ijecs open access www.ijecs.in
International Journal of Engineering and Computer Science
Volume 14 Issue 02 February 2025, Page No. 26853-26864
ISSN: 2319-7242 DOI: 10.18535/ijecs/v14i02.4995

A New Metaheuristic Lung-Inspired Algorithm for Continuous Optimization Problems

Mustafa Kadhim Taqi 1[*], Mueen Mohsin Abbood [2]

[1] Department of Information Techniques, Technical collage of management – Kufa, Al-Furat Al-Awsat Technical University Kufa, Iraq.

[2] Department of Information Techniques, Technical collage of management – Kufa, Al-Furat Al-Awsat Technical University Kufa, Iraq.

Abstract

This paper introduces a metaheuristic algorithm called lung-inspired algorithms (LA). LA is mainly inspired by the breath cycle of the human lung. Three phases namely inhalation, gas exchange, and exhalation are conducted during this cycle, to maximize the saturation of oxygen in the blood as carbon dioxide is removed. This algorithm begins by random initial solutions. The saturation rate would then be determined depending on the objective functions average valuation of all proposals within existing population at each iteration. The inhalation stage explores positions opposite to low saturation solutions in order to identify promising places. However, once they exceed the rate of saturation, solutions are reconsidered. In comparison, the gas exchange process takes advantage of movement operators dependent on the oxygen flux law. Then, by simulating lung exhalation function, weak solutions are eliminated from the population. The best is identified and the saturation rate adjusted when all alternatives were placed in the population. The algorithm is assessed by using twenty-three identified benchmark datasets and corresponding output to many other metaheuristics. This involves the most well-studied metaheuristics GA and PSO. In addition, the newly-designed metaheuristics that includes BAT, GSA, WOA, MPA, and the GWO have also been involved. In almost all the test functions, the output of LA is outperformed.

Keywords: Optimization; Nature-inspired algorithms; Metaheuristics.

1. Introduction

The word metaheuristic identifies higher-level heuristics suggested to solve a broad variety of optimization problems. Recently, researchers have been attempting to mimic nature in technology because nature is the greatest instructor of technology and its architecture and abilities. These two domains are also greatly improved as several new challenges are listed as natural challenges in the information sciences. Simple mapping of nature and technologies is also feasible in the real world. Although the literature has a number of natural algorithms in recent years, we also agree that the best way to map nature and computer science is by improving existing natural algorithms and by implementing new algorithms. This may be the explanation for implementing the algorithm inspired by the lung in this research. In computer science, a metaheuristic can provide a greater degree of heuristic solutions to a problem of optimization. In recent decades, natural algorithms have been particularly suitable to solve various optimization problems amongst all meta-heuristic literature algorithms [1]. Nature-inspired algorithms come from normal physical or biological processes. Nature-inspired algorithms successfully applied in many areas, such as engineering [2-8] data mining [9-14], pattern recognition [15, 16], economics [17-22], industry [23, 24], and scheduling [25-27].

Literature have showed numerous amounts of nature-inspired algorithms such as the Genetic algorithm (GA), which Simulate normal genetic variance and selection operators [28]. Another example is the Particle swarm optimization (PSO) that simulates the swarm behavior [29]. Gravitational search algorithm (GSA) act out gravity law and the idea of mass interactions [30]. Bat algorithm (BA) mimics the echolocation behavior of bats in [31]. Firefly algorithm (FA) simulate the flickering light produced by real fireflies [32]. Whale optimization algorithm (WA) simulate the hunting behavior of humpback whales [33]. In [34], Grey wolf optimizer (GWO) imitates the leadership hierarchy and hunting mechanism of grey wolves in nature. Recently, [35] presents the Marine predators' algorithm (MPA). MPA emulates the widespread foraging strategy namely Lévy and Brownian movements in ocean predators. This is along with optimal encounter rate policy in bio-logical interaction between predator and prey. However, human body organs-inspired algorithms have exposed a significant performance. Table 1 provides some instances of these algorithms.

Table 1: Human Body Organs-Inspired Algorithms Examples

Algorithm	**Abbreviation**	**Inspiration**	**Reference**
Heart algorithm	HA	Simulates the heart process in the human body	[36]
Invasive Tumor Growth Optimization	ITGO	Simulates the principle of invasive tumor growth	[37]
Kidney-inspired Algorithm	KA	Simulates the kidney process in the human body	[38]
Stem Cells Optimization Algorithm	SCO	Simulates the stem cells in the human body	[39]

The performance of a meta-heuristic is mainly affected by balancing exploration and exploitation capabilities of the algorithm [40]. This paper introduces a new optimization algorithm focused on the lung mechanism of human body exploitation and exploration. The remaining paper is organized as follows: Section 2 offers a summary of the human body's overall lung phases. Section 3 defines the specifics of the current LA. This segment also gives the LA's simulated components and its pseudocode. Section 4 provides experimental findings of testing the proposed LA based on twenty-three benchmarks. Section 5 concludes that future extensions are also addressed.

2. The Biological Lungs Process

Lungs are a pair of pyramid-shaped organs. They are spongy and pinkish-gray in color. The main function of the lungs is to get carbon dioxide out of our bodies. Besides, they allow our bodies to take oxygen from the air to fuel things such as muscle tightness and a regeneration capacity in our neurons. To do so, we are using the energy that is stored in carbohydrates, hence, sugar and oxygen must be mixed.

