ECCOMAS

Proceedia

THE HYBRID GLOBAL OPTIMIZATION ALGORITHM ON THE BASIS OF A FIREWORKS ALGORITHM

P. Paździor[1], M. Szczepanik[2]

[1] Silesian University of Technology, Faculty of Mechanical Engineering,
Department of Computational Mechanics and Engineering
ul. Konarskiego 18A, 44-100 Gliwice, Poland
e-mail: pawel.pazdzior@racing.polsl.pl

[2] Silesian University of Technology, Faculty of Mechanical Engineering,
Department of Computational Mechanics and Engineering
ul. Konarskiego 18A, 44-100 Gliwice, Poland
miroslaw.szczepanik@polsl.pl

Abstract

The article shows the composition of the developed hybrid optimization algorithm, as well as the method and results of testing operation. The new algorithm was inspired by the Fireworks Algorithm and the general principle of operation of Evolutionary Algorithms. The goal of developing a new computational solution was to minimize the number of necessary calls to the objective function to reach the global minimum with the desired accuracy. The corresponding cases compared the results obtained for various test functions of two or more variables.

Keywords: Fireworks algorithm, Evolutionary algorithm, Hybrid Algorithm, Optimization, Test functions.

1   INTRODUCTION

Optimization is commonly understood as finding the best solution for specific expectations. From the mathematical point of view, it is about determining the extreme of a given function with specified boundary conditions and dependencies. It does not matter if the optimized problem is related to automation, finance, construction, technology or enterprise management. After describing the problem as a mathematical function (1), it is not important what field it concerns.

$$\min_x\left[f\left(x_1, x_2, \dots, x_j\right)\right] \qquad 34\backslash* \text{ MERGEFORMAT ()}$$

Where $f(x_i)$ is a function of the variables x_1, x_2, $\dots$, x_j with constraints (in particular $<, \leq =, \geq,>$), for each variable x. [1] Based only on numerical optimization with the discretization principle, the solution is a point or a set of points in an n-dimensional space. It is also possible to use the above to maximize the function f by changing the sign of the function f.

In the past, optimization algorithms were compared by the computation time needed to solve a given test problem. However, it was necessary to standardize the results due to different computing abilities of individual computers. [2] Nowadays, due to the enormous hardware diversity and significantly greater computing power of not only workstations, but also personal computers and even mobile devices, it seems reasonable to compare the quality of algorithms by the smallest number of objective function calculations.

2   ALGORITHM

While looking for a solution (of mathematical function) depending on several factors (x), the most reliable approach to find the best solution would be to check all possible solutions and choose the best one. With a small number of possible solutions it can be done even on a sheet of paper. However, when the space of permissible states (all correct solutions) is 10 trillion combinations, checking each of them would be ineffective, and sometimes even impossible.

For this purpose, various search algorithms are used, including algorithms that may not find the best (optimal) solutions, but capable of finding a good one (near optimal). By allowing a small inaccuracy, considerable amounts of computing time and power can be saved. With large spaces of permissible states, values for less than 0,1% of all possible solutions are determined. [3]

2.1 Evolutionary and Fireworks Algorithms

One of the most popular and effective types of algorithms for solving optimization problems are evolutionary algorithms (EA). They mimic evolution, which is a common phenomenon among all living organisms. For this reason, evolutionary algorithms belong to the group of bio-inspired algorithms. The principle of operation seems quite simple, and the quality of the results obtained depend on the adopted criterion for assessing the quality of the solution (a given representative of the population). Quoting the article titled "Artificial life. Biologically inspired algorithms" the meaning of the evolutionary algorithm are explained in a following way. "The idea for evolution in a computer is simple: instead of testing a huge number of all possible solutions to a given problem, create a small population of these solutions, and then mutate them, replace them in parts (cross), evaluate and, above all, duplicate (reproduce) the best ones in the population. " [3]

