mühendislikdergisi
Dicle Üniversitesi Mühendislik Fakültesi

Dicle University Journal of Engineering (DUJE)

web: http://dergipark.gov.tr/dumf

Research Article

A Novel Modified Lévy Flight Distribution Algorithm based on Nelder-Mead Method for Function Optimization

Ahmet Dündar [1], Davut İzci [2], Serdar Ekinci [3], Erdal Eker [4,*]

[1] Vocational School of Social Sciences, Muş Alparslan University, Muş, https://orcid.org/0000-0002-2123-0564
[2] Department of Electronics and Automation, Batman University, Batman, https://orcid.org/0000-0001-8359-0875
[3] Department of Computer Engineering, Batman University, Batman, https://orcid.org/0000-0002-7673-2553
[4] Accounting and Tax Department, Muş Alparslan University, Muş, https://orcid.org/0000-0002-5470-8384

ARTICLE INFO

Article history:

Received 20 February 2021
Received in revised form 22 April 2021
Accepted 1 May 2021
Available online 22 June 2021

Keywords:

Lévy flight distribution, Metaheuristics, Nelder-Mead algorithm, Benchmark functions

ABSTRACT

This paper aims to improve one of the recently proposed metaheuristic approaches known as Lévy flight distribution (LFD) algorithm by adopting a well-known simplex search algorithm named Nelder-Mead (NM) method. Three new strategies were utilized to demonstrate the improved capability of the original LFD algorithm. In the first strategy, NM was run twice as much the number of iterations of LFD after the latter completes its task. In the second strategy, NM was applied after each iterations of LFD instead of waiting for the completion of the latter. Lastly, in the third strategy, NM was applied after each iterations of LFD and run for the total number of current iterations of the latter algorithm. Well-known unimodal and multimodal benchmark functions were adopted, and statistical analysis was performed for performance evaluation. Further assessment was carried out through a nonparametric statistical test. The obtained results have shown the proposed versions of LFD algorithm provide significant performance improvement in general. In addition, the efficiency of the third strategy was found to be better for NM modified LFD algorithm which has greater balance between global and local search stages and can be used as an effective tool for function optimization.

Doi: 10.24012/dumf.955645

Introduction

The optimization techniques have been gaining a greater demand in recent decades due to the need for reliable and effective methods to deal with real-life problems that present increased complexity [1]. In addition to this fact, the increasing number of optimization problems of different fields has led the optimization techniques to be one of the major research areas [2].

Due to inherent disadvantages of deterministic techniques such as being derivative dependent and stagnating in local optimum, this field of research tends to develop different alternative approaches that can avoid those issues and able to solve complex and nonlinear problems effectively [3]. As part of this effort, different metaheuristic algorithms have emerged [4]. The latter approaches have stochastic natures, thus, can explore entire search space by avoiding local optima [5] since they do not need the derivative information.

There are many metaheuristic algorithms which have already been used for different optimization problems [6]. Despite the existing ones and their various applications, it is still quite common to encounter with newer metaheuristics nowadays [7] which is motivated by "No free lunch theorem" [8].

* Corresponding author
Erdal Eker
✉ e.eker@alparslan.edu.tr

As stated by this theorem, there is not any single algorithm that is convenient to solve all existing optimization problems. Therefore, there is a growing appetite for developing newer algorithms that may be quite effective for some of the problems [9]. Lévy flight distribution (LFD) algorithm [10] has been developed as part of the latter effort. The wireless sensor nodes that have connections related to Lévy flight motions have inspired the development of this algorithm.

Having a balance between global and local search stages is an of important feature that make metaheuristic algorithms desired tools to tackle with various problems [3]. However, due to their stochastic nature, it may not always be feasible to offer such integrity in those algorithms. One way of dealing with this issue is to benefit from the feature of existing algorithms instead of developing new ones. Hybridization is a great choice to do so since it allows the combination of existing and complementary algorithms [11]. In this way, a new structure with balanced feature in terms of exploration and exploitation can be achieved.

In terms of LFD algorithm, it has good explorative behavior due to Lévy Flight motions, however, lacks from exploitative structure. Bearing the above discussion in mind, the LFD algorithm can be improved by adopting another complementary approach so that a novel and more capable hybrid algorithm can be obtained. To achieve such a structure, the Nelder-Mead (NM) simplex search method [12] can be used.

The latter is a well-known simplex search method which is quite capable for local search. In terms of hybridization, several examples are available in the literature which adopts NM algorithm. Some of them are firefly algorithm for optimal reactive power dispatch [13], particle swarm optimization for modeling Li-ion batteries for electric vehicles [14], Harris hawks optimization for solving design and manufacturing problems [15], ant lion optimizer algorithm for structural damage detection [16], moth flame optimization for parameter identification of photovoltaic modules [17] and artificial electric field algorithm for optimization problems [18].

Considering the discussion so far, this paper aims to develop different hybrid versions by modifying LFD algorithm using NM method and investigate the promise of those variations for optimization problems. Therefore, three novel strategies were used to achieve NM modified LFD algorithms. All these strategies are discussed in related section of this paper.

In order to evaluate the performances of the original and the modified LFD versions, well-known benchmark functions of unimodal and multimodal features were adopted. Then, statistical analysis was performed using metrics of average, standard deviation, best and worst. Besides, all approaches, including the original LFD algorithm were ranked. In this way, the ability of the algorithms was assessed for global and local searches. Further assessment took place by performing a nonparametric statistical test known as Wilcoxon's signed-rank test to confirm the capability does not occur by chance. The obtained results have shown that hybridizing LFD algorithm with NM method provides a significant performance improvement in general as was expected. Besides, it has also been found that the efficiency is increased if the NM method is applied after each iteration of LFD algorithm and run each time for the total number of LFD algorithm's current iteration.

