We propose a new meta-heuristic algorithm named Levy flights multi-verse optimizer (LFMVO), which incorporates Levy flights into multi-verse optimizer (MVO) algorithm to solve numerical and engineering optimization problems. The Original MVO easily falls into stagnation when wormholes stochastically re-span a number of universes (solutions) around the best universe achieved over the course of iterations. Since Levy flights are superior in exploring unknown, large-scale search space, they are integrated into the previous best universe to force MVO out of stagnation. We test this method on three sets of 23 well-known benchmark test functions and an NP complete problem of test scheduling for Network-on-Chip (NoC). Experimental results prove that the proposed LFMVO is more competitive than its peers in both the quality of the resulting solutions and convergence speed.

Introduction

Optimization problems are common in the field of science and technology [1]. These problems are often nonlinear, multimodal or discontinuous, and very challenging to solve with traditional optimization methods. In the past few years, a number of meta-heuristic algorithms have been successfully developed to solve these problems. These techniques are inspired by natural phenomena or other sources using iterations and stochasticity to generate better solutions for optimization problems [2, 3]. Some popular meta-heuristic algorithms are the Genetic Algorithm (GA) [4], Ant Colony Optimization (ACO) [5], Particle Swarm Optimization (PSO) [6], Differential Evolution (DE) [7], Harmony Search (HS) [8], Artificial Bee Colony(ABC) [9], Cuckoo Search (CS) [10], Gravitational Search Algorithm (GSA) [11], Fruit Fly Optimization algorithm (FOA) [12], Gases Brownian Motion Optimization (GBMO) [13], Symbiotic Organisms Search (SOS) [14], and Moth-flame Optimization (MFO) [15].

Multi-verse optimizer (MVO) is a promising and up-to-date optimization algorithm proposed by [16]. As the name implies, it is inspired by the theory of multi-verse in physics. The three main concepts of the multi-verse theory (white hole, black hole, and wormhole) are the basis for the MVO algorithm. The concepts of white hole and black hole were utilized to explore search spaces by MVO. The wormholes help MVO exploit the search spaces. The
MVO algorithm was first evaluated by 19 challenging test benchmarks. To further evaluate its performance, the MVO was adopted for five practical engineering problems. The experimental results prove that the proposed algorithm can produce very competitive results and outperform other algorithms described in the literature.

However, there are still some issues associated with this algorithm. When wormholes stochastically re-span a number of universes around the best universe achieved over the course of iterations, the MVO is likely to get trapped in the local optima.

To improve the global search ability of MVO and enhance the ability to escape from local optima, MVO is combined with Levy flights (LFMVO) in this paper. Levy flights, proposed by Paul Levy in 1937, are a type of random walk of generalized Brownian motion that include non-Gaussian randomly distributed step sizes for the distance moved. The Levy distribution features long tails, an infinite second order moment and convergence to a non-Gaussian stable distribution [17]. A number of natural and artificial events can be described by Levy flights, e.g., fluid dynamics, earthquake analysis, cooling behavior, diffusion of fluorescent molecules, noise and foraging paths (albatross, bumblebees, deer etc.) [18, 19].

Recently, Levy flights were added to nature-inspired algorithms to enhance their performance [18, 20–24]. In [20], Levy flights were adopted to generate new solutions (new cuckoo) in the cuckoo search. Since its step length is ultimately much longer, the strategy is more efficient for exploring the search space. In [21], the Levy-flight firefly algorithm (LFA) is introduced, which blends Levy flights with the search strategy to enhance the randomization of the firefly algorithm (FA). In [22], a Levy mutation is used in evolutionary algorithms because it is likely to create a new solution that is farther away from its parent solution than Gaussian mutation. Candela et al. [23] used Levy flights as a means to diversify ant colony optimization.

Haklı et al. [18] presented a novel particle swarm optimization algorithm using Levy flight (LFPSO) in which a more efficient search occurs in the search space due to the long jumps executed by the particles. Therefore, the LFPSO is likely to avoid premature convergence and to improve the global search capability.

In this paper, the proposed method performs global search more effectively with random walks. If the universes cannot improve self-solutions, they are re-formed with Levy flights such that the best universe obtained so far is affected and being trapped in local optima is prevented. Experimental results with test benchmarks and test scheduling for NoC show the superiority of the LFMVO compared with the MVO algorithm and other algorithms.

This paper is organized as follows. The next section gives a brief overview of MVO. The following section presents a brief overview of Levy flights. The improved MVO algorithm called LFMVO algorithm is presented and analyzed in the LFMVO section. A comprehensive set of experimental results is provided in the Results section. An NP complete problem of test scheduling for NoC is presented in the Application section. Finally, the conclusions are drawn in last section.

**Brief overview of a multi-verse optimizer**

**Multi-verse theory**

Multi-verse theory is a new and well-known theory in physics. This theory implies the existence of universes beyond the one in which we live [25].

**Multi-verse optimizer**

The concepts of white hole and black hole were utilized to explore search spaces by MVO. Wormholes help MVO exploit the search spaces. In MVO, a solution corresponds to a universe, a variable in the solution corresponds to an object in the universe, the inflation rate of a solution corresponds to the fitness of the solution, and the term time corresponds to the iteration.
A universe with a higher inflation rate is highly probable to have white holes and tend to send objects through white holes, whereas a universe with a lower inflation rate is highly probable to have black holes and tends to receive objects through black holes. The white/black hole tunnels are used to exchange objects between different universes. Despite the inflation rate, objects in all universes have a high probability of shifting to the best universe via wormholes.

A roulette wheel selection (RWS) is adopted to mathematically model the exchange of objects between universes and the white/black hole tunnels. At each iteration, one of the universes is chosen by the RWS to have a white hole based on its inflation rate.

Assume that

\[
U = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\
  x_{21} & x_{22} & \cdots & x_{2d} \\
  \vdots & \vdots & \ddots & \vdots \\
  x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix}
\]

(1)

where \( d \) indicates the number of parameters (variables) and \( n \) denotes the number of universes (candidate solutions):

\[
x_i \begin{cases} x_{ij} & r1 < NI(U_i) \\
  x_{ij} & r1 \geq NI(U_i) \end{cases}
\]

(2)

where \( x_{ij} \) expresses the \( j \)th parameter of the \( i \)th universe, \( x_{ij} \) expresses the \( j \)th parameter of the \( k \)th universe selected by an RWS, \( r1 \in [0, 1] \) denotes a random number, \( U_i \) denotes the \( i \)th universe, and \( NI(U_i) \) denotes a normalized inflation rate of the \( i \)th universe.

The wormhole tunnels are always built between a universe and the best universe constituted so far to provide local changes for each universe and the high probability of refining the inflation rate via wormholes as follows:

\[
x_i \begin{cases} X_i + TDR \times ((ub_j - lb_j) \times r4 + lb_j) & r3 < 0.5 \\
  X_i - TDR \times ((ub_j - lb_j) \times r4 + lb_j) & r3 \geq 0.5 \\
  x_i \end{cases} \quad r2 < WEP \quad r2 \geq WEP
\]

(3)

where \( X_i \) is the \( j \)th parameter of the best universe constituted so far; travelling distance rate (TDR) and wormhole existence probability (WEP) are coefficient; \( ub_j \) and \( lb_j \) are the upper bound and the lower bound of \( j \)th variable, respectively; \( x_i \) denotes the \( j \)th parameter of the \( i \)th universe; and \( r2, r3 \) and \( r4 \) are random numbers in \([0, 1]\).

