

Peter Pechac - Milan Saga - Ardeshir Guran - Leszek Radziszewski \*

# IMPLEMENTATION OF MEMETIC ALGORITHMS INTO STRUCTURAL OPTIMIZATION

*The paper presents an implementation of the memetic algorithm for discrete structural optimization. This algorithm is a combination of genetic algorithm and five local search methods. The proposed memetic algorithm is able to choose the right local search method based on performance which is evaluated in real time during optimization.*

**Keywords:** Memetic algorithm, structural optimization, pattern search, Nelder-Mead method, nonlinear conjugate gradient method, particle swarm optimization, fully stress design.

## 1. Introduction

The term “meme” was coined by evolutionary biologist Richard Dawkins in his book *The Selfish Gene* where it is described as an equivalent to gene in cultural evolution. Examples of memes are tunes, ideas, catch-phrases, clothes fashions, ways of making pots or of building arches. Just as genes propagate themselves in the gene pool by leaping from body to body via sperms or eggs, so memes propagate themselves in the meme pool by leaping from brain to brain via a process which, in the broad sense, can be called imitation [1].

Memetic algorithms represent a group of optimization methods which combine global optimization and local search methods. The global search part is usually based on genetic algorithm or particle swarm optimization. The global optimization part ensures diversity, favors exploration and provides good coverage of the optimization space, while the local search methods favor exploitation and speed up the convergence to local extremes [2]. The choice of local search methods is affected by the problem which is being solved. The no free lunch theorem says that the performance of different methods on different problem classes is not equal [3] and so the choice of the right local optimization method is often left on the user, see - Fig. 1. Problems of engineering mechanics often require computationally expensive evaluation of objective function [4 and 5], so it is crucial to choose the right local search methods. A traditional approach would be that the user selects the local search methods based on his previous experience and observation of their

performance on similar problems. This traditional approach requires the user to have this experience and even then it does not guarantee that the best performance is achieved. The memetic algorithm which is proposed in this paper is using multiple local search methods and the algorithm itself is capable of making the decision which method should be used. This decision is based on performance evaluation.

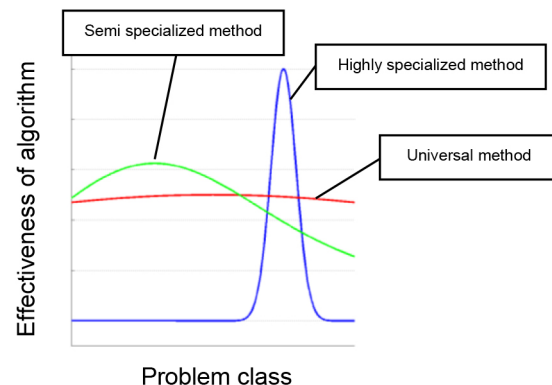


Fig. 1 Effectiveness of algorithm vs. problem class

## 2. Description of the Memetic Algorithm

The global optimization part of memetic algorithm is based on genetic algorithm. The genetic algorithm is using three genetic operators: selection, crossover and mutation. Rank selection

\* <sup>1</sup>Peter Pechac, <sup>1</sup>Milan Saga, <sup>2</sup>Ardeshir Guran, <sup>3</sup>Leszek Radziszewski

<sup>1</sup>Department of Applied Mechanics, Faculty of Mechanical Engineering, University of Zilina, Slovakia

<sup>2</sup>Institute of Structronics, Ottawa, Canada

<sup>3</sup>Faculty of Mechatronics and Machine Building, Kielce University of Technology, Kielce, Poland

E-mail: saga@uniza.sk

is taking into account both the value of objective function and diversity to prevent premature convergence to local extreme. Uniform crossover ensures that the new offspring inheres equal portion of genes from both parents. The principle of uniform crossover is depicted in Fig. 2. Non-uniform Cauchy mutation performs better than other types of mutation because it has potential to make longer jumps [6]. The Cauchy distribution is depicted in Fig. 3. The high probability of mutation in the beginning helps to improve diversity in population and is approaching zero in the final stages. This probability of mutation is guided by exponential cooling schedule known from simulated annealing which can be seen in Fig. 4. In each generation the best 10% of population is carried to the next generation without changes. This elitist approach saves good solutions from mutation and crossover which could be otherwise lost. However, they are also used as parent chromosomes during selection and crossover. The worst 40% of population is deleted prior to the selection. The new population is assembled from the best 10% which were further improved by local search methods, the offspring generated by crossover and mutation and the rest is generated using “white space search” (WSS) algorithm to complete the full size of population.

