

# Locating Energy Storage Devices in Power Grids Using Evolutionary Algorithms

## Author

Jarosław Korpikiewicz

## Keywords

evolutionary algorithms, genetic operators, power grid, energy storage, locating

## Summary

The article presents the premises of the application for locating energy storage devices in the HV power grid. The substation number to which it is to be connected and the value of its power are defined as the storage location. The connected energy storage devices are designed to increase the total capacity of the connected renewable energy sources without compromising any technical constraints of the power grid. The location problem is defined as the process of optimisation with nonlinear constraints implemented in the MATLAB environment, using the application for calculating power distribution. In order to compare the locations with each other, a fixed sum of absolute values of the storage devices' capacity was assumed. Evolutionary algorithms were used to implement the optimisation process. Due to the non-linearity of constraints, a new function of creating the initial population and eight genetic operators were designed. Most of the tests were carried out in two versions, with and without energy storage devices connected, after which the increase in the possibility of introducing additional generation in both variants was compared. Then, many tests were carried out to determine the parameters and select the algorithm version. Based on the results, an application for optimising the location of storage devices was created.

**DOI: 10.12736/issn.2300-3022.2018303**

Received: 26.04.2017

Accepted: 17.07.2017

Available online: 15.02.2019

## 1. Introduction

The purpose of this paper is to examine the possibility of using the chosen optimisation method in order to maximise the safe power output from the RES generation for an HV grid with preset parameters. The basic technical constraints are the overloading of grid elements and the failure to maintain voltage within acceptable limits in grid substations. In order to solve the problem of maximising power output from renewable energy sources, the optimisation problem, all constraints and requirements should be defined. On this basis, a mathematical form of the problem should be formulated. Tools and methods should be chosen based on the analysis of the optimisation problem. The purpose of the paper is not to determine the choice of technology or the storage capacity of storage devices, but only their required installed power and their locations. The problem of the location of storage devices and the location of additional generation has a large impact on the occurrence of possible overloads, voltage changes in substations and the amount of the total active power input into the grid by additional sources. For this reason, during

the optimisation process, the locations of storage devices and the locations of additional sources are searched for, for which the objective function will be optimal. The power grid model is the input data of the application.

## 2. The concept of the computational application for testing the connection capacity of wind farms

To build this system, evolutionary algorithms and the application performing the distribution of active and reactive power in the power system analysed will be used. The use of evolutionary algorithms is dictated by the difficulty of determining the analytical form of the objective function, especially for large network models. Evolutionary algorithms are an attractive optimisation tool especially for problems for which no specialised methods have been developed. Details on genetic/evolutionary algorithms have been presented in monographs [1, 2, 3]. The objective function and constraints are determined on the basis of power distribution simulations. Classic optimisation methods

have additional requirements, e.g. of continuity or differentiability of the objective function, analytical form of the objective function and constraints. To assess the effects of changing the active or reactive power in a given substation, power distribution must be recalculated. It is not possible to take into account the constraints in the analytical form, because possible overloading of grid elements or exceeding the permissible voltage ranges in substations depend on the grid itself (grid model) and its status (active and reactive power values in individual substations, switch states and the state of additional generation). If we want to justify the choice of the optimisation method, we have to analyse its advantages and disadvantages. Advantages of genetic/evolutionary algorithms [3]:

- The method is universal, in order to use the same program in a different problem, it is usually enough to change the objective function and the constraints
- The classic genetic algorithm has a simple design, programming genetic algorithms in a universal programming language is not difficult
- The objective function or nonlinear constraints do not have to be stated in analytical form. Knowing how to determine/calculate it is sufficient information
- Evolutionary algorithms can also cope where the optimised function is subject to noise, changes over time, has many local extremes. Evolutionary algorithms are one of the methods of global optimisation
- To find a solution, we do not need to know anything about the function being optimised. We don't even need to have an objective function at all: we can use genetic algorithms even when the only thing we can say about points in the state space is which of the two (several) solutions is better (when we apply the tournament selection operator)
- Because the genetic algorithm is a randomised algorithm, we can repeat the calculations many times in the hope of getting better results. Individual results will be different because we are receiving an approximate solution.

Disadvantages of these algorithms [3]:

- The method is universal, so not as effective as specialised algorithms can be (in order to improve the efficiency/accuracy of solutions, it is recommended to use hybrid algorithms)
- The method is slower than simple heuristics (e.g. greedy method), although usually more effective. Often, the optimisation of real technical problems may require a large amount of RAM and take a long time, e.g. a dozen or so days
- Success is only possible if the problem is correctly coded and the objective function is selected appropriately. Unfortunately, there is no unambiguous theory of how to achieve it. Often – an in the selection of mutation and crossover parameters – it is up to the programmer's intuition and experience. This requires multiple versions of the application and many simulation tests to be carried out
- Because a genetic algorithm is a randomised algorithm, we are never sure that we have found the optimal solution (advice: settle for an approximate solution). In reality, for example in manufacturing, it is not necessary to find the exact optimum, but to find a better solution than the competition.

In many applications, how long it took to obtain a solution is also important, even at the cost of its accuracy

- Taking into account nonlinear constraints may be difficult to implement. It may require the design of new algorithms for genetic operators. In this case, the method is not straightforward
- Taking into account problem-specific requirements may require the creation of a new function that creates the initial population, functions that implement genetic operators, and a scaling function. Then this method is no longer universal.

Evolutionary algorithms are not without flaws, however, due to the specificity of the constraints, they have been chosen to build the application. In order to accomplish this task, a computational simulation environment was used to enable the analysis of power systems and evolutionary algorithms. Such environment is MATLAB software with the Global Optimization Toolbox installed (containing libraries for creation and simulation of evolutionary algorithms [4]) with MATLAB-based software, used for the analysis of power systems. One of many available applications which can be used as MATLAB'S toolbox is MATPOWER [5]. The creation of such applications (analysis of power systems, genetic algorithms, optimisation methods) in a universal programming language (without the use of ready-made and tested libraries) is very time-consuming. Thanks to the use of a universal computing environment, building a new application can be done in a standard manner used for computational applications, focusing on the problem being solved, using the toolboxes that extend the functionality of this environment. The problems of laborious integration of software from different manufacturers, solving numerical problems should be avoided. The mathematical form of the problem is determined by means of a system of eight relationships:

$$X = \begin{bmatrix} p\_fw(1) \\ \dots \\ p\_fw(w) \\ p\_mag(1) \\ \dots \\ p\_mag(w) \end{bmatrix}$$

– form of the individual (1)

$$\max \left( \sum_{i=1}^w p\_fw(i) \right)$$

– maximising the objective function (2)

$$\prod_{j=1}^w 0 \leq p\_fw(j) \leq FW\_MAX$$

– limitation of generation power (3)

$$\prod_{k=1}^w -MAG\_MIN \leq p\_mag(k) \leq +MAG\_MAX$$

– limitation of storage power (4)

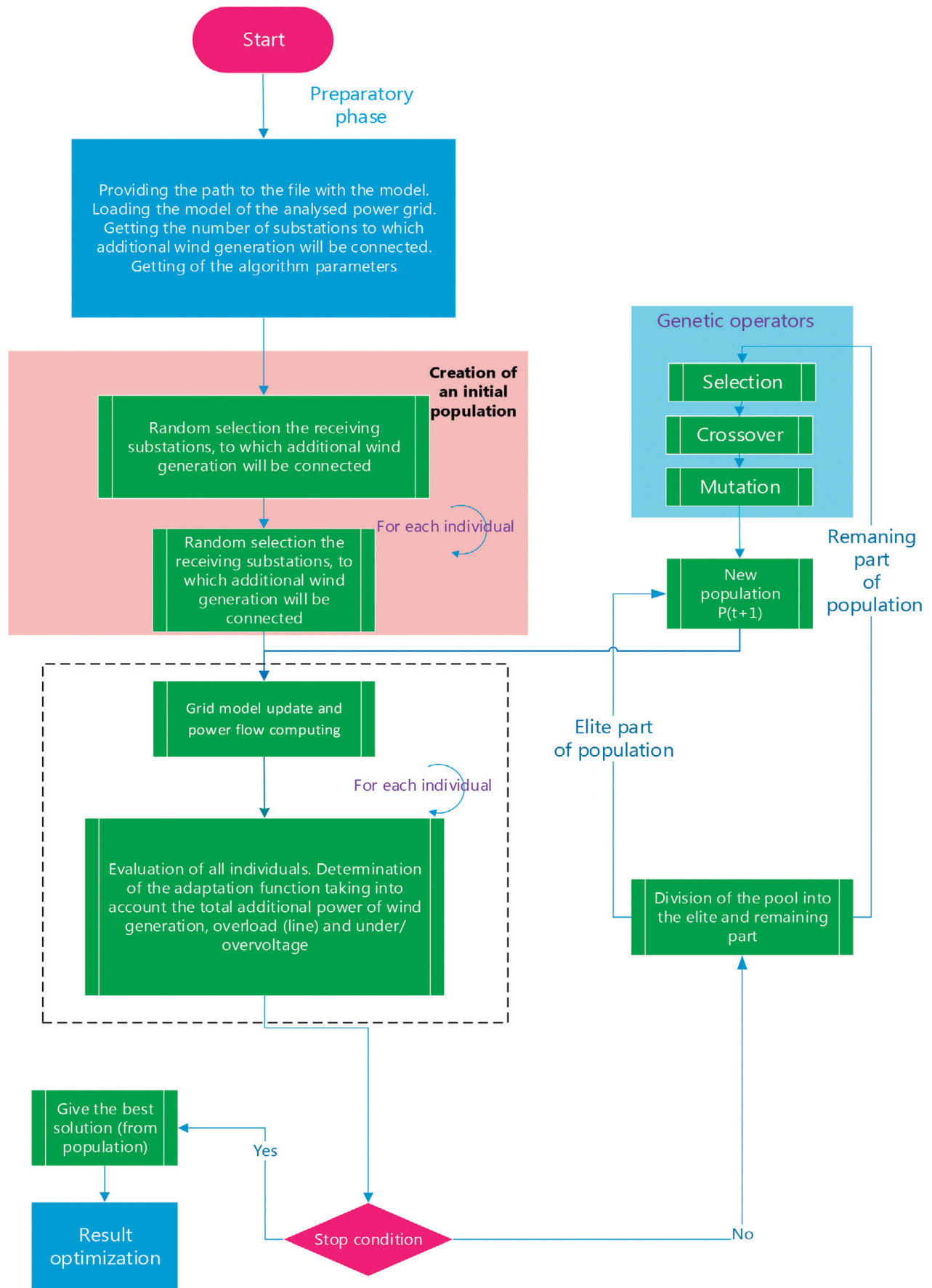


Fig. 1. Diagram of the evolutionary algorithm for determining the maximum total additional power generated in the analysed grid without the participation of the storage device