When breathing in, the lungs bring oxygen into the body and send carbon dioxide when breathing out. Carbon dioxide is the body's cells' waste product. In another words, breathing in process is called inhalation. Breathing out is called exhalation. While, in a process called gas exchange, the lungs add oxygen to the blood and remove carbon dioxide (See Fig. 1).

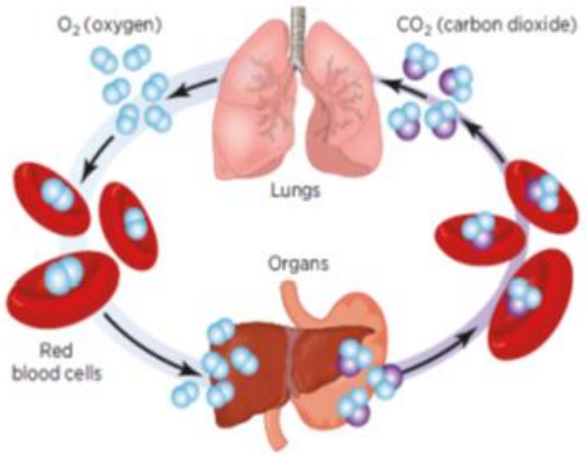

Fig. 1. *Respiratory system [41]*

Besides the lungs, the respiratory system includes airways, muscles, blood vessels, and tissues that render breathing possible. The brain controls your breathing dependent on the need for oxygen. Lung's fundamental physiological function taken from [41]. The lungs process can be summarized as follows:

1. **Inhalation:** As we breathe in the diaphragm contracts, it moves towards the abdomen, causing the lungs to expand and fill with air like a bellow. At this stage, oxygen enters the lungs. Meanwhile, Deoxygenated blood is delivered to the heart by the veins and injected into the pulmonary system, it reaches the right side of the heart and pumped into the lung.
2. **Gas exchange:** This process is carried out as oxygen diffuses from the alveoli into the pulmonary capillaries due to high oxygen concentration in the lungs and low blood pressure. At the same time, the carbon dioxide waste from the breakdown of sugars in the cells of the body diffuses in alveoli from the pulmonary capillaries because there is a high concentration of carbon dioxide in the blood and low concentration in the lungs.
3. **Exhalation:** During this process, carbon dioxide, which is extracted from the deoxygenated blood that is delivered through the heart by the veins into the lung system, diffuses into the lungs and is expelled as we breathe out.

3. Lung-Inspired Algorithm

Basically, the function of the suggested LA is like other population-based algorithms. Furthermore, as the name implies, it mimics lung biological system procedures. In this simulation, three main components of the above lung cycle are used.

The algorithm starts by determining the value of the objective function for all random population of candidate solutions. The value of the objective function represents the oxygen saturation of solutions at each iteration. Then, the oxygen saturation rate (sr), which is the percentage of oxygen-saturated hemoglobin relative to total hemoglobin in the blood is calculated as follows:

$$sr = \frac{\sum_{i=1}^{p} so(x)}{p} \tag{1}$$

Where p is size of population. so(xi) is solution's objective function x_i. This format demonstrates that the saturation rate depends on the objective function values of all population solutions for each iteration. Saturation rate varies to help converge the algorithm. Objective values exceed the global supreme with any iteration and thus, measuring the saturation rate dependent on these solutions offers better solutions.

In the inhalation phase, a portion of the initial population that are less than the oxygen saturation rate is replaced by newly random generated individuals, if they are better than the existing ones. In the gas exchange stage, for all candidate solutions, a new solution is created, if its objective function value lower than the saturation rate (high carbon dioxide concentrations), by moving them based on the best solution found so far. In exhalation phase, poor solution (hemoglobin with low oxygen saturation) will be expelled by using the same movement mechanism as in the gas exchange with opposite side.

Finally, the best solution and the oxygen saturation is updated. This method is continued until the termination requirement is satisfied. Table 2 offers a comparison of LA and the respiratory biological system. The following parts describe the components used in the LA.

Table 2: Lung-Inspired Algorithm Analogy with Lung Biological Scheme

Lung biological scheme	LA
Hemoglobin in red blood cells (Hb)	Solutions
Hemoglobin oxygen saturation (SaO2)	Algorithm objective function (so)
Inhale	Generate new solutions by visiting locations that are opposite to the existing ones
Gas exchange	Moving solutions towards best solution
Exhale	Hb with saturation lower than the saturation rate is replaced with random number

3.1 Population Initialization

As any population-based metaheuristic, LA starts with generating an arbitrary population of candidate solutions. Each solution in the LA population involves the biological pulmonary hemoglobin. For each generated candidate, the objective function is calculated. The best solution so far can therefore be identified based on the value of the objective function.

3.2 Inhalation

In this phase, the continuous flow of blood into lung is appeared by inserting solutions to the existing low saturation ones. The number of insertions is based on the value of irv, which is a constant value of [0, 1] that represents the inspiratory reserve volume in the lung biological system [42]. irv is adjusted in advance. If irv is 0 means the algorithm has no inhalation procedure. Therefore, irv must be greater than 0, equal to or less than 1. However, this study suggested that the location of poor solutions will be replaced with its quasi-opposition location. According to [43], algorithms can be accelerated by using random numbers and their opposites.