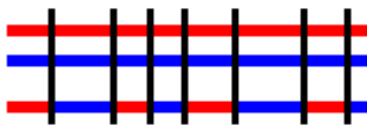


Fig. 2 Uniform crossover

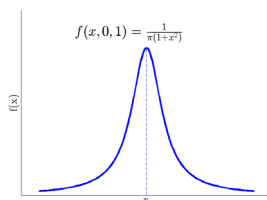


Fig. 3 Cauchy distribution

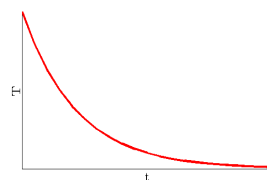


Fig. 4 Exponential cooling schedule

The memetic algorithm has five local search methods at its disposal. These methods include: pattern search method, Nelder-Mead simplex method, Dai-Yuan nonlinear conjugate gradient method with line search, particle swarm optimization and fully

stress design method. The sizes of step parameters of the local search methods are guided by linear cooling schedule (see Fig. 5), which is used in simulated annealing. In each generation of genetic algorithm, 10% of the population is improved by local search methods. The memetic algorithm can choose the local search methods based on their performance.

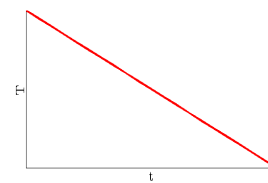


Fig. 5 Linear cooling schedule

In the first iteration all local search methods are tested with equal probability, the performance of methods is then evaluated based on the difference of objective function divided by time spent by the method and averaged for each method. The percentage of candidates improved by local search methods is then divided according to their performance. If the percentage for any method reaches zero it is kept at small non zero value so it has chance to be tested and potentially return to normal use. If the ratio of use of all local search methods reaches zero, the process is restarted and all of them gain the same ratio of use as in the beginning. By always using the best performing method the overall performance is significantly improved. The scheme of the algorithm is in the following Fig. 6.

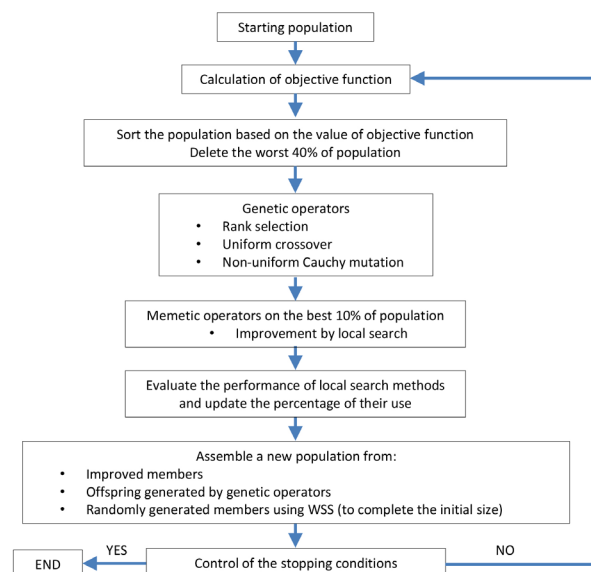


Fig. 6 Scheme of the memetic algorithm

### 3. Properties of Local Search Methods Used in the Memetic Algorithm

The detailed description of the used local search methods would not fit into the scope of this paper, instead only their main properties will be discussed. The memetic algorithm uses five optimization methods as local search methods: pattern search, Nelder-Mead simplex method, Dai-Yuan nonlinear conjugate gradient method, particle swarm optimization method and fully stress design method. Pattern search method and Nelder-Mead simplex method are deterministic comparative optimization methods, thus they are derivative free and can operate on functions that don't have continuous first derivative (noisy functions). However, they can perform inferior to gradient based methods when used on smooth functions with continuous first derivative. Dai-Yuan nonlinear conjugate gradient method is a fast converging deterministic gradient based method, however as all gradient based methods it can have problems with noisy functions as the numerical calculation of gradient can become problematic. Particle swarm optimization is considered as a stochastic global search method, however its properties can be altered by tuning its parameters (inertia, cognitive and social parameter) to favor exploitation over exploration [7]. Fully stress design method is an engineering approach to structural optimization and despite the fact that its use is limited only to stress constrained problems it is very effective. The fully stress design method can very quickly solve the part of problem which requires minimization of weight with respect to stress constraint, on the other hand, the conventional optimization methods would require much more time to find an equivalent solution. More detailed information on the fully stress design method can be found in [8, 9 and 10]. Each local search method is running only for a few iterations.

