Article

An Innovative Hybrid Approach to Global Optimization

Vasileios Charilogis[1],Glykeria Kyrou[2], Ioannis G. 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 established global optimization algorithms. This paper presents a hybrid global optimization method, which produces trial solutions of the objective problem through vector operations derived from methods, such as Genetic Algorithms, Line Search and Differential Evolution. 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. The new method was applied on a wide series of benchmark problems from the recent literature and comparison was made against other established methods of Global Optimization.

Keywords: Optimization; Differential evolution; Genetic algorithm; Line search; Evolutionary techniques; Stochastic methods; Hybrid methods.

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 primary objective of global optimization is to locate the global minimum by thoroughly exploring the relevant range associated with the underlying objective problem. This method of global optimization is focused on identifying the global minimum within a continuous function that spans multiple dimensions. Essentially, the global optimization process is dedicated to seeking out the minimum value of a continuous, multidimensional function, ensuring that the search covers all potential ranges of the problem at hand. The objective is to find the lowest point through systematic exploration of the entire domain of the function, which is defined in a Euclidean space R^n. The optimal value of a function $f : S \to R, S \subset R^n$ is defined as follows:

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

where the set S is defined as follows:

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

Global Optimization refers to algorithms whose main objective is to find the global optimum of a problem. 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]. Optimization methods can be categorized into deterministic [19–21] and stochastic [22–24] based on how they approach solving the problem. The techniques used for deterministic are mainly interval methods [25,26]. Stochastic methods utilize randomness to explore the solution space, while in interval methods, the set S is divided into smaller regions that may contain the global minimum based on certain criteria. Recently, a comparison between deterministic and stochastic methods was proposed by Sergeyev et al [27].

A series of stochastic optimization methods are the so - called evolutionary methods, which attempt to mimic a series of natural processes. Such methods include the Genetic algorithms [28,29], the Differential Evolution method [31,32], Particle Swarm Optimization (PSO) methods [33–35], Ant Colony optimization methods [36,37], the Fish Swarm Algorithm [38], the Dolphin Swarm Algorithm [39], the Whale Optimization Algorithm (WOA) algorithm [40–42] etc. Also, due to the wide spread of parallel computing units, a variety of research papers related to evolutionary techniques appeared that use such processing units [43–45]. In the current work, two evolutionary methods were incorporated in the final algorithm: Genetic Algorithms and the Differential Evolution method was combined into a hybrid optimization method.

Genetic algorithms were formulated by John Holland [46] and his team and they initially generated randomly candidate solutions to an optimization problem. These solutions were modified through a series of operators that mimic natural processes, such as mutation, selection and crossover. Genetic algorithms have been used widely in areas such as networking [47], robotics [48,49], energy topics [50,51] etc. They can be combined with machine learning to solve complex problems, such as neural network training [52,53].

Furthermore, differential evolution (DE) is used in symmetric optimization problems [54,55] 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 [56,57], feature selection [58,59], deep learning [60,61] etc.

Hybrid methods [62,63] 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 proposed a new hybrid PSO-BFGS strategy for the global optimization of multimodal functions [64]. 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 [65].

The current work produces through a series of steps trial solutions using the genetic operators of the Genetic Algorithm as well as solutions identified by a line search procedure. In current work, the Armijo line search is a method is used. This method is incorporated to estimate an appropriate step when updating the trial points, and it was introduced in the work of Armijo[66]. The solutions produced in the previous step are used to formulate new trial solutions using a process derived from Differential Evolution.

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

The proposed method combines some aspects from different optimization algorithms and the main steps are subsequently:

1. **Initialization step.**
   (a) **Set** the population size $N \geq 4$.
   (b) **Set** n the dimension of the benchmark function.

