Article

An Innovative Hybrid Approach to Global Optimization

Vasileios Charilogis[1],Glykeria Kyrou[2], IoannisG. Tsoulos[3,*], Anna Maria Gianni[4]

[1] Department of Informatics and Telecommunications, University of Ioannina, Greece; v.charilog@uoi.gr
[2] Department of Informatics and Telecommunications, University of Ioannina, Greece; g.kyrou@uoi.gr
[3] Department of Informatics and Telecommunications, University of Ioannina, Greece; itsoulos@uoi.gr
[4] Department of Informatics and Telecommunications, University of Ioannina, Greece; am.gianni@uoi.gr
* Correspondence: itsoulos@uoi.gr;

Abstract: Global optimization is critical in engineering, computer science, and various industrial applications, as it aims to find optimal solutions for complex problems. The development of efficient algorithms has emerged from the need for optimization, with each algorithm offering specific advantages and disadvantages. An effective approach to solving complex problems is the hybrid method, which combines different algorithms. This paper presents a hybrid global optimization method, calculating optimal solutions iteratively through vector operations. These operations are based on samples derived either from internal line searches or genetically modified samples in specific subsets of Euclidean space. Additionally, other relevant approaches are explored to enhance the method's efficiency.

Keywords: Optimization, Differential evolution, Genetic algorithm methods, Line search, Kmeans distribution,Evolutionary techniques, Stochastic methods, Hybrid framework.

Citation: Charilogis V.; Kyrou, G.;Tsoulos I.G.; Gianni A.M; An Innovative Hybrid Approach to Global Optimization. *Journal Not Specified* **2023**, *1*, 0. https://doi.org/

Received:
Revised:
Accepted:
Published:

1. Introduction

The basic goal of global optimization is to find the global minimum by searching the appropriate range of the underlying objective problem. The global optimization method aims to find the global minimum of a continuous multidimensional function and is defined as

$$x^* = \arg\min_{x \in S} f(x) \tag{1}$$

with S:

$$S = [a_1, b_1] \times [a_2, b_2] \times \ldots [a_n, b_n]$$

According to literature research there are a variety of real-world problems that can be formulated as global optimization problems, such as problems in mathematics[1–3], physics[4–6], chemistry[7–9], and medicine[10–12], biology[13,14], agriculture[15,16] and economics[17,18]. Global Optimization refers to algorithms whose main objective is to find the global optimum of a problem. Armijo linear search is a method used to find an appropriate step when updating parameters in optimization problems. This method is part of the Armijo criterion, which ensures that the step chosen leads to a sufficient minimization of the cost function[19]. Optimization methods can be categorized into deterministic[20–22] and stochastic[23–25] based on how they approach solving the problem. The techniques used for deterministic are mainly interval methods. In interval methods, the set S is divided into smaller regions that may contain the global minimum using certain criteria. On the other hand stochastic methods use randomness to explore the solution space. A group of stochastic programming techniques that have been proposed to address optimization problems are evolutionary techniques. Genetic Algorithms[26–28] and differential evolution techniques[29,30] also belong to this group.

Genetic algorithms were formulated by John Hollands[31] and his team. Genetic algorithms initially generate random candidate solutions to an optimization problem. They can be combined with machine learning to solve complex problems.

On the other hand, differential evolution (DE) is used in symmetric optimization problems and in problems that are discontinuous and noisy and change over time. After studies, it was observed that differential evolution can be successfully combined with other techniques for machine learning applications, such as classification[32,33], feature selection[34,35], deep learning [36,37]etc.

Hybrid methods [38,39]in global optimization refer to techniques that combine multiple optimization strategies to solve complex problems. These methods aim to take advantage of different approaches to find the global optimum in a more efficient way, particularly when dealing with large-scale problems or strongly nonlinear optimization landscapes. A typical example of a hybrid method is the work of Shutao Li et al who propose a new hybrid PSO-BFGS strategy for the global optimization of multimodal functions[40]. To make the combination more efficient, they proposed an LDI to dynamically start the local search and a repositioning technique to maintain the particle diversity, which can effectively avoid the premature convergence problem. Another innovative hybrid method is the work of M. Andalib Sahnehsaraei et al where a hybrid algorithm using GA operators and PSO formula is proposed was presented through the use of efficient operators, for example, traditional and multiple crossovers, mutation and PSO formula[41].

