4$. For elastic buckling is $k_\sigma = 1$ and for plastic buckling is $k_\sigma = 1 - \left(\frac{\sigma_{c,e} - \sigma_y / 2}{\sigma_{c,e}} \right)^2$ when $\sigma_{c,e} \geq \sigma_y / 2$.

The torsional buckling of flat bar stiffeners prevents the empirical ratio of height to thickness (DNV, 1978) that is normally < 20 .

The ultimate bending strength with respect to multimodal plastic failure modes of plates at the mid of the longer edge of unit plate plastic section modulus $W_{p,p} = \frac{t^2}{4}$ between frames under bending moment $M = k_m \cdot p \cdot s^2$ acting due to lateral pressures p combined with in-plane load σ_L , may be expressed by the following interaction formula (DNV, 1978)

$$\frac{M}{\sigma_y \cdot W_{p,p}} + \frac{1}{\beta} \left(\frac{\sigma_L}{\sigma_y} \right)^2 = \beta.$$

The usage factor β relates the maximal permissible load to the collapse load. Using the

factor $k_{L,p} = \left[\beta - \frac{1}{\beta} \left(\frac{\sigma_L}{\sigma_y} \right)^2 \right]$ to represent the influence of the in-plane stress, the ultimate

lateral pressure on plating accounting for the yield stress σ_y (DNV, 1978) is:

$$p_{u,p,\sigma} = 3 \cdot \left(\frac{t}{s} \right)^2 \cdot \frac{k_{L,p}}{k_s} \cdot \sigma_y \tag{6}$$

The ultimate bending strength of frames under lateral pressure and axial stress is the capability to prevent the plastic failure defined as a three-hinged mechanism (DNV, 1978). For frames with plastic section modulus $W_{f,p}$ including the effective plate flange under bending moment $M = k_m \cdot p \cdot s \cdot \ell^2$ due to lateral pressure p and for small axial stresses σ_x (the shear is usually small) the relation derived from (2) holds (DNV, 1978):

$$p_{f,u,\sigma} = 12 \cdot \frac{W_{f,p}}{s \ell^2} \cdot \varepsilon \cdot \sigma_y \tag{7}$$

where ε is the permissible usage factor.

The ultimate lateral pressure on the whole panel viewed as the orthotropic plate (4), is as shown:

$$p_{b,p,\sigma} = \frac{\sigma_y}{K} \cdot \frac{I_f}{s \cdot \ell^2 \cdot e} \tag{8}$$

Since the transverse in-plane compression of bottom plating is normally small, Fig. 7, it is not likely that buckling of plating occurs at all (DNV, 1978).

The model is a ship hull panel of thickness t , length ℓ , width b which is transversely stiffened by n flat bars of thickness t_w and height h_w at spacing s , Fig. 8. The plate is laterally loaded by pressure p and with in-plane stress σ_L .

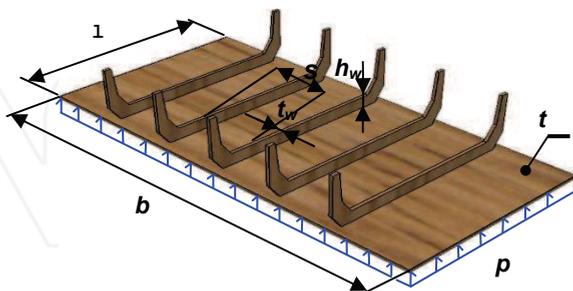


Fig. 8. Panel structural model

The evolutionary design in this example uses the engineering model in order to demonstrate the technical development by employing genetic algorithms aspired with achievement of appropriate safety level as well as with reduction of weight, expenses and production efforts using different materials.

Therefore the stiffened panel design of the case study is basically defined as a general non-linear mathematical programming model of the appropriate ship structure built of the material characterized by material coefficient k and 1 density ρ following section 4 as follows:

- parameters: $p, \ell, b, k, \rho, \sigma_y, \sigma_f, \sigma_s, \tau_s$
- variables: n, t, t_w, h_w

Design goals are the minimization of panel mass m , the minimization of number of transversely stiffening flat bars n which expresses in a simple way the complexity of design or workmanship expenses and finally the minimization of standard deviation of ultimate load carrying capacity taken as a measures of robustness of a structure (Žiha, 2000) $st.dev.(p_{u,p,\sigma}, p_{f,p,\sigma}, p_{b,p,\sigma})$.