WEP is defined as the existence probability of wormholes in universes. To enhance exploitation during the progress of the optimization process, it increases linearly over the iterations.

\[
WEP = W_{min} + I \times \left( \frac{W_{max} - W_{min}}{L} \right)
\]

(4)

where \( W_{min} \) indicates the minimum (commonly set to 0.2), \( W_{max} \) indicates the maximum (commonly set to 1), \( I \) is the current iteration, and \( L \) is the maximum iteration.

TDR is defined as the distance rate by which an object can be teleported by a wormhole around the best universe obtained so far. To gain more precise exploitation/local search
around the best universe, TDR is increased over the iterations.

\[ TDR = 1 - \frac{p^{1/p}}{L^{1/p}} \]  

(5)

where \( p \) (set to 6 in this paper) indicates the exploitation accuracy over the iterations.

The general steps of the MVO algorithm are described as follows. The optimization process starts by creating a set of random universes. At each iteration, objects in the universes with high inflation rates incline to shift to the universes with low inflation rates through white/black holes. Simultaneously, objects in each universe have the chance to randomly teleport to the best universe via wormholes. This process continues until it is terminated by satisfying an end criterion (e.g., maximum iterations).

**Brief overview of Levy flights**

In general terms, Levy flights are a random walk whose step length obeys the Levy distribution. The Levy distribution is often in accordance with a simple power-law formula \( L(s) \sim |s|^{-1-\beta} \), where \( 0 < \beta \leq 2 \) is an index. Mathematically a simple version of the Levy distribution can be described as [18, 20]

\[
L(s, \gamma, \mu) = \begin{cases} 
\sqrt{\gamma} \exp \left( -\frac{\gamma}{2(s-\mu)} \right) \frac{1}{(s-\mu)^{3/2}}, & 0 < \mu < s < \infty \\
0, & s \leq 0
\end{cases}
\]  

(6)

where \( \mu \) denotes a location or shift parameter and \( \gamma > 0 \) denotes a scale parameter.

According to Fourier transform, a Levy distribution can be defined

\[ F(k) = \exp[-z|k|^\beta], \quad 0 < \beta \leq 2 \]  

(7)

where \( \alpha \) indicates skewness or scale factor and \( \beta \) indicates Levy index. The inverse of this integral does not have an analytical form for the general \( \beta \) except for a few special cases.

For the case of \( \beta = 2 \), we have

\[ F(k) = \exp[-z|k|^2] \]  

(8)

whose inverse Fourier transform corresponds to a Gaussian distribution.

For the case of \( \beta = 1 \), we have

\[ F(k) = \exp[-z|k|] \]  

(9)

which corresponds to a Cauchy distribution

\[ p(x, \gamma, \mu) = \frac{1}{\pi \gamma^2 + (x-\mu)^2} \]  

(10)

where \( \mu \) is the location parameter and \( \gamma \) is the scale parameter.

For the general case, the inverse integral

\[ L(s) = \frac{1}{\pi} \int_0^{\infty} \cos(ks)\exp[-z|k|^\beta]dk \]  

(11)

can be evaluated only when \( s \) is large enough. We have

\[ L(s) \to \frac{z\beta \Gamma(\beta)\sin(\pi\beta/2)}{\pi|s|^{1+\beta}}, \quad s \to \infty \]  

(12)
where \( \Gamma(z) \) expresses the Gamma function
\[
\Gamma(z) = \int_0^\infty m^{z-1} e^{-m} dm
\]  

When \( z = n \) is an integer, we have
\[
\Gamma(n) = (n-1)!
\]  

For exploring unknown, large-scale search spaces, Levy flights are superior to Brownian random walks [20, 26].

**The proposed LFMVO algorithm**

In the original MVO algorithm, when wormholes stochastically re-span a number of the solution universes around the best solution achieved over the course of iterations, the MVO is likely to get trapped in the local optima.

If the universes cannot improve self-solutions, they are re-formed with Levy flights such that the best universe obtained so far is affected and being trapped in local optima is prevented.

In the proposed method, when generating new solutions \( U_{i+1} \) (for universe \( i \)), a Levy flight is executed
\[
U_{i+1} = U_i + K \times (Lb + (Ub - Lb) \text{Levy}(x)) \times U_i
\]  

where \( K \) is the Levy weight that controls the impact of the previous universe on the current universe, \( Lb \) is the lower bound of the feasible region, and \( Ub \) is the upper bound of the feasible region. It should be noted that a larger Levy weight inclines to facilitate a global search, while a smaller Levy weight to facilitate a local search. Therefore, the Levy weight \( K \) is crucial to the convergence behavior of MVO. A suitable value for the Levy weight usually provides a balance between global exploration and local exploitation and results in refined solutions. To achieve a trade-off between exploration and exploitation and to accelerate convergence speed, we proposed a Levy weight that linearly decreases over the course of iterations. In the early stages, a relatively large Levy weight is adopted to coarse-tune the whole search area. At the end stages, a relatively small Levy weight is adopted to fine-tune the current search area. This adaptive Levy weight factor (ALWF) is determined as follows.
\[
K = (\text{Max}_\text{Iter} - t) / \text{Max}_\text{Iter}
\]  

where \( \text{Max}_\text{Iter} \) is the maximum iterations, \( t \) is the current iteration.

It is not trivial to generate step size \( s \) samples using Levy flights [27]. There are several approaches to achieve step size samples, but the direct and efficient approach is to adopt the Mantegna algorithm [28]. In Mantegna’s algorithm, the step size \( s \) can be described by
\[
s = \frac{u}{|v|^b}
\]  

where \( u \) and \( v \) are drawn from normal distributions. That is
\[
u \sim N(0, \sigma^2), \quad v = N(0, \sigma_v^2)
\]  

with
\[
\sigma = \left( \frac{\Gamma(1 + \beta) \times \sin(\frac{\pi \beta}{2})}{\Gamma(\frac{1 + \beta}{2}) \times \beta \times 2^{\frac{\beta - 1}{2}}} \right)^{\frac{1}{2}}, \quad \sigma_v = 1
\]
Thus, a simple scheme can be depicted as

\[
\text{Levy}(x) = 0.01 \times \frac{u \times \sigma}{|w|^\beta}
\]

(20)

where \( \beta \) is a constant (= 1.5) and \( \sigma \) is measured in Eq (19).

Based on the above, the pseudo code of the LFMVO is shown in Algorithm 1.

Algorithm 1: LFMVO algorithm
Input: NI (objective function)
\( d \) (number of variables)
\( n \) (number of universes)
\( \text{Lb} = [\text{Lb}_1, \text{Lb}_2, \ldots, \text{Lb}_d] \) (the lower bound of feasible region)
\( \text{Ub} = [\text{Ub}_1, \text{Ub}_2, \ldots, \text{Ub}_d] \) (the upper bound of feasible region)
Max_Iter (maximum number of iterations)
Output: The optimal objective function value \( NI(BU) \) and the optimal solution \( BU \).