3.3 Gas exchange

In this phase, new solutions are created by transferring the solutions to the best solution identified by the algorithm from the previous iteration. That is, by attempting to change the present solution by the best solution. To do so, this study adapted the flexible lung with gas exchange model [44]. In this model, Oxygen diffuses between capillaries and alveoli and the flux of oxygen is estimated by Weibel [45] as follow:

$$flux_i = D_{oi}(P_{oi} - P_{aoi}) \qquad (2)$$

For each ith iteration, D_oi is the oxygen diffusion (transfer factor), in which equals to a uniform random number of distributions between zero and a certain number [46], P_oi is the partial pressure of oxygen in the blood, which represents the current position of the solution in the search space. P_aoi is the partial pressure of oxygen in the alveolar that embody the best solution position? Regarding to equation 2, which represents the Dalton’s Law of partial pressure [44], a negative flux means that gas diffuses from the alveoli to the blood and a positive flux gives the opposite direction. After calculating the flux, the next position of a solution is defined as follows:

$$Hb_{i+1} = Hb_i + flux_i \qquad (3)$$

In Equation 2, Hb is a solution in the LA population (or blood hemoglobin in the capillaries of the biological lung). In addition, pushing the solutions towards the best solution improves the locally convergent potential of the algorithm. The algorithm offers an excellent array of solutions, which imply that the solutions based on current solutions and the best move are accurate.

Please note that a duplicate solution that's structured in the same way as any solution structure already in the population is not allowed into the population, as otherwise the population may come up with all the same solutions, thereby limiting severely the ability of the LA to create new solutions. This is consistent with the fact that we want different architectures for operational control (many different solutions).

3.4 Exhalation

Population solutions exhaled if the saturation threshold cannot be met after a gas exchange possibility was given. In particular, these solutions are exhaled if they are unable to produce sufficient oxygen-saturated hemoglobin after two moves. In this phase, exhaled solution will be substituted with random number.

The inhalation and exhalation, in this algorithm, provides strong experimentation to escape local optimism. On the contrary, the gas exchange operator offers high-exploitation algorithms by heading to the best so far. This is because the gas exchange operator in this algorithm still focuses on the right solution search space. The schematic process and pseudocode of the LA are shown in Fig. 2 and 3, respectively.

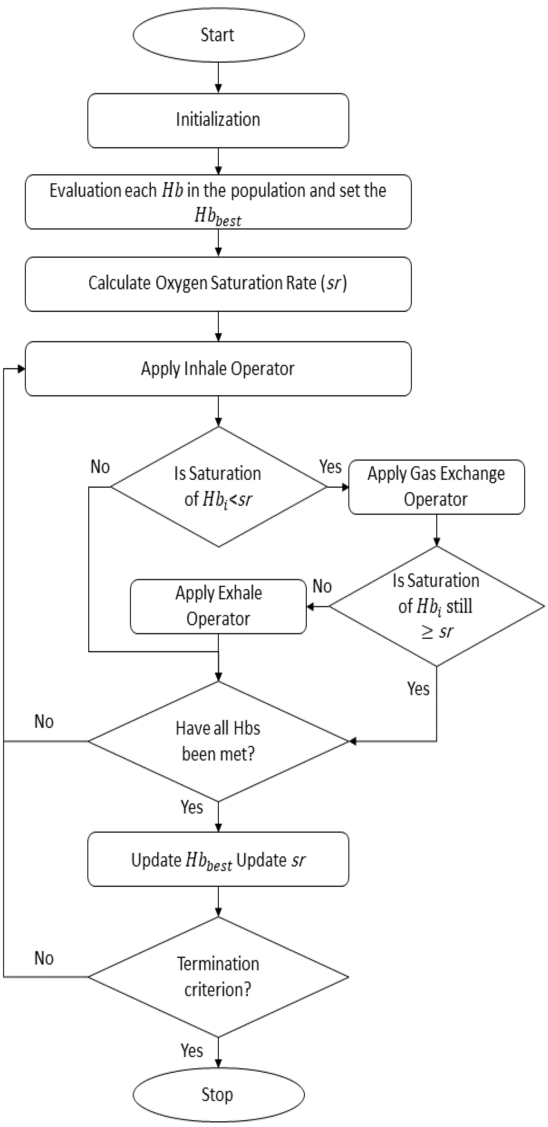

Fig. 2. LA Flowchart

```
set inspiratory reserve volume, irv
set population size, popsize
set number of iterations, numofite
initialize the population
while (ite < numofite)
        evaluate the solutions in the population
        set the best solution, Hb_best
        set the oxygen saturation rate, sr_ite, Eq.1
        for all Hb_i
                if (so_i < sr and not inhaled and irv not exceeded)
                        apply inhalation on Hb_i
                else if (so_i < sr and Hb_i not gas_exchanged)
                        apply gas_exchange on Hb_i
                else if (so_i < sr and Hb_i not exhaled)
                        apply exhalation on Hb_i
                end if
        end for
end while
return Hb_best
```

Fig. 3. LA Pseudocode

4. Simulations and Results

4.1 Validation

The LA can be implemented in any programming language reasonably simple from the pseudo code. However, the proposed algorithm has implemented it in MATLAB. A general context was explored in depth for the computational performance assessment of evolutionary algorithms [47]. In that, Various test functions have been developed over several years for optimization algorithm, and a reasonably detailed analysis of these test functions is available in [48]. These functions are widely used in the studies to test the proposed algorithms efficiency and effectiveness [31, 33, 49-54].

