

A Framework for Optimal Placement of Strain Gauges on Elastic Elements of Force Sensors Using Genetic Algorithms

Sergey I. Gavrilenko^{1*}

¹Bauman Moscow State Technical University, 2nd Baumanskaya str., 5/1, 105005, Moscow, Russia

Abstract. This paper presents a digital education tool for learning the specifics and behavior of a multi-objective genetic algorithm (MOGA) used to solve the problem of optimal placement of strain gauges on the elastic element of a force sensor. The paper formulates the problem statement and specifies how this problem can be solved using the MOGA. For the problem, the design variables are the locations of strain gauges and angles at which they are positioned. The goal functions are the output signal of the sensor and the measurement error from bending moments, which can be caused by the off-centric application of load. The solution algorithm is implemented within a framework that can be used to investigate and learn how parameters of MOGA influence its performance. The framework is used to run computational experiments for the given problem to find the optimal placement of strain gauges on the elastic element of a given force sensor. The performance of the MOGA in solving this problem is compared to that of the traditional approach.

1 Introduction

Currently, digital technologies are increasingly used in education, meaning that all information used in education tends to be transformed into digital form [1-2]. This trend is expressed by taking tests using computers, online universities, and online courses [3]. Moreover, the digitization of educational tools is another critical trend [4-5].

Another trend in engineering education is that engineering becomes more and more complex, and students need to learn the skills of solving optimization problems in their respective domains on knowledge. Unfortunately, there is little emphasis on that in engineering curriculums. Besides, most of the design problems in engineering are inherently multi-objective. So, the task of developing a digital educational tool for learning how a multi-objective engineering problem can be solved using MOGA is relevant.

The paper has a two-fold goal. First, the paper must present a solution of a multi-objective problem from a specific engineering domain, namely, determining the optimal placement of strain gauges on the elastic element of a force sensor. Second, the solution of problem must be implemented as a piece of software for investigating how the parameters of the MOGA influence the speed of convergence and the quality of the obtained Pareto-

* Corresponding author: gavrilenkovsergei@bmstu.ru

optimal solutions. Third, the software must be used to solve the given problem, and performance of MOGA must be compared to that of traditional optimization methods.

2 Development of the framework

The framework is comprised of a Python module for solving the problem of determining the optimal placement of strain gauges on the elastic elements of a given sensor, as well as user interface modules for defining the parameters of the MOGA and post-processing the results. The following subsections formulate the problem and show how this problem is solved with the MOGA using the pymoo library for multi-objective optimization using genetic algorithms [6].

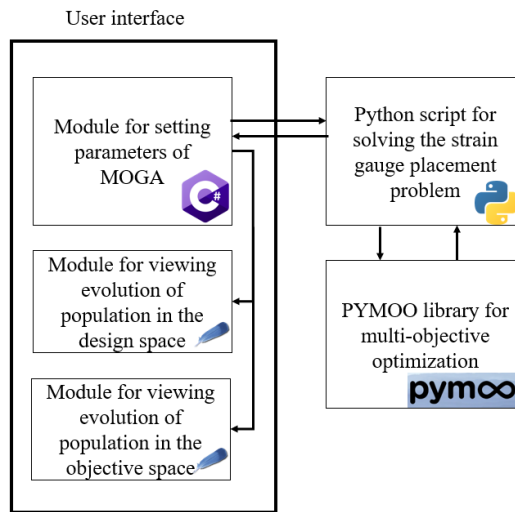


Fig. 1. Framework structure

2.1 Problem statement

A strain-gauged force transducer is a transducer converting applied force to an electrical signal by measuring induced strain using strain gauges connected in the Wheatstone circuit. One of the key characteristics of a strain-gauged force transducer is its output signal at the rated load. The magnitude of the output signal influences the resolution of the instrument and its Signal-to-Noise ratio. Another important characteristic of a force sensor is the immunity to side loads and bending moments that can occur due to incorrect application of force. If the bending moment is applied to the force sensor, it will induce some strain, which will be measured by the strain gauges along with the strain from the vertical load.