The later encapsulates the robustness of design by leveling out the safety apprehended as the maximum lateral pressure that the whole panel and its structural members – plate and stiffeners can withstand (Žiha, 2000) that means avoidance of weak links in the structure.

Finally the design problem is formulated as:

$$\left. \begin{aligned}
 & \min[m(n, k, t, t_w, h_w, \ell, \rho)], \\
 & \min[n], \\
 & \min[st.dev.(p_{u,p,\sigma}, p_{f,p,\sigma}, p_{b,p,\sigma})], \\
 & \text{subject to:} \\
 & W \geq W_{\min}(p, k_m, n, \ell, b, \sigma_{f,a}), \\
 & t \geq t_{\min}(p, k_u, n, p_{u,p,\sigma}, k, \sigma_{p,a}), \\
 & t \geq 2 \text{ mm}, \\
 & \sigma \leq \frac{\sigma_y}{k} \equiv \frac{235}{k}, \\
 & t_w \cdot h_w \geq A_f(p, n, k, \tau_{f,a}), \\
 & \min[p_{u,p,\sigma}, p_{f,p,\sigma}, p_{b,p,\sigma}] \geq p, \\
 & t_w / h_w \leq 20, \\
 & t_w / h_w \geq 10.
 \end{aligned} \right\} \quad (9)$$

At the beginning hard constraints in (9) can hinder the evolutionary process since the majority of the early solutions are infeasible. Consequently, by measuring constraint violations one can rank infeasible solutions.

Later that ranking is added to Pareto frontier of feasible population (Deb, 2001). Constraint violation measure $\Omega(x^{(i)})$ of i -th solution $x^{(i)}$ is derived as summation of normalized violations $\omega_j(x^{(i)})$ (10) (Deb, 2001):

$$\Omega(x^{(i)}) = \sum_{j=1}^8 R_j \omega_j(x^{(i)}) \quad (10)$$

No violations were favored so the weighting factor used is $R = 1$ for all j .

For encoding of chromosomes binary strings were used. Every chromosome consists of four genes which comprise four design variables of the ship hull panel. In addition for the refinement of search the Gray coding was applied (Pahl, 1998).

Design variable	t	n	h_w	t_w
The gene number	1	2	3	4
Available strings per gene	10	10	10	10
Maximum value attainable after mapping [mm]	130	200	430	20

Table 3. The chromosome structure

The emergence of new genes 2, 3, and 4, Table 3, for number of frames, thickness and height of the frame web opens potentials for development of plates stiffened by flat bars. These four characteristics together with the problem parameters define all the other panel properties.

Since the evolution was carried through fixed length chromosomes then the length of the individual genes is also a limitation - constraint put upon the search space, that guide evolution towards reasonable solutions and hopefully speed up the overall search process, Table 3.

Control parameters of the applied NSGA-II algorithm (Deb, 2001) were as follows:

- population size $\lambda = 60$,
- offspring population $\mu = \lambda$,
- uniform crossover (Bäck&Fogel, 2000) - probability $p_c = 1$.
- bit flip mutation probability $p_m = 1/l \cong 0.026$ (Bäck et. al., 2000).

The genetic algorithm tackles the design of the stiffened plate of a contemporary steel ship transversely stiffened panel structure, Fig. 9, of breadth $b=28,8$ m, length $l = 5,17$ m under lateral pressure of $p=0.1$ N/m² according to design loads defined by classification rules (CRS, 2006) using potentials of all the genes, Table 3.

