УДК 004.89
DOI: https://doi.org/10.17721/AIT.2021.03

Maryna Antonevych, orcid.org/0000-0003-3640-7630,
Anna Didyk,
Nataliia Tmienova, orcid.org/0000-0003-1088-9547,
Vitaliy Snytyuk, orcid.org/0000-0002-9954-8767,
Taras Shevchenko National University of Kyiv, Ukraine

Choosing the best parameters for method of deformed stars in n-dimensional space

This paper is devoted to the problem of optimization of a function in n-dimensional space, which, in general case, is polyextreme and undifferentiated. The new method of deformed stars in n-dimensional space was proposed. It is built on the ideas and principles of the evolutionary paradigm. Method of deformed stars is based on the assumption of using potential solutions groups. There by it allows to increase the rate of the accuracy and the convergence of the achieved result. Populations of potential solutions are used to optimize the multivariable function. In contrast to the classical method of deformed stars, we obtained a method that solves problems in n-dimensional space, where the population of solutions consists of 3-, 4-, and 5-point groups. The advantages of the developed method over genetic algorithm, differential evolution and evolutionary strategy as the most typical evolutionary algorithms are shown. Also, experiments were performed to investigate the best configuration of method of deformed stars parameters.

Keywords: *enterprise, technology, optimization, method, experiment.*

For citation: M. Antonevych, A. Didyk, N. Tmienova, V. Snytyuk, “Choosing the best parameters for method of deformed stars in n-dimensional space” *Advanced Information Technology*, vol. 1, p. 24–28, 2021.

INTRODUCTION

The large number of modern practical problems belong to the class of constraint satisfaction problems (CSPs). Such algorithms, as stochastic search, combinatorial optimization methods, and evolutionary algorithms are used to solve these tasks.

In artificial intelligence, the evolutionary algorithm (or EA) is a subset of evolutionary computations [1]. It is general population-based metaheuristic optimization algorithm. All evolutionary algorithms have the basic provisions in the theory of biological evolution - the processes of selection, mutation and reproduction. The environment determines the behavior of elements in population. Such a population evolves in accordance with the selection rules in accordance with the objective function set by the domain.

Thus, each individual in the population is assigned a value for its suitability in the do-main. Only the most suitable species produce offspring. Various types of mutation allow agents to change and adapt to the domain. Thus, such algorithms are classified as adaptive search engines.

PRACTICAL USE

As it is known, evolutionary algorithms are used for problems such as functional optimization and can be easily described with mathematical formulas. EA are used in combinatorial optimization, particularly in solving classical NP-complete problems such as the traveling salesman problem (TSP) [2], number splitting, knapsack packing problem [3], maximally independent set, and graph sketching [4].

Also, other non-classical problems for which EA are used include transportation problems, route calculation, scheduling, planning, location, etc. Evolutionary algorithms are also used to optimize structures and electronic circuits in economics and medicine. In addition, evolutionary algorithms can be used in music and they are being actively explored in Austria, essentially when trying to simulate playing musical instruments by popular people from different years [5].

REVIEW OF EVOLUTIONARY METHODS FOR FUNCTION OPTIMIZATION

As it was said earlier, evolutionary algorithms can be used to solve optimization problems. The most famous algorithms are genetic algorithm (GA), evolutionary strategy (ES), evolutionary programming (EP), genetic programming (GP), etc.

Evolutionary programming was introduced by Lawrence J. Vogel [6]. Like ES and GA, EP has a practical optimization technique when other techniques such as gradient descent (GD) or direct analytic discovery (DAD) are not plausible.

Combinatorial optimization and optimization of functions with real values, in which the optimization surface or fitness landscape is "hard to pass", with many locally optimal solutions, are well suited for evolutionary programming.

Classical EPs have been developed to evolve finite-state automata in such a way that they were capable of solving time series prediction tasks and more generally evolving learning machines [7]. Modern EPs have been applied to solve continuous optimization problems using real-valued representtations [7]. For example, to find the shortest path in a TSP, each solution would be a path. The length of the path (as example, from location a to location b) can be expressed as a number, which in turn would serve as the fitness criteria for the solution [8].