Step 1: Initialization
Create random universes \( U \) using Eq (21)
Initialize WER, TDR, and \( BU \)
\( t = 0 \)

Step 2: Sorting and Normalization
\( SU = \text{Sorted universes} \)
\( NI = \text{Normalize the inflation rate of the universes} \)

Step 3: Iteration
while \( t < \text{Max}_\text{Iter} \)
Evaluate the \( NI(U_i), i = 1, 2, \ldots, n \)
for each universe \( U_i \)
Update WEP and TDR using Eq (4) and Eq (5)
\( BHI = i \)
Update \( U \) using Eq (15)
for each object \( x_{ij} \)
\( r_1 = \text{rand}(0, 1) \)
if \( r_1 < NI(U_i) \)
\( WHI = \text{RWS(}-NI) \)
\( U(BHI, j) = SU(WHI, j) \)
endif
\( r_2 = \text{rand}(0, 1) \)
if \( r_2 < \text{WEP} \)
\( r_3 = \text{rand}(0, 1) \)
\( r_4 = \text{rand}(0, 1) \)
if \( r_4 < 0.5 \)
\( x_{ij} = BU(j) + TDR \times ((UB(j) - LB(j)) \times r_4 + LB(j)) \)
else
\( x_{ij} = BU(j) - TDR \times ((UB(j) - LB(j)) \times r_4 + LB(j)) \)
endif
endif
end for
end for
\( t = t + 1 \)
end while

Step 4: Termination
Output \( BU \) and \( NI(BU) \)

In step 1, the universes are randomly generated in a feasible region using Eq (23) for a given optimization problem. Let \( UP \) represent the universe population, which can be denoted as follows:

\[
UP = \{U_1, U_2, \ldots, U_i, \ldots, U_n\}
\]

(21)
where \( n \) is the number of universes and \( i = 1, 2, \ldots, n \). Each universe \( U_i \) can be expressed as

\[
U_i = (x_{i1}, x_{i2}, \ldots, x_{id})
\]

where \( d \) is the number of variables and \( j = 1, 2, \ldots, d \).

\[
x_{ij} = Lb_j + (Ub_j - Lb_j) \times \text{rand}(0, 1)
\]

where \( Lb_j \) is the lower bound of the \( j \)th variable, \( Ub_j \) is the upper bound of the \( j \)th variable, and \( \text{rand}(0, 1) \) represents a random number in \([0, 1] \).

In step 2, we sort the universe population into a non-decreasing order and normalize the inflation rate (fitness) of the universes.

Step 3 is the process of iterative optimization. First, we evaluate the fitness of all universes \( Ni(U_i) \) using Eq (13). Then, for each universe \( U_i \), we update WEP and TDR using Eq (4) and Eq (5), respectively. Next, we record the black hole index BHI and update the universes \( U \) using Eq (15). After that, we update each object \( x_{ij} \) of the universes using Eq (2) and Eq (3).

In step 4, when the end criterion is satisfied, the optimal objective function value \( Ni(BU) \) and the optimal solution \( BU \) are obtained.

**Experimental Results and Discussion**

To evaluate the performance of the proposed LFMVO algorithm, 23 standard benchmark functions are employed. These functions are well-known and have been widely adopted by many researchers. The functions are shown in Table 1, where \( d \) is the dimension of the function and \( f_{\text{min}} \) represents the optimum value of the function. The optimum values of functions \( f_1-f_{13} \) are zero except for \( f_8 \) which has an optimum value of \(-418.9829\). All the functions \( f_{14}-f_{23} \) have nonzero optimum values. The benchmark functions can be divided into three groups: unimodal benchmark functions (\( f_1-f_7 \)), multi-modal benchmark functions (\( f_8-f_{13} \)), and fixed-dimension multimodal benchmark functions (\( f_{14}-f_{23} \)). The unimodal benchmark functions have one global optimum. However, the multi-modal test functions have a global optimum, and the number of local optima increases exponentially with the dimensions. The fixed-dimension multimodal benchmark functions have only a few local optima.

We set the dimension of the test functions (\( f_1-f_{13} \)) to 40. To have a fair comparison, all algorithms have the same population size (set to 60) and the same maximum number of iterations (set to 600). We run each algorithm 40 times so that we can execute significant statistical analysis (e.g., best, mean and standard deviation). The parameter settings of the algorithms, which are commonly used in the literature, are provided in Table 2. For verification of the results, we compare the LFMVO algorithm with MVO, PSO and MFO, as shown in Tables 3–5.

**Results analysis of unimodal test functions**

Since a unimodal benchmark function has one global optimum, it is suitable for benchmarking the convergence rates (exploitation) of algorithms. Table 3 lists the results of the benchmark functions \( f_1-f_7 \) for different algorithms. First, we rank the algorithm from the smallest mean solution to the largest mean solution. Then, we calculate the average rank with respect to these seven functions and determine the overall rank, as shown in Table 3. From the rank of each function, we see that the LFMVO results are superior to the other algorithms except for \( f_6 \) where the PSO is better. However, LFMVO obtains the overall best rank. The experimental results show that the proposed algorithm has superior performance in terms of exploitation.
Table 1. The benchmark functions used in our experiments.