This analysis tested LA's output with numerous test functions. In particular, two distinct classes of recognized benchmarks are involved. These include unimodal, multimodal, and multimodal with fixed dimension to test LA 's capacity to explore and exploit the function space. Typically, Unimodal research functions (F1-F6) are used to evaluate the exploitation ability, whereas multimodal functions (F7-F13) are used to examine exploration feature of an algorithm. These two groups are checked in 30 variable size. However, the functions for a fixed dimension (F14-F23) suggest LA's exploration potential. Table 3 displays the description of the optimal global component of the test functions that have been used in this evaluation. Dim is a constant value given in the parenthesis, which has been specified in the test function formula.

Table 3: Mathematical Formulation and Properties of Benchmark Functions

Function	Dim	Domain	Global optimum
$F1(x)=\sum_{i=1}^{d} x_i^2$	30	[-100,100]	0
$F2(x)=\sum_{i=1}^{d}\lvert x_i\rvert+\prod_{i=1}^{d}\lvert x_i\rvert$	30	[-100,100]	0
$F3(x)=\sum_{i=1}^{d}(\sum_{j=1}^{d} x_j)^2$	30	[-100,100]	0
$F4(x)=Max_{i=1}^{d}\{\lvert x_i\rvert\}$	30	[-100,100]	0
$F5(x)=\sum_{i=1}^{d-1}\left[100\left(x_{i+1}-x_i^2\right)^2+(x_i-1)^2\right]$	30	[-30,30]	0
$F6(x)=\sum_{i=1}^{d-1}(x_i+0.5)^2$	30	[-100,100]	0
$F7(x)=\sum_{i=1}^{d} ix_i^4+random[0,1]$	30	[-1.28,1.28]	0
$F8(x)=\sum_{i=1}^{d}-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$	30	[-500,500]	$-418.98d$
$F9(x)=10d+\sum_{i=1}^{d}[x_i^2-10\cos(2\pi x_i)]$	30	[-5.12,5.12]	0
$F10(x)=20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_1^d\cos(2\pi x_i)}\right)-\exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right)+20+e$	30	[−32,32]	0
$F11(x)=\sum_{i=1}^{d}\frac{x_i^2}{4000}-\prod_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	[-600,600]	0
$F12(x)=\frac{\pi}{d}\{10\sin(\pi y_1)+\sum_{i=1}^{d-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+(y_d-1)^2\}+\sum_{i=1}^{d}u(x_i,10,100,4)$	30		
$F13(x)=0.1\{\sin^2(3\pi x_1)+\sum_{i=1}^{d-1}(x_i-1)^2[1+\sin^2(3\pi x_i+1)]+(x_d+1)^2[1+\sin^2(2\pi x_d)]\}+\sum_{i=1}^{d}u(x_i,5,100,4)$	30	[-50,50]	0
$F14(x)=(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6})^{-1}$	2	[-65,65]	1
$F15(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_2}\right]^2$	4	[-5,5]	0.00030
$F16(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_1^4$	2	[-5,5]	-1.0316
$F17(x)=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi})\cos x_1+10$	2	[-5,5]	0.398
$F18(x)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]\times[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	2	[-2,2]	3
$F19(x)=\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2)$	3	[1,3]	-3.86
$F20(x)=\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2)$	6	[0,1]	-3.32
$F21(x)=\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	-10.1532
$F22(x)=\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	-10.4028
$F23(x)=\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	-10.5363

4.2 Comparison with Other Algorithms

To compare LA with the cutting-edge algorithms, there were 500 iterations for the method, and also 100 for the population. However, the efficiency of algorithms and the algorithm will achieve a better result when the population size and number of Iterations are advanced. As the algorithm hits the global optimum, the number of function evaluations was collected for 100 runs. Tables 4 and 5 display the average number of tests of 30 outcomes for each test function.

As can be seen in Table 4 and 5, LA findings are compared with the previously published results of the most recognized algorithms, namely the genetic algorithm (GA) [28] and the particle swarm optimization (PSO) [29]. Moreover, as a recent developed metaheuristic, LA also compared with Bat algorithm (BAT) [31], A gravitational search algorithm (GSA) [30], the whale optimization algorithm [33], Marine predators algorithm (MPA) [35], and Grey wolf optimizer (GWO) [34]. Owing to a rational comparison the same criteria are chosen in the literature for such experiments. The regular PSO variant without inertia function is used, although the genetic algorithm has a probability of mutation 0.05 and a probability of crossover of 0.95 without using elitism. Although, many variants of PSO and GA are available, and some variants may perform better than those standards; however, by far, the standard version is the most commonly used. Therefore, in our analogy, we would also use regular algorithms.

4.2.1 Exploitation ability of LA

Because unimodal functions have just one global optimum, they may determine an algorithm's exploitability. Table 4 demonstrates LA and other algorithms using average and standard deviation values in the unimodal test function (TF1-TF7). The findings show that in virtually all research functions LA was able to outperform other methods. These findings indicate the LA's capacity to exploit the search space, which will enable LA converge and use it quickly and accurately.