John H. Holland created the genetic algorithm [9]. A GA is a method for solving problems of both constrained and unconstrained optimization, based on the process of natural selection, which mimics biological evolution. GA can be applied to solve problems that are not well suited for standard optimization algorithms, including problems in which the objective function is non-differentiable, stochastic, discontinuous or highly nonlinear.

Ingo Rechenberg and Hans-Paul Schwefel developed an evolutionary strategy approach [10-11]. Evolutionary strategy is a stochastic global optimization algorithm. Also, ESs can be applied in all fields of optimization including continuous, discrete, combinatorial search spaces without and with constraints as well as mixed search spaces. The ESs can also be applied to a set of objective functions in context of multi-objective optimization.

In the early nineties, a fourth stream appeared - genetic programming. Genetic programming has been successfully used as a tool for automatic programming, automatic problem solving and machine learning. This algorithm is especially useful in areas where the exact shape of a solution is not known in advance or an approximate solution is acceptable (perhaps because it is very difficult to find an exact solution).

Some of the GP applications include symbolic regression, curve fitting, data modeling, classification, feature selection, etc. John R. Koza mentions 76 cases where genetic programming could produce results that compete with results created by humans (the so-called human-competitive results).

Since the 1990s, evolutionary computing has largely become associated with the idea of swarm intelligence [12], and nature-inspired algorithms have become an increasingly significant part of this trend. Swarm intelligence (SI) is concerned with the design of intelligent multi-agent systems by taking inspiration from the collective behavior of social insects as well as from other animal societies [13].

Swarm intelligence methods have been very successful in the area of optimization, which is of great importance for industry and science [13]. Optimization problems are of high importance both for the industrial world as well as for the scientific world.

Examples of practical optimization problems: train scheduling, timetabling, shape optimization, problems from computational biology and telecommunication network design. The research community has simplified many of these problems in order to obtain scientific test cases such as the well-known TSP [13].

PROBLEM FORMULATION

Let’s consider the problem of finding the minimum value of the multivariable function:

$$\begin{aligned} & f(x_1, x_2, \ldots, x_n) \to \min, \\ & \bar{x} = (x_1, x_2, \ldots, x_n), \\ & \bar{x} \in D \subset R^n, \end{aligned} \tag{1}$$

where n is the dimension of the search space, D is some area.

The problem (1) is equivalent to the search problem:

$$\begin{aligned} & Arg \min f(\bar{x}), \\ & x \in D. \end{aligned} \tag{2}$$

In addition, let some unknown point $x^* = (x_1^*, x_2^*, \ldots, x_n^*) \in D$ will be its solution.

Since the problem has an optimization character, the No Free Lunch Theorem (NFL) is applicable to it [14]. The meaning of this theorem is that there is no single method that can find the best solution to any optimization problem.

Thus, the use of evolutionary algorithms does not require strict target functions constraints and does not guarantee the finding of a global optimum. That is why both the enlargement of optimization methods and the research of their effectiveness continue.

Note that the function f can be undifferentiated, polyextreme, besides it can be given tabularly or algorithmically. Therefore, in such cases, the methods of classical optimization are ineffective. Hypothesis is that there is a motion direction of the potential solution of the problems (1), (2), which will allow with high probability and quickly finding the optimal solution of the problem.

Moreover, there is an area in neighborhood of potential solution to be investigated in more detail because it has solutions that are better than the parent's potential solution [15].

We propose to use the idea that underlies all evolutionary algorithms, namely the population approach, which consists of solving problems (1), (2) by evolution the population of potential solutions [16].

The key elements of the solution search for the problems (1), (2) are special transformations like compressions, and rotations of figures formed by the connection of points, which are the potential solutions.

In addition, these groups of points are called deformed stars. For this reason, it is necessary to use a larger number of points grouped in a certain way and to make a directional movement towards finding a potential best solution with a detailed study of the surrounding area.