   (c) **Initialize** the samples $x_i, i = 1, \ldots, N$ using uniform distribution.
2. **Calculation step.**
   (a) **For** $i = 1 \ldots \mathrm{N}$ **do**
      i. **Obtain** sample x_i.
      ii. **Find nearest sample** c_ι **from** x_ι**:**

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

      where $d(x,c)$ is the Euclidean Distance.
      iii. **Set** direction vectors: $p_1 = -\nabla f(x_i)$ end $p_2 = -\nabla f(c_i)$
      iv. **Set** initial step size for Armijo $a = a_0$
      v. **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:

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

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

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

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

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

         where F is the so - called Differential Weight of Differential Evolution algorithm.
         - **If** $y_j \notin \left[a_j, b_j\right]$, then $y_j = x_{i,j}$
      viii. **EndFor**
      - **Set** $r \in [0,1]$ a random number. If $r \leq p_m$ then $x_i = \text{LS}(x_i)$, where $\text{LS}(x)$ is a local search procedure like the BFGS procedure[67].
      - **If** $f(y) \leq f(x)$ then $x = y$, $x^{best} = y$.
   (b) **EndFor**
3. **Check the termination rule stated in** [68], which means that the method checks the difference between the current optimal solution $f_{min}^{(t)}$ and the previous $f_{min}^{(t-1)}$ one. The algorithm terminates when the following:

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

   holds for N_t iterations. The value ϵ is a small positive value. In the conducted experiments the value $\epsilon = 10^{-5}$ was used. If the termination rule of equation 6 does not hold, then the algorithm continues from Step 2.
4. **Return** the sample x^{best} in the population with the lower function value $f\left(x^{best}\right)$.

Initially, the algorithm initializes the trial solutions as well as the necessary parameters. At every iteration, the method constructs trial solutions using genetic algorithms operations as well as the Armijo line search method and these solutions are combined using a process from Differential Evolution, using the following steps:

- Identification of the nearest point c_ι for each sample x_ι.

- Calculation of a sample minLS(x, c) through Armijo line search, between the sample x_i and the samplec_i.
- Generation of the sample using the crossover process of the Genetic Algorithm, between the sample x_ι and the best samplex_i^{best} .
- Computation of the trial point y_i using a process derived from Differential Evolution.

3. Experiments

Settings and benchmark functions

In the random number generator, different seeds were used to ensure the reliability of the experimental results, with the experiments being repeated 30 times. This process of repetition aims to minimize the likelihood of random errors and to enhance the validity of the results. The experiments were conducted on a system with an AMD Ryzen 5950X processor and 128 GB of RAM, operating in a Linux-Debian environment. Additionally, the open-source optimizer "GlobalOptimus" was used, which is a fully developed optimizer and is available for distribution via the link: http://www.github.com/itsoulos/GlobalOptimus (accessed on 17 September 2024). The benchmark functions used in the experimental measurements are presented in Table 1.

Table 1. 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	$f(x) = -\exp(-0.5\sum_{i=1}^{n} x_i^2), \quad -1 \leq x_i \leq 1$	$n = 4, 8, 16, 32$
Gkls[69]	$f(x) = \text{Gkls}(x, n, w)$	$n = 2, 3\, w = 50, 100$
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	$\text{f}(x) = 1 + \frac{1}{200}\sum_{i=1}^{10} x_i^2 - \prod_{i=1}^{10}\frac{\cos(x_i)}{\sqrt{(i)}}$	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[70]	$V_{LJ}(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6\right]$	$n = 9, 15, 21, 30$
Rastrigin	$f(x) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2)$	2
Rosenbrcok	$f(x) = \sum_{i=1}^{n-1}\left(100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right), \quad -30 \leq x_i \leq 30$	$n = 4, 8, 16$
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[71]	$f(x) = -(2.5\prod_{i=1}^{n}\sin(x_i - z) + \prod_{i=1}^{n}\sin(5(x_i - z))), \quad 0 \leq x_i \leq \pi$	$n = 4, 8$
Test2N	$f(x) = \frac{1}{2}\sum_{i=1}^{n} x_i^4 - 16x_i^2 + 5x_i$	$n = 4, 5, 7$
Test30N	$\frac{1}{10}\sin^2(3\pi x_1)\sum_{i=2}^{n-1}\left((x_i - 1)^2(1 + \sin^2(3\pi x_{i+1}))\right) + (x_n - 1)^2(1 + \sin^2(2\pi x_n))$	$n = 3, 4$

The functions used in the conducted experiments have been proposed by various researchers [72,73] in the relevant literature. For a more accurate comparison of the methods, efforts were made to maintain certain parameter values at equal or similar levels. The values for the parameters of the algorithm are presented in Table2, along with some explanation of each parameter.