| Test function                                                                 | n   | Range          | $f_{\text{min}}$ |
|--------------------------------------------------------------------------------|-----|----------------|-----------------|
| $f_1(x) = \sum_{i=1}^{d} x_i^2$                                                | 40  | [-100,100]     | 0               |
| $f_2(x) = \sum_{i=1}^{d} |x_i| + \prod_{i=1}^{d} |x_i|$                                      | 40  | [-10,10]       | 0               |
| $f_3(x) = \sum_{i=1}^{d} (\sum_{j=1}^{d} x_j)^2$                                | 40  | [-100,100]     | 0               |
| $f_4(x) = \max(\{x_i\}, 1 \leq i \leq d)$                                     | 40  | [-100,100]     | 0               |
| $f_5(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2]$               | 40  | [-30,30]       | 0               |
| $f_6(x) = \sum_{i=1}^{d} ((x_i + 0.5)^2 - 10\sin(x_i) + 10$                    | 40  | [-1.28,1.28]   | 0               |
| $f_7(x) = \sum_{i=1}^{d} x_i^4 + \text{random}(0,1)$                           | 40  | [-500,500]     | -418.9829       |
| $f_8(x) = \sum_{i=1}^{d} x_i^4 - 10\cos(2\pi x_i) + 10$                      | 40  | [-5.12,5.12]   | 0               |
| $f_{10}(x) = -20\exp\left(-0.2\sqrt{\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{\prod_{i=1}^{d} \cos(x_i)}\right) + 20 + \theta$ | 40  | [-32.32]       | 0               |
| $f_{11}(x) = \frac{1}{40}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{d}}\right) + 1$ | 40  | [-600,600]     | 0               |
| $f_{12}(x) = \frac{\pi}{2}[10\sin(2\pi x_1) + \sum_{i=1}^{d} (y_i - 1)^2[1 + 10\sin^2(2\pi y_i)] + (y_d - 1)^2] + \sum_{i=1}^{d} u(x_i, 10, 100, 4)$ | 40  | [-50,50]       | 0               |
| $y_i = 1 + \frac{a_{i1}}{1 + \exp\left(\frac{1}{m}\sum_{j=1}^{d} (x_j - a_j)^m\right)}$, $x_i > a$ | 2   | [-65.53,65.53] | 0.998004        |
| $f_{15}(x) = \frac{1}{\prod_{i=1}^{d} \prod_{j=1}^{d} (x_i - a_j)}$            | 4   | [-5,5]         | 0.0003075       |
| $f_{16}(x) = \sum_{i=1}^{d} \left[ \frac{a_i}{b_j} x_i (b_j^2 + b_j x_i) \right]^2$ | 4   | [-50,50]       | -1.0316285      |
| $f_{17}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$ | 2   | [-5,10]        | 0.398           |
| $f_{18}(x) = [x_1^2 + \frac{1}{2}x_2^2 + x_1 - 6]^2 + 10(1 - \frac{1}{10}\cos(10x_2))^2 + \cos(10x_2) + 18 x_2^2 - 36x_2^2]$ | 2   | [-5,10]        | 3               |
| $f_{19}(x) = [-\prod_{i=1}^{d} a_i (x_i - p_i)^{\frac{1}{2}}]^{\frac{1}{2}}$        | 3   | [-0,1]         | -3.86           |
| $f_{20}(x) = -\prod_{i=1}^{d} c_i \exp(-\sum_{i=1}^{d} a_i (x_i - p_i)^{\frac{1}{2}})$ | 6   | [-0,1]         | -3.32           |
| $f_{21}(x) = -\prod_{i=1}^{d} c_i \exp(-\sum_{i=1}^{d} a_i (x_i - p_i)^{\frac{1}{2}})$ | 4   | [0,10]         | -1.01532        |
| $f_{22}(x) = -\prod_{i=1}^{d} c_i \exp(-\sum_{i=1}^{d} a_i (x_i - p_i)^{\frac{1}{2}})$ | 4   | [0,10]         | -1.4029         |
| $f_{23}(x) = -\prod_{i=1}^{d} c_i \exp(-\sum_{i=1}^{d} a_i (x_i - p_i)^{\frac{1}{2}})$ | 4   | [0,10]         | -1.5364         |

Results analysis on multi-modal test functions

The multi-modal test function has a global optimum, and the number of local optima increases exponentially with the dimensions. It is suitable for benchmarking the exploration of algorithms. Table 4 lists the results of benchmark functions $f_8$–$f_{13}$ for different algorithms. We
Table 3. Results of unimodal benchmark functions.

| Functions | Statistics | LFMVO | MVO | PSO | MFO |
|-----------|------------|-------|-----|-----|-----|
|           | Best       | 3.3178e-006 | 0.6989 | 1.5809e-005 | 2.2253 |
| f1        | Mean       | 8.6397e-006 | 1.4317 | 9.6071e-005 | 4.5093e+003 |
|           | STD        | 1.0398e-005 | 0.3460 | 5.9722e-005 | 7.1401e+003 |
| Rank      | 1          | 3          | 2    | 4    |
|           | Best       | 1.5898e-049 | 0.7038 | 0.0020 | 0.6584 |
| f2        | Mean       | 1.5107e-047 | 28.9175 | 0.0324 | 44.1369 |
|           | STD        | 4.2885e-047 | 52.3235 | 0.0329 | 29.2929 |
| Rank      | 1          | 3          | 2    | 4    |
|           | Best       | 1.4079e-005 | 152.5786 | 101.6546 | 9.8165e+003 |
| f3        | Mean       | 6.3973e-005 | 374.9805 | 177.3445 | 3.2987e+004 |
|           | STD        | 4.7714e-005 | 132.9893 | 53.1514 | 1.6263e+004 |
| Rank      | 1          | 3          | 2    | 4    |
|           | Best       | 8.1960e-004 | 0.8698 | 0.9602 | 51.6426 |
| f4        | Mean       | 0.0015 | 2.8432 | 1.4980 | 68.4824 |
|           | STD        | 4.6794e-004 | 1.3255 | 0.2266 | 6.6460 |
| Rank      | 1          | 3          | 2    | 4    |
|           | Best       | 38.7824 | 48.3483 | 34.5072 | 549.6379 |
| f5        | Mean       | 38.9045 | 621.7285 | 115.1391 | 2.0279e+006 |
|           | STD        | 0.0406 | 771.9059 | 74.2353 | 1.2651e+007 |
| Rank      | 1          | 3          | 2    | 4    |
|           | Best       | 8.2804 | 0.9215 | 6.6348e-006 | 3.4127 |
| f6        | Mean       | 8.6489 | 1.3681 | 1.0254e-004 | 4.5101e+003 |
|           | STD        | 0.1465 | 0.3078 | 9.8357e-005 | 7.1438e+003 |
| Rank      | 3          | 2          | 1    | 4    |
|           | Best       | 4.0043e-007 | 0.0169 | 0.0715 | 0.0999 |
| f7        | Mean       | 1.1291e-004 | 0.0367 | 0.2383 | 4.4739 |
|           | STD        | 1.0191e-004 | 0.0116 | 0.0829 | 8.4590 |
| Rank      | 1          | 2          | 3    | 4    |

Average Rank | 1.28 | 2.71 | 2 | 4 |

Overall Rank | 1 | 3 | 2 | 4 |

doi:10.1371/journal.pone.0167341.t003
also adopt the rank scheme used in the previous sub-section. From the rank of each function, we can determine that the LFMVO are superior to those of other algorithms except for f11 and f13, where the PSO outperforms the LFMVO algorithm. Nevertheless, the LFMVO ranks best overall. The experimental results demonstrate that the performance of the LFMVO is highly competitive with respect to exploration and escape from poor local optima.

Results analysis on fixed-dimension multimodal benchmark functions

Compared with functions f8-f13, functions f14-f23 are simpler due to their lower dimension and fewer local minima. The results of benchmark functions f14-f23 for different algorithms are shown in Table 5. Though most algorithms were able to easily reach optima for functions f14-f23, we still rank these algorithms. Each of the algorithms can find the optimum at the best condition. For functions f16-f19, there are no differences among the approaches. From Table 6, we find that the LFMVO reaches better solutions than other algorithms.

Convergence analysis

To investigate the convergence behavior of the proposed algorithm, we compare the convergence curves of the LFMVO, MVO, PSO and MFO for four test functions.