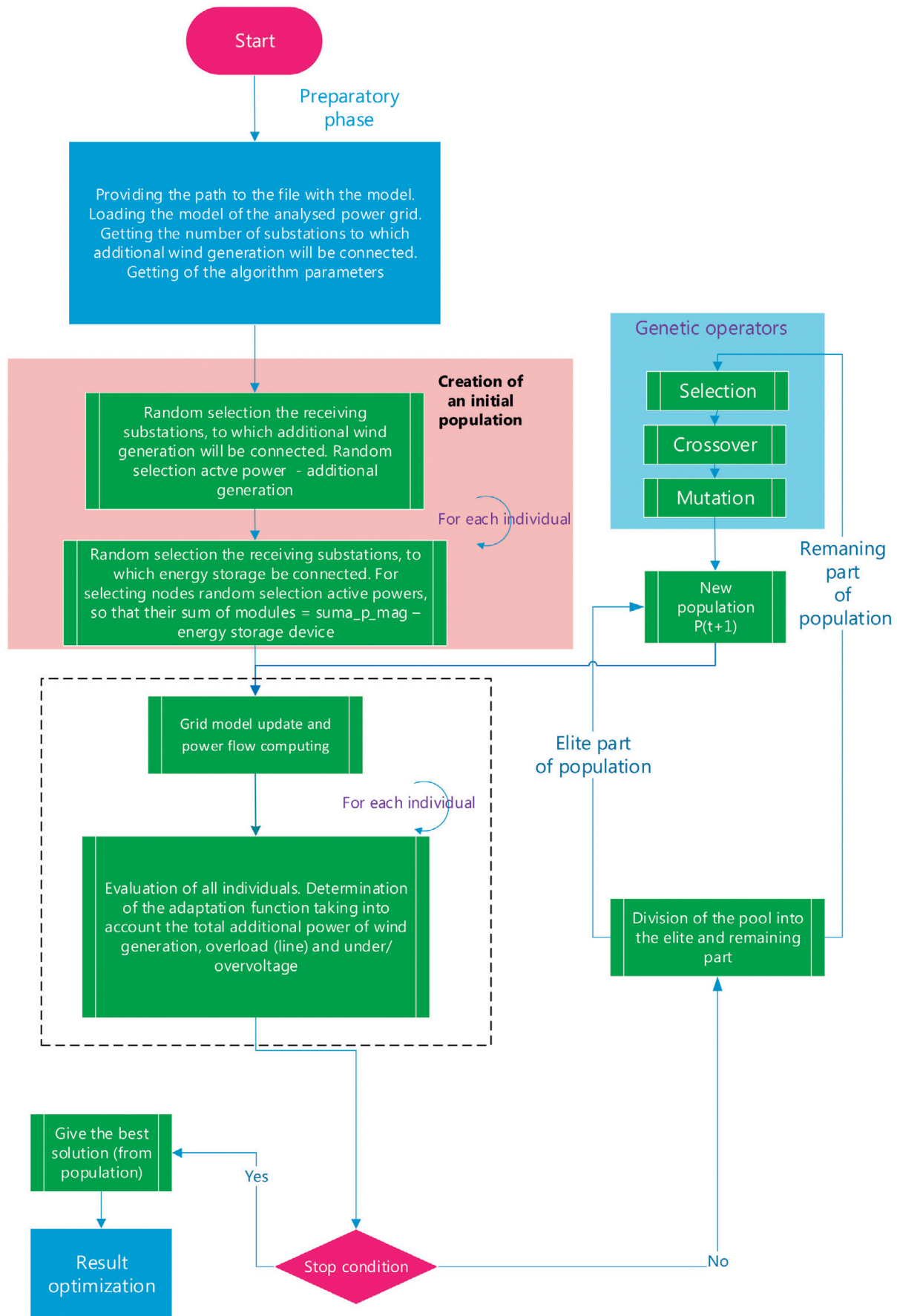


Fig. 2. Diagram of the evolutionary algorithm for determining the maximum total additional power generated in the analysed grid working with energy storage devices

$$\sum_{m=1}^w |p\_mag(m)| = suma\_p\_mag$$

- equality constraint

$$\prod_{l=1}^L S_{obc}(l) \leq S(l)$$

- load capacity constraint

$$\prod_{n=1}^w V_{min}(n) \leq V(n) \leq V_{max}(n)$$

- maintaining the voltage values within the permissible range

$$[S_{obc}, V] = rozpyw(X, mpc)$$

- implementation of distribution calculations

Where:

$X$  – the form of the proposed solution (individual),  $p\_fw$  – vector of active power of additional generation [MW] with the length  $w$ ,  $FW\_MAX$  – maximum active power of the storage device connected to the substation [MW],  $p\_mag$  – vector of active power of the storage device [MW] with length  $w$ ,  $MAG\_MIN$

(5)

(6)

(7)

(8)

and  $MAG\_MAX$  – maximum active power of the storage device connected to the substation in charging and discharging mode [MW],  $w$  – number of grid substations,  $L$  – number of grid lines,  $suma\_p\_mag$  – sum of active power modules of all connected storage devices to the grid [MW],  $S_{obc}$  – load (apparent power) vector of individual lines determined after the distribution of active and passive powers with length  $L$ ,  $S$  – load vector of individual lines (grid model) with length  $L$ ,  $V$  – vector of voltage modules in individual grid substations determined after the distribution of active and passive powers with length  $w$ ,  $V_{min}$ ,  $V_{max}$  – vector of minimum/maximum allowable voltages in individual network nodes, length  $w$ ,  $mpc$  – model of the analysed grid,  $rozpyw$  (distribution) – function updating the grid model in accordance with the form of individual  $X$  (powers of the additional generation  $p\_fw$ , powers of the storage device  $p\_mag$ ), performing the distribution of AC active and passive powers, as a result of which we receive loads of individual lines and voltage values in individual substations. Optimisation constraints can be included in the form of a penalty function or a nonlinear constraint function, etc. The first method consists in changing the form of the objective function [8]. The second method offered by the MATLAB environment is to create a constraint function [4]. The constraint (5) makes it possible to compare two solutions with the same storage capacity (the total sum of the absolute

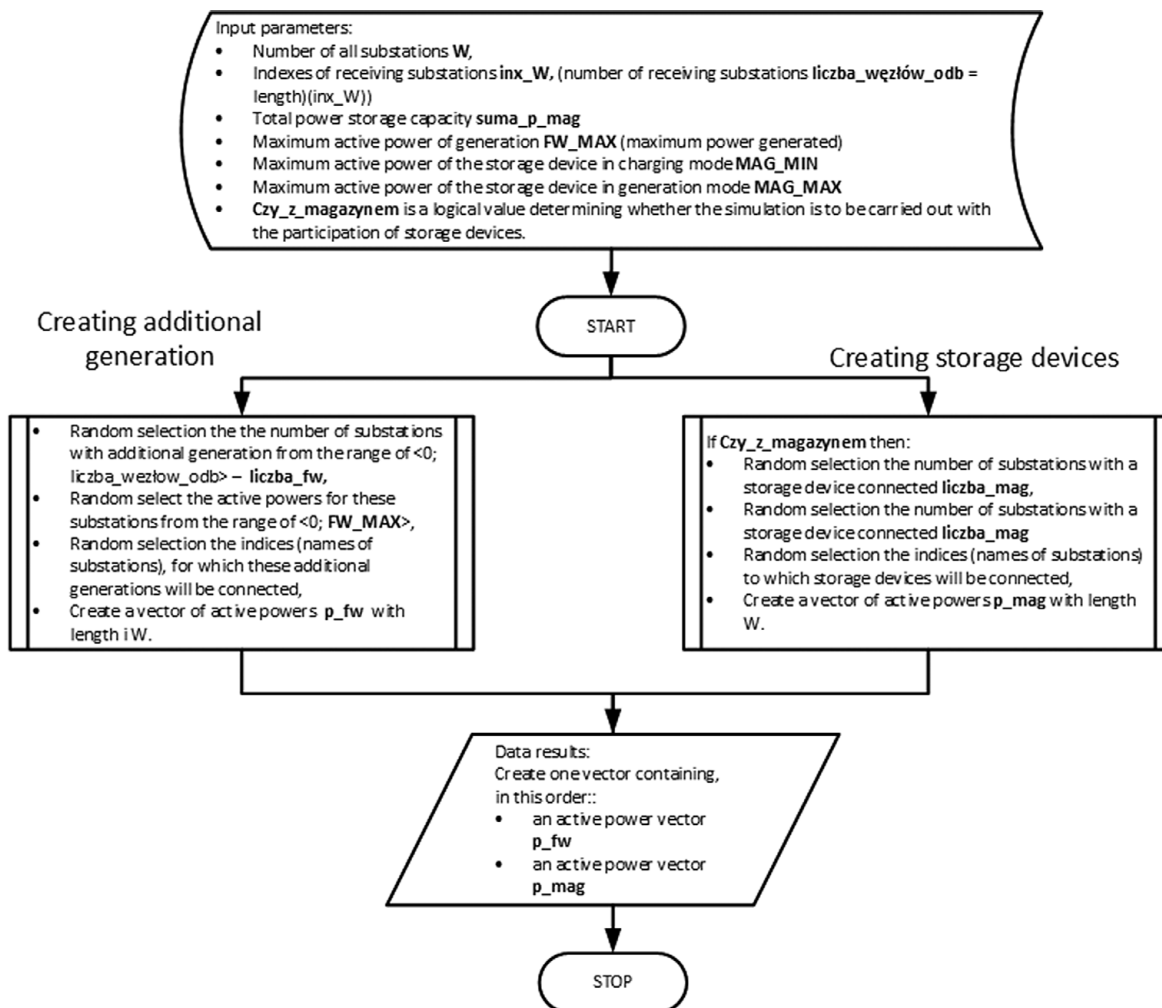


Fig. 3. Block diagram of creating an initial population

values of active powers – regardless of the charging mode, i.e. the negative power or discharging value, i.e. the positive power value).

In order to investigate the impact of storage connection on the possibility of connecting additional generation to the grid, two simulations should be performed.

The first simulation is without the participation of the storage device, its algorithm is shown in Fig. 1. The second algorithm is shown in Fig. 2. Comparison of the results from both simulations will allow to determine whether it is possible in a given grid to increase the maximum total power of the additional generation in a safe manner when energy storage devices are connected to the grid.

The elite strategy has been used in the MATLAB environment. This makes it impossible to “forget” the good solutions found in the generations preceding the last one. However, it can cause problems with premature convergence. Fig. 3 presents one of many presented genetic operators – the operator of creating the initial population. Detailed description and simulations are presented in [7]. A standard 30-substation IEEE grid was selected as the test power grid. The requirement was to determine the loadability of all grid elements in the model.

### 3. Simulations

While testing the effectiveness of the designed genetic operators, the algorithm with the nonlinear constraint function was started six times, with a defined standard selection operator and a scaling function for different application versions (all operators, no operator no. 5, no operator no. 3, no operators no. 3 and 5). The tests were carried out for all possible combinations. Tab. 1 and 2 below show selected simulation results.

The next stage of the simulation was the study of the correctness of the algorithm’s work and its convergence. The study involved running the algorithm 100 times with the nonlinear constraint function (Fig. 4), and with the external penalty function (Fig. 5) – grid operation only with the storage device.

In the next stage of the simulation, an analogous test was carried out, but for the version with the external penalty function, as illustrated in Fig. 5. Another test involved a simulation with different probabilities of implementation of individual genetic operators both in the version with the nonlinear constraint function (Tab. 3), and with the external penalty function (Tab. 4).