The remainder of this paper is divided into the following sections: in section 2, the proposed method is described, in section 3 the experimental results and statistical comparisons are presented, and finally in section 4 some conclusions and guidelines for future improvements are discussed.

2. The overall algorithm

At the start of the process, the necessary initializations are performed for both the initial sample distribution and all parameters related to the hybrid method. The new solution is computed in each iteration of the algorithm through vector operations between neighboring samples, samples obtained from line search, and mutated samples. In each iteration of the algorithm, the following computations are carried out:

- Identification of the nearest point c_ι for each sample x_ι from the initial distribution 2.
- Calculation of the optimal sample $minLS(x, c)$ through line search Armijo between the sample x_i and the samplec_i 3.
- Generation of the sample using the crossover process between the sample x_ι and the best samplex_i^{best} 4.
- Computation of the trial point y_i 5.

It is evident that the formula used to compute the trial sample resembles the one in Differential Evolution (DE), where vector operations involve three random and distinct samples from the initial distribution. In that case, the search is driven by randomness. In the proposed method, however, the search starts from the initial search points and gradually shifts, incorporating information from both the local line search sample and the mutation chromosome, thereby avoiding vector jumps that could reduce the method's efficiency. The comparison between the proposed method and DE is presented in Figure 2.

Algorithm 1 Proposed algorithm

Initialization step.

- **Set** the population size N $\geq$4.
- **Set** n the dimension of the benchmark function.
- **Apply** uniform initial distribution for all samples end initialize them.

Calculation step.

- **Repeat**
  - **For** $i = 1 \ldots \mathrm{N}$ **do**
    - **Set** x as the sample i.
    - **Find nearest sample** c_ι **from** x_ι**:**

$$d(x,c) = \sqrt{\sum_{i=1}^{n} (x_i - c_i)^2} \tag{2}$$

    - **Set** direction vectors: $p_1 = -\nabla f(x_i)$ end $p_2 = -\nabla f(c_i)$
    - **Set** initial step size for Armijo $a = a_0$
    - **Compute** with line search Armijo the sample:
      - **Find** new points using line search $minLS(x,c)$: $x_i^{new} = x_i + ap_1$ and $c_i^{new} = c_i + ap_2$
      - **Adjust** step size a until Armijo condition is met:

$$f(x_i^{new}, c_i{}^{new}) \leq f(x_i, c_i) + c_1 a \nabla f(x_i, c_i)^T (p_1, p_2) \tag{3}$$

    - **Make** sample-child with crossover with random number $g_k \in [0.0, 1.0]$:

$$child(x, x^{best}) = g_k x_k + (1 - g_k) x_k^{best} \tag{4}$$

    - **For** $j = 1, \ldots, n$ **do**
      - **Set** trial vector:

$$y_j = x_j + r \times \left(minLS(x_\iota, c_\iota)_j - child(x_i, c_i)_j \right) \tag{5}$$

      else $y_j = x_j$
    - **EndFor**
    - **Set** $r \in [0,1]$ a random number. If $r \leq p_m$ then $x_i = \mathrm{LS}(x_i)$, where $\mathrm{LS}(x)$ is a local search procedure[42].
    - **If** $f(y) \leq f(x)$ then $x = y$, $x^{best} = y$.
  - **EndFor**
- **Terminate** when the stopping criterion is met. Termination is reached when for the current optimal solution $f_{min}^{(t)}$ is the same as the previous $f_{min}^{(t-1)}$, N_t times[43]:

$$\left| f_{\min}^{(k)} - f_{\min}^{(k-1)} \right| \tag{6}$$

- **Return** the sample x^{best} in the population with the lower function value $f\left(x^{best}\right)$.