That is an idea underlying the method of deformed stars [17]. In this paper, the method is further developed by extending it into groups of 3, 4, and 5 potential solutions (3-, 4- and 5-point stars) [18].

THE PROPOSED ALGORITHM

Step 1. We initialize the parameters of the algorithm, let $t = 0$.

Step 2. Generate m potential solutions in domain D (population P_t).

Step 3. Form w figures (triangles, rectangles or pentagons) $F_i, i = \overline{1, w}$.

Step 4. For each F_i find the vertex in which the function f takes the best value and consider it the best vertex.

Step 5. For each figure F_i find a compressed figure T_i in which the best point is transferred along the line connecting the center of the figure and the best point, and all the others are transferred to it (Figure 1).

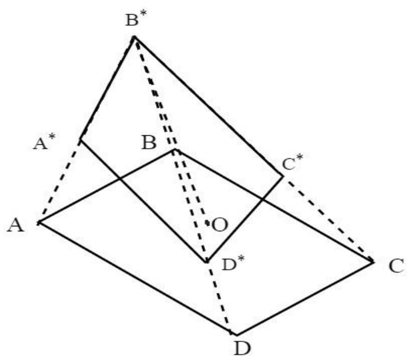

Fig. 1. Figure transformations

Step 6. For each figure F_i find a compressed figure U_i, in which all points are compressed to the best vertex.

Step 7. For each figure F_i find the figure Q_i, obtained by rotating F_i around the best vertex (Figure 2).

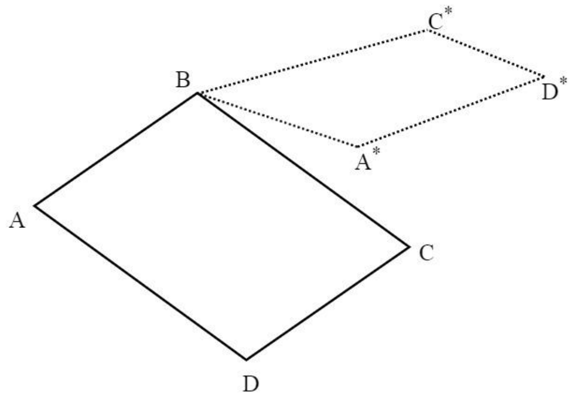

Fig. 2. Figure transformations

Step 8. For each figure F_i find the figure B_i, obtained by rotating F_i around the center of the figure.

Step 9. For each figure F_i find a modified figure R_i.

Step 10. Form a general population P_t, which will contain all the new points created in the previous steps. Thus, $P_t = P_t + T_i + U_i + Q_i + B_i + R_i, i = \overline{1, w}$.

Step 11. For all potential solutions from P_t find the value of the function f and sort the potential solutions from the best to the worst.

Step 12. Leave in P_t only m best solutions and check the fulfillment of the stop criterion.

Step 13. If the stop condition is not met, go to step 3. Otherwise, complete the algorithm and the best element in the population will be considered the best solution. Consider that the stop criterion may be:

- a given number of iterations;
- the average value of fitness function in neighboring populations is less than specified, etc;
- the worst value of fitness function in neighboring populations is less than specified.

THE EXPERIMENTAL RESULTS

Figure 3 shows the experimental results for the Schwefel 2.20 function in 10-dimensional space. The comparative graph of results is presented in Figure 4.

It shows how well-known methods and method of deformed stars (MODS) found a solution during the execution with the stop condition by iterations. As we can see, MODS found a solution on the initial iterations, in contrast to other known methods.