Result parameter	Statistics	Algorithm variants (algorithm run 6 times)			
		8 operators (all)	No operator no. 5	No operator no. 3	No operators no. 3 and 5
Active power of generation with connected storage devices <i>suma_p_fw_m</i> [MW]	Maximum	341.33	341.26	341.08	<b>343.23</b>
	Minimum	337.85	339.24	338.61	<b>340.60</b>
	Average	340.13	340	340	<b>342.25</b>
	Difference = max. – min.	3.48	2.02	2.47	<b>2.63</b>
Active power of generation without connected storage devices <i>suma_p_fw</i> [MW]	Maximum	341.48	332.06	343.08	<b>332.42</b>
	Minimum	338.73	328.95	338.18	<b>330.16</b>
	average	339.97	330.59	341.3	<b>331.52</b>
	Difference = max. – At least	2.75	3.11	4.9	<b>2.26</b>
The difference between the maximum generation with storage devices and maximum generation without them $suma\_p\_fw\_m - suma\_p\_fw$ [MW]		-0.15	9.2	-2	10.81

Tab. 1. Results of six iterations to test the effectiveness of genetic operators using the *selectionroulette* selection function and the *fitscalingrank* scaling function

Result parameter	Statistics	Algorithm variants (algorithm run 6 times)			
		7 operators (all)	No operator no. 5	No operator no. 3	No operators no. 3 and 5
Active power of generation with connected storage devices <i>suma_p_fw_m</i> [MW]	Maximum	339.66	341.48	341.12	<b>342.75</b>
	Minimum	336.99	334.9	337.67	<b>338.78</b>
	average	338.39	339.44	339.32	<b>340.31</b>
	Difference = max. – At least	2.67	6.58	3.45	<b>3.97</b>
Active power of generation without connected storage devices <i>suma_p_fw</i> [MW]	Maximum	341.27	332.68	341.15	<b>332.6</b>
	Minimum	338.29	329.66	337.21	<b>329.75</b>
	average	339.56	331.24	339.21	<b>331</b>
	Difference = max. – At least	2.98	3.02	3.94	<b>2.85</b>
The difference between the maximum generation with storage devices and maximum generation without them $suma\_p\_fw\_m - suma\_p\_fw$ [MW]		-1.61	8.8	-0.03	<b>10.15</b>

Tab. 2. Results of six iterations to test the effectiveness of genetic operators using the *selectionroulette* selection function and the *fitscalingtop* scaling function

### 4. Conclusions

- I. Evolutionary algorithms can be effectively used in order to find the optimisation of the location of the energy storage device and additional generation. However, this requires the design of genetic operators and a function for creating the initial population, taking into account nonlinear constraints. The standard forms of functions implementing genetic operators have been analysed. They are not capable of properly creating and modifying individuals, i.e. not violating the nonlinear constraints of the problem. Thus, evolutionary algorithms for solving technical problems in which there are nonlinear constraints require modifications taking into account the specificity of the problem.
- II. From among the tested versions of the evolutionary algorithm, the algorithm with the external penalty function turned out to be the best. The maximum power of the additional generation of 369.20 [MW] was obtained with the total storage capacity equal to 19 [MW] (Tab. 4). The result of the impact of the operation of the storage device was assessed by comparing it with the result of the optimisation of the additional generation itself (without the storage device). For the tested grid, the maximum power of the additional

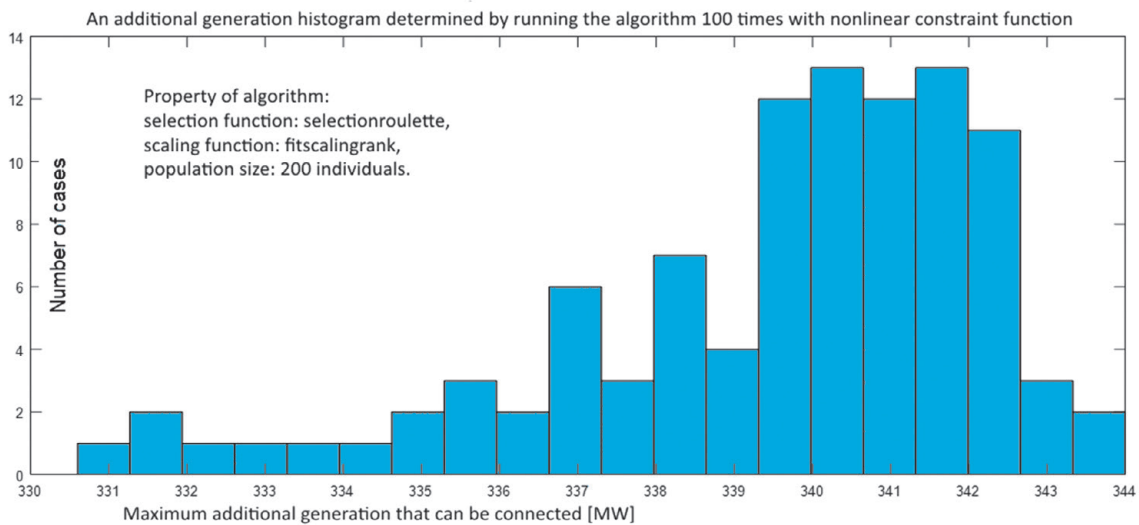


Fig. 4. Additional generation histogram for the algorithm version with nonlinear constraint function

No.	pstwo_mut	pstwo_mut_fw	pstwo_zam_fw	pstwo_zam_mag	pstwo_zam_p	Objective function (with storage device)	Objective function (without storage device)
1	0.81472369	0.90579194	0.12698682	0.91337586	0.63235925	350.09	331.63
2	0.91818459	0.63255766	0.63960491	0.11794852	0.64830387	351.88	332.21
3	0.23048253	0.61061998	0.77947319	0.56706282	0.23407947	351.55	332.79
4	0.88485522	0.18345035	0.22609177	0.60591373	0.94860687	352.83	332.22
5	0.63911062	0.82985738	0.17512177	0.04191938	0.50305758	352.41	332.81
6	0.92049838	0.4880307	0.25213702	0.54685008	0.70505076	352.73	332.90
7	0.46356272	0.88610119	0.65836403	0.7455946	0.21304558	353.12	331.75
8	0.48116548	0.36855	0.53696344	0.06340336	0.30413755	351.59	333.48
9	0.45041356	0.01232514	0.09299071	0.86499249	0.31718746	351.94	332.08
10	0.7551757	0.48476413	0.48879031	0.3981366	0.20264438	352.03	332.97
11	0.61582818	0.06237602	0.26374273	0.16379723	0.11869841	350.99	332.84
12	0.9666019	0.06840756	0.73837416	0.88490846	0.38839793	350.48	333.32
13	0.39193069	0.35246639	0.0508462	0.5282447	0.33001492	352.51	333.26
14	0.14467151	0.45589683	0.41970739	0.87595398	0.90709547	351.03	333.58
15	0.137272	0.41795503	0.81303064	0.57288986	0.43202267	352.37	333.09
16	0.84251773	0.30210997	0.666352	0.1670966	0.58915655	352.71	332.68
17	0.64221802	0.766998	0.62243137	0.01869434	0.01212781	352.51	332.52

The optimal operator probability values for the implementation of operators are highlighted in red (the largest value of the objective function for grids with energy storage devices)

Tab. 3. Results of the simulation of the search for the values of genetic operators' parameters for the algorithm with the nonlinear constraint function

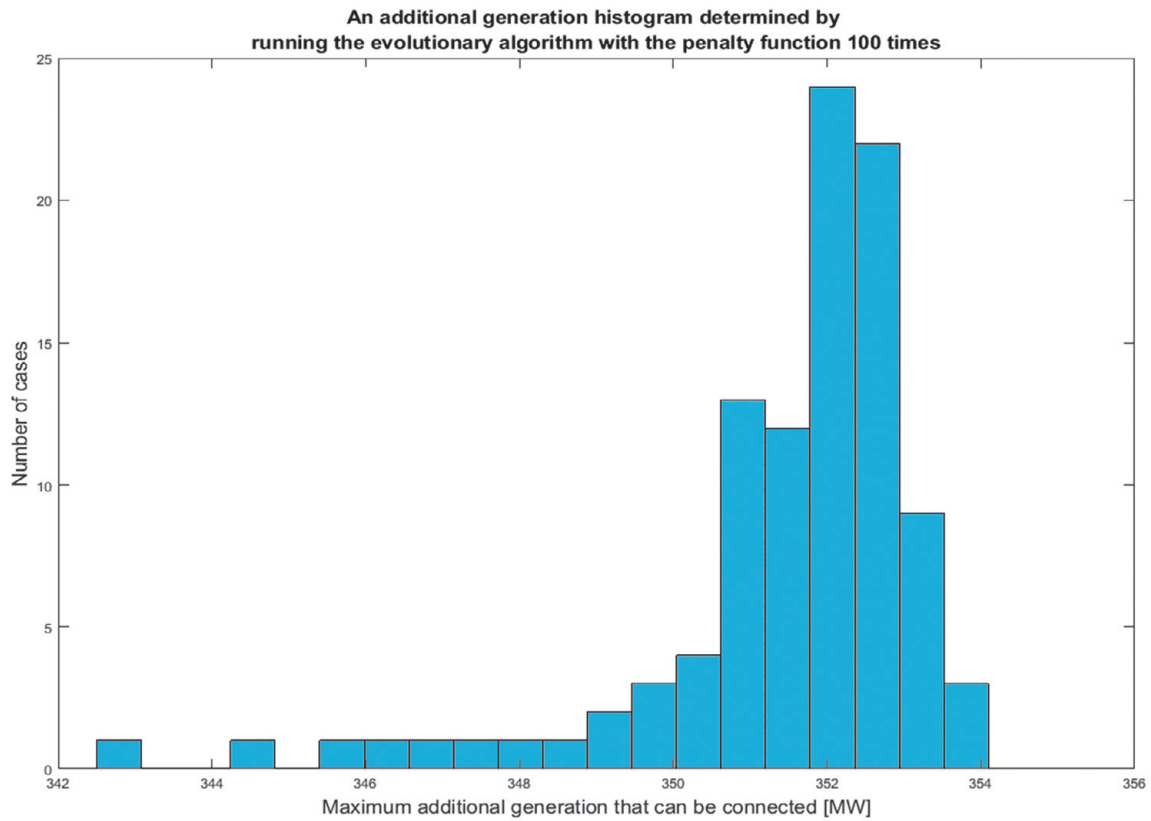


Fig. 5. Additional generation histogram for the algorithm version with external penalty function

No.	pstwo_mut	pstwo_mut_fw	pstwo_zam_fw	pstwo_zam_mag	pstwo_zam_p	Objective function (with storage device)	Objective function (without storage device)
1	0.821579	0.724182	0.835272	0.703187	0.851299	353.98	333.34
2	0.20824	0.788091	0.957554	0.392875	0.270881	352.87	332.88
3	0.586582	0.907114	0.71906	0.09236	0.853622	353.41	333.89
4	0.166876	0.03903	0.195391	0.089744	0.226574	353.54	333.75
5	0.711179	0.859306	0.231377	0.290049	0.24363	354.77	332.96
6	0.860269	0.056848	0.666054	0.744647	0.627422	353.42	333.39
7	0.122104	0.727066	0.700456	0.08631	0.598401	352.30	332.88
8	0.250954	0.081737	0.180487	0.295582	0.451101	353.19	333.22
9	0.553292	0.326571	0.531779	0.348433	0.021119	353.05	333.76
10	0.991492	0.288026	0.548958	0.228424	0.339291	351.75	333.36
11	0.826123	0.676308	0.193865	0.067948	0.877198	353.52	333.31
12	0.070057	0.609323	0.620526	0.30726	0.431354	369.20	333.90
13	0.833288	0.448672	0.247334	0.061462	0.62534	354.18	333.77
14	0.258424	0.820662	0.3408	0.389855	0.408813	352.98	333.29
15	0.650812	0.767099	0.379532	0.776901	0.307508	352.92	334.14
16	0.289789	0.681109	0.522761	0.012858	0.646947	353.06	333.02
17	0.774671	0.069752	0.679253	0.150766	0.201864	352.24	333.83
18	0.499141	0.708158	0.388722	0.864053	0.789496	353.23	333.37
19	0.650764	0.536774	0.7932	0.956314	0.8308	354.21	332.71

The optimal operator probability values for the implementation of operators are highlighted in red (the largest value of the objective function for grids with energy storage devices)

Tab. 4. Results of the simulation of the search for the values of genetic operators' parameters for the algorithm with the external penalty function



generation in the grid without the participation of energy storage devices was 333.9 [MW]. The maximum power of the additional generation could not significantly affect the voltage and load limits.

