ขั้นตอนวิธีการค้นหาของนกกาเหว่าด้วยการหาค่าที่เหมาะสมของค่าความน่าจะเป็นของพารามิเตอร์การกลายพันธุ์ในปัญหาการหาประสิทธิภาพทั่วไป

The Cuckoo Search Algorithm with Suitable Probabilistic Mutation Parameters for Global Optimization Problems

พัชระ นาเสงี่ยม[1] และ คำรณ สุนัติ[1*]

[1]สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น จ.ขอนแก่น 40002

Patchara Nasa-ngium[1] and Khamron Sunat[1*]

[1]Department of Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen, 40002 Thailand.

[*]Corresponding Author, E-mail: khamron_sunat@yahoo.com

Received: 24 July 2019 | Revised: 15 November 2019 | Accepted: 19 November 2019

บทคัดย่อ

ปัจจุบันการแก้ปัญหาเกี่ยวกับการหาคำตอบที่ดีที่สุดได้มีความยากเพิ่มขึ้น เช่น กลุ่มของปัญหาเชิงวัด CEC-2017 จึงมีความจำเป็นต้องมีการพัฒนาเทคนิคใหม่ๆ สำหรับการหาค่าที่เหมาะสม โดยการใช้อัลกอริธึมสมัยใหม่ในการค้นหา เนื่องจากอัลกอริธึมแบบดั้งเดิมไม่สามารถแก้ปัญหาเหล่านี้ได้ งานวิจัยนี้ได้ศึกษาอัลกอริธึมนกกาเหว่าที่ใช้ความใกล้เคียงกันร่วมกับการกลายพันธุ์ที่ใช้ค่าความน่าจะเป็นที่พัฒนามาจากการค้นหาจากนกกาเหว่าดั้งเดิม เพื่อนำมาแก้ปัญหาของขนาดของระยะทางการบินหารังในพื้นที่ค้นหา ซึ่งสามารถแก้ปัญหาเหล่านี้ได้ดีกว่าวิธีการเดิม จากการทดลองพบว่า การค้นหาโดยไม่ใช้ความใกล้เคียงกัน มีประสิทธิภาพมากกว่าอย่างมีนัยสำคัญ และการใช้ค่าความน่าจะเป็น p_k ที่มีค่าเท่ากับ 0.06 ให้ผลการทดลองดีที่สุด ไม่ว่าจะเป็นการใช้กับปัญหาที่มีมิติต่ำหรือสูง ซึ่งได้นำวิธีการดังกล่าวไปเปรียบเทียบกับวิธีอื่น ได้แก่ ABC, CS, PSO, FA, GSA, GWO, MVO, MFO, QPSO, LCA, NNCS เพื่อค้นหาประสิทธิภาพที่เพิ่มขึ้นของวิธีการที่นำเสนอ

ABSTRACT

As the complexity of optimization problems have increased over the last few decades, such as the benchmark functions established by the Congress on Evolutionary Computation-2017 (CEC-2017), the development of new optimization techniques has become evident more than previously. Modern algorithms are required because conventional algorithms are inadequate to solve complicated problems. The Nearest Neighbor Cuckoo Search (NNCS), with probabilistic mutation, is studied in this work. It is the improved cuckoo search algorithm using the topology of the nearest-neighbor population and probabilistic mutation to fix the step-size problem in a search space. The proposed algorithm can solve this problem without using any NN topology, and it provides a better result than the NNCS. The p_k of 0.06 was selected for both low and high dimensional problems. The proposed method has been compared with other previously-reported algorithms such as ABC, CS,

PSO, FA, GSA, GWO, MVO, MFO, QPSO, LCA, NNCS in order to investigate the improvement of efficiency over the original CS.

คำสำคัญ: การค้นหาของนกกาเหว่า การบินแบบเลวี ขั้นตอนวิธีจากธรรมชาติ

Keywords: Cuckoo search, Lévy flight, Nature-inspired algorithm

1. INTRODUCTION

Nature-inspired optimization algorithms are widely used in many classes of applications (Binitha and santhya, 2012; Fister et al., 2013; Yang, 2014). Significant efforts have been made to modify these algorithms to improve their searching abilities for a globally optimal solution, such as new distribution, adaptivity, nearest neighbor, hybrid, etc. A simple benchmark is generally employed to evaluate the performance of optimization algorithms and to understand the behavior of the algorithm. Various strategies have been proposed to modify the existing algorithms. Some strategies are a combination of several existing optimization algorithms.

In the past decade, the cuckoo search algorithm (Yang and Deb, 2009) has been improved using many techniques or by using ideas from other algorithms. Furthermore, many algorithm search skills were upgraded by using the Lévy flight distribution (Mantegna and Stanley, 1994). In 2016, the cuckoo search was tested on CEC2014 (Liang et al., 2013) and CEC2017 (Awad et al., 2016). Both benchmark functions are rotated and shifted with complicated functions, and are as complex as the real-world problem. Compared with the flower pollination algorithm, the tested cuckoo search was a better algorithm (Binh et ai., 2018). However, based on a comparison with other competitors, the cuckoo search is not the best algorithm. There are various other improved versions of cuckoo search that can find a better solution than the original CS.