In the Fireworks Algorithm (FWA), the explosion corresponds to one iteration of the algorithm in which the sparks radiate from the source. For the next iteration of the algorithm (new explosion), another explosion occurs in the place where the best spark was located,

and the situation repeats according to the flowchart shown in Fig. 1. In the beginning, the objective function should be determined - number of variables, variability amplitude and steps of discretization of each x_i , constraints and dependencies, stopping conditions of the algorithm and other control parameters. Then, using a semi-random number generator, a set of variables is drawn for each of the sparks and the objective function is calculated. The next item in the scheme is a conditional block, that checks whether any of the stopping conditions of the algorithm has been reached. If "YES", the set of design variables X from the actual best solution is the final solution. If "NO", the algorithm starts another iteration narrowing the state space around the best solution. [4]
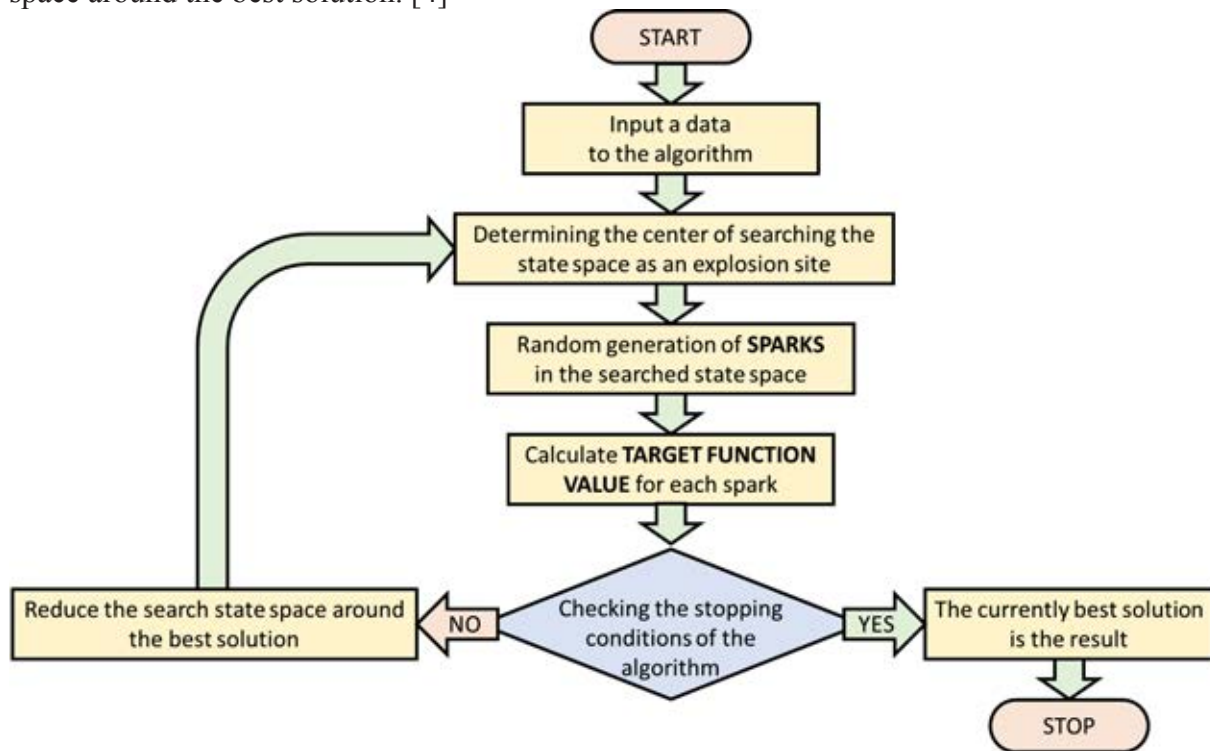

Figure 1: A simplified block diagram of the FWA.[5]

2.2 The hybrid global optimization algorithm

As a part of the doctoral dissertation, the FWA was developed for the optimization of technological processes based on the simulation of the injection process. FWA belong to a less popular type of algorithms. For this reason, in order to improve the quality (and speed) of the results obtained, it was decided to apply a completely new approach and create a hybrid algorithm, connecting FWA with EA, which is one of the most popular optimization algorithms. If the popularity is assessed by the number of searched positions in the Google search engine, even such niche algorithms as the polar bear algorithm are more popular than FWA (Table 1).

As in the theory of the origin of the universe, everything starts with a big bang [6] in which FWA generates a number of state space points (sparks) for which the objective function is calculated. It will continue to work like FWA, except for a certain probability to trigger usage of EA part. In this case, if population generated by earlier algorithm iterations is large enough, instead of generating a point in the state space by FWA, it is generated by EA. This is particularly advantageous when analyzing the space close to the optimal solution. Simplified diagram of hybrid algorithm working shown in the figure 2.