3. Experiments

Settings and benchmark functions

To ensure the reliability of the experimental results, the experiments were repeated 30 times using different seeds for the random number generator. The hardware used included an AMD Ryzen 5950X processor and 128 GB of RAM. For the software, the Debian Linux operating system was employed in combination with the open-source optimizer GLOBALOPTIMUS, available at [http://www.github.com/itsoulos]. The test functions used in the experiments are presented in Table 2.For a more accurate comparison of the methods, efforts were made to maintain certain parameter values at equal or similar levels. The parameters used are presented in Table1.

Table 1. Parameters of optimization methods settings

PARAMETER	VALUE	EXPLANATION
N_i	200	Number of samples for all methods
N_k	200	Maximum number of iterations for all methods
SR	$\left\lvert f_{\min}^{(k)} - f_{\min}^{(k-1)} \right\rvert$	Best fitness: Stopping rule for all methods
LSR	0.05 (5%)	Local search rate for all methods
N_t	12	Similarity max count for all methods
F	0.8	Differential weight for DE
CR	0.9	Crossover Probability for DE
C_1, C_2	0.5	Parameters of PSO
G_c	0.1 (10%)	Crossover rate for Genetic
G_m	0.05 (5%)	Mutation rate for Genetic

Table 2. The benchmark functions used in the conducted experiments.

NAME	FORMULA	DIMENSION
BF1	$f(x) = x_1^2 + 2x_2^2 - \frac{3}{10}\cos(3\pi x_1) - \frac{4}{10}\cos(4\pi x_2) + \frac{7}{10}$	2
BF2	$f(x) = x_1^2 + 2x_2^2 - \frac{3}{10}\cos(3\pi x_1)\cos(4\pi x_2) + \frac{3}{10}$	2
BF3	$f(x) = x_1^2 + 2x_2^2 - \frac{3}{10}\cos(3\pi x_1)\cos(4\pi x_2) + \frac{3}{10}$	2
BRANIN	$f(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10$	2
CAMEL	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4, \quad x \in [-5,5]^2$	2
Easom	$f(x) = -\cos(x_1)\cos(x_2)\exp\left((x_2 - \pi)^2 - (x_1 - \pi)^2\right)$	2
ELP	$f(x) = \sum_{i=1}^{n} (10^6)^{\frac{i-1}{n-1}} x_i^2$	$n = 10, 20, 30$
Exp	$f(x) = -\exp(-0.5\sum_{i=1}^{n} x_i^2), \quad -1 \le x_i \le 1$	$n = 4, 8, 16, 32$
Gkls[44]	$f(x) = \text{Gkls}(x, n, w)$	$n = 2, 3\, w = 50, 100$
Griewank2	$f(x) = 1 + \frac{1}{200}\sum_{i=1}^{2} x_i^2 - \prod_{i=1}^{2} \frac{\cos(x_i)}{\sqrt{(i)}}$	2
Griewank10	$\mathrm{f}(x) = 1 + \frac{1}{200}\sum_{i=1}^{10} x_i^2 - \prod_{i=1}^{10} \frac{\cos(x_i)}{\sqrt{(i)}}$	10
Hansen	$f(x) = \sum_{i=1}^{5} i\cos[(i-1)x_1 + i]\sum_{j=1}^{5} j\cos[(j+1)x_2 + j]$	2
Hartman3	$f(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	3
Hartman6	$f(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	6
Potential[45]	$V_{LJ}(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$	$n = 9, 15, 21, 30$
Rastrigin	$f(x) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2)$	2
Rosenbrcok	$f(x) = \sum_{i=1}^{n-1}\left(100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right), \quad -30 \le x_i \le 30$	$n = 4, 8, 16$
Shekel5	$f(x) = -\sum_{i=1}^{5} \frac{1}{(x-a_i)(x-a_i)^T + c_i}$	4
Shekel7	$f(x) = -\sum_{i=1}^{7} \frac{1}{(x-a_i)(x-a_i)^T + c_i}$	4
Shekel10	$f(x) = -\sum_{i=1}^{10} \frac{1}{(x-a_i)(x-a_i)^T + c_i}$	4
Sinusoidal[46]	$f(x) = -(2.5\prod_{i=1}^{n}\sin(x_i - z) + \prod_{i=1}^{n}\sin(5(x_i - z))), \quad 0 \le x_i \le \pi$	$n = 4, 8$
Test2N	$f(x) = \frac{1}{2}\sum_{i=1}^{n} x_i^4 - 16x_i^2 + 5x_i$	$n = 4, 5, 7$
Test30N	$\frac{1}{10}\sin^2(3\pi x_1)\sum_{i=2}^{n-1}\left((x_i - 1)^2(1 + \sin^2(3\pi x_{i+1}))\right) + (x_n - 1)^2(1 + \sin^2(2\pi x_n))$	$n = 3, 4$