Lévy Flight Distribution Algorithm

Wireless sensor networks having a Lévy flight (LF) motions related connection is the main inspiration for LFD algorithm [10]. Mathematically, the LFD algorithm is initialized by calculating the Euclidean distance (E_{dist}) between adjacent nodes which determines the position replacement of sensor nodes. To locate a sensor node, LF is performed. In such a case, the sensor node is placed close to another one with lower number of neighbors or in a position that has no sensor node. The latter behavior increases the effectiveness of the exploration. Two important parameters for generating random walks are the step length and the direction of the walk. To determine the step length (sl), the following equation can be used where β is the Lévy distribution index having limits of $0 < \beta \leq 2$.

$$sl = \frac{U}{|V|^{1/\beta}} \quad (1)$$

The parameters of U and V, given in the above equation, can be determined using (2).

$$U \sim N(0, \sigma_u^2), \qquad V \sim N(0, \sigma_v^2) \tag{2}$$

σ_u and σ_v represent standard deviation and calculated as in (3):

$$\sigma_u = \left(\frac{\zeta(1+\beta) \times \sin(\pi\beta/2)}{\zeta((1+\beta)/2) \times \beta \times 2^{(\beta-1)/2}}\right)^{1/\beta}, \quad \sigma_v = 1 \tag{3}$$

where ζ is a function having the following definition for an integer z.

$$\zeta(z) = \int_0^\infty t^{z-1} e^{-t} dt \tag{4}$$

The E_{dist} value is calculated as in (5) for the locations of adjacent agents (X_i and X_j):

$$E_{dist}(X_i, X_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{5}$$

where X_i and X_j positions are represented by (x_i, y_i) and (x_j, y_j), respectively. A pre-defined threshold value is compared with the value of E_{dist} through iterations. The positions of search agents are re-adjusted using (6) for smaller E_{dist} values than the threshold.

$$X_j(t+1) = L_f\left(X_j(t), X_L, L_b, U_b\right) \tag{6}$$

In the above equation, t is used for the number of iterations whereas L_b and U_b denote the lowest and the highest limits of the search space. L_f function represents the sl value and the LF direction. X_L is used as LF direction since it represents the position of the agent with the lowest number of neighbors. In order to increase the exploration capability, the agent of X_j is moved towards the agent with the lowest number of neighbors using (7).

$$X_j(t+1) = L_b + (U_b - L_b) rd() \tag{7}$$

In the above equation, $rd()$ is used to generate uniformly distributed random numbers R in a range of [0, 1]. The following identification helps discovering the search space with more opportunities:

$$R = rd(), C_{scalar} = 0.5 \tag{8}$$

where C_{scalar} is the comparative scalar value with R in each position update of X_j. The value of R is checked and compared with C_{scalar}. In case of smaller R values than C_{scalar} values (6) is executed whereas (7) is used otherwise. The position of X_i is updated using (9) and (10).

$$X_i(t+1) = T_{pos} + \alpha_1 \times T_{FitN} + rd() \times \alpha_2 \times \left(\left(T_{pos} + \alpha_3 X_L\right)/2 - X_i(t)\right) \tag{9}$$

$$X_i^{new}(t+1) = L_f\left(X_i(t+1), T_{pos}, L_b, U_b\right) \tag{10}$$

X_i position is calculated using (9) whereas (10) provides the final position of X_i. The solution with the best fitness value (target position) is denoted by T_{pos}. The parameters of α_1, α_2 and α_3 are used to represent random numbers of $0 < \alpha_1, \alpha_2, \alpha_3 \leq 10$. The following gives the total target fitness of neighbors (T_{FitN}) around $X_i(t)$ where the neighbor index and neighbor position of $X_i(t)$ are denoted by k and X_k, respectively.

$$T_{FitN} = \sum_{k=1}^{NN} \frac{D(k) \times X_k}{NN} \tag{11}$$

The total number of $X_i(t)$ neighbors is represented by NN. $D(k)$ denotes the fitness degree for each neighbor and given by (12).

$$D(k) = \frac{\delta_1\left(V - Min(V)\right)}{Max(V) - Min(V)} + \delta_2 \tag{12}$$

$$V = \frac{Fitness\left(X_j(t)\right)}{Fitness\left(X_i(t)\right)}, \delta_1 > 0 \; and \; \delta_2 \leq 1 \tag{13}$$

A detailed flowchart of LFD algorithm is demonstrated in Figure 1.

Nelder-Mead Method

This algorithm is a simplex search method and developed to solve nonlinear functions using gradient-free computations [19]. An optimal point of X_1 is determined by generating $p+1$ points of X_1, X_2, … X_{p+1}. Then, the respective fitness function values of $f(X_1)$, $f(X_2)$,…$f\left(X_{p+1}\right)$ are evaluated and sorted in ascending order. Four scalar coefficients of reflection (ρ), expansion (γ) contraction (β) and shrinkage (δ) are used to replace the worst point of X_{p+1}. The computed fitness values allow determination of the best (X_1), the worst (X_{p+1})

and the centroid ($\bar{X}$) points. To identify the reflection, X_{rf}, (14) is used:

$$X_{rf} = \bar{X} + \rho(\bar{X} - X_{p+1}) \quad (14)$$

The reflection point is expanded using (15):

$$X_{ex} = \bar{X} + \gamma(X_{rf} - \bar{X}) \quad (15)$$

where X_{ex} denotes the expansion point and replaces the worst value for $f(X_{ex}) < f(X_{rf})$. Otherwise, this point is replaced by X_{rf}. The contraction step is performed for $f(X_p) \leq f(X_{rf})$. An outer contraction (X_{outc}) is generated using (16) to obtain the fitness value of $f(X_{outc})$ in case of $f(X_{rf}) < f(X_{p+1})$.