- III. A characteristic feature of optimisation methods with the external penalty function is that solutions may slightly violate some of the constraints. The size of these violations can be reduced by applying the appropriate high values of the penalty coefficients. In the presented solution, there were no voltage exceedances, and the overload concerned only two lines with a maximum of 1.7% of their load capacity (Tab. 5). However, the best solution obtained during the examination of the influence of other parameters does not show any exceedances.
- IV. In the version of the evolutionary algorithm with nonlinear constraints, no solutions violating any constraints were obtained. This is the advantage of this type of algorithm.
- V. The evolutionary algorithm uses a distribution algorithm in order to find the optimisation of the location of the energy storage device and additional generation. It is used to determine whether the solution obtained does not overload any grid component and whether the voltage values in all substations are within acceptable ranges. Depending on the version of the algorithm, the distribution was performed in the nonlinear constraint function or in the objective function.
- VI. The version of the evolutionary algorithm with nonlinear constraints turned out to be not as effective as the version with the external penalty function. This is due to the fact that in both versions there are different conditions for stopping the algorithm in the MATLAB environment. The version of the algorithm with nonlinear constraints is stopped based on numerical derivative constraints. Algorithms of this type are performed using a small number of generations. Thus, the

problem of premature convergence appears, which disqualifies this type of algorithm for most applications. Typically, evolutionary algorithms require about 100 generations to obtain solutions close to the extreme value. Our application with the nonlinear constraint algorithm implemented only 6 generations.

- VII. Evolutionary or genetic algorithms are optimisation methods that find solutions near the value of the global extreme. They do not guarantee finding the exact extreme value. However, in most applications, finding a solution with a small error is not a problem. In the analysed problem, an error below 1 [MW] is of no practical significance.
- VIII. The evolutionary algorithm designed is sensitive to the values of the probabilities of implementation of individual operators.
- IX. Evolutionary or genetic algorithms are randomised methods. This results in the fact that the obtained results for the same input data can be different if the algorithm is run several times. This is a feature of all randomised methods. The evolutionary algorithm designed is convergent, which is confirmed by the histogram of running the algorithm 100 times (Fig. 5). The histogram shows that 46% of the obtained results are in the range of 351.8–352.9 [MW], but there are single solutions with much smaller values. Therefore, in order to optimise, the algorithm should be run several times and the best solution obtained should be considered as the result.
- X. Time optimisation of the algorithm and testing of other factors on the effectiveness and convergence of the evolutionary algorithm requires further research. Efficiency could increase the creation of applications in a universal programming language based on a designed algorithm, which will allow optimal use of RAM and other computer resources. MATLAB environment is a universal tool enabling quick prototyping of computational/control algorithms.

Substation number	Active power [MW]	Part of vector	Substation number	Active power [MW]	Part of vector	Substation number	Active power [MW]	Part of vector	Substation number	Active power [MW]	Part of vector
1	0	p_fw	16	20	p_fw	1	0	p_mag	16	-0,632	p_mag
2	0		17	10,811		2	0				
3	1,183		18	20		3	-8,625				
4	12,748		19	1,289		4	0				
5	13,152		20	10,327		5	-8,352				
6	15,641		21	9,730		6	0				
7	18,243		22	0		7	0				
8	12,758		23	0		8	0				
9	18,084		24	20		9	-1,331				
10	13,922		25	19,968		10	0				
11	20		26	20		11	0				
12	0		27	0		12	0				
13	0		28	20		13	0				
14	16,406		29	20		14	0				
15	20		30	19,716		15	0				

Tab. 5. An example of a solution obtained from an evolutionary algorithm with a penalty function during convergence testing

XI. In the case of analysing a larger-sized electric grid, the algorithm with the penalty function will determine several substations in which storage devices should be installed. The algorithm with the parameters changed (grid model for statistical simulation of renewable generation work and power demand) will identify substations that effectively increase the connectivity of the renewable generation, which is the issue of optimising the location of storage devices in the actual grid.

These substations will be referred to as characteristic substations. Not every power grid has such substations. The solution to this algorithm is the active power vector of the installed renewable generations (wind farms, photovoltaic power plants) and the vector of required capacity of installed energy storage devices. The sign of finding a solution with characteristic substations will be the fact that the objective function of this individual should be much smaller than other individuals in the population, and the fact that there were no significant constraints in any state. The task of the developed system with the changed model is to find characteristic substations in the grid. This will require not only finding the best solution, but also analysing the remaining individuals of the population.

#### REFERENCES

---

1. Goldberg D.E., "Algorytmy genetyczne i ich zastosowania" [Genetic algorithms and their applications], Wydawnictwa Naukowo-Techniczne, Warsaw 1995.

2. Michalewicz Z., "Algorytmy genetyczne + struktury danych = programy ewolucyjne" [Genetic algorithms + data structures = evolutionary programs], Wydawnictwo Naukowo-Techniczne, Warsaw 2010.
3. Abras J., "Wykłady z algorytmów ewolucyjnych" [Lectures on evolutionary algorithms] Wydawnictwa Naukowo-Techniczne, Warsaw 2001.
4. Global Optimization Toolbox, User's guide. Matlab R2016a, The MathWorks, Inc. 2016.
5. Zimmerman R.D., Murillo-Sanches C.E., Matpower 5.1. User's manual, Power. System Engineering Research System, 20 March 2015.
6. Matlab, Object – oriented programming, Matlab R2016a, The MathWorks, Inc. 2016.
7. Korpikiewicz J. et al., "Opracowanie metodyki wyboru lokalizacji zasobników energii w sieci WN w celu zwiększenia zdolności przyłączenia energetyki odnawialnej do systemu dystrybucyjnego" [Development of a methodology for the selection of energy storage locations in the HV grid in order to increase the capacity of connecting renewable energy to the distribution system], statutory work of the Institute of Power Engineering Gdańsk Branch OGS-68/16, Institute of Power Engineering, Gdańsk 2016.
8. Stachurski A., Wierzbicki A., "Podstawy optymalizacji" [Fundamentals of optimization], Oficyna Wydawnicza Politechniki Warszawskiej, Warsaw 2001.

*The article was created as part of the research project GEKON1/02/213880/30/2015.*

---

#### Jarosław Korpikiewicz

Gdańsk Branch of the Institute of Power Engineering  
 e-mail: j.korpikiewicz@ien.gda.pl

Graduated with master's degree in control engineering and robotics from the Faculty of Electrical and Control Engineering of Gdańsk University of Technology (2002). He has worked in the industry as an automation and IT specialist. He is currently employed at the Gdańsk branch of the Institute of Power Engineering as a specialist. His interests include the application of artificial intelligence methods in electrical power engineering, voltage control in substations, research and implementation of fuzzy logic in embedded systems, design of controllers. In addition, he is involved in renewable energy and energy storage (Smart Grid). Mr. Korpikiewicz is a PhD student at the Department of Ship Automation at the Faculty of Electrical Engineering of the Maritime University of Gdynia. He is a member of the Gdańsk Branch of the Association of Polish Electrical Engineers and a member of the Gdańsk Branch of the Polish Society of Theoretical and Applied Electrotechnics.

This is a supporting translation of the original text published in this issue of "Acta Energetica" on pages 33–42. When referring to the article please refer to the original text.

PL

## Lokalizacja zasobników energii w sieciach elektroenergetycznych za pomocą algorytmów ewolucyjnych

### Autor

Jarosław Korpikiewicz

### Słowa kluczowe

algorytmy ewolucyjne, operatory genetyczne, sieć elektroenergetyczna, magazyn energii, lokalizacja

### Streszczenie

W artykule przedstawiono założenia aplikacji do lokalizacji zasobników energii w sieci elektroenergetycznej WN. Jako lokalizację magazynu zdefiniowano numer węzła, do którego ma być przyłączony, oraz wartość jego mocy. Przyłączone magazyny energii mają za zadanie zwiększenie mocy sumarycznej przyłączonych odnawialnych źródeł energii, przy nienaruszaniu żadnego z ograniczeń technicznych sieci elektroenergetycznej. Problem lokalizacji zdefiniowano jako proces optymalizacji z ograniczeniami nieliniowymi zrealizowany w środowisku MATLAB, wykorzystującym aplikację do obliczeń rozplywów mocy. Chcąc porównać ze sobą lokalizację, założono stałą sumę wartości bezwzględnych mocy magazynów. Do realizacji optymalizacji wykorzystano algorytmy ewolucyjne. Ze względu na nieliniowość ograniczeń zaprojektowano nową funkcję tworzenia populacji początkowej oraz osiem operatorów genetycznych. Większość badań wykonywano w dwóch wersjach z przyłączonymi magazynami energii i bez nich, po czym porównywano wzrost możliwości wprowadzenia dodatkowej generacji w obu wariantach. Następnie wykonywano wiele testów i badań w celu ustalenia parametrów i wyboru wersji algorytmu. Na podstawie wyników stworzono aplikację do optymalizacji lokalizacji zasobników.

Data wpływu do redakcji: 26.04.2017

Data akceptacji artykułu: 17.07.2017

Data publikacji online: 15.02.2019

### 1. Wprowadzenie

Zadaniem tej pracy jest zbadanie możliwości zastosowania wybranej metody optymalizacji w celu maksymalizacji bezpiecznego wyprowadzenia mocy z generacji OZE dla zadanej parametrami sieci WN. Podstawowymi ograniczeniami technicznymi są przeciążenia elementów sieci oraz niedotrzymanie napięć w dopuszczalnych granicach w węzłach sieci. W celu rozwiązania problemu maksymalizacji wyprowadzenia mocy ze źródeł OZE należy zdefiniować problem optymalizacji, wszystkie ograniczenia i wymagania. Na tej podstawie należy sformułować postać matematyczną problemu. Na podstawie analizy problemu optymalizacyjnego należy dokonać wyboru narzędzi i metod. Celem pracy nie jest wyznaczenie doboru technologii lub pojemności magazynów, lecz tylko ich wymaganych mocy zainstalowanych i ich lokalizacji. Problem lokalizacji magazynów oraz lokalizacji dodatkowej generacji ma duży wpływ na powstawanie ewentualnych przeciążeń, zmiany napięć w węzłach czy wielkość sumy mocy czynnej oddanej do sieci przez dodatkowe źródła. Z tego powodu podczas optymalizacji poszukiwane są jednocześnie lokalizacje magazynów oraz lokalizacje dodatkowych źródeł, dla których funkcja celu będzie optymalna. Model sieci elektroenergetycznej jest daną wejściową aplikacji.