Test function	Global minimum	1000 iterations, 10 launches, accuracy = 10^{-5}					
		GA	ES	DE	MODS-3	MODS-4	MODS-5
	f	f	f	f	f	f	f
Schwefel 2.20	0	0.143	194.1793	79.7148	0.0	0.0	0.0
Worst fitness-function between populations, epsilon = $1*10^{-10}$							
Schwefel 2.20	0	8.4094	191.1157	283.9126	0.0	0.0	0.0
Average fitness-function between populations, epsilon = $1*10^{-10}$							
Schwefel 2.20	0	3.6056	244.8562	170.2587	0.0	0.0	0.0

Fig. 3. Comparison of the results of methods under three conditions of completion

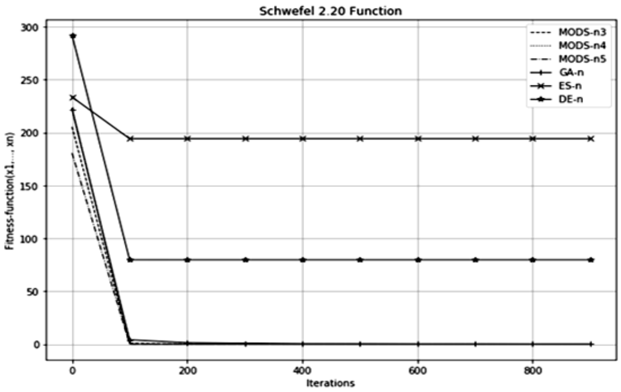

Fig. 4. Convergence plots for the Schwefel 2.20 function, the condition by iterations, coefficient = 2

CHOOOSING THE BEST PARAMETERS FOR MODS

As we can see, first experiments were made with the parameters, which are used to perform transformations in MODS populations, equal to 2, for all stop conditions. Moreover, MODS with these parameters was used for comparison with well-known methods in Figure 3.

To investigate the best configuration of the MODS, it was decided to conduct testing, considering the different values of the input parameters of the method. For the study, it was decided to set the values of all parameters, which are used to perform transformations in MODS populations, equal to 1.5.

After conducting experiments, it was found that even with the change of parameters, MODS was able to find the correct extremes of the tested functions. Thus, we can say that when changing the parameters of the method, the solution will still be found. However, it is important to note, that for coefficients = 1.5 the method worked faster. Therefore, the change in the value of the parameter affected the speed of finding the result. The comparative graph of results is presented in Figure 5. It shows how well-known methods and MODS (but, in this case, with parameters, that are equal to 1.5) found a solution with the execution of the stop condition by iterations.

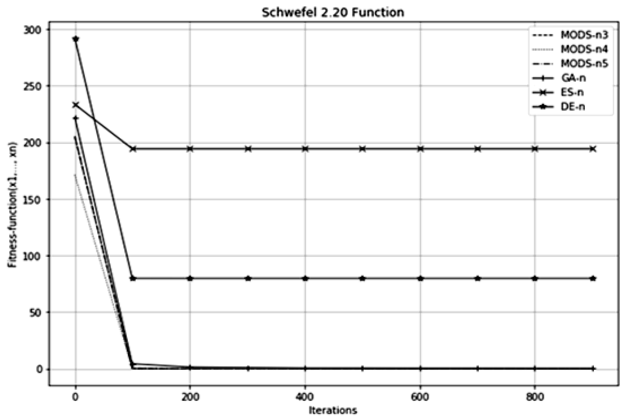

Fig. 5. Convergence plots for the Schwefel 2.20 function, the condition by iterations, coefficients = 1.5

CONCLUSION

The large number of modern practical problems belong to the class of constraint satisfaction problems (CSPs). Such algorithms, as stochastic search, combinatorial optimization methods, and evolutionary algorithms are used to solve these tasks.

We propose to use the idea that underlies all evolutionary algorithms, namely the population approach, which consists of solving problems (1), (2) by evolution the population of potential solutions. The algorithm is called method of deformed stars.

The key elements of the solution search for the problems (1), (2) are special transformations like compressions, and rotations of figures formed by the connection of points, which are the potential solutions. In addition, these groups of points are called deformed stars. MODS belongs to the class of evolutionary algorithms and makes it possible to take into account the surface of the function under study. Its advantages are the speed of convergence and the accuracy of the result compared to other evolutionary methods.

The obtained experimental results allow us to conclude that the proposed method is applicable to solve problems of finding optimal (suboptimal) values, including undifferentiated functions. In implementing the method of deformed stars, significantly fewer steps performed in the wrong direction, in contrast to genetic algorithm, the method of differential evolution and evolutionary strategy as representatives of classic evolutionary paradigm. The accuracy of the obtained solutions is, on average, higher than that of competing algorithms due to a deeper study of the solution search area.