$$X_{outc} = \bar{X} + \beta(X_{rf} - \bar{X}) \quad (16)$$

The point of X_{p+1} is replaced by X_{outc}, then the iterations are terminated for $f(X_{outc}) \leq f(X_{rf})$. Otherwise, the shrinkage occurs in the next action. An inner contraction (X_{inc}), provided in (17), may also be constructed in the contraction step to obtain fitness of $f(X_{inc})$ for $f(X_{p+1}) \leq f(X_{rf})$.

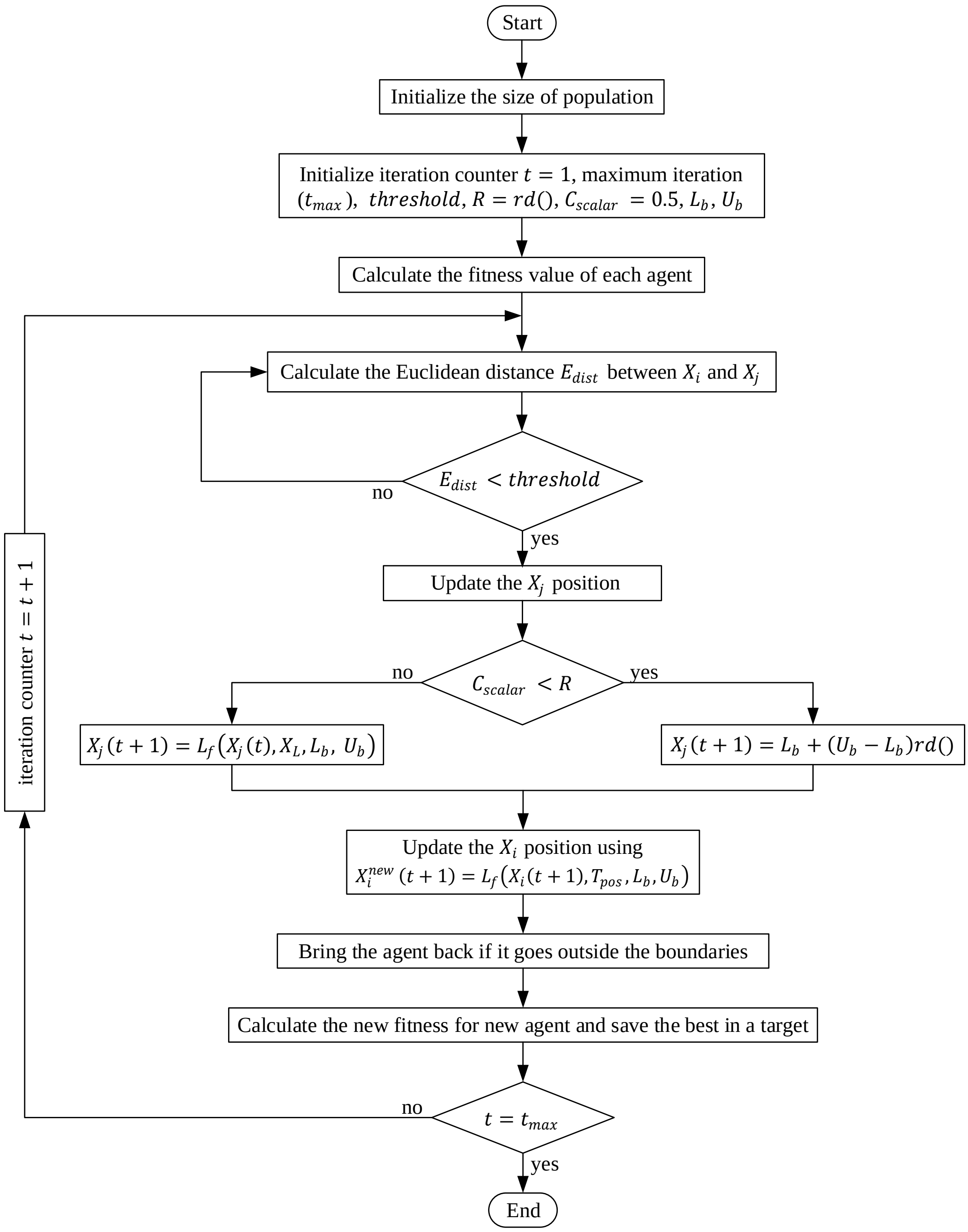

Figure 1. *A detailed flowchart of LFD algorithm*

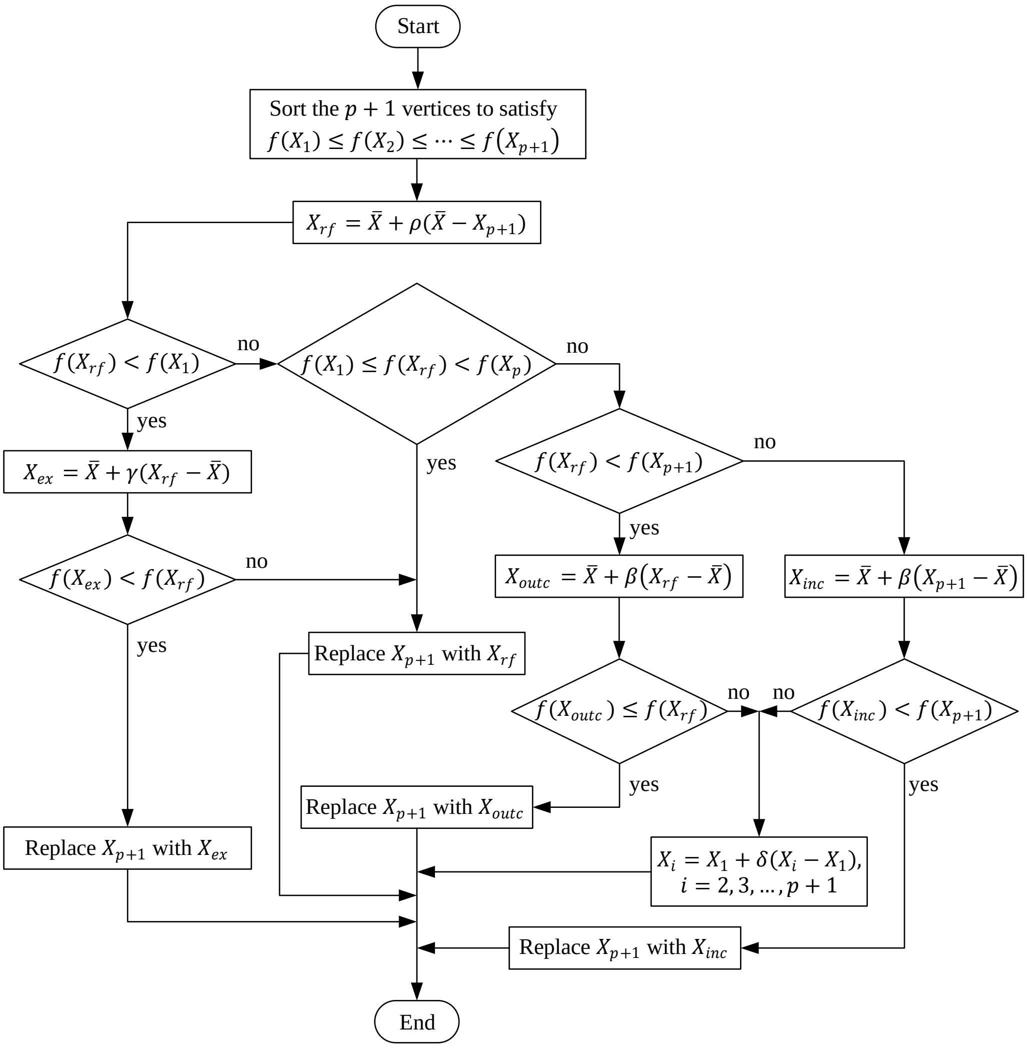

Figure 2. *Flowchart of NM method*

$$X_{inc} = \bar{X} + \beta\left(X_{p+1} - \bar{X}\right) \quad (17)$$

The point of X_{p+1} is replaced by X_{inc}, then the iterations are terminated for $f(X_{inc}) < f\left(X_{p+1}\right)$. Otherwise, the shrinkage occurs. The shrinkage step is the final operation which constructs new points using (18).

$$X_i = X_1 + \delta(X_i - X_1), \quad i = 2, 3, \ldots, p+1 \quad (18)$$

The flowchart of NM simplex method is provided in Figure 2.

Proposed Hybrid Strategies

This section provides information about different approaches to hybridize LFD algorithm with NM simplex search method. In order to improve the performance of the original LFD algorithm, three different strategies were employed to adopt NM method for modifying LFD algorithm. To have a fair assessment, a dimension of 30, maximum iterations of 500 and a population size of 50 were adopted for all approaches.

In the first proposed strategy, LFD algorithm is performed. Then, the NM method is applied after LFD algorithm completes its task entirely. The NM is run twice as much the number of iterations in this strategy which means the NM is performed for 1000 iterations since the chosen number of iterations was 500. This strategy was named as LFDNM-S1.