However, if the construction of the elastic element is perfectly symmetrical, and the strain gauges are located accurately, and their gauge factors are identical, the influence of the bending moment will be eliminated by the topology of the Wheatstone bridge. The nature of this phenomenon was thoroughly elaborated in [7]. Briefly, the sensitivity to the bending moment arises when the responses of the individual arms of the Wheatstone bridge to the strain from the bending moment are not anti-symmetrical. Such a response can be caused by errors in positioning strain gauges on the elastic element, the variation of the gauge factor across strain gauges of the circuit, or a non-anti-symmetrical strain field from the bending moment. Although the paper only considers bending moments, the same can be

$$S = \sqrt{(S_{avg}^{M_x})^2 + (S_{avg}^{M_y})^2 + (S_{std}^{M_x})^2 + (S_{std}^{M_y})^2} \tag{3}$$

where $S_{avg}^{M_x}$, $S_{avg}^{M_y}$ are the average values of the sensitivity to M_x and M_y , respectively, and $S_{std}^{M_x}$, $S_{std}^{M_y}$ are the standard deviation values of the sensitivity to M_x and M_y , respectively. The simulation is conducted according to the method described in [10].

The parameters of the probabilistic simulation for calculating the sensitivity to M_x and M_y are as follows:

- Positioning tolerance: ± 0.05 mm;
- Angular positioning tolerance: $\pm 0.5^\circ$;
- Gauge factor tolerance: $\pm 1.5\%$.

The SGs are generic constantan (gauge factor is 2) linear strain gauges type LA11 produced by HBM [11].

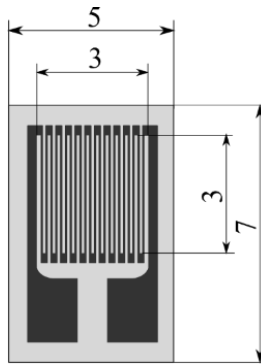


Fig. 3. Framework structure.

Figure 4 shows contour plots of strain state components on the surface A when the force sensor is loaded separately by F , M_x and M_y . The strain data is obtained using FEA system Code_Aster [7].