### 2. Koncepcja aplikacji obliczeniowej do badania zdolności przyłączeniowych farm wiatrowych

Do realizacji budowy tego systemu zostaną wykorzystane algorytmy ewolucyjne oraz aplikacja wykonująca rozplyw mocy czynnej i bierniej badanego systemu

elektroenergetycznego. Zastosowanie algorytmów ewolucyjnych jest podyktowane trudnością wyznaczenia w postaci analitycznej funkcji celu, zwłaszcza dla modeli dużych sieci. Algorytmy ewolucyjne stanowią atrakcyjne narzędzie optymalizacji zwłaszcza dla problemów, dla których nie ma opracowanych specjalizowanych metod. Szczegóły o algorytmach genetycznych/ewolucyjnych zostały zaprezentowane w monografiach [1, 2, 3]. Funkcja celu i ograniczenia wyznaczane są na podstawie symulacji – rozplywów mocy. Klasyczne metody optymalizacji posiadają dodatkowe wymagania, np. ciągłości lub różniczkowalności funkcji celu, funkcji celu i ograniczeń w postaci analitycznej. Żeby ocenić skutki zmiany mocy czynnej lub bierniej w danym węzle, należy wykonać ponownie rozplyw mocy. Nie jest możliwe uwzględnienie ograniczeń w postaci analitycznej, ponieważ ewentualne przeciążenia elementów sieci lub przekroczenia dopuszczalnych przedziałów napięć w węzłach zależne są od samej sieci (modelu sieci) i jej stanu (wartości mocy czynnych i biernych w poszczególnych węzłach, stanów łączników oraz stanu dodatkowej generacji). Chcąc uzasadnić wybór metody optymalizacyjnej, należy przeanalizować jej wady i zalety. Zalety algorytmów genetycznych/ewolucyjnych [3]:

- Metoda jest uniwersalna, aby tego samego programu użyć w innym problemie, przeważnie wystarczy zmienić funkcję celu i ograniczenia
- Klasyczny algorytm genetyczny ma prostą konstrukcję, oprogramowanie algorytmów genetycznych w uniwersalnym języku programowania nie jest trudne
- Funkcja celu lub ograniczenia nieliniowe nie muszą być podane w postaci

analitycznej. Wystarczającą informacją jest to, że wiemy, jak ją wyznaczyć/policzyć

- Algorytmy ewolucyjne potrafią poradzić sobie również tam, gdzie optymalizowana funkcja jest zaszumiona, zmienia się w czasie, ma wiele ekstremów lokalnych. Algorytmy ewolucyjne są jedną z metod optymalizacji globalnej
- Chcąc znaleźć rozwiązanie, nie musimy prawie nic wiedzieć o optymalizowanej funkcji. Funkcji celu może nawet w ogóle nie być: możemy wykorzystywać algorytmy genetyczne nawet wtedy, gdy jedyną rzeczą, jaką potrafimy powiedzieć o punktach przestrzeni stanów, jest to, które z dwóch (kilku) rozwiązań jest lepsze (gdy zastosujemy operator selekcji turniejowej)
- Ponieważ algorytm genetyczny jest algorytmem randomizowanym, możemy powtarzać obliczenia wielokrotnie w nadziei otrzymania lepszych wyników. Poszczególne wyniki będą się różniły, ponieważ otrzymujemy rozwiązanie przybliżone.

Wady tych algorytmów [3]:

- Metoda jest uniwersalna, więc nie tak skuteczna, jak bywają algorytmy specjalizowane (w celu poprawy wydajności/dokładności rozwiązań zaleca się stosować algorytmy hybrydowe)
- Metoda jest wolniejsza od prostych heurystyk (np. metody zachłannej), choć zwykle skuteczniejsza. Często optymalizacje rzeczywistych problemów technicznych mogą wymagać dużej ilości pamięci RAM oraz trwać długo, np. kilkanaście dni
- Sukces jest możliwy wyłącznie przy prawidłowym zakodowaniu problemu i odpowiednim dobraniu funkcji celu. Niestety,

This is a supporting translation of the original text published in this issue of "Acta Energetica" on pages 33–42. When referring to the article please refer to the original text.

PL

nie ma jednoznacznej teorii mówiącej, jak to robić. Jest to często – podobnie jak dobór parametrów mutacji i krzyżowania – sprawa wyczucia i doświadczenia programisty. Wymaga to wykonania wielu wersji aplikacji i przeprowadzenia wielu badań symulacyjnych

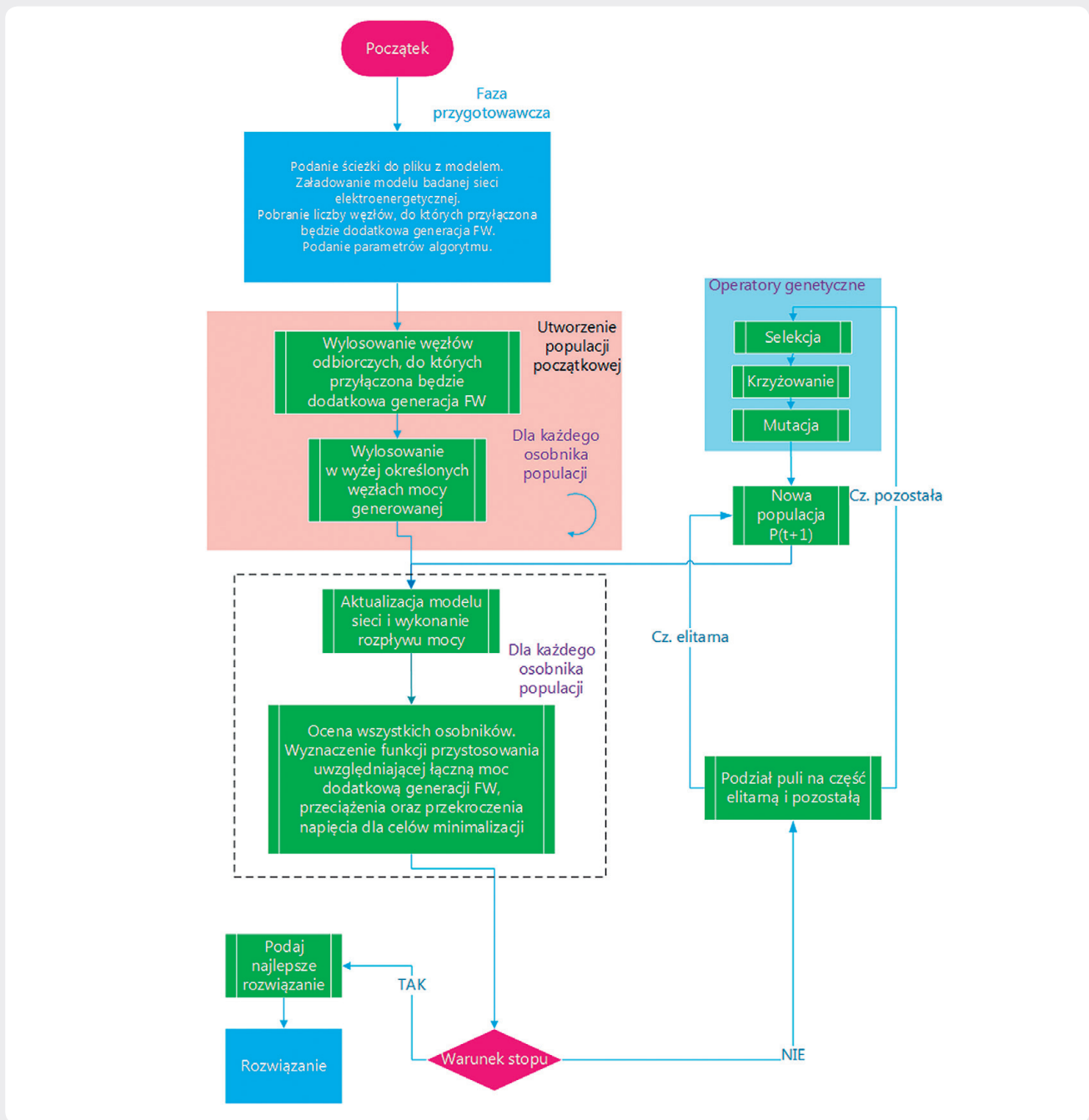
- Ponieważ algorytm genetyczny jest algorytmem randomizowanym, nigdy nie mamy pewności, że znaleźliśmy rozwiązanie optymalne (rada: zadowol się rozwiązaniem przybliżonym). W rzeczywistości, np. przy produkcji, nie jest konieczne znalezienie dokładnego optimum, lecz znalezienie rozwiązania lepszego niż konkurencja. W wielu

zastosowaniach ważny jest również czas uzyskania rozwiązania nawet kosztem jego dokładności

- Uwzględnienie nieliniowych ograniczeń może być trudne w realizacji. Może ono wymagać zaprojektowania nowych algorytmów dla operatorów genetycznych. Wówczas nie jest to metoda prosta
- Uwzględnienie wymagań specyficznych dla problemu może wymagać utworzenia nowej funkcji tworzącej populację początkową, funkcji realizujących operatory genetyczne, funkcji skalującej. Wówczas metoda ta nie jest już uniwersalna.

Algorytmy ewolucyjne nie są pozbawione wad, jednak ze względu na specyfikę

ograniczeń zostały wybrane do budowy aplikacji. W celu realizacji tego zadania zostało wykorzystane środowisko obliczeniowo-symulacyjne umożliwiające realizację analiz systemów elektroenergetycznych oraz algorytmów ewolucyjnych. Takim środowiskiem jest oprogramowanie MATLAB z zainstalowanym pakietem Global Optimization Toolbox (zawierającym biblioteki do tworzenia i symulacji algorytmów ewolucyjnych [4]) oraz aplikacje MATLABA (ang. *MATLAB – based software*), służące do analizy systemów elektroenergetycznych. Jedną z wielu dostępnych tego typu aplikacji jako przyborek (*toolbox*) MATLABA



Rys. 1. Schemat algorytmu ewolucyjnego do wyznaczenia maksymalnej łącznej dodatkowej mocy generowanej w badanej sieci bez udziału magazynu

This is a supporting translation of the original text published in this issue of "Acta Energetica" on pages 33–42. When referring to the article please refer to the original text.