The convergence curves of functions f2 and f7 (unimodal test functions) are illustrated in Fig 1 and Fig 2. The convergence curves show that the LFMVO algorithm can successfully improve the fitness of all universes and find a better solution over the course of iterations. The

Table 4. Results of multi-modal benchmark functions.

| Functions | Statistics | LFMVO | MVO | PSO | MFO |
|-----------|------------|-------|-----|-----|-----|
| f8        | Best       | -1.7623e+004 | -1.1953e+004 | -4.7331e+003 | -1.3585e+004 |
|           | Mean       | -1.4886e+004 | -1.0285e+004 | -8.6352e+003 | -1.1249e+004 |
|           | STD        | 5.4972e+003  | 792.9835      | 1.2979e+003  | 1.1557e+003  |
|           | Rank       | 1           | 3             | 4              | 2             |
| f9        | Best       | 0           | 107.0923      | 50.7752        | 117.6214      |
|           | Mean       | 0           | 171.9808      | 75.9237        | 217.1745      |
|           | STD        | 0           | 35.2283       | 14.5150        | 42.4181       |
|           | Rank       | 1           | 3             | 2              | 4             |
| f10       | Best       | 2.2204e-014 | 0.7559        | 0.0024         | 1.0129        |
|           | Mean       | 2.4558e-013 | 1.7515        | 0.0941         | 16.9435       |
|           | STD        | 6.2728e-013 | 0.5808        | 0.2946         | 5.8916        |
|           | Rank       | 1           | 3             | 2              | 4             |
| f11       | Best       | 2.5505e-007 | 0.6177        | 2.1519e-007    | 1.0064        |
|           | Mean       | 0.0172      | 0.8146        | 0.0060         | 43.9017       |
|           | STD        | 0.0056      | 0.0668        | 0.0080         | 67.4905       |
|           | Rank       | 2           | 3             | 1              | 4             |
| f12       | Best       | 7.3649e-008 | 0.0351        | 1.0481e-007    | 3.7687        |
|           | Mean       | 0.0008      | 2.3601        | 0.0019         | 1.9200e+007   |
|           | STD        | 0.0036      | 1.3190        | 0.0123         | 6.8287e+007   |
|           | Rank       | 1           | 3             | 2              | 4             |
| f13       | Best       | 0.0384      | 0.0953        | 2.6017e-006    | 17.2521       |
|           | Mean       | 0.0921      | 0.1809        | 0.0050         | 1.0252e+007   |
|           | STD        | 0.0417      | 0.0898        | 0.0082         | 6.4837e+007   |
|           | Rank       | 2           | 3             | 1              | 4             |
|           | Average Rank | 1.33         | 3             | 2              | 3.66          |
|           | Overall Rank | 1           | 3             | 2              | 4             |

doi:10.1371/journal.pone.0167341.t004
### Table 5. Results of fixed-dimension multi-modal benchmark functions.

| Functions | Statistics | LFMVO | MVO | PSO | MFO |
|-----------|------------|-------|-----|-----|-----|
| f14       | Best       | 0.9980| 0.9980| 0.9980| 0.9980 |
|           | Mean       | 0.9980| 0.9980| 1.6429| 1.4923 |
|           | STD        | 0     | 0    | 0.9107| 1.2829 |
|           | Rank       | 1     | 1    | 3    | 2    |
| f15       | Best       | 3.3355e-004| 3.0828e-004| 3.0803e-004| 5.7996e-004 |
|           | Mean       | 2.4005e-004| 5.2835e-004| 8.0481e-004| 9.4639e-004 |
|           | STD        | 7.3980e-004| 0.0117  | 2.0810e-004| 3.7484e-004 |
|           | Rank       | 1     | 2    | 3    | 4    |
| f16       | Best       | -1.0316| -1.0316| -1.0316| -1.0316 |
|           | Mean       | -1.0316| -1.0316| -1.0316| -1.0316 |
|           | STD        | 0     | 0    | 0    | 0    |
|           | Rank       | 1     | 1    | 1    | 1    |
| f17       | Best       | 0.39789| 0.39789| 0.39789| 0.3979 |
|           | Mean       | 0.39789| 0.39789| 0.39789| 0.3979 |
|           | STD        | 0     | 0    | 0    | 0    |
|           | Rank       | 1     | 1    | 1    | 1    |
| f18       | Best       | 3.0000| 3.0000| 3.0000| 3.0000 |
|           | Mean       | 3.0000| 3.0000| 3.0000| 3.0000 |
|           | STD        | 0     | 0    | 0    | 0    |
|           | Rank       | 1     | 1    | 1    | 1    |
| f19       | Best       | -3.8628| -3.8628| -3.8628| -3.8628 |
|           | Mean       | -3.8628| -3.8628| -3.8628| -3.8628 |
|           | STD        | 0     | 0    | 0    | 0    |
|           | Rank       | 1     | 1    | 1    | 1    |
| f20       | Best       | -3.3220| -3.3220| -3.3220| -3.3220 |
|           | Mean       | -3.2619| -3.2675| -3.2655| -3.2249 |
|           | STD        | 0.0609| 0.0610| 0.0601| 0.0557 |
|           | Rank       | 3     | 1    | 1    | 1    |
| f21       | Best       | -10.1532| -10.1532| -10.1532| -10.1532 |
|           | Mean       | -7.7833| -7.6859| -7.5651| -7.6339 |
|           | STD        | 2.5717| 2.8304| 2.8225| 3.0626 |
|           | Rank       | 1     | 2    | 4    | 3    |
| f22       | Best       | -10.4029| -10.4029| -10.4029| -10.4029 |
|           | Mean       | -9.8284| -8.2990| -9.2505| -8.2915 |
|           | STD        | 2.8397| 2.7932| 2.3655| 3.2963 |
|           | Rank       | 1     | 3    | 2    | 4    |
| f23       | Best       | -10.5364| -10.5364| -10.5364| -10.5364 |
|           | Mean       | -10.3201| -8.7133| -10.3335| -9.5119 |
|           | STD        | 2.2845| 2.8921| 1.2830| 2.5077 |
|           | Rank       | 2     | 4    | 1    | 3    |

Average Rank 1.30 1.70 1.90 2.30
Overall Rank 1 2 3 4

doi:10.1371/journal.pone.0167341.t005

### Table 6. Basic Information of Benchmark Circuits.

| Benchmark | Number of Cores |
|-----------|----------------|
| d695      | 10             |
| p22810    | 28             |
| p93791    | 32             |

doi:10.1371/journal.pone.0167341.t006
The results of f2 and f7 demonstrate that the proposed algorithm has a very fast convergence speed. The reason is that objects in the universes with high inflation rates incline to shift to the universes with low inflation rates through white/black holes, so the fitness of all universes is get better over the course of iterations. Moreover, the proposed Levy flights phase can produce universes with long jumps that leads to quick convergence toward hopeful areas of the search spaces.

The convergence curves of functions f9 and f10 (multimodal test functions) are illustrated in Fig 3 and Fig 4. The graphical results of f9 and f10 (multimodal test functions) show the superior local optima avoidance and global search ability of the LFMVO algorithm. The reason is that regardless of inflation rate, wormholes incline to exist stochastically in any universe which drive universes maintain the diversity over the course of iterations. In addition, Levy flights stage has the ability to escape from local optima and converge to the global optimum rapidly. We have provided the explanation between convergence and application in sub-section Convergence analysis.

The above results demonstrate the superior performance of the LFMVO algorithm in solving different benchmark functions compared with well-known algorithms. To further investigate the performance of the proposed LFMVO algorithm, a real engineering problem, which proved to be an NP complete problem, is solved in the following section.

**The application of the LFMVO on NoC test scheduling optimization**

In this section, we apply the LFMVO to practical engineering applications to investigate the applicability and feasibility of the proposed algorithm. We estimate the performance of the
LFMVO in terms of an engineering design problem—an NoC test scheduling problem. We describe the engineering applications generally and present the relevant mathematical models in the following paragraphs.