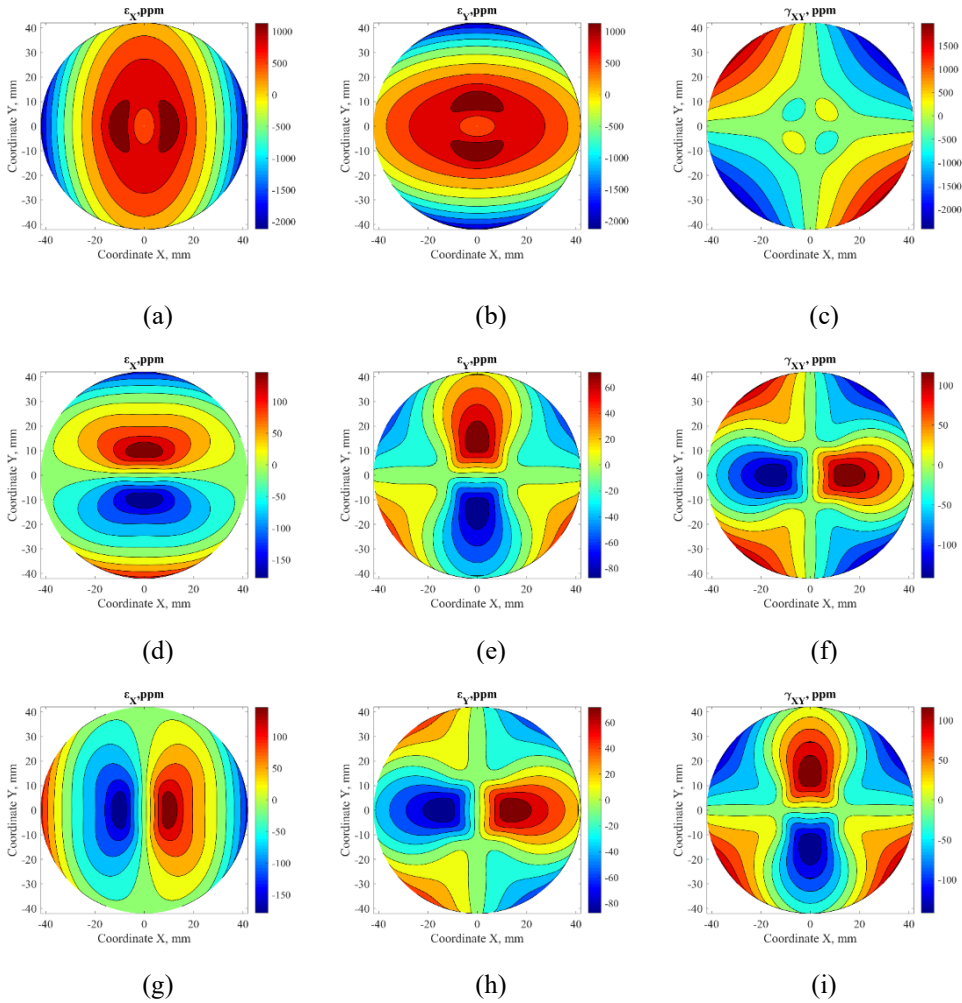


Fig. 4. Strain state components when sensor is loaded with F (a, b, c), Mx (d, e, f), My (g, h, i).

2.2 Solution of the problem using a MOGA

There are many libraries for multi-objective optimization using genetic algorithms. Deb and Blank gave a thorough review of them in [6]. In this study, the pymoo library is chosen [12]. This is an open-source library under the Apache 2.0 license. The library supports the most popular MOGAs, and it supports user-defined optimization problems. The library is written in Python. In [7], the procedure for calculating the sensitivity to bending moments was written in Octave language. For this study, the procedure is ported to Python, which is an easy task given the similarity between Octave and Python.

To define a user-defined problem, a class PlacementProblem is created. This is a child class of the Problem class [12]. PlacementProblem contains all procedures requires to calculate GF.

For a single solution with the design variable vector X, GF is evaluated as follows. First, it is checked that none of the gauges violate topological constraints, i.e., none of the

gauges overlap, and none of the gauges go beyond the boundaries of the surface for mounting SGs. Figure 5 presents possible violations of the topological constraints.

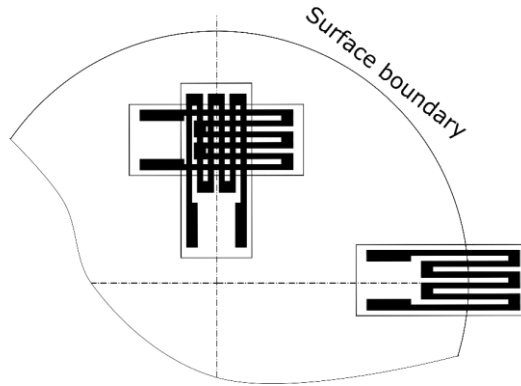


Fig. 5. Possible violations of topological constraints: overlapping of gauges and violation of the boundary of the surface appointed for mounting SGs.

If the solution satisfied the topological constraints, \mathbf{GF} is calculated using the procedure presented in [7].

The problem formulated above solved using NSGA II [13]. This is one of the most known MOGAs. NSGA II was used on many multi-objective problems [14-18]. Using NSGA II, the problem is solved as follows. First, the strain data required for calculating \mathbf{GF} is loaded from a text file obtained from FEA. Second, the *PlacementProblem* class is initialized. Lastly, the algorithm is executed and the results of the algorithm are stored in text files for postprocessing. This procedure is implemented as a Python script. The input data for the script is as follows:

1. Population size;
2. Number of generations until termination;
3. Crossover type;
4. Crossover probability;
5. Mutation probability;

The output data of the script is the following:

1. Evolution of design variables;
2. Evolution about goal functions;
3. Values of indicator functions for each generation.