Algorithm	founded position in the Google

Genetic Algorithm	164 000 000
Evolutionary Algorithm	59 500 000
Ant Algorithm	27 000 000
Bees Algorithm	9 370 000
Particle swarm Algorithm	9 360 000
Polar bear optimization Algorithm	3 120 000
Fireworks Algorithm	887 000

Table 1: The popularity of algorithms according to the number of founded position in the Google search engine (as of January 9, 2021).
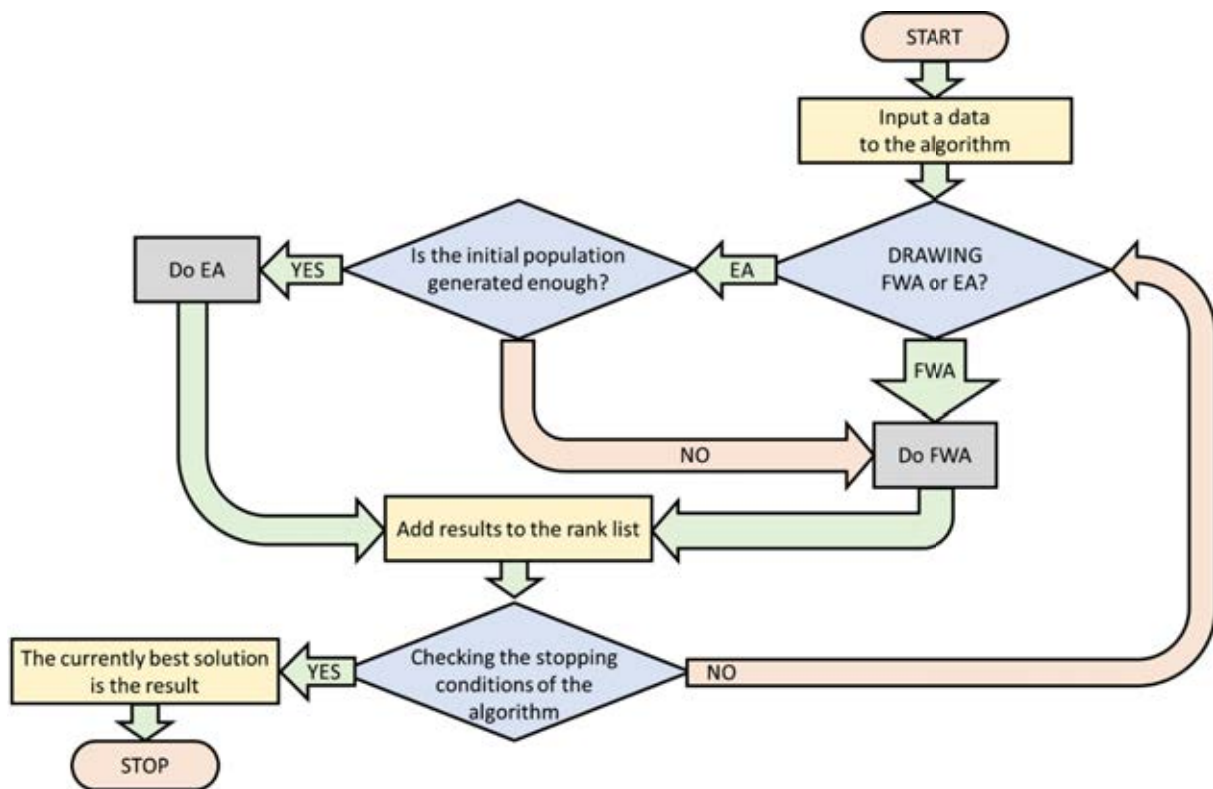

Figure 2: A simplified block diagram of the hybrid algorithm.

After each iteration of either EA or FWA a ranking list of the best solutions is updated, similar to a typical method of ranking selection. However, all individuals from the list have the same weights (there is no difference between better and worse), and the number of items on the list is a parameter that controls the work of EA. Further, similarly to the selection method ($\mu + \lambda$), parents are randomly selected [7], however, the selection is made from the current ranking list common to all past solutions since the generation of the first population.[8][9]

One individual is selected for the mutation operator, which is mutated and further treated as any other individual in the population. Individuals do not die and do not become outdated as the number of iterations of the algorithm increases. An individual ceases to be included in EA operations only when it falls out of the ranking list of the best solutions. Two individuals are always selected for crossing from the ranking list, thanks to which two offspring are always created.