In the second proposed strategy, unlike the first one, the NM method is applied after each iterations of LFD algorithm instead of waiting for the completion of the latter algorithm.

However, the NM method is performed for $d + 1$ iterations where d is the dimension of the problem. This strategy was named as LFDNM-S2.

The third proposed strategy is the last approach that was adopted for modifying LDF algorithm using NM method. Similar to the second strategy, the NM method is applied after each iterations of LFD algorithm in this strategy, as well. However, the implementation of NM method lasts for l iterations after each iteration of the LFD algorithm where l is the current iteration of the latter algorithm. That means, for example, the NM method would run for 10 iterations if it is implemented after the 10^{th} iteration of LFD algorithm, and run for 11 iterations after 11^{th} iteration of LFD algorithm and so on. The last strategy was named as LFDNM-S3.

Experiments and Discussions

The performance validation of the original and NM modified versions of LFD algorithms together with the parameter settings are presented in this section. The performances of the respective algorithms were tested using well-known four unimodal and four multimodal test functions provided in the following subsection. The algorithms were tested against each other using a set of fixed parameters for the sake of fair comparison. Therefore, a swarm size (search agents) of 50 and maximum iterations of 500 along with dimension (n) of 30 were adopted for all algorithms. Besides, each algorithm was performed on each test function for 30 independent runs.