The indicator functions are the Inverted Generational Distance Plus (IGD+) and Hypervolume [19]. IGD+ shows how close the obtained Pareto set is to reference Pareto set, and HV shows the hypervolume dominated by the current Pareto set. Calculating IGD+ requires a reference Pareto set. Unlike classical test problems for MOGA with known exact Pareto sets [20], this study is limited to the approximation of the exact Pareto set to be used as reference. The approximation is made by sampling the design space with a dense mesh using a Sobol sequence. The number of elements in the Sobol sequence is 500000. Figure 6 shows the approximation of the Pareto set used as a reference set for calculating IGD+.

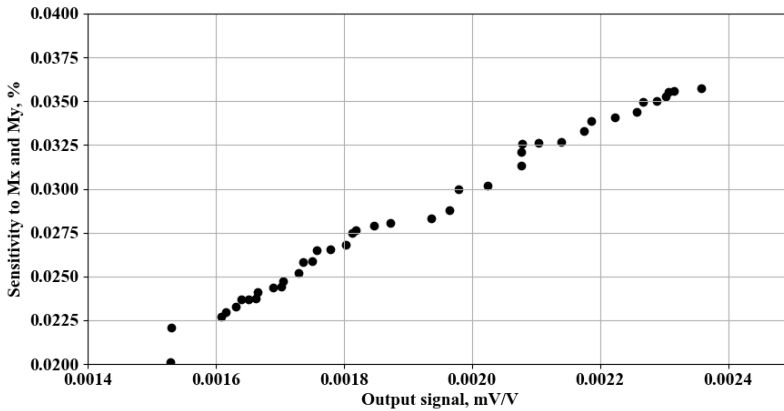


Fig. 6. User interface of the module for managing algorithm settings.

The information about the reference Pareto set is stored in a text file similar to FEA strain data. To ease the interaction with the script, a user interface is built around it. The interface has controls for setting the parameters of the MOGA and running it, Figure 7.

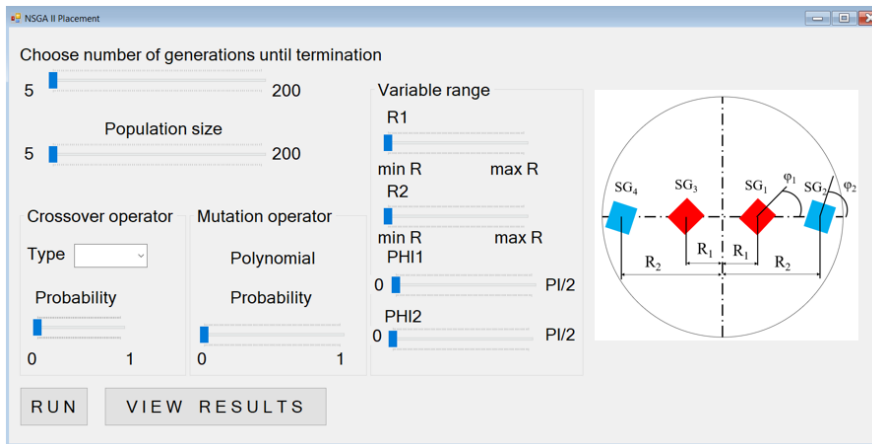


Fig. 7. User interface of the module for managing algorithm settings.

The results can be investigated using two postprocessing modules. The first module, Fig. 7, shows the evolution of the population in the objective space using plots of performance indicators and scatter plots. Other modules, Fig.8, show the evolution of the population in the design space using histograms of design variables. The modules for postprocessing results are also capable of animating the process of population evolution.