Table 4: Results of Unimodal Benchmark Functions

Function		**LA**	**PSO**	**GA**	**BAT**	**MPA**	**WOA**	**GWO**	**GSA**
F1	Ave	1.16E-97	0.0409	1.095	3.86E+04	4.04E-23	2.79E-72	8.43E-28	3.25E+00
	Std	1.05E-94	0.0416	0.4896	1.03E+04	4.59E-23	1.49E-71	1.03E-27	6.39E+00
F2	Ave	3.95E-60	0.0659	0.106	2.42E+07	2.51E-13	2.03E-51	9.32E-17	1.41E+00
	S	7.0	0.	0.0	6.5	2.6	8.3	6.0	1.0

	Std	2E-60	0864	498	3E+07	9E-13	6E-51	5E-17	3E+00
F3	Ave	3.96E-73	4236.3	25,187.3	9.14E+04	1.35E-04	4.73E+04	5.02E-06	9.25E+02
	Std	8.77E-74	1217.9	5243.43	3.68E+04	3.86E-04	1.55E+04	5.89E-06	2.31E+02
F4	Ave	4.50E-53	9.335	35.619	6.99E+01	3.95E-09	3.80E+01	5.31E-07	6.35E+00
	Std	2.39E-53	1.0119	9.4072	8.32E+00	2.74E-09	2.38E+01	5.10E-07	1.93E+00
F5	Ave	2.89E+01	310.39	715.98	1.01E+08	2.52E+01	2.79E+01	2.70E+01	8.95E+01
	Std	5.62E-02	430.60	634.71	5.16E+07	4.24E-01	5.02E-01	8.05E-01	7.67E+01
F6	Ave	7.11E-13	0.0589	0.925	3.92E+04	4.98E-08	2.95E-01	7.00E-01	3.05E-07
	Std	5.81E-01	0.1217	0.5063	1.12E+04	5.30E-08	1.74E-01	3.44E-01	1.67E-06
F7	Ave	1.37E-03	0.0665	0.1130	6.45E+01	1.21E-03	3.47E-03	1.90E-03	8.03E-02
	Std	4.45E-05	0.0123	0.0355	3.54E+01	7.95E-04	4.07E-03	1.08E-03	3.63E-02

4.2.2 Exploration ability of LA

There are many local optima multimodal test functions, which increase dramatically in various measures (design variables). It is beneficial to obtain more than one optimum while determining the exploration potential of an algorithm. Functions in the range of TF8 to TF23 are high-and-fixed (low) multimodal. Table 5 summarizes the outcomes of LA and other algorithms in the experiment. The table shows LA gains from excellent exploration capabilities relative to other approaches. LA could outdo all algorithms on half of the high-dimensional multimodal functions and, on the other half, the results are comparable with high-performance optimization methods.

Table 5: Results of Multimodal (High Dimensional) Benchmark Functions

Function		LA	PSO	GA	BAT	MPA	WOA	GWO	GSA
F8	Ave	-1.30E+04	-13,594.1	-10,815.3	-6.18E+03	-8.90E+03	-1.06E+04	-5.80E+03	-2.59E+03
	Std	5.60E+02	811.3	992.1	4.00E+03	4.08E+02	1.76E+03	1.01E+03	5.21E+02
F9	Ave	0.00E+00	0.000	78.42	3.75E+02	1.23E-09	5.68E-15	1.81E+00	3.01E+01
	Std	0.00E+00	0.000	16.44	4.69E+01	6.72E-09	2.29E-14	3.35E+00	9.86E+00
F10	Ave	8.88E-16	9.69E-12	1.204	1.98E+01	1.58E-12	4.68E-15	9.56E-14	3.23E-02
	Std	0.00E+00	6.13E-12	0.729	4.58E-01	9.68E-13	2.94E-15	1.32E-14	1.70E-01
F11	Ave	0.00E+00	0.000	0.0128	3.85E+02	0.00E+00	0.00E+00	2.83E-03	2.80E+01
	Std	0.00E+00	0.000	0.0130	9.71E+01	0.00E+00	0.00E+00	7.33E-03	6.67E+00
F12	Ave	3.25E-01	0.0085	0.0319	1.92E+08	1.75E-05	2.07E-02	5.07E-02	1.86E+00
	St	2.30E-	0.005	0.056	1.22E	7.31E-	1.33E-	4.82E-	9.42E-