Experimental results

In Table 3, the average of objective function calls for each method is presented. At the end of the table, the total sum of calls for each method is listed. The number in parentheses indicates the percentage of executions in which the global optimum was successfully found. The absence of this number signifies that the global minimum was computed in every independent run with 100% success. Additionally, Figure 1 is derived from the data provided in Table 3.

From Figure 3, it is evident that the proposed method demonstrates significantly better performance compared to DE, with a substantial reduction in cost (144,024 versus 296,929). The t-test confirmed the statistically significant difference (p-value < 0.05). Compared to IPSO, the proposed method also outperforms, with lower values in almost all functions (144,024 versus 195,297). The Kruskal-Wallis test further confirmed the statistical significance of this difference (p-value < 0.05). Regarding GENETIC, the proposed method proves to be more efficient, with a lower total cost (144,024 versus 171,842), and the t-test indicates a statistically significant superiority.

In conclusion, the proposed method is more efficient compared to DE, IPSO, and GENETIC. The differences in objective function costs are statistically significant, as confirmed by both parametric (t-test) and non-parametric (Kruskal-Wallis) analyses, demonstrating its superiority in solving the specific functions.

The Figure 4a presents the performance of the ELP problem in terms of function calls across various dimensions, from 10 to 100. It is evident that as the dimensionality increases, the number of function calls required to solve the problem also rises. Specifically, for a dimension of 10, the algorithm requires 2,820 function calls, whereas for a dimension of 100, it demands 23,062 function calls, which is approximately eight times higher. This increase in function calls as the problem's dimensionality grows suggests that the complexity of the ELP problem scales significantly with dimensionality. The relationship between dimensions and function calls appears to be nonlinear, indicating that higher dimensions introduce additional computational challenges. For instance, moving from 10 to 20 dimensions more than doubles the required function calls, from 2,820 to 5,337, while the increase between 90 and 100 dimensions, although significant, shows a slightly smaller relative increase (19,598 to 23,062). The Figure 4b shows the response times of the ELP problem across various dimensions, from 10 to 100, measured in seconds. As the dimensionality increases, the execution time also rises significantly. For instance, with 10 dimensions, the execution time is only 0.224 seconds, while for 100 dimensions, it escalates to 32.537 seconds, marking a more than 145-fold increase. This substantial growth in execution time as the problem's dimensionality increases highlights the complexity and computational demands of higher-dimensional problems. The relationship between dimensions and time appears to be nonlinear, as seen in the sharp rise in time as dimensions grow. For example, moving from 10 to 20 dimensions results in more than doubling the time, from 0.224 to 0.577 seconds, while increasing from 90 to 100 dimensions causes a larger jump, from 21.897 to 32.537 seconds.