The NoC design paradigm has been proposed as an alternative to the traditional System-on-Chip (SoC) design paradigm for the next generation of complex Very Large Scale Integration (VLSI) [30]. The NoC is composed of IP cores, routers, resource interfaces and interconnection links. Due to the packet-switching network, the NoC provides high performance interconnection to embedded IP cores. However, testing embedded cores for NoC-based systems poses new challenges compared to traditional SoC [31].

Like traditional bus-based SoC, the general issues of the NoC system testing are composed of the test architecture design (test wrapper and TAM) and the test scheduling approaches. The test wrapper is the logic added around an embedded core, which is used to isolate the embedded core from surrounding logic and to offer test access to the core via a TAM. The TAM is the physical mechanism used to transport test stimuli and test responses for the cores. The test scheduling approach is employed to decide the test organization that targets test efficiency while considering all test constraints [32].

Testing is usually executed using automated test equipment (ATE), which offers test stimuli and estimates the test responses. ATE provides a limited number of tester pins (test channels) that can be used to send data to/receive data from the core-under-test (CUT). Inefficient use of tester pins (tester channels) has a negative impact on test cost [31].

In testing the embedded cores of NoC, we aim to minimize the test time while satisfying the test pins constraints and power constraints. The test time depends on the test architecture (test
wrapper and TAM) and the corresponding test schedule with the test pins and power constraints. Here, we consider only the test wrapper proposed in [33] and a dedicated test access mechanism (TAM). The test scheduling problem for the NoC system can be defined as follows: in a NoC system, given the set parameters of cores Co, such that each core has a test time $T(c)$ associated with the TAM width, the maximum test channels $N_t$ for NoC, and the maximum power limit $PoL$ for NoC, develop a test schedule, such that 1) $N_t$ is not violated, 2) $PoL$ is not violated, 3) the overall test time is minimized [18].

The embedded cores in a TAM are tested in series, and different TAMs are tested in parallel. The total test time is the sum of all the maximum test times for all the TAMs that are tested in parallel.

We introduce binary variables $y_{ij}$ (1 $\leq$ i $\leq$ N and 1 $\leq$ j $\leq$ B) that are used to determine the assignment of cores to TAMs in the NoC. Each core in the system must be assigned to exactly one TAM.

We can formulate this unity condition by $y_{ij}$ defined in Eq (24) with the unity condition formulated in Eq (25).

$$y_{ij} = \begin{cases} 
1, & \text{if core } i \text{ is assigned to TAM } j \\
0, & \text{otherwise}
\end{cases}$$

(24)

$$\sum_{j=1}^{B} y_{ij} = 1, \ 1 \leq i \leq N$$

(25)

Fig 3. Sample graphs for convergence process comparison of LFMVO, MVO, PSO, and MFO over function $f_9$. doi:10.1371/journal.pone.0167341.g003
The time needed to test all cores on TAM $j$ is given by
\[ \sum_{i=1}^{N} T_i(w_j)y_{ij} \] (26)

Since all the TAMs can be tested in parallel, the overall test time equals
\[ T_{\text{sum}} = \max_{1 \leq j \leq B} \sum_{i=1}^{N} T_i(w_j)y_{ij} \] (27)

The core test time is associated with the transmit bandwidth of the test data. Assuming that core $i$ is assigned to TAM bandwidth $w$, the test time $T_i(w_j)$ is defined by Eq (28):
\[ T_i(w_j) = \left(1 + \max(S_{\text{in}}, S_{\text{out}})\right) \times np + \min(S_{\text{in}} + S_{\text{out}}) \] (28)

where $S_{\text{in}}$ ($S_{\text{out}}$) denotes the length of the longest wrapper scan-in (scan-out) chain for a core and $np$ denotes the number of test vectors. $T_i(w_j)$ is calculated with a Best Fit Decreasing (BFD) algorithm for wrapper design from [31].

The total test pins used by the cores cannot exceed $Pin_{\text{max}}$ during the whole test process. We can formulate the test pins used, $Pin'_{\text{used}}$, during time slot $t$ as follows:
\[ Pin'_{\text{used}} = \sum_{i=1}^{N} Pin_i \cdot \hat{x}_i \leq Pin_{\text{max}} \] (29)

where $Pin_{\text{max}}$ is the total number of test pins available for testing.
\( \lambda^i_t \) is defined by Eq (30):

\[
\lambda^i_t = \begin{cases} 
1, & \text{if } \text{TS}_i \leq t \leq \text{TE}_i \\
0, & \text{otherwise}
\end{cases}
\]  

(30)

where \( \text{TS}_i \) and \( \text{TE}_i \) are the test start time and test end time of core \( i \), respectively.

Although increasing the number of TAMs can effectively shorten the test time and reduce the test cost, it can lead to increasing test power. Therefore, to ensure the viability of the test, power must be constrained during the test.

In any test slot \( i \), power consumption must satisfy

\[
P_{\text{st}}^i = \sum_{i=1}^{N} P_{\text{test}}^i \cdot \lambda^i_t \leq P_{\text{max}}^i
\]  

(31)

where \( P_{\text{test}}^i \) is the power consumption on core \( i \), and \( P_{\text{max}}^i \) is the maximum power consumption allowed for the system.

Therefore, test scheduling for NoC can be formulated as follows:

\[
\min T_{\text{sum}} = \max_{1 \leq j \leq B} \sum_{i=1}^{N} T_i(w_j) \gamma_{ij}
\]

s.t. \( \sum_{i=1}^{N} P_{\text{test}}^i \cdot \lambda^i_t \leq P_{\text{max}}^i \)

\[
\sum_{i=1}^{N} P_{\text{in}}^i \cdot \lambda^i_t \leq P_{\text{in}}_{\text{max}}
\]  

(32)

For the experiments, we used three SOCs from the ITC’02 SoC Test Benchmarks [34]: d695, p22810 and p93791 (see Table 6). We change the problem structure to use much bigger cases for the sake of observing convergence. In others words, we artificially constructed a hybrid system named hy629 including one d695, one p22810 and one p93791.

To compare conveniently, we used the same parameters as in the previous section. Every algorithm was run independently 40 times, and the best results of each algorithm for d695, p22810, p93791 and hyd629 are expressed in Tables 7–10, respectively.

The shortest test results times among the four algorithms are indicated by bold font for each method. From Table 7, we find that the four algorithms obtain the same results in most cases because d695 has the smallest scale among the three benchmarks. However, as the scale increases, the proposed algorithm yields the smaller test time in each category than the three reference methods. The experimental results of Tables 8–10 verify this statement. To further investigate the performance of the LFMV0 (especially on the border/critical cases), we show the boxplots of the four algorithms for the different test benchmarks in Figs 5–8. Figs 5–7 show the condition \( P_{\text{in}}_{\text{max}} = 256 \) and \( P_{\text{max}} = 100\% \). Fig 8 shows the condition \( P_{\text{in}}_{\text{max}} = 512 \) and \( P_{\text{max}} = 100\% \). From Figs 5–8, we can see that the LFMVO outperforms other algorithms with respect to robustness and optimization accuracy. The reason of the superior results of LFMVO on application is that this proposed algorithm efficiently gains a balance between exploration and exploitation. For one, the concepts of white/black holes and Levy flights promote exploration, which can maximize the efficiency of resource searches in uncertain search space. For another, adding the existence of wormholes guarantees exploitation around the most hopeful area of the search space. In general, our proposed algorithm yields the higher performance in each statistical parameter than the three reference methods.
Table 7. Experimental results for d695 with different test pins.