PL

jest MATPOWER [5]. Stworzenie takich aplikacji (analiza systemów elektroenergetycznych, algorytmy genetyczne, metody optymalizacji) w uniwersalnym języku programowania (bez wykorzystania gotowych i przetestowanych bibliotek) jest bardzo czasochłonne. Dzięki wykorzystaniu uniwersalnego środowiska obliczeniowego budowanie nowej aplikacji może się odbywać w sposób standardowy dla aplikacji obliczeniowych, koncentrując się na rozwiązywanym problemie, wykorzystując przybory rozszerzające funkcjonalność tego środowiska. Unika się problemów pracochłonnej integracji

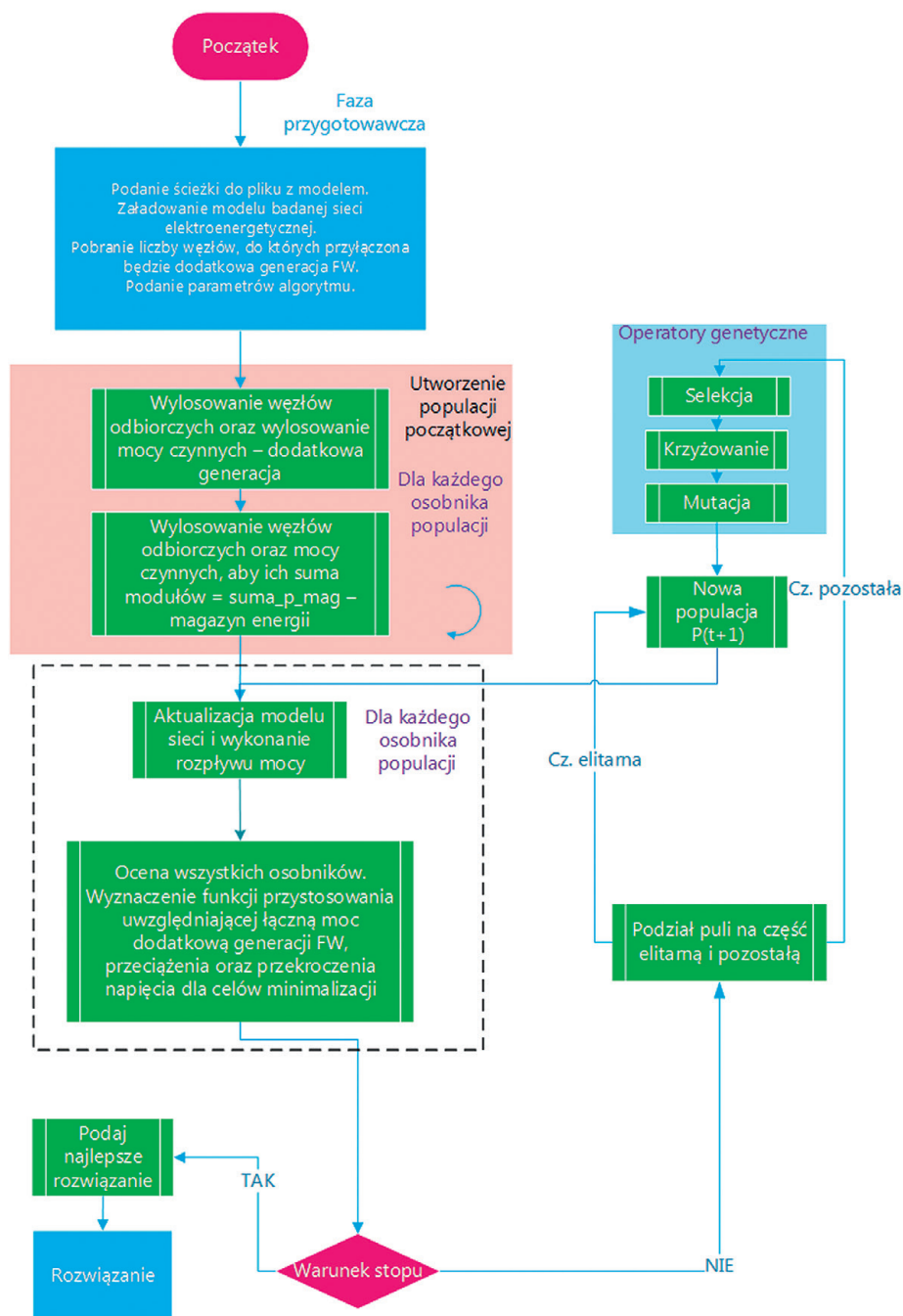
oprogramowania różnych producentów, rozwiązywania problemów numerycznych. Matematyczna postać problemu jest określona za pomocą układu ośmiu zależności:

$$X = \begin{bmatrix} p\_fw(1) \\ \dots \\ p\_fw(w) \\ p\_mag(1) \\ \dots \\ p\_mag(w) \end{bmatrix} \text{ - postać osobnika (1)}$$

$$\max \left( \sum_{i=1}^w p\_fw(i) \right) \text{ - maksymalizacja funkcji celu (2)}$$

$$\prod_{j=1}^w 0 \leq p\_fw(j) \leq FW\_MAX \text{ - ograniczenie na moc generacji (3)}$$

$$\prod_{k=1}^w -MAG\_MIN \leq p\_mag(k) \leq +MAG\_MAX \text{ - ograniczenie na moc magazynu (4)}$$



Rys. 2. Schemat algorytmu ewolucyjnego do wyznaczenia maksymalnej łącznej dodatkowej mocy generowanej w badanej sieci współpracującej z magazynami energii

This is a supporting translation of the original text published in this issue of "Acta Energetica" on pages 33–42. When referring to the article please refer to the original text.

PL

$$\sum_{m=1}^w |p\_mag(m)| = suma\_p\_mag$$

– ograniczenie równościowe

(5)

$$\prod_{l=1}^L S_{obc}(l) \leq S(l)$$

– ograniczenie na obciążalność

(6)

$$\prod_{n=1}^w V_{min}(n) \leq V(n) \leq V_{max}(n)$$

– utrzymanie wartości napięć w dopuszczalnym zakresie

(7)

$$[S_{obc}, V] = rozpyw(X, mpc)$$

– realizacja obliczeń rozpywowych

(8)

gdzie:

$X$  – postać proponowanego rozwiązania (osobnika),  $p\_fw$  – wektor mocy czynnej dodatkowej generacji [MW] o długości  $w$ ,  $FW\_MAX$  – maksymalna moc czynna magazynu przyłączona do węzła [MW],  $p\_mag$  – wektor mocy czynnej magazynu [MW] o długości  $w$ ,  $MAG\_MIN$  i  $MAG\_MAX$  – maksymalna moc

czynna magazynu przyłączonego do węzła w trybie ładowania i rozładowania [MW],  $w$  – liczba węzłów sieci,  $L$  – liczba linii sieci,  $suma\_p\_mag$  – suma modułów mocy czynnych wszystkich przyłączonych magazynów do sieci [MW],  $S_{obc}$  – wektor obciążenia (mocy pozornych) poszczególnych linii wyznaczonych po rozplywie mocy czynnych i biernych o długości  $L$ ,  $S$  – wektor obciążalności poszczególnych linii (model sieci) o długości  $L$ ,  $V$  – wektor modułów napięć w poszczególnych węzłach sieci wyznaczonych po rozplywie mocy czynnych i biernych o długości  $w$ ,  $V_{min}$ ,  $V_{max}$  – wektor minimalnych/maksymalnych dopuszczalnych napięć w poszczególnych węzłach sieci, o długości  $w$ ,  $mpc$  – model badanej sieci,  $rozpyw$  – funkcja uaktualniająca model sieci zgodnie z postacią osobnika  $X$  (moce dodatkowej generacji  $p\_fw$ , moce magazynu  $p\_mag$ ), wykonująca rozplyw mocy czynnych i biernych AC, w wyniku którego otrzymujemy obciążenia poszczególnych linii oraz wartości napięć w poszczególnych węzłach.

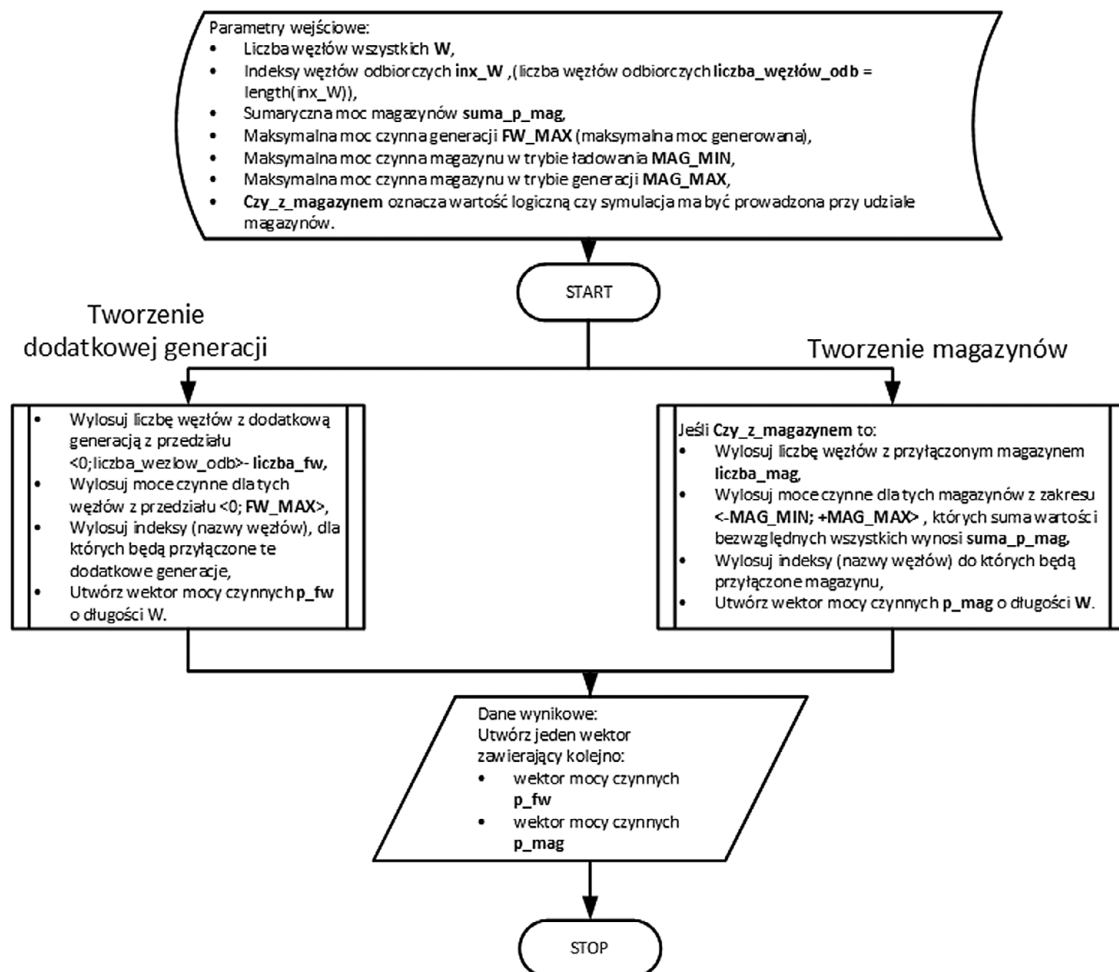
Ograniczenia optymalizacji mogą zostać uwzględnione w postaci np. funkcji kary lub nieliniowej funkcji ograniczeń. Pierwsza metoda polega na zmianie postaci funkcji celu [8]. Druga metoda oferowana przez

środowisko MATLAB polega na utworzeniu funkcji ograniczeń [4]. Ograniczenie (5) umożliwia porównanie dwóch rozwiązań o tych samych wielkościach magazynu (łącznie sumie wartości bezwzględnych mocy czynnych – niezależnie od trybu pracy ładowania, tj. wartość ujemna mocy albo rozładowania, tj. wartość dodatnia mocy).

W celu zbadania wpływu przyłączenia magazynów na możliwości przyłączenia dodatkowej generacji do sieci należy wykonać dwie symulacje.

Pierwsza symulacja jest bez udziału magazynu, jej algorytm pokazano na rys. 1. Natomiast algorytm drugiej pokazano na rys. 2. Porównanie wyników z obu symulacji umożliwi stwierdzenie, czy w danej sieci jest możliwy wzrost maksymalnej sumarycznej mocy dodatkowej generacji w sposób bezpieczny, gdy do sieci przyłączono magazyny energii.