Table 2. Parameters of optimization methods settings

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

Experimental results

In Table 3, the average function calls for each method and every objective function is presented. Description of table captions and values:

1. The column FUNCTION represents the name of the used objective problem.
2. The column DE represents the average function calls for the Differential Evolution optimization technique.
3. the column IPSO stands for the application of the Improved PSO algorithm as suggested by Charilogis and Tsoulos [74] to the objective problems.
4. The column GENETIC represents the application of the Genetic Algorithm to the objective problem.
5. The total sum of calls for each method is listed at the end of the table.
6. The success rate is indicated by the number in parentheses, which shows the executions in which the global optimum was successfully found. The absence of this number implies that the global minimum was computed with 100% success in all independent runs.

Table 3. Comparison of average function calls of proposed 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

At the end of every global optimization procedure, a BFGS variant of Powell [75] is utilized to improve the discovered solution and identify with certainty a local minimum. Additionally, Figure 1 presents a statistical comparison between the used optimization methods and the data provided from Table 3.

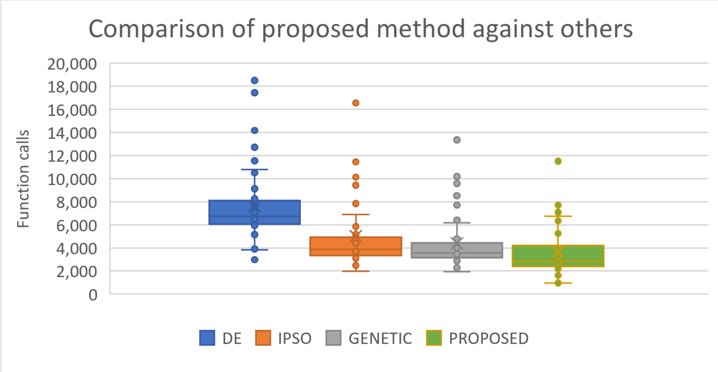


Figure 1. Comparison of function calls of proposed method against others

The proposed method seems to significantly reduce the required number of functional calls compared to other techniques, and in many functions this reduction can exceed 50%. This trend is also confirmed by Figure 2, where a statistical comparison is presented between universal optimization methods used in the experiments.