	d	01	2	0	+08	05	02	02	01
F13	Ave	0.00E+00	0.4901	0.419	4.35E+08	9.39E-03	5.62E-01	6.67E-01	8.55E+00
	Std	1.10E-01	0.1932	0.5814	2.16E+08	1.62E-02	2.74E-01	2.45E-01	5.48E+00
F14	Ave	3.75E-19	0.9980	2.1825	2.24E+01	9.98E-01	2.48E+00	3.91E+00	5.99E+00
	Std	3.58E+00	2.47E-16	2.0085	3.61E+01	8.76E-17	2.17E+00	3.92E+00	3.67E+00
F15	Ave	3.80E-04	3.07E-04	5.61E-04	6.24E-02	3.07E-04	7.57E-04	3.74E-03	4.82E-03
	Std	8.96E-03	4.09E-15	4.38E-04	5.13E-02	6.47E-15	3.76E-04	7.57E-03	2.18E-03
F16	Ave	-1.03E+00	−1.0316	−1.0316	-6.37E-01	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	Std	3.06E-02	4.46E-16	6.64E-16	3.91E-01	7.98E-17	1.38E-09	1.99E-08	1.03E-16
F17	Ave	3.98E-01	0.3979	0.3979	9.61E-01	3.98E-01	3.98E-01	3.99E-01	3.98E-01
	Std	2.00E-01	9.12E-15	0.000	6.72E-01	6.59E-14	1.33E-05	4.10E-03	0.00E+00
F18	Ave	3.00E+00	3.000	3.000	2.59E+01	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	Std	7.66E+00	1.95E-15	1.38E-15	2.54E+01	1.70E-15	1.05E-03	3.81E-05	4.30E-15
F19	Ave	-3.86E+00	−3.8628	−3.8628	-3.50E+00	-3.86E+00	-3.85E+00	-3.86E+00	-3.86E+00
	Std	1.07E-01	2.42E-15	2.68E-15	2.65E-01	3.08E-16	1.41E-02	1.83E-03	1.11E-16
F20	Ave	-3.28E+00	−3.3220	−3.2625	-1.99E+00	-3.32E+00	-3.26E+00	-3.27E+00	-3.32E+00
	Std	4.00E-01	1.14E-11	6.05E-02	6.37E-01	8.59E-12	1.10E-01	7.60E-02	3.21E-16
F21	Ave	-1.02E+01	−10.1532	−5.3010	-8.63E-01	-1.02E+01	-7.78E+00	-8.97E+00	-4.99E+00
	Std	2.06E+00	2.53E-11	2.9288	5.80E-01	3.27E-11	3.02E+00	2.46E+00	3.45E+00
F22	Ave	-1.04E+01	−10.4029	−7.0716	-1.04E+00	-1.04E+01	-8.49E+00	-1.02E+01	-9.89E+00
	Std	2.16E+00	2.81E-11	3.6840	3.66E-01	6.11E-11	2.76E+00	1.09E+00	1.59E+00
F23	Ave	-1.05E+01	−10.5364	−7.2467	-1.25E+00	-1.05E+01	-7.71E+00	-9.99E+00	-9.87E+00
	Std	2.96E+00	3.89E-11	3.6582	5.20E-01	3.68E-11	3.32E+00	2.06E+00	2.14E+00

In most benchmarks with fixed dimension, LA has hit the global optimization near high-performance optimizers. The reason of LA's high exploration is because of its inhalation and exhalation phases.

4.2.3 Convergence analysis of LA

The convergence curve of BAT, MPA, WOA, GWO, and GSA for some of the test functions is showing in Fig. 3. Please notice that the best-so-far average is the best solution obtained so far in each iteration over 30 runs. Also, curves for PSO and GA were not included due to their poor performance compared to other methods appeared in this study. It can be observed that LA's search members appear to search extensively for promising design spaces and leverage the strongest. In most cases, search members shift abruptly in the early stages of optimization and eventually converge. Such activity is recognized by [16], who says it will ensure that a population-based algorithm ultimately converges to a point in a search area. LA is equally comparable with other state-of-the-art meta-heuristic algorithms.

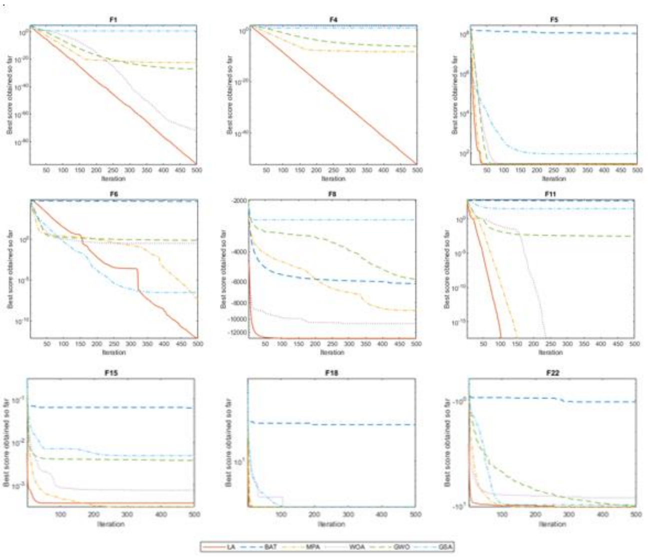

Fig. 3. LA Averaged Convergence Curve on Different Mathematical Benchmark Functions

The results, however, show two distinct, observable LA behavioral patterns in optimizing test functions. First, LA algorithm convergence appears to intensify as optimization grows. This is due to the gas exchange and exhale mechanisms proposed for LA that allows it to look for the promising search space regions in the initial steps of iteration and converge more rapidly to the optimum after passing nearly half of iterations. This is apparent in F1, F4, and F6. The second behavior is a fast convergence from initial iteration phases as shown in F5, F8, F11, F15, F18, and F22. This is presumably attributed to LA's ability to find a successful answer in the initial steps of iteration. Overall, LA 's progress rate appears strong in addressing difficult problems.