| Pinmax  | Pmax  | Test Time |
|---------|-------|-----------|
|         |       | LFMVO     | MVO | PSO | MFO |
| 256     | 100%  | 9869      | 9869| 9869| 9869|
| 256     | 50%   | 9869      | 9869| 9869| 9869|
| 256     | 30%   | 13164     | 13164| 13164| 13164|
| 256     | 20%   | 20163     | 20163| 20503| 20528|
| 192     | 100%  | 12663     | 12663| 12663| 12663|
| 192     | 50%   | 12663     | 12663| 12663| 12663|
| 192     | 30%   | 13428     | 13428| 13428| 13428|
| 192     | 20%   | 20188     | 21022| 21010| 20751|
| 128     | 100%  | 18869     | 18869| 18869| 18869|
| 128     | 50%   | 18869     | 18869| 18869| 18869|
| 128     | 30%   | 18869     | 18869| 18869| 18869|
| 128     | 20%   | 21401     | 21989| 21989| 21401|

doi:10.1371/journal.pone.0167341.t007

Table 8. Experimental results for p22810 with different test pins.

| Pinmax  | Pmax  | Test Time |
|---------|-------|-----------|
|         |       | LFMVO     | MVO | PSO | MFO |
| 256     | 100%  | 136400    | 137436| 138239| 137586|
| 256     | 50%   | 135998    | 137336| 139132| 138049|
| 256     | 30%   | 135907    | 139229| 139939| 139388|
| 256     | 20%   | 136341    | 140669| 141498| 139944|
| 192     | 100%  | 180942    | 181393| 181161| 181283|
| 192     | 50%   | 180952    | 181284| 182496| 181800|
| 192     | 30%   | 180954    | 181510| 184108| 181995|
| 192     | 20%   | 181376    | 181376| 184111| 182137|
| 128     | 100%  | 271331    | 271360| 271429| 271343|
| 128     | 50%   | 271333    | 271365| 271529| 271347|
| 128     | 30%   | 271332    | 271407| 271644| 271356|
| 128     | 20%   | 271340    | 271376| 271655| 271420|

doi:10.1371/journal.pone.0167341.t008

Table 9. Experimental results for p93791 with different test pins.

| Pinmax  | Pmax  | Test Time |
|---------|-------|-----------|
|         |       | LFMVO     | MVO | PSO | MFO |
| 256     | 100%  | 306233    | 310681| 317253| 312840|
| 256     | 50%   | 307278    | 313360| 342174| 310842|
| 256     | 30%   | 366506    | 373062| 397351| 387611|
| 256     | 20%   | 534687    | 564496| 567322| 553547|
| 192     | 100%  | 407861    | 410490| 415412| 410118|
| 192     | 50%   | 407856    | 408154| 408666| 410896|
| 192     | 30%   | 408246    | 415666| 453412| 422115|
| 192     | 20%   | 551352    | 568489| 577414| 573920|
| 128     | 100%  | 611745    | 611777| 611762| 611766|
| 128     | 50%   | 611748    | 611766| 611759| 611801|
| 128     | 30%   | 611748    | 611778| 611774| 611865|
| 128     | 20%   | 611789    | 636431| 658180| 611974|

doi:10.1371/journal.pone.0167341.t009
Conclusions

A new algorithm named LFMVO is proposed in this paper, and it improves the performance of the MVO by incorporating Levy flights. The Levy flights component is introduced to enhance the global search ability of the MVO and its ability to escape from local optima. Experimental results on three sets of 23 well-known benchmark functions have verified that the proposed LFMVO has outstanding performance in speed of convergence and precision of the solution for global optimization in most cases. A real engineering application using NoC test scheduling optimization confirms that our proposed algorithm outperforms several state-of-the-art algorithms. This superior performance proves that the Levy flights are a promising way of strengthening the searching performance of MVO. Current studies implies that the LFMVO is a powerful and universal approach; it should be investigated further in several applications of engineering optimization problems, such as cloud computing, big data, smart city and vehicular named data networks [35–45]. Our future work is to extend the LFMVO to these fields.

Table 10. Experimental results for hybrid systems hyd629 with different test pins.

| Pinmax | Pmax  | LFMVO | MVO  | PSO  | MFO  |
|--------|-------|-------|------|------|------|
| 512    | 100%  | 243749| 261107| 280109| 271678|
| 512    | 50%   | 245305| 265048| 282099| 271730|
| 512    | 30%   | 246828| 264513| 284344| 284127|
| 512    | 20%   | 247289| 275809| 285549| 284513|

Fig 5. The boxplot of LFMVO, MVO, PSO, and MFO for d695.

doi:10.1371/journal.pone.0167341.t010

doi:10.1371/journal.pone.0167341.g005
Fig 6. The boxplot of LFMVO, MVO, PSO, and MFO for p22810.

doi:10.1371/journal.pone.0167341.g006

Fig 7. The boxplot of LFMVO, MVO, PSO, and MFO for p93791.

doi:10.1371/journal.pone.0167341.g007
Acknowledgments

We are grateful to Jing Ling and Mengyi Jia for valuable comments and providing us some simulated figures.

Author Contributions

Conceptualization: CH ZL CX.
Data curation: TZ AZ CX.
Formal analysis: TZ AZ.
Funding acquisition: CH ZL CX.
Investigation: TZ AZ CX.
Methodology: CH AZ CX.
Project administration: CH ZL CX.
Resources: CH ZL TZ.
Software: ZL TZ AZ.
Supervision: ZL CX.
Validation: TZ AZ.
Visualization: CH ZL TZ.
Writing – original draft: CH ZL CX.
Writing – review & editing: CH ZL TZ AZ CX.