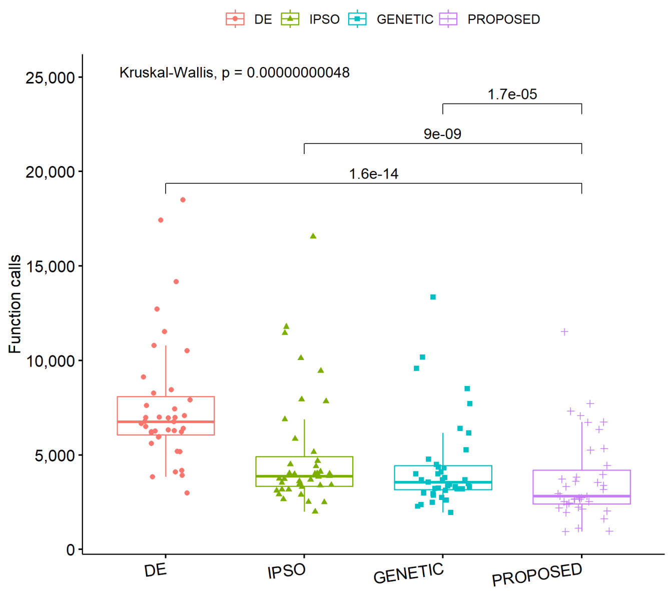


Figure 2. Statistical comparison of proposed method against others

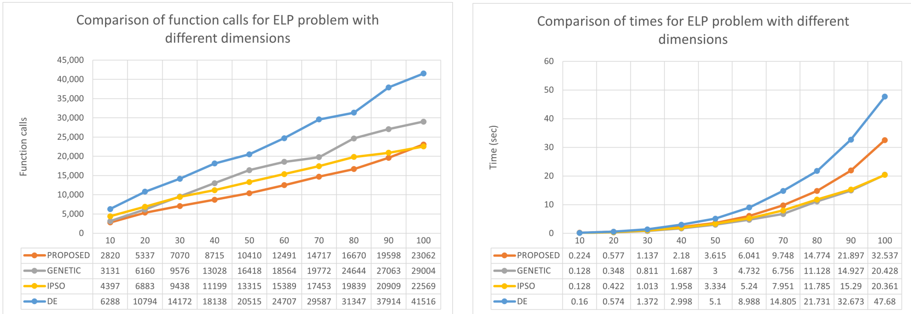
From the analysis presented in this Figure 2, it is abundantly clear that the proposed optimization method exhibits a markedly superior performance when juxtaposed with the DE technique. This superiority is evidenced by a significant reduction in computational cost, with the proposed method requiring only 144,024 function evaluations as opposed to the 296,929 required by DE. To substantiate this observation, a t-test was conducted, which confirmed the presence of a statistically significant difference, indicated by a p-value of less than 0.05. In a comparative assessment with the iPSO method, the proposed approach also demonstrates enhanced efficacy, achieving lower function evaluation costs across nearly all tested functions, yielding a cost of 144,024 compared to IPSO's 195,297. The Kruskal-Wallis test was employed to further validate the statistical significance of this observed difference, with results again confirming a p-value of less than 0.05. When evaluated against the Genetic Algorithm (GENETIC), the proposed method continues to display greater efficiency, as evidenced by a lower total cost of 144,024 in contrast to GENETIC's 171,842. The t-test results further reinforce this finding, highlighting a statistically significant advantage in favor of the proposed method.

In conclusion, the proposed optimization technique demonstrates enhanced efficiency relative to DE, IPSO, and GENETIC methodologies. The disparities in objective function costs are not only noteworthy but are also statistically significant, as corroborated by both parametric analysis through the t-test and non-parametric analysis via the Kruskal-Wallis test. This evidence collectively underscores the proposed method's superior capability in effectively addressing the specific optimization functions evaluated in this study.

An additional experiment was executed for the High Elliptic function, which is defined as:

$$f(x) = \sum_{i=1}^{n} \left(10^6\right)^{\frac{i-1}{n-1}} x_i^2$$

where the parameter n defines the dimension of the function. In this particular experiment, the proposed optimization method was systematically applied to a specific mathematical function as the dimension n underwent a transition from 10 to 100. Figure 3a illustrates the performance metrics of the proposed method concerning this function, focusing specifically on the number of function calls made across various dimensional settings. It becomes increasingly evident that as the dimensionality of the problem expands, the corresponding number of function calls necessary to effectively solve the problem also experiences a significant increase. To be more specific, when examining the case for a dimension of 10, the algorithm requires a total of 2,820 function calls to arrive at a solution. In stark contrast, for a dimension of 100, the algorithm demands a staggering 23,062 function calls, which represents an increase of approximately eight times the number of calls required at the lower dimensional level. This notable escalation in the number of function calls as the dimensionality of the problem grows strongly suggests that the inherent complexity of the ELP problem scales considerably with increasing dimensionality. Furthermore, the relationship between the number of dimensions and the number of function calls appears to exhibit a nonlinear pattern. This indicates that as the dimensions increase, additional computational challenges are introduced, complicating the problem-solving process. For instance, the transition from 10 to 20 dimensions results in more than a doubling of the required function calls, escalating from 2,820 to 5,337. Conversely, the increase observed between 90 and 100 dimensions, although still significant, demonstrates a somewhat smaller relative increase, changing from 19,598 to 23,062. Additionally, Figure 3b provides a visual representation of the response times associated with solving the ELP problem across the various dimensions, ranging from 10 to 100, with the time measured in seconds. As anticipated, with the increase in dimensionality, the execution time also rises dramatically. For example, when working with 10 dimensions, the execution time is relatively short, clocking in at only 0.224 seconds. However, when the dimensionality is increased to 100, the execution time escalates sharply to 32.537 seconds, marking an astonishing increase of more than 145 times the initial time. This substantial growth in execution time as the dimensionality of the problem increases underscores the inherent complexity and computational demands associated with higher-dimensional optimization problems. Moreover, the relationship between dimensions and execution time also appears to be nonlinear, as evidenced by the sharp increase in time observed as dimensions grow. For example, when moving from 10 to 20 dimensions, the time required more than doubles, rising from 0.224 seconds to 0.577 seconds. In contrast, the transition from 90 to 100 dimensions results in an even larger jump in execution time, from 21.897 seconds to 32.537 seconds, further illustrating the significant challenges posed by higher-dimensional problems.