Several parameters are used to control the operation of the evolution algorithm. The first one is the probability of the occurrence of a mutation M <0.1> where 0 means no mutation at all and the operation of crossing two individuals each time, and 1 each mutation of one individual without crossing. The second parameter is the number of mutations - the number of genes (variables) mutated once in an individual.

Like for the mutation, the number of genotype divisions is determined in the case of crossing, noted further by KM $\in$ <2, n / 2>. KM should be interpreted as follows: If in a given population the genotype consists of 24 genes, and KM = 2, then as a result of one crossing, 2 individuals are created with 12 genes of each parent. When KM = 3, the emerging individuals have 16 genes of one of the parents and 8 genes of the other. When KM = 4, individuals have 18 genes from one parent and 6 genes from the other parent, and so on.

The last parameter controlling the implemented evolution algorithm is the size of the ranking list, referred to as RK <2, ∞). RK = 50 means that only the best 50 solutions from the ranking list will be considered a population for the crossover and mutation operators. RK = 2 will mean that only the best 2 solutions are subject to constant mutation and crossing.

Creation of this hybrid firework-evolution optimization algorithm has been based on a semi-random probability of occurrence of an operation characteristic for FWA or EA in each iteration.

2.3 Steering parameters

The only additional parameter is a value denoted by E $\in$ <0.1>. It should be understood as the probability of the algorithm operating in a way characteristic for EA, i.e. crossing or mutation. When E = 0, then the algorithm is completely firework and no crossover and mutation operations take place. When E = 1 the algorithm is completely evolutionary and FWA is only used to complete the ranking list and thus generate an initial population, equal to RK parameter.

Table 2 presents the parameters controlling the operation of the hybrid and evolutionary algorithm. The presented data ranges do not take into account the limitations of discretization and the capacity of the data types of variables.

Symbols	Amplitude	opis
j	<1,∞)	Number of variables of the analyzed problem
E	<0,1>	Probability of using mutation or crossing instead of FWA
M	<0,1>	Probability of mutation (only when E was >0)
RK	<2,∞)	The size of the ranking list / the size of EA population
KM	<2,j/2>	The proportion of inheritance of genes by descendants
MM	<1,j-1)	The number of mutated genes per one mutation process

Table 2: Selected parameters controlling the hybrid and evolutionary algorithm..

3   TESTING FUNCTIONS

In order to check the differences in efficiency of the algorithms, they were tested with\ the use of selected test functions. The functions were selected to check the influence of the number of variables (F2 step function), state space size (F1 parabolic function), the precision of the results achieved (F3 Beale's function) and the chance of finding the global minimum in functions with many local minima (F4 Rastrigin function) as and low-variety functions (F5 Easom function).

3.1 Parabolic function

It is one of the simplest test function of j variables, having a real result for every x belonging to the set of real numbers (figure 3). It has one global and local minimum together (2).

$$F_1(x_1, x_2, \ldots, x_j) = \sum_{i=1}^{j} x_i^2$$
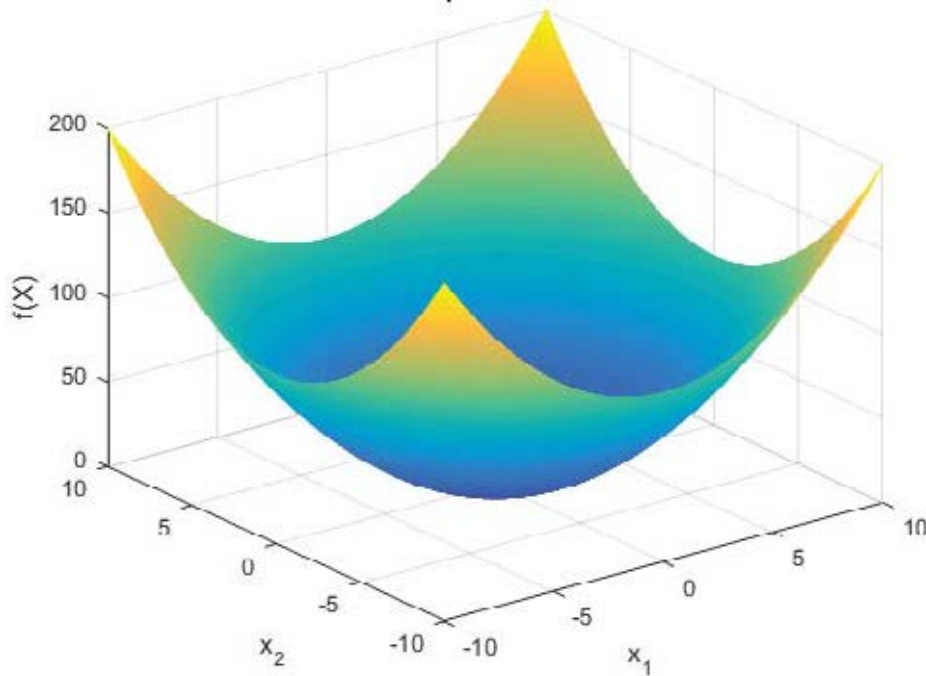$$min(F_1) = F_1(0,0,\ldots,0) = 0 \tag{2}$$