### 4. Numerical Test

The algorithm was tested on a truss structure subjected to stress and displacement constraint. The goal is to minimize price of used material, while satisfying the stress and displacement constraints. Three materials with different price and yield strength

were used. The used materials were structural steels S235, S275 and S355. The safety coefficient for stress was  $k = 3$  so the stress limits  $\sigma_L$  for each material were 78.33, 91.66 and 118.33 MPa respectively. The prices  $p$  for ton of material were estimated 450, 500 and 600 euro. Density was same for all materials  $\rho = 7850 \text{ kg m}^{-3}$ . The displacement limit was 20 mm. The Young's modulus is the same for all three materials  $E = 210\,000 \text{ MPa}$ .

Objective function is calculated as follows

$$f(\mathbf{x}) = \sqrt{\sum_{i=1}^m \rho_i V_i p_i}. \quad (1)$$

If the stress constraint is violated, the objective function is penalized as follows

$$f(\mathbf{x}) = f(\mathbf{x}) + f(\mathbf{x}) \left( \frac{(\sigma_{\max} - \sigma_L)}{\sigma_L} \right). \quad (2)$$

Similarly, if the displacement constraint is violated, the objective function is penalized as follows

$$f(\mathbf{x}) = f(\mathbf{x}) + f(\mathbf{x}) \left( \frac{(u_{\max} - u_L)}{u_L} \right). \quad (3)$$

The structure was divided into 6 optimization groups. The optimization variables were 6 cross-section areas and 10 geometrical parameters. The sizes of cross-section area were divided into 361 discrete values from 500 to 2500 mm<sup>2</sup>. For the purpose of keeping the number of optimization variables as low as possible the design variables for cross-sections were combinations of materials and cross-section areas so the total number of available cross-sections was 1083. The geometrical optimization variables were heights of the structure defining its shape. The nodes between the top and bottom of structure were always positioned in the middle of the total height.

The structure was loaded by its own mass and forces of magnitude 15 kN. Boundary conditions and optimization variables are depicted in the following Fig. 7 of the structure using the shape after optimization.

The size of population was set to 1000, maximum number of generations was 500 and the number of iterations of local search in one generation was 2. The resulting axial stresses after optimization are depicted in the following Fig. 8.

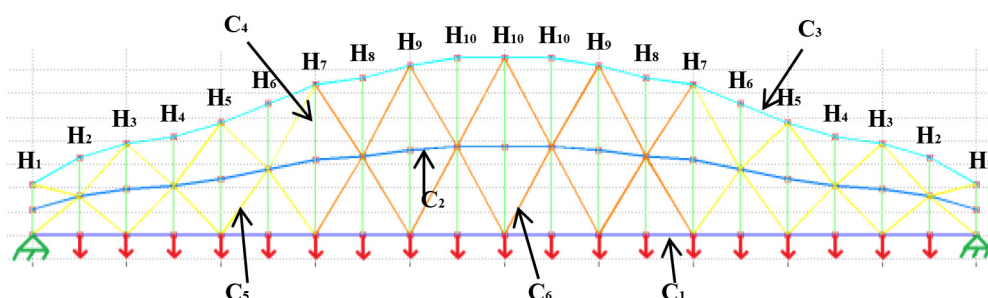


Fig. 7 Boundary conditions (red - loading forces, green - deformation constraints) and optimization variables (c - cross-section, h - height)

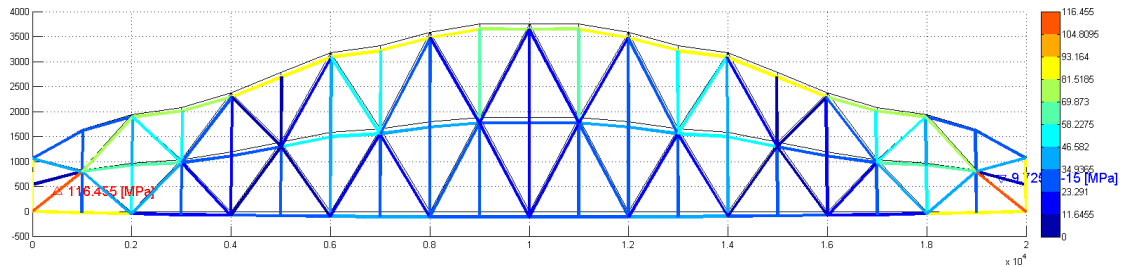


Fig. 8 Axial stress [MPa] after optimization

The history of local search methods use is in the next Fig. 9.