Table 3. Comparison of function calls of porpoded method against others

FUNCTION	DE	IPSO	GENETIC	PROPOSED
BF1	8268	4113	4007	3951
BF2	7913	3747	3793	3382
BF3	6327	3305	3479	2736
BRANIN	4101	2522	2376	1622
CAMEL	5609	2908	2869	2027
EASOM	2978	1998	1958	969
ELP10	6288	4397	3131	2820
ELP20	10794	6883	6160	5337
ELP30	14172	9438	9576	7070
EXP4	5166	3177	2946	2370
EXP16	6498	3477	3250	2654
EXP32	7606	3728	3561	2652
GKLS250	3834	2495	2280	1115
GKLS350	3919	2658	2612	945(93)
GOLDSTEIN	6781	3856	3687	2676
GRIEWANK2	7429(96)	3168	4500(96)	2453(50)
GRIEWANK10	18490	7942	6409	6351
HANSEN	4185	2892	3209	2525
HARTMAN3	5190	3103	2751	1945
HARTMAN6	5968	3688	3219	2832
POTENTIAL3	6218	5154	4351	3537
POTENTIAL5	9119	10128	7704	6735
POTENTIAL6	10509(76)	11780(46)	10177(70)	7706(76)
POTENTIAL10	12721(96)	16550(86)	13357	11517
RASTRIGIN	6216	3539	4106	2125(86)
ROSENBROCK4	8452	5858	3679	4442
ROSENBROCK8	11530	7843	5269	6726
ROSENBROCK16	17432	11450	8509	7310
SHEKEL5	6662	3886	3325	3722
SHEKEL7	6967	4009	3360	3817
SHEKEL10	6757	3985	2488	3317
SINU4	5953	3409	2990	2192
SINU8	6973	3995	3441	3171
SINU16	6979	4680	4320	5250
TEST2N4	6396	3390	3330	2235
TEST2N5	6271	3604	4000(96)	2530(93)
TEST2N7	7074	4020(96)	4775(73)	2939(50)
TEST30N3	6178	4018	3210	2728
TEST30N4	7006	4504	3678	3593
TOTAL SUM	296929	195297	171842	144024

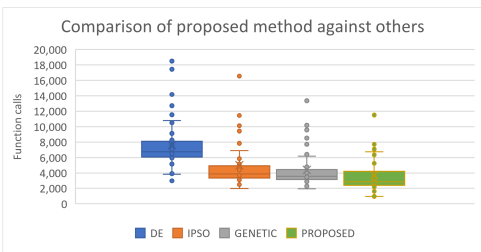


Figure 1. Comparison of function calls of proposed method against others

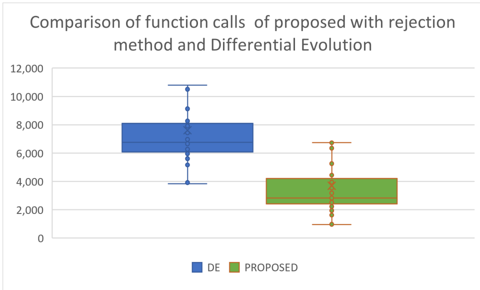


Figure 2. Comparison of proposed method against DE

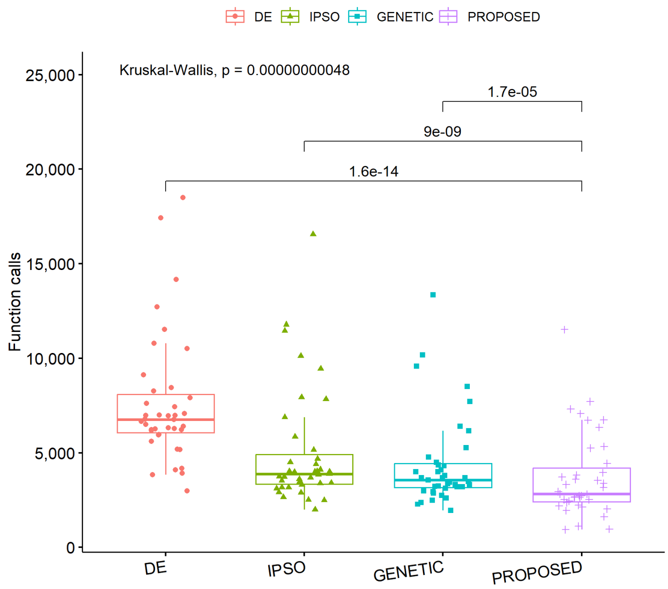
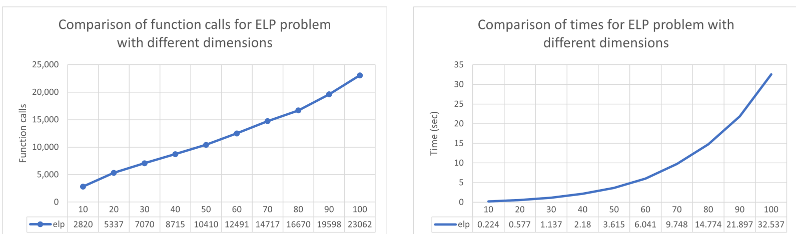