Figure 3: A plot of F1 function of two variables.

3.2 Step function

This function is perfect for testing the behavior of optimization algorithms with a large number of variables. It has a small range of variation for each x_j of j variables, and each x_j belongs to integers, but then the function may not have one global minimum. It has one global minimum only when all x_j belong to natural numbers and only such a variant of this function was considered (3).

$$F_2(x_1, x_2, \ldots, x_j) = \sum_{i=1}^{j} x_i \vee$$
$$min(F_2) = F_2(0,0,\ldots,0) = 0 \tag{3}$$
$$0 \leq x_i \leq 10$$

3.3 Beale's function

Unimodal function of two variables with one global minimum (4). It is characterized by a flat area in the middle of the function and steeply slops in areas further away from the global minimum. It allows you to check the algorithm's capabilities in terms of the precision of the results (figure 4).

$$F_3(x_1, x_2) = (1,5 - x_1 + x_1 x_2)^2 + (2,25 - x_1 + x_1 x_2{}^2)^2 + (2,625 - x_1 + x_1 x_2{}^3)^2$$

$$min(F_3) = F_3(3; 0,5) = 0 \qquad (4)$$

$$-4,5 \le x_i \le 4,5$$
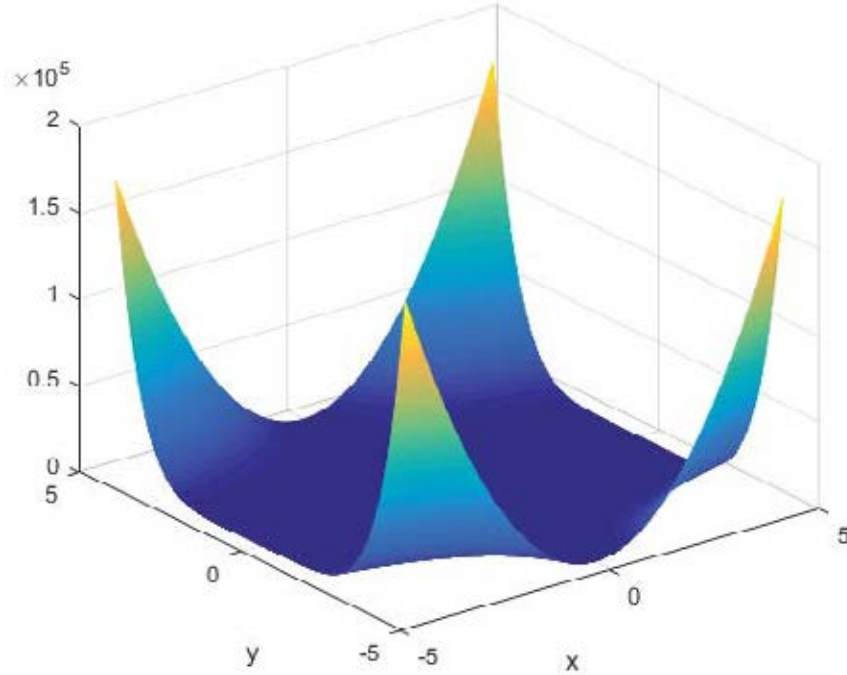
Figure 4: Plot of the F3 function for two variables.