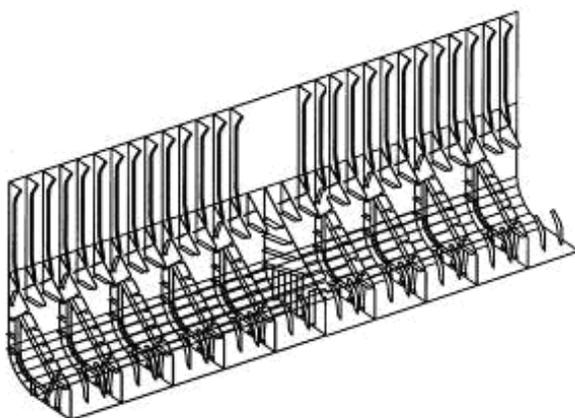


Fig. 9. The modern ship hull transversely framed side structure

The computation results of one out of many iterative trials with repeatable outcomes on standard personal computers are presented as the 3-D Pareto frontier plot n - m - $st.dev.$, Fig. 10.

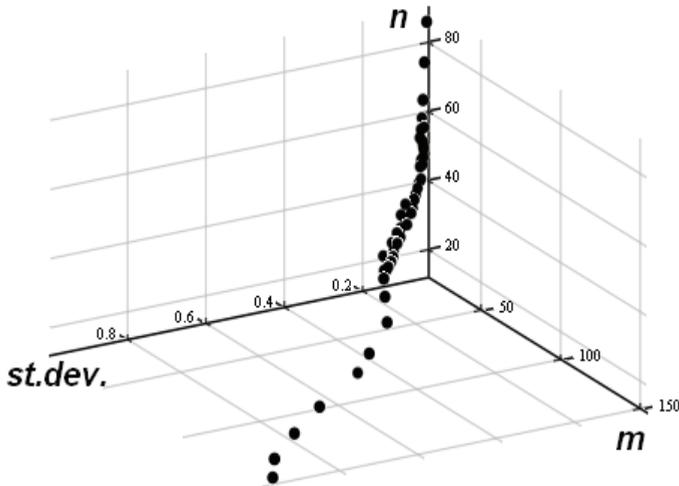


Fig. 10. 3-D Pareto frontier plot

After the full gene potential of chromosome, Table 3, is being unleashed more up to date solutions evolved. The obtained results after 8000 iterations are plotted on Figs. 10. - 14.