W środowisku MATLAB zastosowano strategię elitarną. Uniemożliwia to „zapomnienie” dobrych rozwiązań znalezionych w generacjach poprzedzających ostatnią. Jednak może powodować problemy z przedwczesną zbieżnością. Na rys. 3 zaprezentowano jeden z wielu zaprezentowanych operatorów genetycznych – operator tworzenia populacji początkowej. Szczegółowy opis



Rys. 3. Schemat blokowy tworzenia populacji początkowej

This is a supporting translation of the original text published in this issue of "Acta Energetica" on pages 33–42. When referring to the article please refer to the original text.

PL

oraz symulacje zaprezentowano w [7]. Jako elektroenergetyczną sieć testową wybrano standardową sieć IEEE 30-węzłową. Wymaganiem było określenie w modelu obciążalności wszystkich elementów sieci.

### 3. Symulacje

Podczas badania skuteczności zaprojektowanych operatorów genetycznych uruchamiano algorytm z funkcją ograniczeń nieliniowych sześciokrotnie, przy określonym standardowym operatorze selekcji oraz funkcji skalującej dla różnych wersji aplikacji (wszystkie operatory, bez operatora nr 5, bez operatora nr 3, bez operatorów 3 i 5). Badania wykonano dla wszystkich możliwych kombinacji. Obok w tab. 1 i 2 przedstawiono niektóre wyniki symulacji.

Kolejnym etapem symulacji było badanie poprawności pracy algorytmu oraz jego zbieżności. Badanie polegało na 100-krotnym uruchomieniu algorytmu z nieliniową funkcją ograniczeń (rys. 4), jak i z zewnętrzną funkcją kary (rys. 5) – praca sieci tylko z magazynem.

W następnym etapie symulacji wykonano analogiczne badanie, lecz dla wersji z zewnętrzną funkcją kary, co zobrazowano na rys. 5. Innym badaniem była symulacja przy różnych prawdopodobieństwach realizacji poszczególnych operatorów genetycznych zarówno w wersji z nieliniową funkcją ograniczeń (tab. 3), jak i z zewnętrzną funkcją kary (tab. 4).

### 4. Wnioski

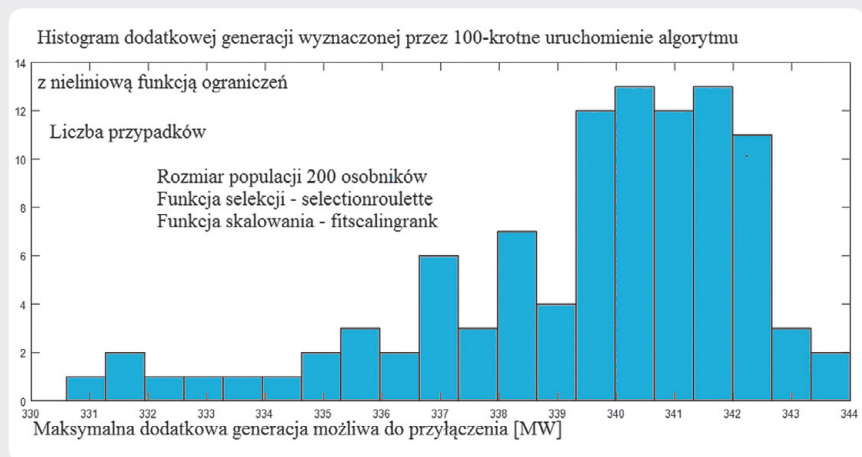
- I. W celu znalezienia optymalizacji lokalizacji magazynu energii oraz dodatkowej generacji można skutecznie wykorzystać algorytmy ewolucyjne. Wymaga to jednak zaprojektowania operatorów genetycznych oraz funkcji tworzenia populacji początkowej, uwzględniających ograniczenia nieliniowe. Przeanalizowano standardowe postacie funkcji realizujące operatory genetyczne. Nie są one w stanie poprawnie tworzyć i modyfikować osobników, tj. nienaruszających ograniczeń nieliniowych problemu. Zatem algorytmy ewolucyjne do rozwiązywania problemów technicznych, w których występują ograniczenia nieliniowe, wymagają modyfikacji uwzględniających specyfikę problemu.
- II. Spośród zbadanych wersji algorytmu ewolucyjnego najlepszy okazał się algorytm z zewnętrzną funkcją kary. Uzyskano maksymalną moc dodatkowej generacji 369,20 [MW] przy łącznej mocy magazynów równą 19 [MW] (tab. 4). Wynik wpływu pracy magazynu oceniano, porównując go z wynikiem optymalizacji samej generacji dodatkowej (bez magazynu). Dla testowanej sieci maksymalna moc dodatkowej generacji w sieci bez udziału magazynów energii wyniosła 333,9 [MW]. Maksymalna moc dodatkowej generacji nie mogła naruszać w sposób znaczący ograniczeń napięciowych i obciążeniowych.
- III. Cechą charakterystyczną metod optymalizacyjnych z zewnętrzną funkcją kary jest to, że rozwiązania mogą

Parametr wynikowy	Statystyka	Warianty algorytmu (6-krotne uruchomienie algorytmu)			
		8 operatorów (wszystkie)	bez operatora nr 5	bez operatora nr 3	bez operatorów nr 3 i 5
Moc czynna generacji z przyłączonymi magazynami $\text{suma\_p\_fw\_m}$ [MW]	max.	341,33	341,26	341,08	343,23
	min.	337,85	339,24	338,61	340,60
	średnia	340,13	340	340	342,25
	Różnica = max. – min.	3,48	2,02	2,47	2,63
Moc czynna generacji bez magazynów $\text{suma\_p\_fw}$ [MW]	max.	341,48	332,06	343,08	332,42
	min.	338,73	328,95	338,18	330,16
	średnia	339,97	330,59	341,3	331,52
	Różnica = max. – min.	2,75	3,11	4,9	2,26
Różnica maksymalnej generacji z magazynami i maksymalnej generacji bez nich $\text{suma\_p\_fw\_m} - \text{suma\_p\_fw}$ MW [MW]		-0,15	9,2	-2	10,81

Tab. 1. Wyniki sześciokrotnej iteracji w celu zbadania skuteczności operatorów genetycznych, wykorzystując funkcję selekcji *selectionroulette* oraz funkcję skalowania *fitscalingrank*

Parametr wynikowy	Statystyka	Warianty algorytmu (6-krotne uruchomienie algorytmu)			
		7 operatorów (wszystkie)	bez operatora nr 5	bez operatora nr 3	bez operatorów nr 3 i 5
Moc czynna generacji z przyłączonymi magazynami $\text{suma\_p\_fw\_m}$ [MW]	max.	339,66	341,48	341,12	342,75
	min.	336,99	334,9	337,67	338,78
	średnia	338,39	339,44	339,32	340,31
	Różnica = max. – min.	2,67	6,58	3,45	3,97
Moc czynna generacji bez magazynów $\text{suma\_p\_fw}$ [MW]	max.	341,27	332,68	341,15	332,6
	min.	338,29	329,66	337,21	329,75
	średnia	339,56	331,24	339,21	331
	Różnica = max. – min.	2,98	3,02	3,94	2,85
Różnica maksymalnej generacji z magazynami i maksymalnej generacji bez nich $\text{suma\_p\_fw\_m} - \text{suma\_p\_fw}$ MW [MW]		-1,61	8,8	-0,03	10,15

Tab. 2. Wyniki sześciokrotnej iteracji w celu zbadania skuteczności operatorów genetycznych – wykorzystując funkcję selekcji *selectionroulette* oraz funkcję skalowania *fitscalingtop*



Rys. 4. Histogram generacji dodatkowej dla wersji algorytmu z nieliniową funkcją ograniczeń

This is a supporting translation of the original text published in this issue of "Acta Energetica" on pages 33–42. When referring to the article please refer to the original text.

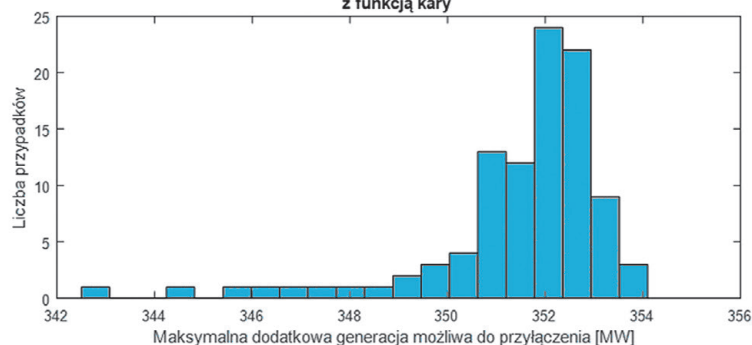
PL

Lp.	pstwo_mut	pstwo_mut_fw	pstwo_zam_fw	pstwo_zam_mag	pstwo_zam_p	f-cja celu (z magazynem)	f-cja celu (bez magazynu)
1	0,81472369	0,90579194	0,12698682	0,91337586	0,63235925	350,09	331,63
2	0,91818459	0,63255766	0,63960491	0,11794852	0,64830387	351,88	332,21
3	0,23048253	0,61061998	0,77947319	0,56706282	0,23407947	351,55	332,79
4	0,88485522	0,18345035	0,22609177	0,60591373	0,94860687	352,83	332,22
5	0,63911062	0,82985738	0,17512177	0,04191938	0,50305758	352,41	332,81
6	0,92049838	0,4880307	0,25213702	0,54685008	0,70505076	352,73	332,90
7	0,46356272	0,88610119	0,65836403	0,7455946	0,21304558	353,12	331,75
8	0,48116548	0,36855	0,53696344	0,06340336	0,30413755	351,59	333,48
9	0,45041356	0,01232514	0,09299071	0,86499249	0,31718746	351,94	332,08
10	0,7551757	0,48476413	0,48879031	0,3981366	0,20264438	352,03	332,97
11	0,61582818	0,06237602	0,26374273	0,16379723	0,11869841	350,99	332,84
12	0,9666019	0,06840756	0,73837416	0,88490846	0,38839793	350,48	333,32
13	0,39193069	0,35246639	0,0508462	0,5282447	0,33001492	352,51	333,26
14	0,14467151	0,45589683	0,41970739	0,87595398	0,90709547	351,03	333,58
15	0,137272	0,41795503	0,81303064	0,57288986	0,43202267	352,37	333,09
16	0,84251773	0,30210997	0,666352	0,1670966	0,58915655	352,71	332,68
17	0,64221802	0,766998	0,62243137	0,01869434	0,01212781	352,51	332,52

Kolorem czerwonym zaznaczono optymalne wartości prawdopodobieństw wykonania operatorów (największa wartość funkcji celu dla sieci z magazynami energii)

Tab. 3. Wyniki symulacji poszukiwania wartości parametrów operatorów genetycznych dla algorytmu z nieliniową funkcją ograniczeń

Histogram dodatkowej generacji wyznaczonej przez 100-krotne uruchomienie algorytmu ewolucyjnego z funkcją kary



Rys. 5. Histogram generacji dodatkowej dla wersji algorytmu z zewnętrzną funkcją kary

w sposób nieznaczny naruszać któreś z ograniczeń. Wielkość tych naruszeń można zmniejszyć, stosując odpowiednie duże wartości współczynników kary. W zaprezentowanym rozwiązaniu nie wystąpiły przekroczenia napięciowe, a przeciążenie dotyczyło tylko dwóch linii o wielkości nie większej niż 1,7% ich obciążalności (tab. 5). Natomiast najlepsze rozwiązanie uzyskane podczas badania wpływu pozostałych parametrów nie wykazuje żadnych przekroczeń.