Figure 3. Statistical comparison of proposed method against others



(a) ELP: Function calls **(b)** ELP: Times

Figure 4. Different variations of the ELP problem

4. Conclusions

The proposed optimization method demonstrates significantly better performance compared to the other methods (DE, IPSO, GENETIC) in terms of objective function calls, with fewer calls indicating better efficiency. This suggests that the proposed method is more efficient, achieving optimal solutions with fewer objective function evaluations, which is crucial in problems where each call is computationally expensive. Statistical tests, including both the t-test and Kruskal-Wallis, confirm that the differences in the number of calls between the proposed method and the others are statistically significant (p-value < 0.05). This indicates that the proposed method not only uses fewer resources but also achieves reliable results with improved efficiency. In summary, the proposed method clearly excels in efficiency compared to DE, IPSO, and GENETIC, significantly reducing the number of objective function calls and optimizing computational cost. Potential improvements to the algorithm could involve identifying samples that contribute more effectively toward finding the optimal solution. Additionally, since this is a novel optimization method, exploring alternative termination criteria or different initial sample distributions may lead to enhanced performance.

Author Contributions: V.C., I.G.T. and V.S. conceived the idea and methodology and supervised the technical part regarding the software. V.C. conducted the experiments, employing several datasets, and provided the comparative experiments. I.G.T. performed the statistical analysis. V.S. and all other authors prepared the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Institutional Review Board Statement: Not applicable.

Acknowledgments: The experiments of this research work were performed at the high performance computing system established at Knowledge and Intelligent Computing Laboratory, Department of Informatics and Telecommunications, University of Ioannina, acquired with the project "Educational Laboratory equipment of TEI of Epirus" with MIS 5007094 funded by the Operational Programme "Epirus" 2014–2020, by ERDF and national funds.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Intriligator, M. D. (2002). Mathematical optimization and economic theory. Society for Industrial and Applied Mathematics.
2. Cánovas, M. J., Kruger, A., Phu, H. X., & Théra, M. (2020). Marco A. López, a Pioneer of Continuous Optimization in Spain. Vietnam Journal of Mathematics, 48, 211-219.
3. Mahmoodabadi, M. J., & Nemati, A. R. (2016). A novel adaptive genetic algorithm for global optimization of mathematical test functions and real-world problems. Engineering Science and Technology, an International Journal, 19(4), 2002-2021.
4. E. Iuliano, Global optimization of benchmark aerodynamic cases using physics-based surrogate models, Aerospace Science and Technology **67**, pp.273-286, 2017.
5. Q. Duan, S. Sorooshian, V. Gupta, Effective and efficient global optimization for conceptual rainfall-runoff models, Water Resources Research **28**, pp. 1015-1031 , 1992.
6. L. Yang, D. Robin, F. Sannibale, C. Steier, W. Wan, Global optimization of an accelerator lattice using multiobjective genetic algorithms, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **609**, pp. 50-57, 2009.
7. S. Heiles, R. L. Johnston, Global optimization of clusters using electronic structure methods, Int. J. Quantum Chem. **113**, pp. 2091– 2109, 2013.
8. W.H. Shin, J.K. Kim, D.S. Kim, C. Seok, GalaxyDock2: Protein–ligand docking using beta-complex and global optimization, J. Comput. Chem. **34**, pp. 2647– 2656, 2013.