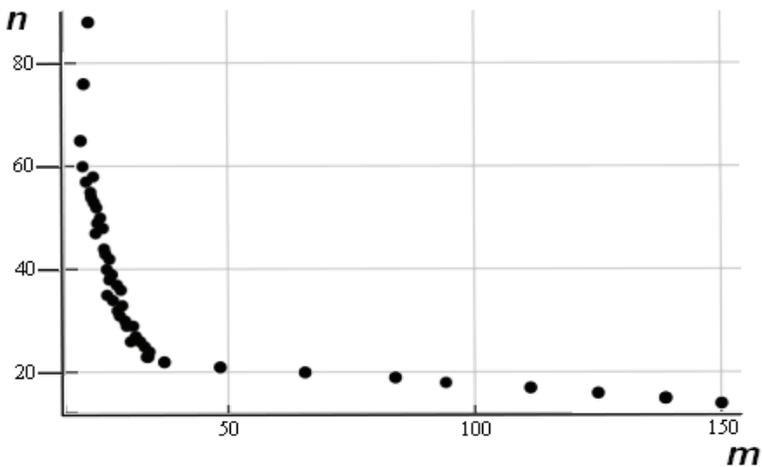


Fig. 11. n - m plot

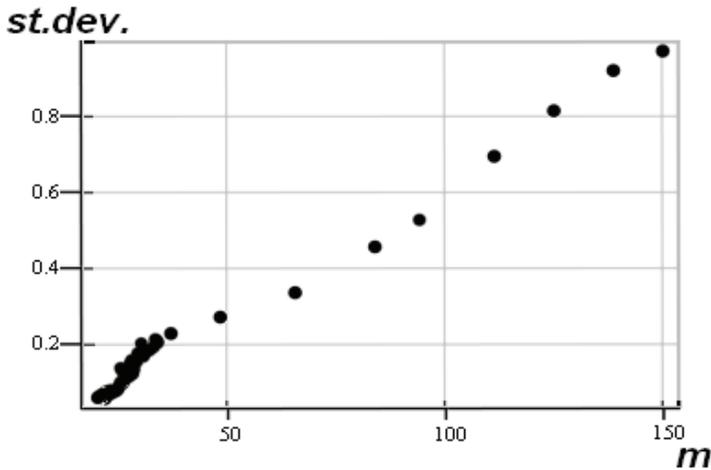


Fig. 12. st.dev.-m plot

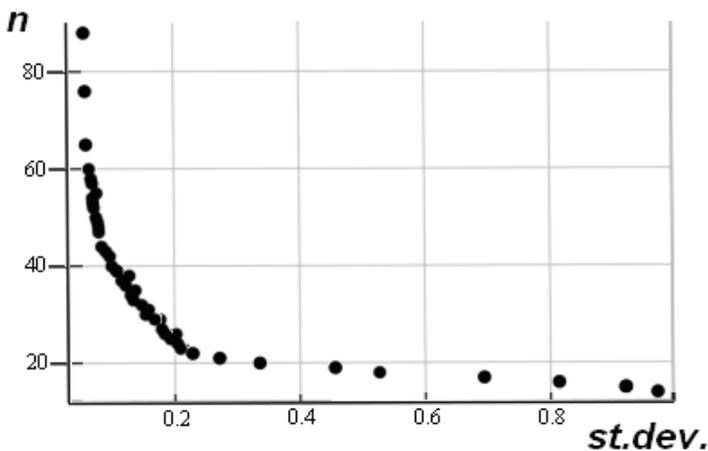


Fig. 13. st.dev.-n plot

The aim of the illustrative example is to interpret the optimization results obtained by evolutionary algorithm as the effects of social and environmental conditions on the development of technical structures. It is comprehensible on one hand, Fig. 11, how the expensive workmanship related to the number of stiffeners irrespective to the material expenses and other technical requirements may yield to preferable solutions of thicker plates with smaller number of stiffeners, even simple plates without stiffeners, regardless of the overall mass of the panel. On the other hand, the socio-environmental condition of expensive material or technical request for light structures irrespective to the workmanship expenses leads to solution of thinner plates with greater number of stiffeners. For highly efficient light-weight structures when the material and workmanship expenses are irrelevant, just the minimal mass, thinner plates with a greater number of stiffeners of

higher class material are preferable. The mathematical model incorporates the assumption of the importance of robustness when the environmental conditions imply uncertainties. The robustness is considered as the minimal variation among safety measures of different failure modes (Žiha, 2000) (inter frame plate bending (6), frame bending (7), overall panel yield (8) and effect of shear stresses (3)). In Fig. 12 it is shown how the request for maximum robustness (minimal standard deviation of safety measures) in this example leads to solution of minimal mass panel that satisfies the prescribed safety level. Moreover the increase of robustness followed by diminution of mass is affordable only by significant increase in workmanship efforts due to large number of built-in stiffeners, Fig. 13. Implementing the ancient conditions of expensive (unavailable) material (except for example wood) and tough workmanship (no experience and tools available) into the mathematical model the solutions points to least expensive plane plate, Fig. 11, without stiffening as the primitive carved-out logs, Fig. 14.

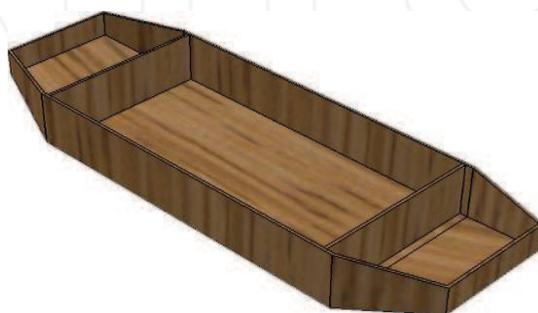


Fig. 14. The primitive boat structure

Finally, the contemporary engineering model resulting in four genes, Table 1, in the last run degenerates to the one single primitive gene number 1, having the plate thickness for the only property. The design model is used in its most degenerative form appropriate to early days of shipbuilding and lack of engineering knowledge and experience. As a final consequence, the mathematical model points to un-stiffened 125 millimeter thick plating, Fig. 14, as the least workmanship demanding solution although inappropriate for now days practice. The only affordable outcome of one primitive gene is the simple un-stiffened plate of minimal thickness appropriate to ancient conditions for carved-out logs that satisfies the past and modern safety requirements, Fig 15.