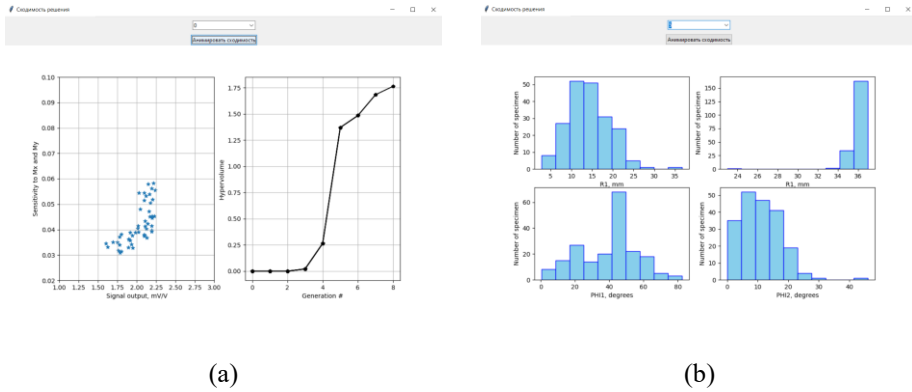


Fig. 8. User interface of the module for investigating the population evolution in the objective (a) and design (b) spaces, results for a given generation can be chosen using a combo box.

3 Approbation of the framework

The efficacy of the framework is checked by solving the problem of determining the optimal placement of strain gauges on the elastic element of the force sensor presented in Section 2.1. First, the problem is solved using the developed framework with different parameters of the MOGA. Second, the problem is solved using a conventional approach to multi-objective problems. Lastly, the quality of the obtained Pareto-optimal solutions is compared using a quality indicator. The quality indicator is the generational distance (IGD+), which shows how close the obtained Pareto set to the reference Pareto set.

3.1 Solving the problem using the framework

To test the influence of the parameters of NSGA II on the quality of the obtained Pareto set and the rate of convergence, the problem is solved several times with different settings of NSGA II. The vector of parameters of the algorithms is as follows:

$$\mathbf{P} = [N_{pop} \quad Crossover]^T \tag{4}$$

where N_{pop} is population size, $Crossover$ is the crossover operator. The mutation operator is PM, and the probabilities of crossover and mutation are 0.95. The algorithm terminates after 20 generations. The population is initialized by Latin Hypercube sampling. Table 1 specifies the possible values of N_{pop} , and Crossover [12] (Simulated Binary Crossover, Uniform Crossover, One-point Crossover).

Table 1. Possible values of the parameters of NSGA II.

Parameter name	Set of possible values
N_{pop}	{10, 20, 40}
Crossover type	{"SBX", "UX", "1-point"}

Based on Table 1, 9 combinations of \mathbf{P} are built. For each combination, the algorithm is executed 20 times, as MOGAs, as well as the simulation for evaluating the sensitivity to M_x and M_y are probabilistic by nature. Mean values, as well as standard deviations for IGD++ are calculated for all generations.

Figure 9 shows the heatmap of the mean and standard deviation values of IGD+ for all combinations. The distributions are plotted for the last generation.

Based on Fig. 9 (a), the Pareto set obtained by NSGA II is closest to the reference set if $N_{pop} = 40$. Besides, the greater N_{pop} is the smaller the variation of results.

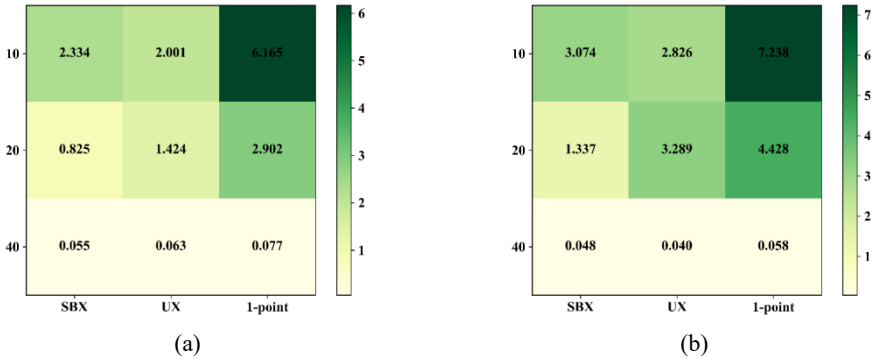


Fig. 9. Heatmap of mean (a) and standard deviation (b) values of IGD+ for the last generation of the algorithm.