The parameter values for LFD were chosen to be 2 for threshold, 0.5 for C_{scalar}, 1.5 for β, 10 for α_1, 0.00005 for α_2, and 0.005 for α_3 along with 0.9 for δ_1 and 0.1 for δ_2 [10]. In terms of NM method, the parameter values were chosen to be 1 for ρ, 2 for γ, 0.5 for β and 0.5 for δ [19].

In terms performance evaluation of the algorithms for global and local search abilities, the statistical values of average, standard deviation (Sdev), best and worst were used. Besides, the algorithms were ranked. In addition to those statistical metrics, a nonparametric statistical test known as Wilcoxon's signed-rank test [20] was also performed for further assessment of the algorithms. The adopted statistical metrics of average, Sdev, best and worst can mathematically be defined as given in (19), (20), (21) and (22), respectively.

$$\text{Average} = \frac{\sum_{i=1}^{M}(f_i)}{M} \tag{19}$$

$$\text{Sdev} = \sqrt{\frac{1}{M-1}\sum_{i=1}^{M}(f_i - Average)^2} \tag{20}$$

$$Best = \min_{1 \le i \le M} f_i \tag{21}$$

$$\text{Worst} = \max_{1 \le i \le M} f_i \tag{22}$$

where M is the number of runs and f_i is the function fitness value.

Benchmark Functions

The following benchmark functions listed in Table 1 have been adopted for this study. The related table contains unimodal test functions of Sphere, Schwefel 2.22, Rosenbrock and Step together with multimodal benchmark functions of Schwefel, Rastrigin, Ackley and Griewank. Those are all well-known test functions with different properties and present a good environment for performance evaluation of the algorithm such that the exploitation and the exploration capabilities of the algorithm can be assessed using unimodal and multimodal functions, respectively [21].

For example, the unimodal benchmark functions have one global optimum with no local optima and are good for assessment of exploitation ability of the algorithms. On the other hand, the multimodal benchmark functions have considerable number of local optima which make them good for assessing the exploration capability of the algorithms. The properties of both unimodal (Sphere, Schwefel 2.22, Rosenbrock, Step) and multimodal (Schwefel, Rastrigin, Ackley, Griewank) benchmark functions can also be seen visually as demonstrated in Figure 3. The performance of the proposed NM modified LFD algorithms together with the original LFD algorithm was tested against each other using those benchmark functions.

In terms of implementation of the algorithms on these benchmark functions, the related search domains listed in Table 1 for the respective test functions along with a dimension of 30 for each function were adopted. Then the population size, and number of iterations were defined, and the related algorithms were tested against those test functions in terms of statistical performance. The results were then compared with each other. Figure 4 shows the implementation steps in brief.

Table 1. *Unimodal and multimodal test functions*

Name	Description	n	Search domain	Optimum
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100, 100]$	0
Schwefel 2.22	$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-10, 10]$	0
Rosenbrock	$f_3(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	30	$[-30, 30]$	0
Step	$f_4(x) = \sum_{i=1}^{n} (x_i + 0.5)^2$	30	$[-100, 100]$	0
Schwefel	$f_5(x) = -\sum_{i=1}^{n} \left(x_i \sin(\sqrt{\lvert x_i \rvert})\right)$	30	$[-500, 500]$	$-418.9829 \times n$
Rastrigin	$f_6(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]$	0
Ackley	$f_7(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	$[-32, 32]$	0
Griewank	$f_8(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]$	0

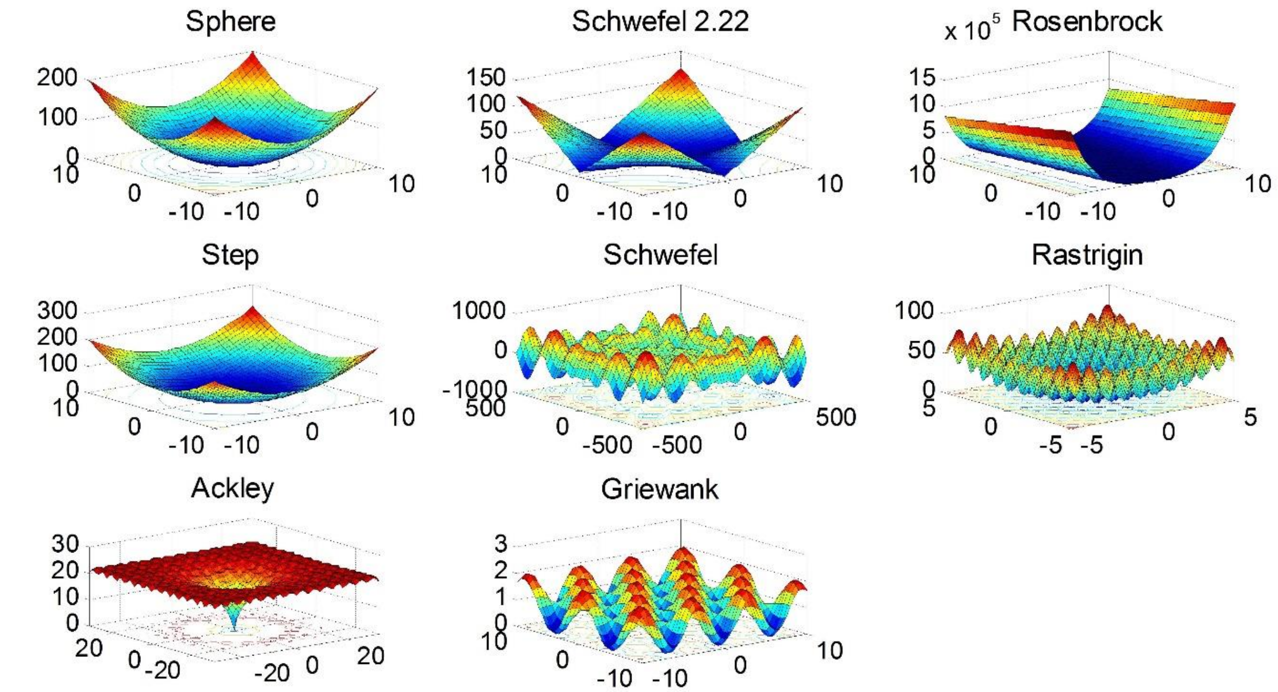

Figure 3. *Surface plots of the two-variable benchmark functions used in experiment*