IV. W wersji algorytmu ewolucyjnego z nieliniowymi ograniczeniami nie uzyskiwano rozwiązań naruszających którekolwiek ograniczenia. Jest to zaleta tego typu algorytmu.

V. W celu znalezienia optymalizacji lokalizacji magazynu energii oraz dodatkowej

generacji algorytm ewolucyjny wykorzystuje algorytm rozprawy. Służy on do stwierdzenia, czy uzyskane rozwiązanie nie powoduje przeciążenia się któregoś elementu sieci oraz czy wartości napięć we wszystkich węzłach są w dopuszczalnych zakresach. W zależności od wersji algorytmu rozprawy był wykonywany w nieliniowej funkcji ograniczeń lub w funkcji celu.

VI. Wersja algorytmu ewolucyjnego z nieliniowymi ograniczeniami okazała się nie tak skuteczna jak wersja z zewnętrzną funkcją kary. Jest to spowodowane tym, że w obu wersjach w środowisku MATLAB występują inne warunki zatrzymania algorytmu. Wersja algorytmu z nieliniowymi ograniczeniami zatrzymuje się na podstawie

numerycznych pochodnych ograniczeń. Algorytmy tego rodzaju wykonują się przez niewielką liczbę generacji. Uwidacznia się zatem problem przedwczesnej zbieżności, co dyskwalifikuje ten rodzaj algorytmu do większości zastosowań. Typowo algorytmy ewolucyjne wymagają ok. 100 generacji w celu uzyskania rozwiązania w pobliżu wartości ekstremalnej. Nasza aplikacja z algorytmem z nieliniowymi ograniczeniami realizowała tylko 6 generacji.

VII. Algorytmy ewolucyjne lub genetyczne są metodami optymalizacji znajdującymi rozwiązania w pobliżu wartości ekstremum globalnego. Nie gwarantują one znalezienia dokładnej wartości ekstremalnej. Jednak w większości zastosowań znalezienie rozwiązania z niewielkim błędem nie stanowi problemu. W rozpatrywanym problemie błąd poniżej 1 [MW] nie ma praktycznego znaczenia.

VIII. Zaprojektowany algorytm ewolucyjny jest wrażliwy na wartości prawdopodobieństw realizacji poszczególnych operatorów.

IX. Algorytmy ewolucyjne lub genetyczne są metodami randomizowanymi. Skutkuje to tym, że uzyskiwane wyniki dla tych samych danych wejściowych przy kilkukrotnym uruchomieniu algorytmu mogą się różnić. Jest to cecha wszystkich metod randomizowanych. Zaprojektowany algorytm ewolucyjny oznacza się zbieżnością, co potwierdza histogram z 100-krotnego uruchomienia algorytmu (rys. 5). Z histogramu wynika, że 46% uzyskanych wyników znajduje się w przedziale 351,8–352,9 [MW], lecz istnieją



This is a supporting translation of the original text published in this issue of "Acta Energetica" on pages 33–42. When referring to the article please refer to the original text.

PL

Lp.	pstwo_mut	pstwo_mut_fw	pstwo_zam_fw	pstwo_zam_mag	pstwo_zam_p	f-cja celu (z magazynem)	f-cja celu (bez magazynu)
1	0,821579	0,724182	0,835272	0,703187	0,851299	353,98	333,34
2	0,20824	0,788091	0,957554	0,392875	0,270881	352,87	332,88
3	0,586582	0,907114	0,71906	0,09236	0,853622	353,41	333,89
4	0,166876	0,03903	0,195391	0,089744	0,226574	353,54	333,75
5	0,711179	0,859306	0,231377	0,290049	0,24363	354,77	332,96
6	0,860269	0,056848	0,666054	0,744647	0,627422	353,42	333,39
7	0,122104	0,727066	0,700456	0,08631	0,598401	352,30	332,88
8	0,250954	0,081737	0,180487	0,295582	0,451101	353,19	333,22
9	0,553292	0,326571	0,531779	0,348433	0,021119	353,05	333,76
10	0,991492	0,288026	0,548958	0,228424	0,339291	351,75	333,36
11	0,826123	0,676308	0,193865	0,067948	0,877198	353,52	333,31
12	0,070057	0,609323	0,620526	0,30726	0,431354	369,20	333,90
13	0,833288	0,448672	0,247334	0,061462	0,62534	354,18	333,77
14	0,258424	0,820662	0,3408	0,389855	0,408813	352,98	333,29
15	0,650812	0,767099	0,379532	0,776901	0,307508	352,92	334,14
16	0,289789	0,681109	0,522761	0,012858	0,646947	353,06	333,02
17	0,774671	0,069752	0,679253	0,150766	0,201864	352,24	333,83
18	0,499141	0,708158	0,388722	0,864053	0,789496	353,23	333,37
19	0,650764	0,536774	0,7932	0,956314	0,8308	354,21	332,71

Kolorem czerwonym zaznaczono optymalne wartości prawdopodobieństw wykonania operatorów (największa wartość funkcji celu dla sieci z magazynami energii)

Tab. 4. Wyniki symulacji poszukiwania wartości parametrów operatorów genetycznych dla algorytmu z zewnętrzną funkcją kary

pojedyncze rozwiązania o znacznie mniejszych wartościach. Zatem w celu optymalizacji należy kilkakrotnie uruchomić algorytm i jako wynik uznać najlepsze uzyskane rozwiązanie.

X. Optymalizacja czasowa algorytmu oraz badanie innych czynników na skuteczność i zbieżność algorytmu ewolucyjnego wymaga dalszych

badan. Wydajność mogłoby podnieść utworzenie aplikacji w uniwersalnym języku programowania na podstawie zaprojektowanego algorytmu, co umożliwi optymalne wykorzystanie pamięci RAM i pozostałych zasobów komputera. Środowisko MATLAB jest uniwersalnym narzędziem umożliwiającym szybkie prototypowanie

algorytmów obliczeniowych/sterowania.

XI. W przypadku badania sieci elektroenergetycznej o większym rozmiarze algorytm z funkcją kary wyznaczy kilka węzłów, w których powinny być zainstalowane magazyny. Natomiast algorytm z uzmiennionymi parametrami (model sieci dla statystycznej

Nr węzła	Moc czynna [MW]	Część wektora	Nr węzła	Moc czynna [MW]	Część wektora	Nr węzła	Moc czynna [MW]	Część wektora	Nr węzła	Moc czynna [MW]	Część wektora
1	0	p_fw	16	20	p_fw	1	0	p_mag	16	-0,632	p_mag
2	0		17	10,811		2	0		17	0	
3	1,183		18	20		3	-8,625		18	-0,0043	
4	12,748		19	1,289		4	0		19	0	
5	13,152		20	10,327		5	-8,332		20	0	
6	15,641		21	9,730		6	0		21	-0,0737	
7	18,243		22	0		7	0		22	0	
8	12,758		23	0		8	0		23	0	
9	18,084		24	20		9	-1,331		24	0	
10	13,922		25	19,968		10	0		25	0	
11	20		26	20		11	0		26	0	
12	0		27	0		12	0		27	0	
13	0		28	20		13	0		28	0	
14	16,406		29	20		14	0		29	0	
15	20		30	19,716		15	0		30	0	

Tab. 5. Przykład otrzymanego rozwiązania z algorytmu ewolucyjnego z funkcją kary w trakcie badania zbieżności

This is a supporting translation of the original text published in this issue of "Acta Energetica" on pages 33–42. When referring to the article please refer to the original text.

PL

symulacji pracy generacji odnawialnej i zapotrzebowania na moc odbiorców) wyznaczy węzły, które bez względu na stan sieci skutecznie zwiększają przyłączalność generacji odnawialnej, co jest zagadnieniem optymalizacji lokalizacji magazynów w rzeczywistej sieci. Węzły te nazwiemy węzłami charakterystycznymi. Nie każda sieć elektroenergetyczna posiada takie węzły. Rozwiązaniem tego algorytmu jest wektor mocy czynnych zainstalowanych generacji odnawialnej (farm wiatrowych, elektrowni fotowoltaicznych) oraz wektor wymaganych mocy zainstalowanych magazynów energii. Oznaką znalezienia rozwiązania z węzłami charakterystycznymi będzie fakt, iż funkcja celu tego osobnika powinna być znacznie mniejsza niż innych osobników w populacji, oraz fakt, że w żadnym stanie nie wystąpiły znaczące ograniczenia. Zadaniem opracowanego systemu z uzmiennionym

modelem jest znajdowanie węzłów charakterystycznych w sieci. Wymagać to będzie nie tylko znalezienia najlepszego rozwiązania, lecz również analizy pozostałych osobników populacji.

#### Bibliografia

1. Goldberg D.E., Algorytmy genetyczne i ich zastosowania, Wydawnictwa Naukowo-Techniczne, Warszawa 1995.
2. Michalewicz Z., Algorytmy genetyczne + struktury danych = programy ewolucyjne, Wydawnictwo Naukowo-Techniczne, Warszawa 2010.
3. Abras J., Wykłady z algorytmów ewolucyjnych, Wydawnictwa Naukowo-Techniczne, Warszawa 2001.
4. Global Optimization Toolbox, User's guide. Matlab R2016a, The MathWorks, Inc. 2016.
5. Zimmerman R.D., Murillo-Sanches C.E., Matpower 5.1. User's manual, Power

System Engineering Research System, 20 marca 2015.

6. Matlab, Object – oriented programming, Matlab R2016a, The MathWorks, Inc. 2016.
7. Korpikiewicz J. i in., Opracowanie metody wyboru lokalizacji zasobników energii w sieci WN w celu zwiększenia zdolności przyłączenia energetyki odnawialnej do systemu dystrybucyjnego, praca statutowa Instytutu Energetyki Oddział Gdańsk OGS-68/16, Instytut Energetyki, Gdańsk 2016.
8. Stachurski A., Wierzbicki A., Podstawy optymalizacji, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2001.

Artykuł powstał w ramach projektu badawczego GÉKON1/02/213880/30/2015.

#### Jarosław Korpikiewicz

mgr inż

Instytut Energetyki Instytut Badawczy Oddział Gdańsk

e-mail: j.korpikiewicz@ien.gda.pl

Ukończył studia magisterskie na kierunku automatyka i robotyka na Wydziale Elektrotechniki i Automatyki Politechniki Gdańskiej (2002). Pracował w przemyśle jako automatyk oraz informatyk. Obecnie jest zatrudniony w gdańskim oddziale Instytutu Energetyki na stanowisku specjalisty. Jego zainteresowania obejmują zastosowania metod sztucznej inteligencji w elektroenergetyce, regulację napięcia w stacji elektroenergetycznej, badania i implementację logiki rozmytej w systemach wbudowanych, projektowanie regulatorów. Ponadto zajmuje się energetyką odnawialną oraz magazynami energii (Smart Grid). Doktorant w Katedrze Automatyki Okrętowej Wydziału Elektrycznego Akademii Morskiej w Gdyni. Członek Gdańskiego Oddziału Stowarzyszenia Elektryków Polskich. Członek Gdańskiego Oddziału Polskiego Towarzystwa Elektrotechniki Teoretycznej i Stosowanej.