5. Conclusion

In this article, we have suggested an original optimization algorithm based on human lung breathing cycle. The algorithm starts with the calculation of a saturation rate dependent on the average value of objective functions for the current iteration's population solutions. In this algorithm, three elements in The lung pulmonary system biological composition is imitated. These are inhalation, gas exchange, and exhalation. In the lung inhalation, solutions with saturation lower than the saturation rate are reconsidered in each iteration by opposite the random solution. This emulates the continuous of blood flow into the lung. Afterward, A defined movement operators, based on the flux of oxygen as said by Dalton's Law of partial pressure [44], are applied to solutions with saturation lower than the saturation rate, simulating the lung's gas exchange function. if the solution doesn't meet the saturation rate, it excluded from the population, acting out the lung's exhale function. After putting all options in the population, the strongest is found and the concentration rate is modified. The algorithm 's implementation and its performance for twenty-three benchmarks have shown that the Algorithm is efficient in relation to other algorithms.

The key benefit of this algorithm is the balance between exploitation, given by new solution generation in gas exchange and exhalation process that focus on the best search space field, and exploration, offered by the inhalation. This advantage makes greater convergence of the algorithm. LA shows fast convergence from initial iteration phases.

These encouraging findings indicate that this research could be expanded to conduct a broader comparison of a wider variety of algorithms utilizing higher-dimensional test functions. In addition, as we research the issue of continuous optimization, the lung-inspired algorithm would also be a significant extension to the discrete search problems. Additional tests should be made in order to expand the method to certain practical issues such as scheduling and time tabling, image processing as well as engineering challenges in order to determine how effective these problems are solved.

References

1. Dressler, F. and O.B.J.C.N. Akan, A survey on bio-inspired networking. 2010. 54(6): p. 881-900.
2. Diwania, S., et al., Photovoltaic–thermal (PV/T) technology: a comprehensive review on applications and its advancement. 2020. 11(1): p. 33-54.
3. Sulaiman, M.H., et al., Barnacles Mating Optimizer: A new bio-inspired algorithm

for solving engineering optimization problems. 2020. 87: p. 103330.
4. Alsarraf, J., et al., Application of PSO–ANN modelling for predicting the exergetic performance of a building integrated photovoltaic/thermal system. 2020. 36(2): p. 633-646.
5. Guo, J. and H. Li. RSWI: a rescue system with wifi sensing and image recognition. in Proceedings of the ACM Turing Celebration Conference-China. 2019.
6. Zhang, Y., et al. Review of Gait Recognition Methods. in 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC). 2019. IEEE.
7. Jiang, W., H. Dieter Schotten, and J.y.J.M.L.f.F.W.C. Xiang, Neural Network–Based Wireless Channel Prediction. 2020: p. 303-325.
8. Caraka, R.E., et al., Employing Best Input SVR Robust Lost Function with Nature-Inspired Metaheuristics in Wind Speed Energy Forecasting. 2020.
9. Jaddi, N.S., S. Abdullah, and A.R.J.I.S. Hamdan, Multi-population cooperative bat algorithm-based optimization of artificial neural network model. 2015. 294: p. 628-644.
10. Wei, B., et al., Visual interaction networks: A novel bio-inspired computational model for image classification. 2020. 130: p. 100-110.
11. Taqi, M.K., R.J.J.o.T. Ali, and A.I. Technology, Obka-Fs: An Oppositional-Based Binary Kidneyinspired Search Algorithm for Feature Selection. 2017. 95(1).
12. Sharma, M. and P.J.A.o.C.M.i.E. Kaur, A Comprehensive Analysis of Nature-Inspired Meta-Heuristic Techniques for Feature Selection Problem. 2020: p. 1-25.
13. Al-Yaseen, W.L.J.I.I.J.o.C.S., Improving Intrusion Detection System by Developing Feature Selection Model Based on Firefly Algorithm and Support Vector Machine. 2019. 46(4).
14. Taghian, S. and M.H.J.I.J.o.C.S.E. Nadimi-Shahraki, A Binary Metaheuristic Algorithm for Wrapper Feature Selection. 2019. 8(5): p. 168-172.
15. Connolly, J.-F., E. Granger, and R. Sabourin, An adaptive classification system for video-based face recognition. Information Sciences, 2012. 192: p. 50-70.
16. Martarelli, N.J., M.S.J.S. Nagano, and E. Computation, Unsupervised feature selection based on bio-inspired approaches. 2020. 52: p. 100618.
17. Huang, Y. and Z.J.S.o.T.T.E. He, Carbon price forecasting with optimization prediction method based on unstructured combination. 2020: p. 138350.
18. Chai, S., et al., A Hybrid Forecasting Model for Nonstationary and Nonlinear Time Series in the Stochastic Process of CO2 Emission Trading Price Fluctuation. 2020. 2020.
19. Wang, J., et al., An innovative random forest-based nonlinear ensemble paradigm of improved feature extraction and deep learning for carbon price forecasting. 2020: p. 143099.
20. Reddy, N.S. Particle Swarm Optimized Neural Network for Predicting Customer Behaviour in Digital Marketing. in 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA). 2020. IEEE.
21. Honarvar, M., M. Alimohammadi Ardakani, and M.J.A.i.O.R. Modarres, A Particle Swarm Optimization Algorithm for Solving Pricing and Lead Time Quotation in a Dual-Channel Supply Chain with Multiple Customer Classes. 2020. 2020.
22. Rhuggenaath, J., et al. A PSO-based algorithm for reserve price optimization in online ad auctions. in 2019 IEEE Congress on Evolutionary Computation (CEC). 2019. IEEE.
23. Dhiman, G. and A.J.E.A.o.A.I. Kaur, Stoa: a bio-inspired based optimization