3.4 Rastrigin function

One of the most difficult nonlinear multimodal test functions for optimization algorithms (5). It is often impossible to find the global minimum of this function due to the extremely large number of local minima (figure 5 and 6).

$$F_4(x_1, x_2, \dots, x_j) = 10n + \sum_{i=1}^{j} [x_i^2 - 10cos(2\pi x_i)]$$

$$min(F_4) = F_4(0,0, \dots, 0) = 0 \qquad (5)$$

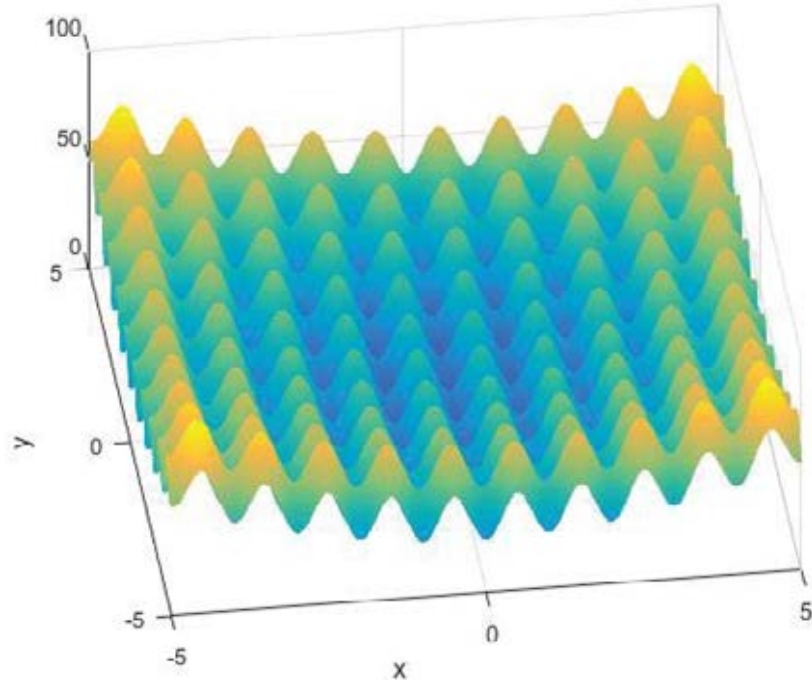$$-5.12 \le x_i \le 5.12$$

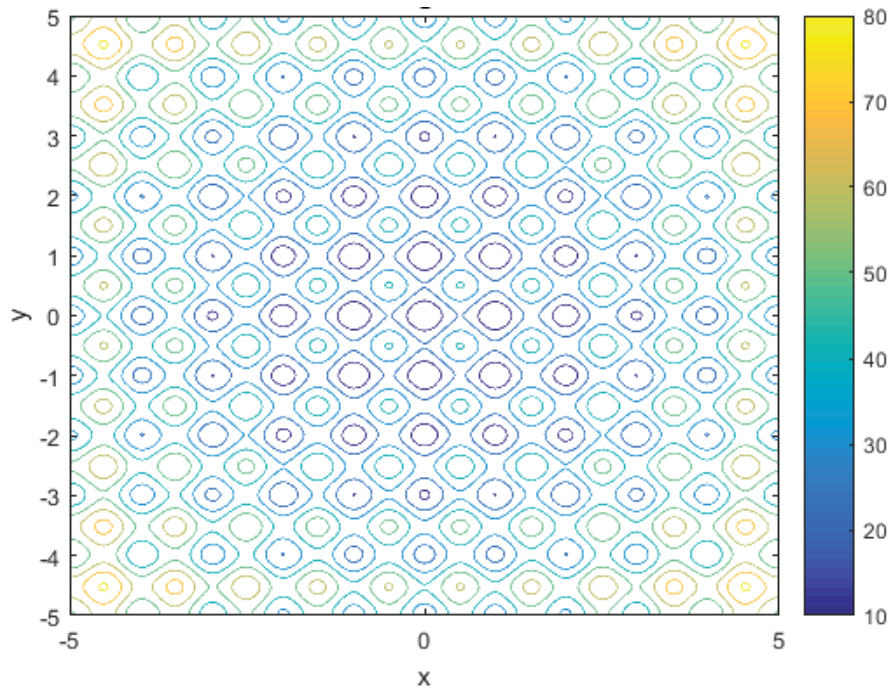Figure 5: Plot of the F4 function for two variables.