It is also important to note that in the course of research it was found that when changing the parameters of the MODS solution will still be found.

REFERENCES

[1] Vikhar, P. A., "Evolutionary algorithms: A critical review and its future prospects", in *Proc.* 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), 2016, pp. 261–265.

[2] Arthur E. Carter, Cliff T. Ragsdale, "A new approach to solving the multiple traveling salesperson problem using genetic algorithms", *European Journal of Operational Research,Volume 175,* Issue 1, pp. 246–257, 2006.

[3] Kierkosz, I., Luczak, M. "A hybrid evolutionary algorithm for the two-dimensional packing problem". *Cent Eur J Oper Res 22,* 2014, pp. 729–753.

[4] Olariu Stephan, Zomaya Albert Y, *Handbook of Bio-inspired Algorithms and Applications,* Chapman Hall/Crc Computer Information Science, 2006.

[5] Madsen, S. T. and Widmer, G., "Evolutionary Search for Musical Parallelism", Applications of Evolutionary Computing, in *Proc.* EvoWorkshops, 2005, LNCS 3449 p., pp. 488–497.

[6] Lawrence J. Fogel, "Evolutionary programming", Alexandria, Virginia, U.S.: *National Science Foundation (NSF),* 1960.

[7] Zaineb Chelly Dagdia, Miroslav Mirchev, Chapter 15 – When Evolutionary Computing Meets Astro- and Geoinformatics, Knowledge Discovery in Big Data from Astronomy and Earth Observation, Elsevier, 2020, pp. 283–306.

[8] R.G.S. ASTHANA, CHAPTER 6 – Evolutionary Algorithms and Neural Networks, In Academic Press Series in Engineering, Soft Computing and Intelligent Systems, Academic Press, 2000, pp. 111–136.

[9] John Henry Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor: University of Michigan Press, 1975.

[10] Rechenberg, Ingo, "Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution", PhD thesis, Frommann-Holzboog, 1973.

[11] Hans-Paul Schwefel, "Numerische Optimierung von Computer-Modellen", PhD thesis, 1974.

[12] Beni, Gerardo, and Jing Wang. "Swarm intelligence in cellular robotic systems." *Robots and biological systems: towards a new bionics?*, Springer, Berlin, Heidelberg, 1993. pp. 703–712.

[13] Blum, Christian & Li, Xiaodong, *Swarm Intelligence in Optimization,* 2008.

[14] D.H. Wolpert, W.G. Macready, "No Free Lunch Theorems for Optimization", in *Proc.* IEEE Transactions on Evolutionary Computation 1, 1997, p.67.

[15] Tmienova N., Snytyuk V, "Method of Deformed Stars for Global Optimization". in *Proc.* 2020 IEEE 2nd International Conference on System Analysis & Intelligent Computing (SAIC), 2020, pp. 1–4.

[16] M. Antonevych, A. Didyk, V. Snytyuk, "Optimization of Functions of Two Variables by Deformed Stars Method", in *Proc.* 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT), 2019, pp. 475–480.

[17] V. Snytyuk, "Method of Deformed Stars for Multi-extremal Optimization. One- and Two-Dimensional Cases", in *Proc.* International Conf. Mathematical Modeling and Simulation of Systems, Advances in Intelligent Systems and Computing, Springer, Cham, 2019, vol 1019.

[18] M. Antonevych, A. Didyk, V. Snytyuk, "Choice of better parameters for method of deformed stars in n-dimensional case", in *Proc.* IT&I, 2020, pp. 17–20.