9. A. Liwo, J. Lee, D.R. Ripoll, J. Pillardy, H. A. Scheraga, Protein structure prediction by global optimization of a potential energy function, Biophysics **96**, pp. 5482-5485, 1999.
10. Eva K. Lee, Large-Scale Optimization-Based Classification Models in Medicine and Biology, Annals of Biomedical Engineering **35**, pp 1095-1109, 2007.
11. Y. Cherruault, Global optimization in biology and medicine, Mathematical and Computer Modelling **20**, pp. 119-132, 1994.
12. Houssein, E. H., Hosney, M. E., Mohamed, W. M., Ali, A. A., & Younis, E. M. (2023). Fuzzy-based hunger games search algorithm for global optimization and feature selection using medical data. Neural Computing and Applications, 35(7), 5251-5275.
13. Banga, J.R. Optimization in computational systems biology. BMC Syst. Biol. 2008, 2, 47.
14. Beites, T.; Mendes, M.V. Chassis optimization as a cornerstone for the application of synthetic biology based strategies in microbial secondary metabolism. Front. Microbiol. 2015, 6, 159095.
15. Filip, M.; Zoubek, T.; Bumbalek, R.; Cerny, P.; Batista, C.E.; Olsan, P.; Bartos, P.; Kriz, P.; Xiao, M.; Dolan, A.; et al. Advanced computational methods for agriculture machinery movement optimization with applications in sugarcane production. Agriculture 2020, 10, 434
16. Zhang, D.; Guo, P. Integrated agriculture water management optimization model for water saving potential analysis. Agric. Water Manag. 2016, 170, 5–19.
17. Intriligator, M.D. Mathematical Optimization and Economic Theory; SIAM: Philadelphia, PA, USA, 2002
18. Dixit, A.K. Optimization in Economic Theory; Oxford University Press: Oxford, MA, USA, 1990.
19. Armijo, L. (1966). Minimization of functions having Lipschitz continuous first partial derivatives. Pacific Journal of mathematics, 16(1), 1-3.
20. Ion, I. G., Bontinck, Z., Loukrezis, D., Römer, U., Lass, O., Ulbrich, S., ... & De Gersem, H. (2018). Robust shape optimization of electric devices based on deterministic optimization methods and finite-element analysis with affine parametrization and design elements. Electrical Engineering, 100(4), 2635-2647.
21. Cuevas-Velásquez, V., Sordo-Ward, A., García-Palacios, J. H., Bianucci, P., & Garrote, L. (2020). Probabilistic model for real-time flood operation of a dam based on a deterministic optimization model. Water, 12(11), 3206.
22. Pereyra, M., Schniter, P., Chouzenoux, E., Pesquet, J. C., Tourneret, J. Y., Hero, A. O., & McLaughlin, S. (2015). A survey of stochastic simulation and optimization methods in signal processing. IEEE Journal of Selected Topics in Signal Processing, 10(2), 224-241.
23. Hannah, L. A. (2015). Stochastic optimization. International Encyclopedia of the Social & Behavioral Sciences, 2, 473-481.
24. Kizielewicz, B., & Sałabun, W. (2020). A new approach to identifying a multi-criteria decision model based on stochastic optimization techniques. Symmetry, 12(9), 1551.
25. Chen, T., Sun, Y., & Yin, W. (2021). Solving stochastic compositional optimization is nearly as easy as solving stochastic optimization. IEEE Transactions on Signal Processing, 69, 4937-4948.
26. D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing Company, Reading, Massachussets, 1989.
27. Z. Michaelewicz, Genetic Algorithms + Data Structures = Evolution Programs. Springer - Verlag, Berlin, 1996.
28. Charilogis, V., Tsoulos, I. G., & Stavrou, V. N. (2023). An Intelligent Technique for Initial Distribution of Genetic Algorithms. Axioms, 12(10), 980.
29. Charilogis, V., Tsoulos, I. G., Tzallas, A., & Karvounis, E. (2022). Modifications for the differential evolution algorithm. Symmetry, 14(3), 447.
30. Storn, R. On the usage of differential evolution for function optimization. In Proceedings of the North American Fuzzy Information Processing, Berkeley, CA, USA, 19–22 June 1996; pp. 519–523.
31. Holland, J.H. Genetic algorithms. Sci. Am. 1992, 267, 66–73.
32. Maulik, U.; Saha, I. Automatic Fuzzy Clustering Using Modified Differential Evolution for Image Classification. IEEE Trans. Geosci. Remote Sens. 2010, 48, 3503–3510.
33. Zhang, Y.; Zhang, H.; Cai, J.; Yang, B. A Weighted Voting Classifier Based on Differential Evolution. Abstr. Appl. Anal. 2014, 2014, 376950.
34. Hancer, E. Differential evolution for feature selection: A fuzzy wrapper–filter approach. Soft Comput. 2019, 23, 5233–5248.
35. Vivekanandan, T.; Iyengar, N.C.S.N. Optimal feature selection using a modified differential evolution algorithm and its effectiveness for prediction of heart disease. Comput. Biol. Med. 2017, 90, 125–136.