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

Figure 3. Different variations of the ELP problem

In each iteration of the algorithm, a trial point is calculated through vector operations, similar to the process of optimization using differential evolution. The main difference between the proposed method compared to differential evolution lies in the fact that the samples for calculating the trial point are selected from nearby regions of the initial distribution, rather than being chosen randomly. However, the performance of the two methods differs in their ability to find optimal solutions, as shown in Figure4. The extraneous samples of Figure 4 have been removed.

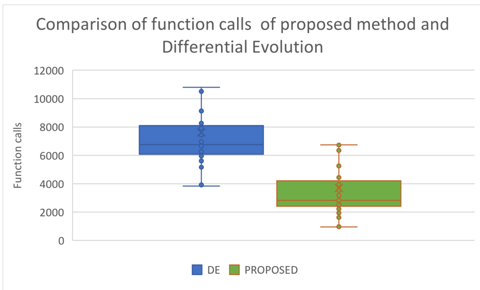


Figure 4. Comparison of proposed method against DE

4. Conclusions

An innovative global optimization method has been proposed in this research paper, which leverages techniques derived from well-established optimization strategies. More specifically, the new method incorporates genetic operators from Genetic Algorithms alongside the Linear Search method to generate candidate solutions for the given objective functions. These candidate solutions are then combined to create new solutions utilizing approaches inspired by the Differential Evolution method. To validate the effectiveness of this new optimization approach, a comprehensive series of experiments were conducted on various problems sourced from the existing literature. Additionally, numerical comparisons were made with recognized global optimization techniques to provide a clear benchmark. The results indicate that the proposed optimization method exhibits significantly superior performance when compared to alternative methods, particularly in terms of the number of calls made to the objective function. Fewer calls to the objective function suggest better overall efficiency, highlighting the proposed method's ability to achieve optimal solutions with fewer evaluations. This efficiency is particularly critical in scenarios where each function call is computationally expensive. Statistical analyses, including both the t-test and the Kruskal-Wallis test, confirm that the observed differences in the number of calls between the proposed method and other methods are statistically significant, with a p-value of less than 0.05. This finding not only underscores the reduced resource consumption of the proposed method but also affirms that it delivers reliable results with enhanced efficiency.

In summary, the proposed method stands out in terms of efficiency when compared to other optimization techniques, significantly decreasing the number of objective function calls and optimizing overall computational cost. Potential enhancements to the algorithm could involve identifying specific samples that contribute more effectively to the discovery of the optimal solution. Furthermore, since this method represents a novel approach to optimization, exploring alternative termination criteria or varying the initial sample distributions could lead to even greater performance improvements. By refining these aspects, the method could further bolster its efficiency and effectiveness in solving complex optimization problems.

Author Contributions: The conceptualization of the idea and the design of the methodology, as well as the supervision of the technical aspects related to the software, were undertaken by V.C., I.G.T., and G.K. The experiments were conducted using various datasets, and the comparative results were presented by V.C., G.K. and A.M.G. The statistical analysis was carried out by V.C. The manuscript was prepared by G.K. and the other authors. All authors have reviewed and approved the final published version.

Funding: No external funding was received for this research.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Institutional Review Board Statement: Not applicable.