algorithm for industrial engineering problems. 2019. 82: p. 148-174.
24. Ramli, M.R., et al. Bio-inspired Service Provisioning Scheme for Fog-based Industrial Internet of Things. in 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). 2019. IEEE.
25. Kumar, M., et al., A comprehensive survey for scheduling techniques in cloud computing. 2019. 143: p. 1-33.
26. Singh, H., S. Tyagi, and P. Kumar, Scheduling in cloud computing environment using metaheuristic techniques: A survey, in Emerging Technology in Modelling and Graphics. 2020, Springer. p. 753-763.
27. Shukor, S.A., I.M. Shaheed, and S.J.I.J.A.S.E.I.T. Abdullah, Population initialisation methods for fuzzy job-shop scheduling problems: issues and future trends. 2018. 8(4-2): p. 1820-1828.
28. Holland, J.H., adaptation in natural and artificial systems, university of michigan press, ann arbor,". 1975. 100.
29. Eberhart, R. and J. Kennedy. A new optimizer using particle swarm theory. in MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. 1995. Ieee.
30. Rashedi, E., H. Nezamabadi-Pour, and S. Saryazdi, GSA: a gravitational search algorithm. Journal of Information sciences, 2009. 179(13): p. 2232-2248.
31. Yang, X.-S., A new metaheuristic bat-inspired algorithm, in Nature inspired cooperative strategies for optimization (NICSO 2010). 2010, Springer. p. 65-74.
32. Yang, X.-S., Firefly algorithm, Levy flights and global optimization, in Research and development in intelligent systems XXVI. 2010, Springer. p. 209-218.
33. Mirjalili, S. and A. Lewis, The whale optimization algorithm. Advances in engineering software, 2016. 95: p. 51-67.
34. Mirjalili, S., S.M. Mirjalili, and A.J.A.i.e.s. Lewis, Grey wolf optimizer. 2014. 69: p. 46-61.
35. Faramarzi, A., et al., Marine predators algorithm: A nature-inspired Metaheuristic. 2020: p. 113377.
36. Hatamlou, A.J.P.i.A.I., Heart: a novel optimization algorithm for cluster analysis. 2014. 2(2-3): p. 167-173.
37. Tang, D., et al., ITGO: Invasive tumor growth optimization algorithm. 2015. 36: p. 670-698.
38. Jaddi, N.S., et al., Kidney-inspired algorithm for optimization problems. 2017. 42: p. 358-369.
39. Taherdangkoo, M., M. Yazdi, and M.H. Bagheri. Stem cells optimization algorithm. in International Conference on Intelligent Computing. 2011. Springer.
40. Pelusi, D., et al., Improving exploration and exploitation via a Hyperbolic Gravitational Search Algorithm. 2020. 193: p. 105404.
41. Cedar, S.J.N.T., Every breath you take: the process of breathing explained. 2018. 114(1): p. 47-50.
42. Fletcher, M., Nurses lead the way in respiratory care. 2007. 103(24): p. 42-43.
43. Rahnamayan, S., H.R. Tizhoosh, and M.M.J.A.S.C. Salama, Opposition versus randomness in soft computing techniques. 2008. 8(2): p. 906-918.
44. Ben-Tal, A.J.J.o.T.B., Simplified models for gas exchange in the human lungs. 2006. 238(2): p. 474-495.
45. Weibel, E.R., Symmorphosis: on form and function in shaping life. 2000: Harvard University Press.
46. Li, Q., S.-Y. Liu, and X.-S.J.A.S.C. Yang, Influence of initialization on the performance of metaheuristic optimizers. 2020: p. 106193.
47. Shilane, D., et al., A general framework for statistical performance comparison of evolutionary computation algorithms. 2008. 178(14): p. 2870-2879.

48. Bäck, T., D.B. Fogel, and Z.J.R. Michalewicz, Handbook of evolutionary computation. 1997. 97(1): p. B1.
49. Mahmood, M., B.J.P.o.E. Al-Khateeb, and N. Sciences, The blue monkey: A new nature inspired metaheuristic optimization algorithm. 2019. 7(3): p. 1054-1066.
50. Jian, C., et al., An improved chaotic bat swarm scheduling learning model on edge computing. 2019. 7: p. 58602-58610.
51. Mirjalili, S., et al., Binary bat algorithm. 2014. 25(3-4): p. 663-681.
52. Hussien, A.G., et al., New binary whale optimization algorithm for discrete optimization problems. 2020. 52(6): p. 945-959.
53. Abdel-Basset, M., L. Abdle-Fatah, and A.K.J.C.C. Sangaiah, an improved Lévy based whale optimization algorithm for bandwidth-efficient virtual machine placement in cloud computing environment. 2019. 22(4): p. 8319-8334.
54. Abd Elaziz, M. and S.J.K.-B.S. Mirjalili, A hyper-heuristic for improving the initial population of whale optimization algorithm. 2019. 172: p. 42-63.