Exploitation Capability

As mentioned in the previous subsection, the unimodal functions ($f_1(x)$, $f_2(x)$, $f_3(x)$, $f_4(x)$) provided in Table 2 can help assessing the local search capability of the algorithms under consideration [22]. It can easily be spotted that the average values for all test functions obtained by the LFDNM-S3 algorithm (shown in bold) are well below the other values achieved by the other algorithms. In addition, the LFDNM-S3 algorithm has achieved better values in terms of other statistical metrics. Besides, the constructed LFDNM-S3 algorithm has also been ranked the first, as well. The obtained results clearly show the third proposed strategy for NM modified LFD algorithm has a strong competitiveness in terms of exploitation.

Exploration Capability

In terms of assessment of global search capability, the multimodal functions ($f_5(x)$,

$f_6(x)$, $f_7(x)$, $f_8(x)$) provided in Table 3 can be used [22]. Similar to local search ability, the LFDNM-S3 algorithm has also achieved better results than the other algorithms in terms of all statistical metrics. Besides, the constructed LFDNM-S3 algorithm has also been ranked the first for multimodal functions, as well. The obtained results clearly show the LFDNM-S3 algorithm's strong competitiveness in terms of exploration, as well. Considering both global and local search capabilities, it can be concluded that the third proposed strategy for improving LFD algorithm through NM method has a good balance in terms of exploration and exploitation.

Table 2. *Results of the unimodal functions*

Function	Measure	Basic LFD	LFDNM-S1	LFDNM-S2	LFDNM-S3
$f_1(x)$	Average	1.5220E−07	3.6282E−08	1.1919E−08	**1.4697E−42**
	Sdev	5.4715E−08	1.2058E−08	9.0745E−09	3.1618E−42
	Best	8.0419E−08	1.9756E−08	1.5254E−09	1.7217E−44
	Worst	2.3698E−07	5.7184E−08	3.9174E−08	1.2261E−41
	Rank	4	3	2	1
$f_2(x)$	Average	3.0625E−04	1.3472E−04	3.7902E−05	**2.0705E−27**
	Sdev	5.9240E−05	2.7256E−05	8.9938E−06	2.3826E−27
	Best	1.8925E−04	9.9580E−05	2.2527E−05	4.7253E−29
	Worst	3.8394E−04	2.0202E−04	6.0360E−05	9.0123E−27
	Rank	4	3	2	1
$f_3(x)$	Average	27.8977	27.0939	23.6612	**0.4302**
	Sdev	0.1187	0.2811	0.1142	0.1802
	Best	27.7517	26.5653	23.4629	0.2515
	Worst	28.0630	27.6308	23.8389	0.9410
	Rank	4	3	2	1
$f_4(x)$	Average	1.1480	0.5076	2.3949E−06	**1.7721E−07**
	Sdev	0.2349	0.2549	8.0920E−07	4.1048E−08
	Best	0.5739	0.1592	6.8912E−07	5.6266E−08
	Worst	1.4370	0.9830	4.3665E−06	2.3634E−07
	Rank	4	3	2	1

Table 3. *Results of the multimodal functions*

Function	Measure	Basic LFD	LFDNM-S1	LFDNM-S2	LFDNM-S3
$f_5(x)$	Average	−4.3960E+03	−7.8148E+03	−7.6089E+03	**−8.2171E+03**
	Sdev	283.6078	391.0008	661.1682	772.6186
	Best	−4.8243E+03	−8.3413E+03	−8.9003E+03	−9.8013E+03
	Worst	−3.8051E+03	−7.0629E+03	−6.5018E+03	−7.4562E+03
	Rank	4	2	3	1
$f_6(x)$	Average	2.8745	0.4463	0.0665	**2.2578E−11**
	Sdev	3.8130	0.5498	0.1319	8.6173E−12
	Best	1.5259E−05	2.9721E−06	2.0053E−06	1.2108E−11
	Worst	13.2109	1.6631	0.5054	4.3315E−11
	Rank	4	3	2	1
$f_7(x)$	Average	8.8859E−05	4.4928E−05	2.9489E−05	**2.1202E−12**
	Sdev	1.1628E−05	9.3227E−06	8.5630E−06	5.2423E−13
	Best	7.0401E−05	3.5371E−05	8.3122E−06	1.4611E−12
	Worst	1.0701E−04	7.3164E−05	4.2522E−05	3.0349E−12
	Rank	4	3	2	1
$f_8(x)$	Average	3.7917E−07	1.0317E−07	4.5251E−08	**5.8538E−14**
	Sdev	1.2776E−07	5.0798E−08	2.4069E−08	1.8298E−14
	Best	1.8918E−07	4.3454E−08	1.4857E−08	4.1078E−14
	Worst	6.7854E−07	2.0082E−07	9.5037E−08	1.0003E−13
	Rank	4	3	2	1