Figure 6: The contour of the function F4 for 2 variables.

3.5 Easom function

Unusual function with one funnel-shaped (figure 7 and 8) global minimum in the center of the state space and a flat part around it (6).

$$F_5(x_1, x_2) = -cos(x_1)cos(x_2)exp\big(-((x_1 - \pi)^2 + (x_x - \pi)^2)\big)$$

$$min(F_5) = F_5(\pi; \pi) = -1 \qquad (6)$$

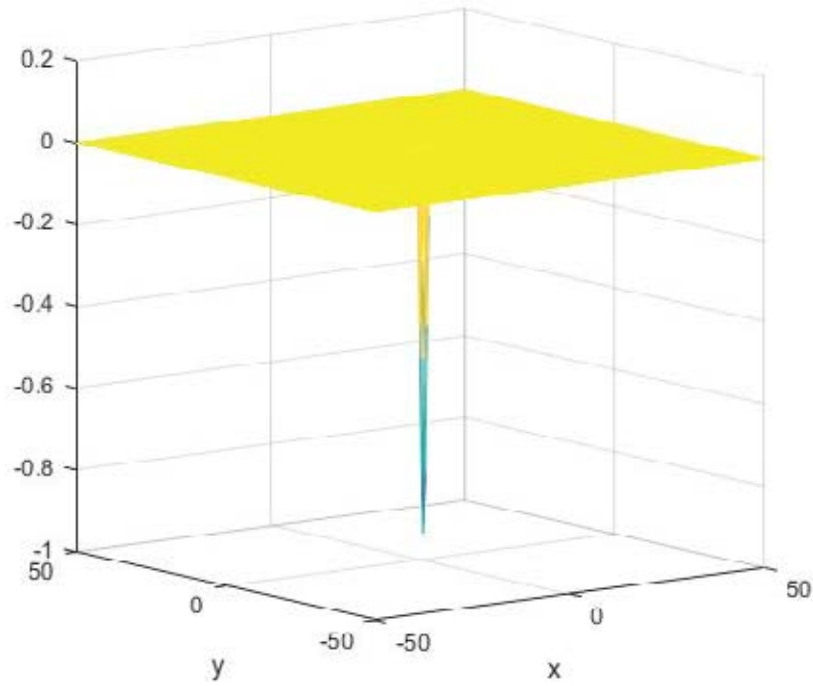$$-100 \leq x_i \leq 100$$

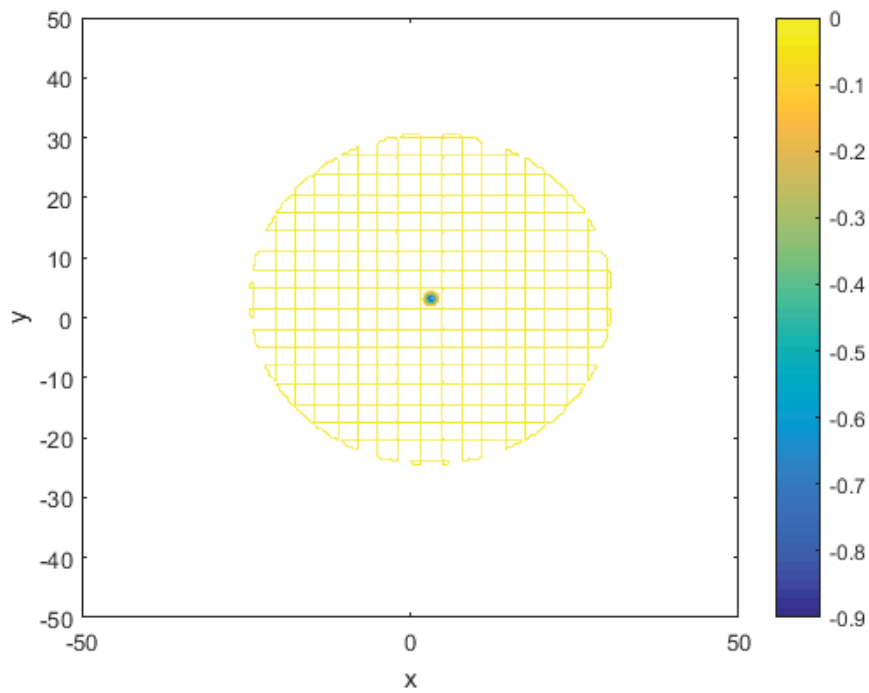Figure 7: Plot of the F5 function for two variables.