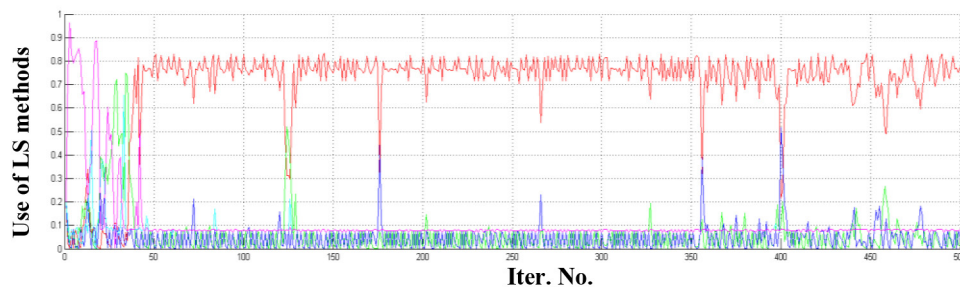


Fig. 9 Use of local search methods vs. iteration step (red - PS, green - NM, blue - NCG, cyan - PSO, magenta - FSD)

The history of objective function can be seen in Fig. 10.

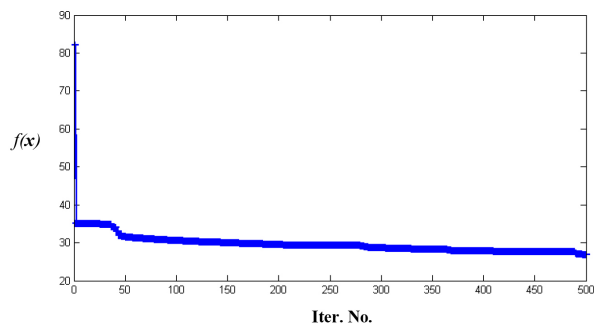


Fig. 10 Objective function vs. iteration step

The value of objective function in optimum found by the algorithm was 26.9 and mass of the structure was 1345.3 kg.

Furthermore, due to stochastic nature of the components which are included in the memetic algorithms, a statistical test was carried out to compare the performance of MA with automated choice of local search method vs. MA with single local search method. We tested it against all five LS methods which have been used in the proposed algorithm.

The size of the population was set to 100, each local search method was allowed to run 2 iteration steps in one generation of GA. The algorithm stopped when it found solution with value of objective function lower than 40. The performance is compared based on the time spent by the algorithm. To provide statistical

significance we run each combination 500 times. The results showing percentage of samples vs. computational time can be seen on the following histograms (Figs. 11 - 16).