References
1. Rezaee Jordehi A. A chaotic-based big bang–big crunch algorithm for solving global optimisation problems. Neural Computing and Applications. 2014; 25:1329–1335.
2. Gandomi AH and Yang X. Chaotic bat algorithm. Journal of Computational Science. 2014; 5:224–232.
3. Liu C, Du W and Wang W. Particle Swarm Optimization with Scale-Free Interactions. PLoS ONE. 2014; 9:e97822. doi: 10.1371/journal.pone.0097822 PMID: 24859007
4. Holland J. Adaptation in Natural and Artificial Systems. Ann Arbor: University of Michigan Press; 1975.
5. Coloni A, Dorigo M and Maniezzo V. Distributed optimization by ant colonies. Proceedings of the 1st European Conference on Artificial Life; 1991 Dec 11–13; Paris, France: MIT Press; 1992.
6. Kennedy J and Eberhart R. Particle swarm optimization. Proceedings of the IEEE international conference on neural networks; 1995 Nov 27-Dec 01; University of Western Australia, Perth, Australia: IEEE; 1995.
7. Storn R and Price K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. Journal Of Global Optimization. 1997; 11:341–359.
8. Geem ZW, Kim JH and Loganathan GV. A new heuristic optimization algorithm: Harmony search. Simulation. 2001; 76:60–68.
9. Karaboga D and Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal Of Global Optimization. 2007; 39:459–471.
10. Yang X and Deb S. Cuckoo Search via Lévy Flights. 2009 World Congress on Nature & Biologically Inspired Computing; 2009 Dec 09–12; Coimbatore, India: IEEE; 2009.
11. Rashedi E, Nezamabadi-pour H and Saryazdi S. GSA: A Gravitational Search Algorithm. Information Sciences. 2009; 179:2232–2248.
12. Pan W. A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example. Knowledge-Based Systems. 2012; 26:69–74.
13. Abdechiri M, Meybodi MR and Bahrami H. Gases Brownian Motion Optimization: an Algorithm for Optimization (GBMO). Applied Soft Computing. 2013; 13:2932–2946.
14. Cheng M and Prayogo D. Symbiotic Organisms Search: A new metaheuristic optimization algorithm. Computers & Structures. 2014; 139:98–112.
15. Mirjalili S. Moth–flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowledge-Based Systems. 2015; 89:228–249.
16. Mirjalili S, Mirjalili SM and Hatamloiu A. Multi-Verse Optimizer: a nature-inspired algorithm for global optimization. Neural Computing and Applications. 2016; 27:495–513.
17. Al-Tememy AA, Spencer JW and Ralph JF. Levy Flights for Improved Ladar Scanning. 2010 IEEE International Conference on Imaging Systems and Techniques; 2010 July 1–2; Thessaloniki, Greece: IEEE; 2010.
18. Hakli H and Uğuz H. A novel particle swarm optimization algorithm with Levy flight. Applied Soft Computing. 2014; 23:333–345.
19. Chen Y. Research and simulation on Levy flight model for DTN. 2010 3rd International Congress on Image and Signal Processing; 2010 Oct 16–18; Yantai, China:IEEE; 2010.
20. Yang X and Deb S. Multiobjective cuckoo search for design optimization. Computers & Operations Research. 2013; 40:1616–1624.
21. Yang XS. Firefly algorithm, Levy flights and global optimization. In: Bramer M., Ellis R., Petridis M. (Eds.), Research and Development in Intelligent Systems XXVI: Springer, London; 2010, p. 209–218.
22. C. Y. Lee XY. Evolutionary algorithms with adaptive Levy mutations. Proceedings of the 2001 Congress on Evolutionary Computation, 2001 May 27–30; Seoul, South Korea: IEEE; 2001.
23. Candela R, Cottone G, Seiemi GF, Sanseverino ER. Composite Laminates Buckling Optimization through Levy Based Ant Colony Optimization. 23rd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems; 2010 Jun 01–04; Cordoba, Spain: Springer; 2012.
24. Saadi Y, Yanto ITR, Herawan T, Balakrishnan V, Chiroma H and Rismawan A. Ringed Seal Search for Global Optimization via a Sensitive Search Model. PLoS ONE. 2016; 11:e144371.
25. Tegmark M. Parallel universes. In: Barrow JD, Davies PCW, Harper CL Jr (eds) Science and ultimate reality: Quantum theory, cosmology, and complexity: Cambridge University Press; 2004. pp. 459–491.
26. Nurzaman SG, Matsumoto Y, Nakamura Y, Shirai K, Koizumi S and Ishiguro H. From Lévy to Brownian: A Computational Model Based on Biological Fluctuation. PLoS ONE. 2011; 6:e16168. doi: 10.1371/journal.pone.0016168 PMID: 21304911

27. Yang X and Deb S. Multiobjective cuckoo search for design optimization. Computers & Operations Research. 2013; 40:1616–1624.

28. Mantegna RN. Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes. Physical Review E. 1994; 49:4677–83.

29. Clerc M and Kennedy J. The particle swarm—Expllosion, stability, and convergence in a multidimensional complex space. IEEE Transactions On Evolutionary Computation. 2002; 6:58–73.

30. Liu C, Iyengar V, Iyengar V, Pradhan DK and Pradhan DK. Thermal-aware testing of network-on-chip using multiple-frequency clocking. Proceedings of the 24th IEEE VLSI Test Symposium; 2006 Apr 30-May 04; Berkeley, CA; 2006.

31. Richter M and Chakrabarty K. Optimization of Test Pin-Count, Test Scheduling, and Test Access for NoC-Based Multicore SoCs. IEEE Transactions On Computers. 2014; 63:691–702.

32. Hu C, Li Z, Xu C and Jia M. Test Scheduling for Network-on-Chip Using XY-Direction Connected Subgraph Partition and Multiple Test Clocks. Journal of Electronic Testing-Theory And Applications. 2016; 32:31–42.

33. Iyengar V, Chakrabarty K and Marinissen EJ. Test Wrapper and Test Access Mechanism Co-Optimization for System-on-Chip. Journal of Electronic Testing: Theory and Applications. 2002; 18:213–230.

34. Marinissen EJ, Iyengar V and Chakrabarty K. A set of benchmarks for modular testing of SOCs. International Test Conference; 2002 OCT 07–10; Baltimore, MD: IEEE; 2002.

35. Ahmed SH, Bouk SH, Yaqub MA, Kim D, Song H, Lloret J. CODIE: Controlled Data and Interest Evaluation in Vehicular Named Data Networks. IEEE Transactions on Vehicular Technology. 2016; 65:3954–63.

36. Cordeschi N, Shojafar M, Amendola D, Baccarelli E. Energy-efficient adaptive networked datacenters for the QoS support of real-time applications. The Journal of Supercomputing. 2015; 71:448–78.

37. Shojafar M, Cordeschi N, Baccarelli E. Energy-efficient Adaptive Resource Management for Real-time Vehicular Cloud Services. IEEE Transactions on Cloud Computing. Forthcoming 2016.

38. Cordeschi N, Shojafar M, Baccarelli E. Energy-saving self-configuring networked data centers. Computer Networks. 2013; 57:3479–91.

39. Wei W, Fan X, Song H, Fan X, Yang J. Imperfect Information Dynamic Stackelberg Game Based Resource Allocation Using Hidden Markov for Cloud Computing. IEEE Transactions on Services Computing. Forthcoming 2016.

40. Li W, Santos I, Delicato FC, Pires PF, Pirmez L, Wei W, et al. System modelling and performance evaluation of a three-tier Cloud of Things. Future Generation Computer Systems. Forthcoming 2016.

41. Lv Z, Yin T, Song H, Chen G. Virtual Reality Smart City Based on WebVRGIS. IEEE Internet of Things Journal. Forthcoming 2016.

42. Lo Ai A. Tawalbeh WBHS. A Mobile Cloud Computing Model Using the Cloudlet Scheme for Big Data Applications. 2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE); 2016 June 27–29; Washington, DC; IEEE; 2016.

43. Zhang Y, Liu S, Zhang R, Wei W, Song H, Li W, et al. A New Multi-service Token Bucket-Shaping Scheme Based on 802.11e. 2015 International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI); 2015 Oct 22–23; Beijing, China; IEEE; 2015.

44. Butun I, Erol-Kantarci M, Kantarci B, Song H. Cloud-Centric Multi-Level Authentication as a Service for Secure Public Safety Device Networks. IEEE Communications Magazine. 2016; 54:47–53.

45. Houbing Song DRSJ. Cyber-Physical Systems Foundations, Principles and Applications Paperback. Burlington, Massachusetts: Morgan Kaufmann; 2016.