Fig. 15. Carved-out log

7. Conclusion

The first case study in this chapter brings a special way of genotype encoding well suited for mechanisms of genetic algorithm operators. Furthermore, it considered how the structural model is decoded from such a genotype. The aim of this study of the new “mesh-like” approach was a consequence of formerly detected deficiencies of the existing methods. The presented advanced methods for the truss structure optimization work perfectly, but are specialized either for the structure properties optimization (Hasançeb, 2007), (Coello&Christiansen, 2000), or the structure topology optimization (Jakiela at all, 2000), (Hamdam, Schoenauer, 2002) and (Kim&Weck, 2004). Shape annealing and shape grammars applied for the structural optimization (Shea&Cagan, 1998) offered an alternative, but a GAs are found as suitable alternatives. For the reasons of simplicity, at this point of research the search space is reduced. The struggle genetic algorithm applied for single objective uses a fixed length chromosome, and the number of nodes is therefore user-defined. The cross-section area of trusses is fixed. Future work will include an introduction of these present parameters as variables with suitable coding and will aim at defining load vectors. To enhance speed the parallelization of algorithm is being considered in future too. With all which has been accomplished, moving away from the single to the multi-objective optimization makes a natural step ahead in evolving truss structures.

The evolutionary design supports normally the contemporary progressive engineering reasoning aspired with achievement of highly efficient products providing socially acceptable safety levels and appropriately lower costs by employing genetic algorithms. However, the second case study in this chapter indicates how the reverse process to the technical progress can reconstruct the origins of contemporary products using evolutionary algorithms on engineering models in two manners. The simplest way is the replication of primitive conditions, such as for example lack of experience, unavailability of appropriate material and technology. Introduction of past conditions into up to date mathematical models corroborates early solutions based on past engineering practice. Reconstruction of past social and environmental conditions may lead to primitive solutions appropriate to early human’s engineering but it does not characterize only the evolutionary algorithms. Another way is the simplification or degeneration of the design model that is in terms of genetic algorithms, deactivating or removing more complex genes from the chromosomes that might be viewed as a particular feature of evolutionary algorithms.

Evolutionary design approach upholds that the technical progress goes on if the existing gene potentials are activated or the new evolutionary potentials based on additional knowledge are introduced.

The optimization search by genetic algorithms may be viewed as time-condensed best-practice that in reverse order can back-trace the engineering development either by replicating past condition or by omission of chromosomes introduced into evolutionary models by growth of engineering experience.

The chapter demonstrated that the evolution of structural systems is successfully driven forward towards the fittest structures by the synthesizing criteria of most uniform responsiveness. The changing of external circumstances that provoke uniform structural response is interpretable as the robustness criteria. The research takes up the most commonly used statistical measures of data dispersion to define the structural system robustness for practical purposes. The most conveniently measure appears the minimal coefficient of variations of internal forces under displaced external loads. Shortly, the thesis is that the fittest structure is the robust one.

8. Acknowledgment

This research was supported by two projects of the Ministry of science, education and sports of the Republic of Croatia.