Figure 8: The contour of the function F5 for 2 variables.

4   RESULTS

The test functions of the optimization algorithms presented in the previous subsection were solved by the developed algorithms. The only indicator of the quality of the solution was the number of calls to the objective function necessary to achieve the correct result. Each case was calculated 10 times. The value used for comparison was the arithmetic mean of the indicators, which was presented in the table 3.

EA	FWA	Hybrid
	F1 2 var. ; $-1000 \leq x_i \leq 1000$	
4976	*77,8*	157
	F1 5 var.; $-100 \leq x_i \leq 100$	
5706	253	374
	F2 100 var.	
12801	*13560*	*10370*
	F2 1000 var.	
13228	*13161*	*8184*
	F3 5 var.	
1323	184	156
	F4 2 var.	
345	144	140
	F4 5 var.	
5454	3363	3286
	F5 1000 var.	
9990	3898	951

Table 3: Selected results of objective function calling for evolutionary and fireworks and hybrid algorithm.

5   CONCLUSIONS

Above all it should be emphasized that the results achieved by individual algorithms strictly depend on the programmer, the way of implementing the known scheme of activities in computational technology. This means that another algorithm of the same type may produce different results, but the trend should be saved.

In a common evolutionary algorithm, a single mutation can be a milestone in the search for the optimum and slowly reaching the best result thanks to the crossing operation turns out to be effective. The implemented FWA fits fantastically in the parabolic function, where, due to the narrowing of the search space with successive iterations, it easily tends to the optimum. The use of a hybrid algorithm in this case is slightly behind the pure FWA. However, algorithms are rarely used to search for solutions to quadratic functions, and the situations of their application usually are much more complicated. In the case of more difficult test functions, such as the Easom or Rastrigin functions, the use of the hybrid algorithm gives spectacular results.

The use of EA has a significant advantage over the others in the form of a small number of parameters controlling the algorithm. In the case of FWA, it was almost necessary to perform preliminary calculations in order to select the appropriate parameters of the algorithm. The use of hybrid algorithm made this task even more difficult, because in order to achieve the best results, it was necessary to select more than 15 parameters, not including the number, ranges of variability and the degree of discretization of variables, etc.

The works presented in this article are part of the doctoral dissertation and are further developed for this purpose. The optimization algorithm developed by the authors was used

to optimize the technological processes of plastic injection and the production of composite elements by the RTM method. The positive results of using the algorithm in practice are a solid basis for continuing work and further development of the algorithm.

6   ACKNOWLEDGEMENT

REFERENCES

[1] W. Findeisen, J. Szymanowski i A. Wierzbicki, *Computational Optimization Methods*. Warszawa, Warsaw University of Technology Publishing Houses, 1972, pp. 7.

[2] J. Goliński, Optimization methods in technical design. Warszawa, WNT, 1974, pp. 201-202.

[3] M. Komosiński, Artificial life. Biologically inspired algorithms. *SCIENCE*, n 4, pp. 7-21, 2008.

[4] P. Paździor, M. Szczepanik, Development of global optimization fireworks algorithm. *Student Scientific Conference „Computer Methods – 2020"*, 2020, Gliwice.

[5] K. S. Kishore, C. N. Sarat i S. B. Himansu, Forecasting currency exchange rate time series with fireworks algorithm-based higher order neural network, with special attention to training data enrichment. *Computer Science*, vol. 21, n 4, pp. 466-468, 2020.

[6] E. Howell, What Is the Big Bang Theory?. *SPACE*, 07 11 2017.

[7] K. Murawski, Evolutionary calculations - genesis and application. *Bulletin of the Institute of Automation and Robotics WAT*, 15, 2001.

[8] T. Back and H.P. Schwefel, Evolutionary algorithms: Some very old strategies for optimization and adaptation. *New Computing Techniques in Physics Research II*, Perret-Gallix, D., editor (World Scientific, Singapore, 1992).

[9] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGAII. *Evolutionary Computation, IEEE Transactions* on 6(2), 182–197 Apr (2002