Figure 10 shows the history of convergence for NSGA II with crossover operator UX and different population sizes.

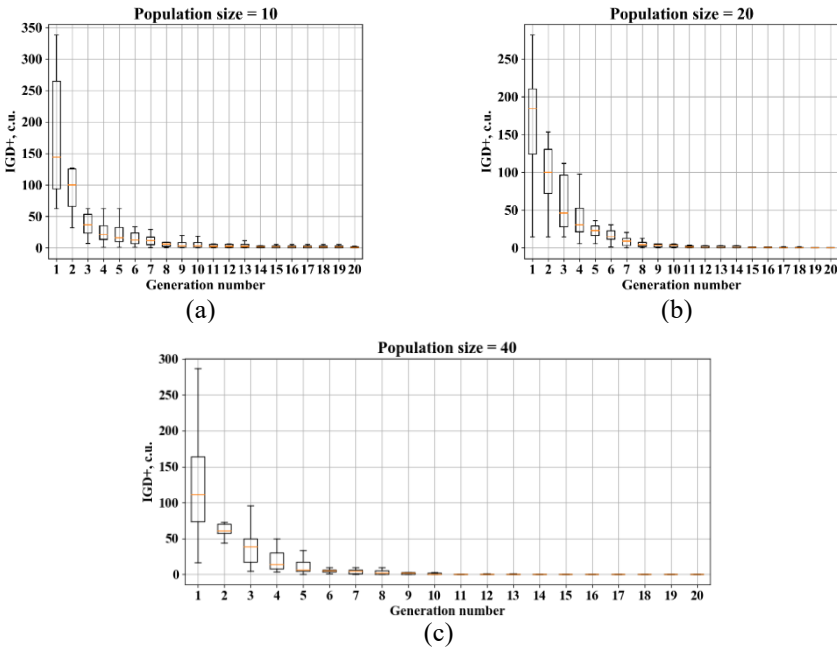


Fig. 10. Convergence of NSGA II with UX and population size of 10 (a), 20 (b), 40(c).

It is seen from the plots, that the greater the population, the less the variation from run to run.

Figure 11 shows the distribution of Pareto optimal solution in the design space for NSGA II with UX, $N_{pop} = 10$, the number of generations to termination is 50.

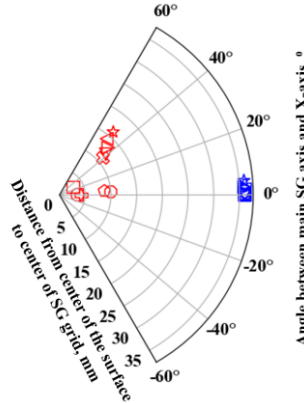


Fig. 11. Distribution of Pareto optimal solutions in the design space, red and blue markers correspond to SGs experiencing tension and compression, respectively.

3.2 Solving the problem using a traditional approach and comparing performance with NSGA II

The problem is solved using a traditional approach by sampling the design space with a Sobol sequence and determining the Pareto set, followed by calculating IGD+ based on the reference Pareto set. The number of elements in the Sobol sequence is 1000.

For benchmarking, NSGA II is executed for $N_{pop} = 20$ and for each crossover type, the number of generations to termination is 20. Table 2 presents the benchmarking results. For NSGA II, the average values of IGD+ are shown.

Table 2. Benchmarking results.

Parameter name	ICD+
Traditional approach	12.12
NSGA II + SBX	0.83
NSGA II + UX	1.42
NSGA II + 1-point	2.90

Based on benchmarking results, Fig. 12, the efficacy of NSGA II in solving the problem of determining the optimal placement of strain gauges on the elastic element of force sensors is confirmed.