Стаття надійшла до редколегії 10.09.2021

Вибір найкращих параметрів для методу деформованих зірок у n-вимірному просторі

Досліджено задачу оптимізації функції в n-*вимірному просторі, яка, у загальному випадку, є поліекстремальною і недиференційованою. Запропоновано новий метод деформованих зірок у* n-*вимірному просторі. Указаний метод побудований на ідеях і принципах еволюційної парадигми і заснований на припущенні використання потенційних груп розв'язків. Це дозволяє підвищити швидкість точності та збіжності досягнутого результату. Для оптимізації функції від багатьох змінних використовуються популяції потенційних рішень. Отримано метод, який застосовують для розв'язання задач в* n-*вимірному просторі, де сукупність розв’язків складається з 3-, 4- та 5-точкових груп. Показано переваги розробленого методу перед генетичним алгоритмом, диференціальною еволюцією та еволюційною стратегією, як найбільш типовими еволюційними алгоритмами. Також проведено експерименти для дослідження найкращої конфігурації параметрів методу деформованих зірок.*

Ключові слова: *технологія, оптимізація, метод, експеримент.*

Марина Антоневич. Студентка факультету інформаційних технологій Київського національного університету імені Тараса Шевченка. Київ, Україна. Сфера наукових інтересів: наука про дані, еволюційні алгоритми, машинне навчання, глибоке навчання, комп’ютерний зір, інтелектуальний аналіз даних.

Maryna Antonevych. Student Faculty of Information Technology, Taras Shevchenko National University of Kyiv. Kyiv, Ukraine.
Scientific interests: data science, evolutionary algorithms, machine learning, deep learning, computer vision, data mining.

Наталія Тмєнова. Кандидат фізико-математичних наук, доцент, заступник декана з навчально-виховної роботи. Вища освіта: Київський національний університет імені Тараса Шевченка, магістр соціальної інформатики, викладач математики та інформатики. Київ, Україна. Кандидатська дисертація: 01.05.04 — Системний аналіз і теорія оптимальних рішень, «Граничні теореми для керованих систем». Сфера наукових інтересів: обробка природної мови, технології електронного навчання, дослідження систем масового обслуговування, дослідження випадкових процесів.

Nataliia Tmienova. Candidate of Physical and Mathematical Sciences (PhD), Associate Professor, Deputy Dean of Academic Studies.
Higher Education: Taras Shevchenko National University of Kyiv, Master in Social Informatics, Teacher of Mathematics and Informatics. Kyiv, Ukraine.
Defended Thesis: PhD: 01.05.04 — System analysis and theory of optimal decisions, “The limit theorems for controlled systems”.
Scientific interests: natural language processing, e-learning technologies, queuing systems research, random processes research.

Анна Дідик. Студентка факультету інформаційних технологій Київського національного університету імені Тараса Шевченка. Київ, Україна. Сфера наукових інтересів: наука про дані, еволюційні алгоритми, веброзробка, вебдизайн, машинне навчання, глибоке навчання.

Anna Didyk. Student Faculty of Information Technology, Taras Shevchenko National University of Kyiv. Kyiv, Ukraine.
Scientific interests: data science, evolutionary algorithms, web-development, web-design, machine learning, deep learning.

Віталій Снитюк. Доктор технічних наук, професор, декан факультету інформаційних технологій.
Вища освіта: Київський національний університет імені Тараса Шевченка. Київ, Україна. Математик, викладач.
Кандидатська дисертація: 05.13.12 – Системи автоматизації проєктних робіт, «Методи зменшення невизначеності на початкових етапах проєктування систем із змінною структурою».
Докторська дисертація: 05.13.06 – Інформаційні технології, «Еволюційні технології прийняття рішень в умовах невизначеності». Сфера наукових інтересів: прийняття рішень в умовах невизначеності, обчислювальний інтелект, еволюційне моделювання.

Vitaliy Snytyuk. Doctor of Technical Science, Professor, Dean of the Faculty of Information Technology.
Higher education: Taras Shevchenko National University of Kyiv. Kyiv, Ukraine. Mathematician, Lecturer.
Defended Theses: PhD 05.13.12 – Systems of automation of design works, “Methods to reduce uncertainty on the initial stages of systems design with variable structure”. Doctoral 05.13.06 – Information technology, “Evolutionary technologies of decision making under uncertainty”.
Scientific interests: decision making under uncertainty, computational intelligence, evolutionary modelling.