Acknowledgments: The experiments for this research were conducted using the high-performance computing system installed at the Knowledge and Intelligent Computing Laboratory, Department of Informatics and Telecommunications, University of Ioannina, which was acquired through the project "Educational Laboratory Equipment of TEI of Epirus" with MIS 5007094, funded by the Operational Programme "Epirus" 2014–2020, through ERDF and national funds

Conflicts of Interest: No conflicts of interest are declared by the authors.

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. 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.
20. 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.
21. 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.
22. Hannah, L. A. (2015). Stochastic optimization. International Encyclopedia of the Social & Behavioral Sciences, 2, 473-481.
23. Kizielewicz, B., & Sałabun, W. (2020). A new approach to identifying a multi-criteria decision model based on stochastic optimization techniques. Symmetry, 12(9), 1551.
24. 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.
25. M.A. Wolfe, Interval methods for global optimization, Applied Mathematics and Computation **75**, pp. 179-206, 1996.
26. T. Csendes and D. Ratz, Subdivision Direction Selection in Interval Methods for Global Optimization, SIAM J. Numer. Anal. **34**, pp. 922–938, 1997.
27. Y.D. Sergeyev, D.E. Kvasov, M.S. Mukhametzhanov, On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget. Sci Rep **8**, 453, 2018.
28. D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing Company, Reading, Massachussets, 1989.
29. Z. Michaelewicz, Genetic Algorithms + Data Structures = Evolution Programs. Springer - Verlag, Berlin, 1996.
30. Charilogis, V., Tsoulos, I. G., & Stavrou, V. N. (2023). An Intelligent Technique for Initial Distribution of Genetic Algorithms. Axioms, 12(10), 980.
31. R. Storn, K. Price, Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, Journal of Global Optimization **11**, pp. 341-359, 1997.
32. J. Liu, J. Lampinen, A Fuzzy Adaptive Differential Evolution Algorithm. Soft Comput **9**, pp.448–462, 2005.
33. J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of ICNN'95 - International Conference on Neural Networks, 1995, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968.
34. Riccardo Poli, James Kennedy kennedy, Tim Blackwell, Particle swarm optimization An Overview, Swarm Intelligence **1**, pp 33-57, 2007.
35. Ioan Cristian Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Information Processing Letters **85**, pp. 317-325, 2003.
36. M. Dorigo, M. Birattari and T. Stutzle, Ant colony optimization, IEEE Computational Intelligence Magazine **1**, pp. 28-39, 2006.
37. K. Socha, M. Dorigo, Ant colony optimization for continuous domains, European Journal of Operational Research 185, pp. 1155-1173, 2008.