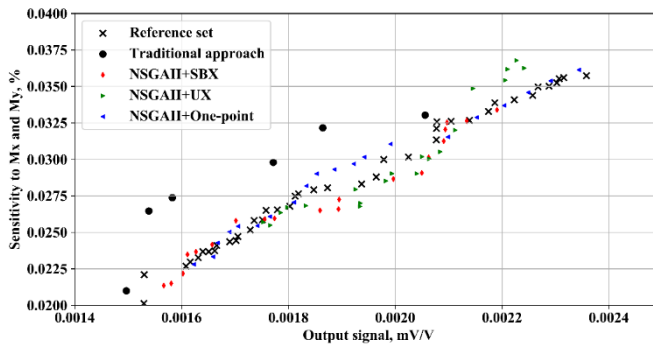


Fig. 12. Results obtained from NSGA II with different settings and from the traditional approach.

The Python module the framework is based upon can be used as a part of the system for automation and control of the process of managing lifecycle of strain-gauged force transducers described in previous papers [21, 22]. Moreover, the formulation of the placement problem can be readjusted to include such goal functions as nonlinearity or temperature error from non-uniform heating.

4 Conclusions

This paper presents a digital education tool for studying the performance of a multi-objective genetic algorithm used to solve the problem of optimal placement of strain gauges on the elastic element of a force sensor. The paper formulates the problem statement, and specifies how this problem can be solved using the MOGA NSGA II. The solution algorithm is implemented within a framework that can be used to investigate and learn how parameters of NSGA II influence its performance. The framework is used to run computational experiments for the given problem of determining the optimal placement of strain gauges on the elastic element of a given force sensor. Compared to a traditional approach, NSGA II gives better results in terms of Pareto set at fewer computational costs. Python modules used to build the framework can be used as submodules in a system for optimal design of strain-gauged force sensor, whereby better characteristics of strain gauges can be obtained just by changing placement of strain gauges without changing the elastic element.

References

1. E.V. Smirnova, A.A. Dobrjkov, A.P. Karpenko, V.V. Syuzev, *Mentally structured educational technology and engineers preparation quality management*, Commun. Comput. Inf. Sci. **754**, pp. 119-132 (2017)
2. A.I. Vlasov, L.V. Juravleva, V.A. Shakhnov, *Visual environment of cognitive graphics for end-to-end engineering project-based education*, J. Appl. Eng. Sci. **17**, pp. 99–106 (2019)
3. I.G. Doronkina, E.E. Krasnovskiy, E.V. Zakharova, P.V. Ulianishchev, L.V. Ulyanishcheva, *Academic cloud services: Innovative solutions and international practice*, J. Adv. Res. Dyn. Control Syst. **11**, pp. 65–72 (2019)