36. Deng, W.; Liu, H.; Xu, J.; Zhao, H.; Song, Y. An Improved Quantum-Inspired Differential Evolution Algorithm for Deep Belief Network. IEEE Trans. Instrum. Meas. 2020, 69, 7319–7327.
37. Wu, T.; Li, X.; Zhou, D.; Li, N.; Shi, J. Differential Evolution Based Layer-Wise Weight Pruning for Compressing Deep Neural Networks. Sensors 2021, 21, 880.
38. H. Badem, A. Basturk, A. Caliskan, M.E. Yuksel, A new hybrid optimization method combining artificial bee colony and limited-memory BFGS algorithms for efficient numerical optimization, Applied Soft Computing **70**, pp. 826-844, 2018.
39. A.A. Nagra, F. Han, Q.H. Ling, An improved hybrid self-inertia weight adaptive particle swarm optimization algorithm with local search, Engineering Optimization **51**, pp. 1115-1132, 2018.
40. Li, S., Tan, M., Tsang, I. W., & Kwok, J. T. Y. (2011). A hybrid PSO-BFGS strategy for global optimization of multimodal functions. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 41(4), 1003-1014.
41. Andalib Sahnehsaraei, M., Mahmoodabadi, M. J., Taherkhorsandi, M., Castillo-Villar, K. K., & Mortazavi Yazdi, S. M. (2015). A hybrid global optimization algorithm: particle swarm optimization in association with a genetic algorithm. Complex System Modelling and Control Through Intelligent Soft Computations, 45-86.
42. , Convergence properties of the BFGS algoritm. SIAM Journal on Optimization **13**, pp. 693-701, 2002.
43. Charilogis, V.; Tsoulos, I.G. Toward an Ideal Particle Swarm Optimizer for Multidimensional Functions. Information 2022, 13, 217.
44. M. Gaviano, D.E. Ksasov, D. Lera, Y.D. Sergeyev, Software for generation of classes of test functions with known local and global minima for global optimization, ACM Trans. Math. Softw. **29**, pp. 469-480, 2003.
45. J.E. Lennard-Jones, On the Determination of Molecular Fields, Proc. R. Soc. Lond. A **106**, pp. 463–477, 1924.
46. Z.B. Zabinsky, D.L. Graesser, M.E. Tuttle, G.I. Kim, Global optimization of composite laminates using improving hit and run, In: Recent advances in global optimization, pp. 343-368, 1992.