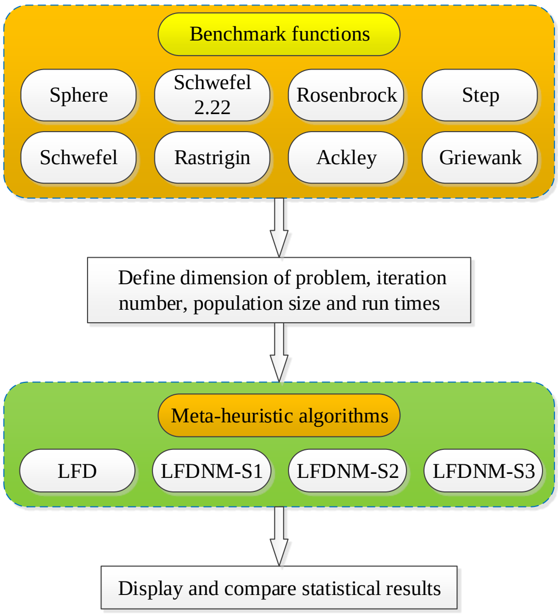

Figure 4. *Optimization stage for the benchmark functions*

The Wilcoxon's Test

The Wilcoxon's signed-rank test was used in this study as a non-parametric statistical test which was performed to have a meaningful conclusion for the performances of the proposed hybrid strategies since this test helps evaluating the significance level between different algorithms [23]. The respective p values of the algorithms can be obtained along with the calculations of $T+$ and $T-$ related to comparisons between two algorithms using Wilcoxon's signed-rank test. Table 4 demonstrates the respective test results on 8 benchmark functions in 30 runs for the original and the proposed NM modified LFD algorithms. In the table, the sign of '$+$' indicates statistically significant difference, thus, better performance of the algorithm and the sign of '$-$' indicates vice versa. In case of no statistically significant difference between the compared algorithms, the sign of '$=$' is used.

Evaluating the demonstrated results, the LFDNM-S3 algorithm can clearly be seen to be significantly superior to the original LFD and LFDNM-S2 algorithms for all test functions. The comparison between LFDNM-S3 and the LFDNM-S1 algorithms shows no significant difference only for $f_5(x)$, however, for the rest of the functions LFDNM-S3 has clear superiority.

Table 4. *Wilcoxon signed-rank test results for LFDNM-S3 vs LFD, LFDNM-S1 and LFDNM-S2*

Function	**LFDNM-S3 vs. LFD**				**LFDNM-S3 vs. LFDNM-S1**				**LFDNM-S3 vs. LFDNM-S2**			
	***p* value**	$T+$	$T-$	W	***p* value**	$T+$	$T-$	W	***p* value**	$T+$	$T-$	W
$f_1(x)$	1.7181E−06	465	0	+	1.7181E−06	465	0	+	1.7181E−06	465	0	+
$f_2(x)$	1.7181E−06	465	0	+	1.7094E−06	465	0	+	1.7181E−06	465	0	+
$f_3(x)$	1.7181E−06	465	0	+	1.7181E−06	465	0	+	1.7181E−06	465	0	+
$f_4(x)$	1.7181E−06	465	0	+	1.7181E−06	465	0	+	1.7181E−06	465	0	+
$f_5(x)$	1.7344E−06	465	0	+	0.2208	292	173	=	3.0481E−04	408	57	+
$f_6(x)$	1.7181E−06	465	0	+	1.7181E−06	465	0	+	1.7181E−06	465	0	+
$f_7(x)$	1.7181E−06	465	0	+	1.7181E−06	465	0	+	1.7181E−06	465	0	+
$f_8(x)$	1.7181E−06	465	0	+	1.7181E−06	465	0	+	1.7181E−06	465	0	+

Conclusion

In this work, novel approaches for improving the capability of the original LFD algorithm were proposed and discussed through modifications using NM method. The NM method was implemented after LFD algorithm in each of the strategies, however, by following different patterns. In the first approach, NM was run twice as much the number of iterations of LFD algorithm after the LFD algorithm finishes its task. In the second approach, NM was run after each iterations of LFD algorithm instantly whereas for the third approach, NM was run for the total number of current iterations after each iteration of the LFD algorithm. Four unimodal and four multimodal test functions were used to observe the performance of the algorithms through statistical and nonparametric statistical analyses. The obtained results have shown the modified versions of LFD algorithm can provide better capability in general. In addition, the efficiency of the third approach was found to be better for NM modified LFD algorithm since it has demonstrated a greater balance between exploration and exploitation phases. Therefore, the latter can be used as an effective tool for optimization problems. Bearing the obtained results in mind, the constructed algorithms have the potential to be used for several different real-life optimization problems for future works. Some of them can be listed as controlling an automatic voltage regulator system, regulating the speed of a direct current motor, and operating a magnetic levitation system.

References

[1] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, "The Arithmetic Optimization Algorithm," *Comput. Methods Appl. Mech. Eng.*, vol. 376, p. 113609, Apr. 2021, doi: 10.1016/j.cma.2020.113609.

[2] W. Zhao, L. Wang, and Z. Zhang, "Supply-Demand-Based Optimization: A Novel Economics-Inspired Algorithm for Global Optimization," *IEEE Access*, vol. 7, pp. 73182–73206, 2019, doi: 10.1109/ACCESS.2019.2918753.

[3] D. Izci, S. Ekinci, E. Eker, and M. Kayri, "Improved Manta Ray Foraging Optimization Using Opposition-based Learning for Optimization Problems," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Jun. 2020, pp. 1–6, doi: 10.1109/HORA49412.2020.9152925.