4. N.M. Mezhenay, O.V. Pugachev, *On perception of computer algebra systems and Microsoft excel by engineering students*, Probl. Educ. 21st Century. **77**, pp. 379–395 (2019).
5. N.M. Mezhenay, O.V. Pugachev, *On the results of using interactive education methods in teaching probability theory*, Probl. Educ. 21st Century. **76**, pp. 678–692 (2018)
6. J. Blank, K. Deb, pymoo: Multi-objective Optimization in Python. Preprint at <http://arxiv.org/abs/quant-ph/0208066> (2020).
7. S. Gavryushin, V. Godzikovsky, S. Gavrilenko, *Investigation of the sensitivity of a strain gauge force sensor to bending moment*, In: E.A. Mikrin, D.O. Rogozin, A.A. Aleksandrov, V.A. Sadovnichy, I.B. Fedorov, V.I. Mayorova, (eds.), Proceedings of the XLIII Academic Space Conference: dedicated to the memory of academician S.P. Korolev and other outstanding Russian scientists , Pioneers of space exploration, Moscow, AIP (2019). <https://doi.org/10.1063/1.5133317>
8. X. Li, H. He, H. Ma, *Structure design of six-component strain-gauge-based transducer for minimum cross-interference via hybrid optimization methods*, Structural and Multi-disciplinary Optimization, **60**(1), pp. 301-314 (2019).
9. A.R. Tavakolpour-Saleh, A.R. Setoodeh, M.A. Gholamzadeh, *novel multi-component strain-gauge external balance for wind tunnel tests: Simulation and experiment*, Sensors and Actuators A: Physical, **247**, pp. 172-186 (2016).
10. A.R. Tavakolpour-Saleh, M.R. Sadeghzadeh, *Design and development of a three-component force/moment sensor for underwater hydrodynamic tests*, Sensors and Actuators A: Physical, **216**, pp. 84-91 (2014).
11. Catalog of HBM strain gauges, <https://www.hbm.cz/wp-content/uploads/s1266.pdf>, last accessed 2019/11/02.
12. Website of the pymoo library, <https://pymoo.org/>, last accessed 2019/11/02.
13. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE Trans. Evol. Comput. **6**, pp. 182–197 (2002). <https://doi.org/10.1109/4235.996017>
14. N. Agrawal, G.P. Rangaiah, A.K. Ray, S.K. Gupta, *Design stage optimization of an industrial low-density polyethylene tubular reactor for multiple objectives using NSGA-II and its jumping gene adaptations*, Chem. Eng. Sci. **62**, pp. 2346–2365 (2007). <https://doi.org/10.1016/j.ces.2007.01.030>
15. E.G. Bekele, J.W. Nicklow, *Multi-objective automatic calibration of SWAT using NSGA-II*, J. Hydrol. **341**, pp. 165–176 (2007). <https://doi.org/10.1016/j.jhydrol.2007.05.014>
16. S. Honda, T. Igarashi, Y. Narita, *Multi-objective optimization of curvilinear fiber shapes for laminated composite plates by using NSGA-II*, Compos. Part B Eng. **45**, pp. 1071–1078 (2013). <https://doi.org/10.1016/j.compositesb.2012.07.056>
17. Y. Li, S. Liao, G. Liu, *Thermo-economic multi-objective optimization for a solar-dish Brayton system using NSGA-II and decision making*, Int. J. Electr. Power Energy Syst. **64**, pp. 167–175 (2015). <https://doi.org/10.1016/j.ijepes.2014.07.027>
18. Z. Moravej, F. Adelnia, F. Abbasi, *Optimal coordination of directional overcurrent relays using NSGA-II*, Electr. Power Syst. Res. **119**, pp. 228–236 (2015). <https://doi.org/10.1016/j.epsr.2014.09.010>
19. S. Chand, M. Wagner, *Evolutionary many-objective optimization: a quick-start guide*, Surv. Oper. Res. Manag. Sci. **20**, pp. 35–42 (2015)

20. K. Deb, *Multi-objective genetic algorithms: problem difficulties and construction of test problems*, *Evol. Comput.* 7, pp. 205–230 (1999). <https://doi.org/10.1162/evco.1999.7.3.205>
21. S.I. Gavrilentov, S.S. Gavriushin, V.A. Godzikovsky, *Multicriteria approach to design of strain gauge force transducers*, Proceedings of the Joint IMEKO TC1-TC7-TC13-TC18 Symposium, St. Petersburg, IOP Publishing (2019). <https://doi.org/10.1088/1742-6596/1379/1/012010>
22. S.I. Gavrilentov, S.S. Gavryushin, *Development and Performance Evaluation of a Software System for Multi-objective Design of Strain Gauge Force Sensors*, Zh. Hu, S. Petoukhov, M. He, (eds.), Proceedings of the International Symposium on Computer Science, Digital Economy and Intelligent Systems CSDEIS 2019, Moscow, Springer (2020). https://doi.org/10.1007/978-3-030-39216-1_21