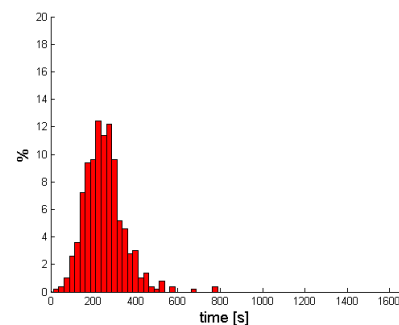


Fig. 11 GA + PS

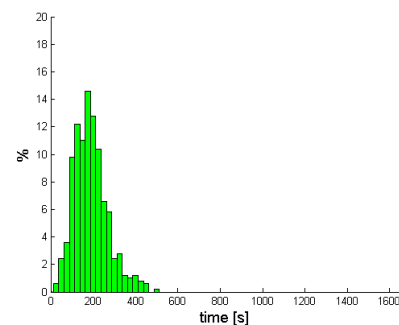


Fig. 12 GA + NM

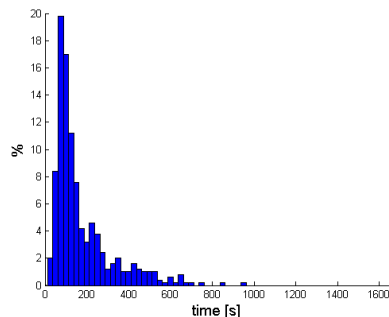


Fig. 13 GA + NCG

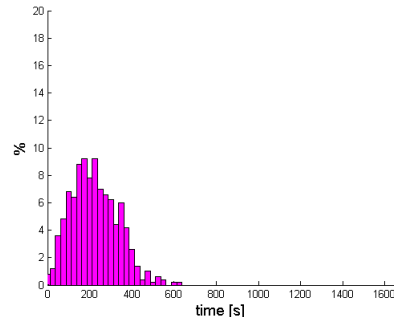


Fig. 15 GA + FSD

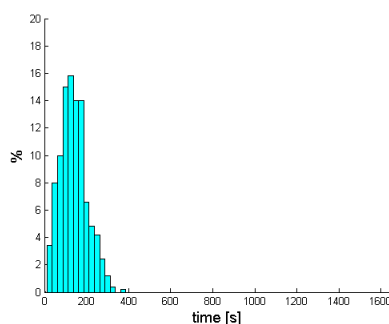


Fig. 14 GA + PSO

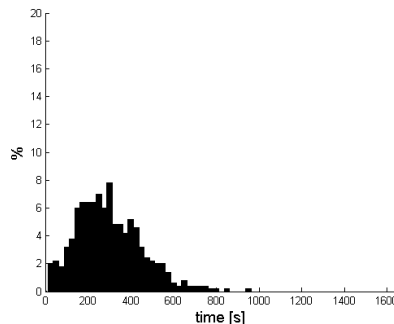


Fig. 16 GA + [PS, NM, NCG, PSO, FSD]

In Fig. 9 you can see that the algorithm first selects FSD method, then Nelder- Mead method and finally the pattern search method. Nonlinear conjugate gradient method and particle swarm optimization method were used only scarcely. Pattern search and Nelder-Mead method are both deterministic and gradient free. Comparison of combinations of GA and single local search methods shows that the Nelder-Mead method is better than simple pattern search, however the automatic choice of local search methods chose pattern search over Nelder-Mead.

## 5. Conclusion

The best performing combinations were GA+PSO and GA+NCG so they can be considered as the best choices for solution of similar problems. The automatized choice of local search method has not improved performance on this particular

test problem as we expected. Further research and testing on multiple test problems is required to definitely confirm whether the proposed idea of automatized choice of local search method improves the performance or not. Its performance could be affected by repeated use of deterministic local search methods on the same solution which did not improve the solution but rather wasted the computational time. The proposed method could be improved by keeping the track of operations carried out on each solution so the algorithm would not use the same deterministic method repeatedly on the same solution and also by using meta-optimization to adjust the parameters of local search methods.

## Acknowledgements

This work has been supported by grant project VEGA No. 1/0234/13.

## References

- [1] DAWKINS, R.: *The Selfish Gene*. New York: Oxford University Press, 1976.
- [2] MOSCATO, P., COTTA, C.: *A Gentle Introduction to Memetic Algorithms*, Handbook of Metaheuristics, 105-144, Springer US, 2003.
- [3] WOLPERT, D., MACREADY, W.: No Free Lunch Theorems for Optimization, *IEEE Trans. Evol. Comput.*, vol. 1, pp. 67 -82 1997.
- [4] A. CZAN, E. TILLOVA, J. SEMCER, J. PILC: Surface and Subsurface Residual Stresses after Machining and their Analysis by x-ray Diffraction. *Communications - Scientific Letters of University of Zilina*, vol. 15, No. 2, 2013, 69-76, ISSN 1335-4205.

- [5] ZMINDAK, M., NOVAK, P., PASTOREK, P.: *Finite Element Analysis of Delamination in Steel-reinforced Concrete Reinforced by Fiber Composite (in Slovak)*, Experimental and Computational Methods in Engineering, Litomerice, 2015.
- [6] YAO, X., LIU, Y., LIN, G.: Evolutionary Programming Made Daster, *IEEE Transaction on Evolutionary Computation*, vol. 3, No. 2, 82-102, 1999.
- [7] CERVANTES, A.: *Constriction Factors and Parameters*, 2005, Retrieved from <http://tracer.uc3m.es/tws/psa/parameters.html>.
- [8] SAGA, M., PECHAC, P., VASKO, M.: FSD and its Modifications for Shell Finite Elements. *American J. of Mechanical Engineering*, 1(7), 2013, 427-433.
- [9] SAGA, M., DUDINSKY, M., PECHAC, P.: Optimization of thin Shell Structures using FSD Algorithms, 2012, *Communications - Scientific Letters of the University of Zilina*, 14 (3), 32-38.
- [10] SAPIETOVA, A., DEKYS, V., VASKO, M.: A Numerical Modeling Rotating Machine Having Unbalance and the Measuring of its Dynamical Properties, *Metalurgija/Metallurgy*, vol. 49/2010, No. 2, 503-507.