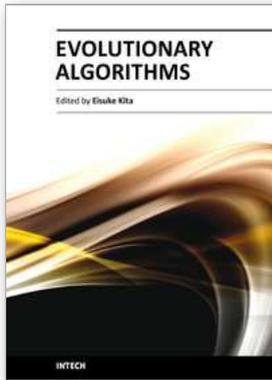
9. References

- Bäck T.; Fogel D.B., Michalewicz Z. (2000). *Evolutionary Computation 1, Basic Algorithms and Operators* Institute of Physics Publishing, Bristol and Philadelphia.
- Bäck T., Fogel D. B., Michalewicz Z. *Evolutionary Computation, Advanced Algorithms and Operators*, Institute of Physics Publishing, Bristol and Philadelphia; 2000.
- Bentley, P. (1999). *Evolutionary Design by Computers*, Morgan Kaufmann.
- Chomsky, N. (1957). *Syntactic Structures*, Mouton and Co.
- Coello, C.A.C.; Christiansen, A.D. (2000). *Multiobjective Optimization of Trusses using Genetic Algorithms*, Computers and Structures, Elsevier.
- CRS (2006). *Rules for the Classification of Sea-Going Ships-Part 2 Hull*, Croatian Register of Shipping, Split.
- Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons Ltd.
- Gero, J.S.; Louis, S.L. (1995). Improving Pareto optimal designs using genetic algorithms, *Microcomputers in Civil Engineering*, , pp 241–249.
- DNV (1978). *Ship' Load and Strength Manual*, Det Norske Veritas, Hovik.
- Goldberg, D.E. (1989). *Genetic Algorithms in Search Optimization and Machine Learning*, Addison Wesley Longman Inc.
- Goldberg, D. E. (2002). *Design of Innovatinon*, Kluwer academic Publishers.
- Golinski, J. (1970). Optimal Synthesis Problems Solved by Means of Nonlinear Programming and Random Methods, *Journal of Mechanisms*, pp 287-309.
- Hamda, H.; Schoenauer, M. (2002). Topological optimum Design with Evolutionary Algorithms, *Journal of Convex Analysis*, pp 503-517,
- Hasançeb, Q. (2007). Optimization of truss bridges within a specified design domain using evolution strategies, *Engineering Optimization*, Taylor and Francis.
- Hubka, V.; Eder, W.E. (1992). *Engineering Design: General Procedural Model of engineering Design*, Springer-Verlag Berlin Heidelberg.
- Hughes, O. (1972.) *Rational Ship Structural Design*, SNAME, New Jersey.
- Jakiela, M.J.; Chapman, C.; Duda, J.; Adewuya, A.; Saitou, K. (2000). Continuum structural topology design with genetic algorithms, *Computational. Methods in Applied Mechanical Engineering*, Elsevier, pp 339-356
- Kim, Y.I.; De Weck, O. (2004). Progressive structural topology optimization by variable chromosome length genetic algorithm, *CJK-OSM3*.
- Pahl, G.; Beitz, W. (1988). *Engineering Design - A Systematic Approach*, Springer Verlag;
- Shea, K.; Cagan, J. (1998). *Generating Structural Essays from Languages of Discrete Structures*, Artificial Intelligence in Desgin, Kluwer Academic Publishers, pp 365-384.
- Tan K.C.; Lee T.H.; Khor E.F. (2002). *Evolutionary Algorithms for Multi-Objective Optimization: Performance Assessments and Comparisons*, Artificial Intelligence Review, Kluwer Academic Publishers,; pp 253–290.

- Yockey, H. (2005). *Information theory, Evolution and the Origin of Life*, Cambridge University Press.
- Zienkiewicz, O. C. (1971). *The Finite Element Method in Engineering Science*, Mc Graw-Hill.
- Wood, K.L.; Otto, K.N. (1999). *Product Design Techniques in Reverse Engineering, Systematic design, and New Product Development*, Prentice-Hall, NY.
- Žiha, K. (2000). Redundancy and Robustness of Systems of Events, *Probabilistic Eng. Mechanics*, Vol. 15, pp. 347-357.

INTECH

INTECH



Evolutionary Algorithms

Edited by Prof. Eisuke Kita

ISBN 978-953-307-171-8

Hard cover, 584 pages

Publisher InTech

Published online 26, April, 2011

Published in print edition April, 2011

Evolutionary algorithms are successively applied to wide optimization problems in the engineering, marketing, operations research, and social science, such as include scheduling, genetics, material selection, structural design and so on. Apart from mathematical optimization problems, evolutionary algorithms have also been used as an experimental framework within biological evolution and natural selection in the field of artificial life.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Tino Stanković, Kalman Žiha and Dorian Marjanovic (2011). Tracing Engineering Evolution with Evolutionary Algorithms, Evolutionary Algorithms, Prof. Eisuke Kita (Ed.), ISBN: 978-953-307-171-8, InTech, Available from: <http://www.intechopen.com/books/evolutionary-algorithms/tracing-engineering-evolution-with-evolutionary-algorithms>

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