[4] A. F. Nematollahi, A. Rahiminejad, and B. Vahidi, "A novel meta-heuristic optimization method based on golden ratio in nature," *Soft Comput.*, vol. 24, no. 2, pp. 1117–1151, 2020, doi: 10.1007/s00500-019-03949-w.

[5] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, "Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems," *Appl. Intell.*, 2020, doi: 10.1007/s10489-020-01893-z.

[6] E. Eker, M. Kayri, S. Ekinci, and D. Izci, "A New Fusion of ASO with SA Algorithm and Its Applications to MLP Training and DC Motor Speed Control," *Arab. J. Sci. Eng.*, Feb. 2021, doi: 10.1007/s13369-020-05228-5.

[7] W. Zhao, L. Wang, and Z. Zhang, "Artificial ecosystem-based optimization: a novel nature-inspired meta-heuristic algorithm," *Neural Comput. Appl.*, pp. 1–43, 2019.

[8] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, 1997, doi: 10.1109/4235.585893.

[9] D. Izci and S. Ekinci, "Comparative Performance Analysis of Slime Mould Algorithm For Efficient Design of Proportional–Integral–Derivative Controller," *Electrica*, vol. 21, no. 1, pp. 151–159, Jan. 2021, doi: 10.5152/electrica.2021.20077.

[10] E. H. Houssein, M. R. Saad, F. A. Hashim, H. Shaban, and M. Hassaballah, "Lévy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 94, p. 103731, 2020, doi: 10.1016/j.engappai.2020.103731.

[11] S. Ekinci, D. Izci, and B. Hekimoğlu, "Optimal FOPID Speed Control of DC Motor via Opposition-Based Hybrid Manta Ray Foraging Optimization and Simulated Annealing Algorithm," *Arab. J. Sci. Eng.*, vol. 46, no. 2, pp. 1395–1409, Feb. 2021, doi: 10.1007/s13369-020-05050-z.

[12] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *Comput. J.*, vol. 7, no. 4, pp. 308–313, Jan. 1965, doi: 10.1093/comjnl/7.4.308.

[13] A. Rajan and T. Malakar, "Optimal reactive power dispatch using hybrid Nelder-Mead simplex based firefly algorithm," *Int. J. Electr. Power Energy Syst.*, vol. 66, pp. 9–24, 2015, doi: 10.1016/j.ijepes.2014.10.041.

[14] T. Mesbahi, F. Khenfri, N. Rizoug, K. Chaaban, P. Bartholomeüs, and P. Le Moigne, "Dynamical modeling of Li-ion batteries for electric vehicle applications based on hybrid Particle Swarm–Nelder–Mead (PSO–NM) optimization algorithm," *Electr. Power Syst. Res.*, vol. 131, pp. 195–204, 2016, doi: https://doi.org/10.1016/j.epsr.2015.10.018.

[15] A. R. Yıldız, B. S. Yıldız, S. M. Sait, S. Bureerat, and N. Pholdee, "A new hybrid Harris hawks-Nelder-Mead optimization algorithm for solving design and manufacturing problems," *Mater. Test.*, vol. 61, no. 8, pp. 735–743, Aug. 2019, doi: 10.3139/120.111378.

[16] C. Chen and L. Yu, "A hybrid ant lion optimizer with improved Nelder–Mead algorithm for structural damage detection by improving weighted trace lasso regularization," *Adv. Struct. Eng.*, vol. 23, no. 3, pp. 468–484, Sep. 2020, doi: 10.1177/1369433219872434.

[17] H. Zhang, A. A. Heidari, M. Wang, L. Zhang, H. Chen, and C. Li, "Orthogonal Nelder-Mead moth flame method for parameters identification of photovoltaic modules," *Energy Convers. Manag.*, vol. 211, p. 112764, 2020, doi: https://doi.org/10.1016/j.enconman.2020.112764.

[18] D. Izci, S. Ekinci, S. Orenc, and A. Demiroren, "Improved Artificial Electric Field Algorithm Using Nelder-Mead Simplex Method for Optimization Problems," in *2020 4th International Symposium on Multidisciplinary Studies and Innovative*

Technologies (ISMSIT), Oct. 2020, pp. 1–5, doi: 10.1109/ISMSIT50672.2020.9255255.

[19] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder-Mead simplex method in low dimensions," *SIAM J. Optim.*, vol. 9, no. 1, pp. 112–147, 1998, doi: 10.1137/S1052623496303470.

[20] W. Zhao, L. Wang, and Z. Zhang, "Atom search optimization and its application to solve a hydrogeologic parameter estimation problem," *Knowledge-Based Syst.*, vol. 163, pp. 283–304, 2019, doi: 10.1016/j.knosys.2018.08.030.

[21] M. Jamil and X. S. Yang, "A literature survey of benchmark functions for global optimisation problems," *Int. J. Math. Model. Numer. Optim.*, vol. 4, no. 2, pp. 150–194, 2013, doi: 10.1504/IJMMNO.2013.055204.

[22] W. Zhao, Z. Zhang, and L. Wang, "Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications," *Eng. Appl. Artif. Intell.*, vol. 87, p. 103300, Jan. 2020, doi: 10.1016/j.engappai.2019.103300.

[23] N. Zhang, C. Li, R. Li, X. Lai, and Y. Zhang, "A mixed-strategy based gravitational search algorithm for parameter identification of hydraulic turbine governing system," *Knowledge-Based Syst.*, vol. 109, pp. 218–237, 2016, doi: 10.1016/j.knosys.2016.07.005.