38. M. Neshat, G. Sepidnam, M. Sargolzaei, et al, Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications, Artif Intell Rev **42**, pp. 965–997, 2014.
39. Tq. Wu, M. Yao, Jh. Yang, Dolphin swarm algorithm, Frontiers Inf Technol Electronic Eng **17**, pp. 717–729, 2016.
40. Mirjalili, S., Lewis, A.: The whale optimization algorithm. Adv. Eng. Softw. 95, 51–67 (2016)
41. Nasiri, J., & Khiyabani, F. M. (2018). A whale optimization algorithm (WOA) approach for clustering. Cogent Mathematics & Statistics, 5(1), 1483565.
42. Gharehchopogh, F. S., & Gholizadeh, H. (2019). A comprehensive survey: Whale Optimization Algorithm and its applications. Swarm and Evolutionary Computation, 48, 1-24.
43. Y. Zhou and Y. Tan, "GPU-based parallel particle swarm optimization," 2009 IEEE Congress on Evolutionary Computation, pp. 1493-1500, 2009.
44. L. Dawson and I. Stewart, "Improving Ant Colony Optimization performance on the GPU using CUDA," 2013 IEEE Congress on Evolutionary Computation, 2013, pp. 1901-1908, doi: 10.1109/CEC.2013.6557791.
45. Barkalov, K., Gergel, V. Parallel global optimization on GPU. J Glob Optim 66, 3–20 (2016).
46. Holland, J.H. Genetic algorithms. Sci. Am. 1992, 267, 66–73.
47. Y.H. Santana, R.M. Alonso, G.G. Nieto, L. Martens, W. Joseph, D. Plets, Indoor genetic algorithm-based 5G network planning using a machine learning model for path loss estimation, Appl. Sci. **12**, 3923. 2022.
48. X. Liu, D. Jiang, B. Tao, G. Jiang, Y. Sun, J. Kong, B. Chen, Genetic algorithm-based trajectory optimization for digital twin robots, Front. Bioeng. Biotechnol **9**, 793782, 2022.
49. K. Nonoyama, Z.Liu, T. Fujiwara, M.M. Alam, T. Nishi, Energy-efficient robot configuration and motion planning using genetic algorithm and particle swarm optimization, Energies **15**, 2074, 2022.
50. K. Liu, B. Deng, Q. Shen, J. Yang, Y. Li, Optimization based on genetic algorithms on energy conservation potential of a high speed SI engine fueled with butanol–gasoline blends, Energy Rep. **8**, pp. 69–80, 2022.
51. G. Zhou, S. Zhu, S. Luo, Location optimization of electric vehicle charging stations: Based on cost model and genetic algorithm, Energy **247**, 123437, 2022.
52. J. Arifovic, R. Gençay, Using genetic algorithms to select architecture of a feedforward artificial neural network, Physica A: Statistical Mechanics and its Applications **289**, pp. 574-594, 2001.
53. F. H. F. Leung, H. K. Lam, S. H. Ling, P. K. S. Tam, Tuning of the structure and parameters of a neural network using an improved genetic algorithm, IEEE Transactions on Neural Networks **14**, pp. 79-88, 2003.
54. Y.H. Li, J.Q. Wang, X.J. Wang, Y.L. Zhao, X.H. Lu, D.L. Liu, Community Detection Based on Differential Evolution Using Social Spider Optimization, Symmetry **9**, 2017.
55. W. Yang, E.M. Dilanga Siriwardane, R. Dong, Y. Li, J. Hu, Crystal structure prediction of materials with high symmetry using differential evolution, J. Phys.: Condens. Matter **33** 455902, 2021.
56. Maulik, U.; Saha, I. Automatic Fuzzy Clustering Using Modified Differential Evolution for Image Classification. IEEE Trans. Geosci. Remote Sens. 2010, 48, 3503–3510.
57. Zhang, Y.; Zhang, H.; Cai, J.; Yang, B. A Weighted Voting Classifier Based on Differential Evolution. Abstr. Appl. Anal. 2014, 2014, 376950.
58. Hancer, E. Differential evolution for feature selection: A fuzzy wrapper–filter approach. Soft Comput. 2019, 23, 5233–5248.
59. 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.
60. 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.
61. 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.
62. 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.
63. 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.

64. 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.
65. 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.
66. Armijo, L. (1966). Minimization of functions having Lipschitz continuous first partial derivatives. Pacific Journal of mathematics, 16(1), 1-3.
67. , Convergence properties of the BFGS algoritm. SIAM Journal on Optimization **13**, pp. 693-701, 2002.
68. Charilogis, V.; Tsoulos, I.G. Toward an Ideal Particle Swarm Optimizer for Multidimensional Functions. Information 2022, 13, 217.
69. 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.
70. J.E. Lennard-Jones, On the Determination of Molecular Fields, Proc. R. Soc. Lond. A **106**, pp. 463–477, 1924.
71. 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.
72. M. Montaz Ali, Charoenchai Khompatraporn, Zelda B. Zabinsky, A Numerical Evaluation of Several Stochastic Algorithms on Selected Continuous Global Optimization Test Problems, Journal of Global Optimization **31**, pp 635-672, 2005.
73. C.A. Floudas, P.M. Pardalos, C. Adjiman, W. Esposoto, Z. Gümüs, S. Harding, J. Klepeis, C. Meyer, C. Schweiger, Handbook of Test Problems in Local and Global Optimization, Kluwer Academic Publishers, Dordrecht, 1999.
74. V. Charilogis, I.G. Tsoulos, Toward an Ideal Particle Swarm Optimizer for Multidimensional Functions , Information **13**, 217, 2022.
75. Powell, M.J.D. A Tolerant Algorithm for Linearly Constrained Optimization Calculations. Math. Program. 1989, 45, 547–566.