Nowadays, the dimension of the optimization problem involved in many scientific research fields is getting higher. Some complicated problems contain a large set of parameters so that the conventional algorithms may be inadequate for reaching global optimization. Therefore, the development of a new algorithm for large-sized problems is of great interest. This research introduces a probabilistic mutation parameter for the CS algorithm, especially in cases of complicated problems.

2. THE CUCKOO SEARCH ALGORITHM WITH SUITABLE PROBABILISTIC MUTATION PARA-METER

In 2009, Yang and Deb proposed CS. The algorithm uses cuckoo bird behavior for reproducing the parasitic reproduction actions of a single species of cuckoo birds and an individual host species. Besides, the traditional CS uses Lévy flight as the key search principle. The search process contains three simple rules:

1. Each cuckoo randomly chooses a host nest to hatch and brood only once.
2. The best nests are extant through the next generation.
3. The number of hosts, representing convenience, and the probability of the hosts noticing the alien eggs, are fixed. If the cuckoo's egg is found, the host bird may abandon the nest to establish a new nest at a new location.

In 2016, Wang et al. (2016) modified CS to the Nearest Neighbour Cuckoo Search (NNCS) algorithm with Probabilistic Mutation so that ideas within the algorithm would use the topology of nearest neighbour to find near positions and reference the LFRW phase. This modification, therefore, cut off Lévy Flight behaviour in order to produce a random number by using zero with probabilistic mutation (see Equation 6).

With both algorithms, the researchers have proposed a simple, modified algorithm by using a traditional CS framework, and realizing a partial reduction in the Lévy Flight random number by NNCS to find a suitable parameter of probabilistic mutation (p_k), called CSPK. The variable parameter can be defined as a percentage of occurance, such as in "CSPKxx in that xx is a percent of probabilistic mutation." For example, CSPK06 is 6% and uses p_k = 0.06 in LFRW. Figure 1 shows a random number from a Lévy Flight random number, and is reduced by p_k variants that show 5,000 random numbers. For the first full random number by Lévy, one can see the frequncy of a random jumping step; but if the Lévy behavior plot is reduced, one can see a less aggressive jump in the points, which is a useful feature for the proposed alogorithm. Figure 2 shows the algorithm flowchart that is only replaced with p_k in LFRW.

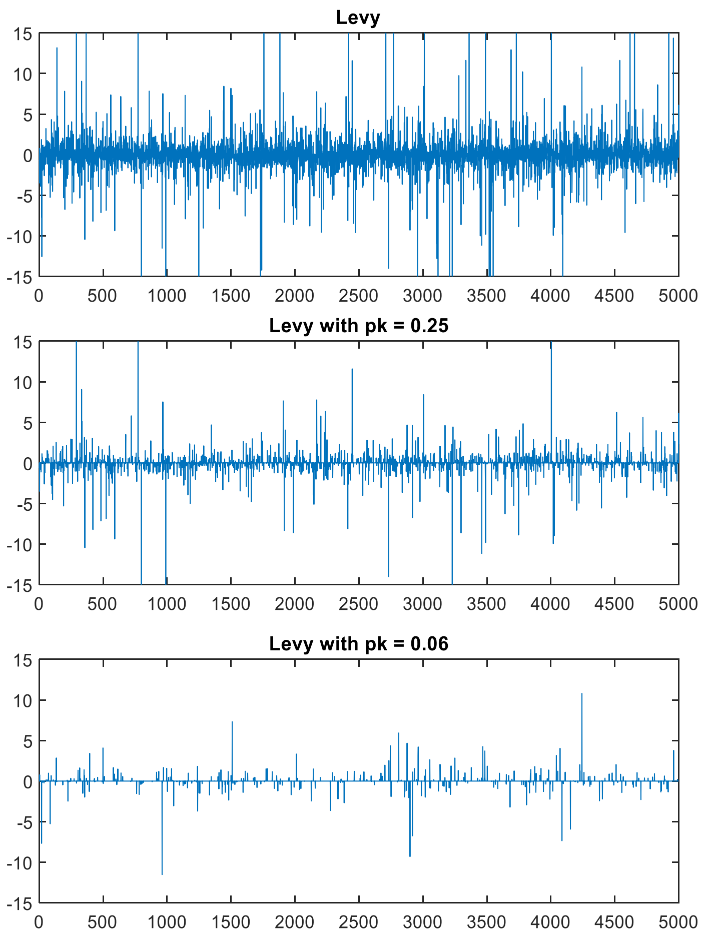

Figure 1 Lévy Flights random numbers, generated from 1 to 5,000 without p_k (1st) and with p_k (2nd, 3rd). In this case, p_k parameters are 0.25 and 0.06.

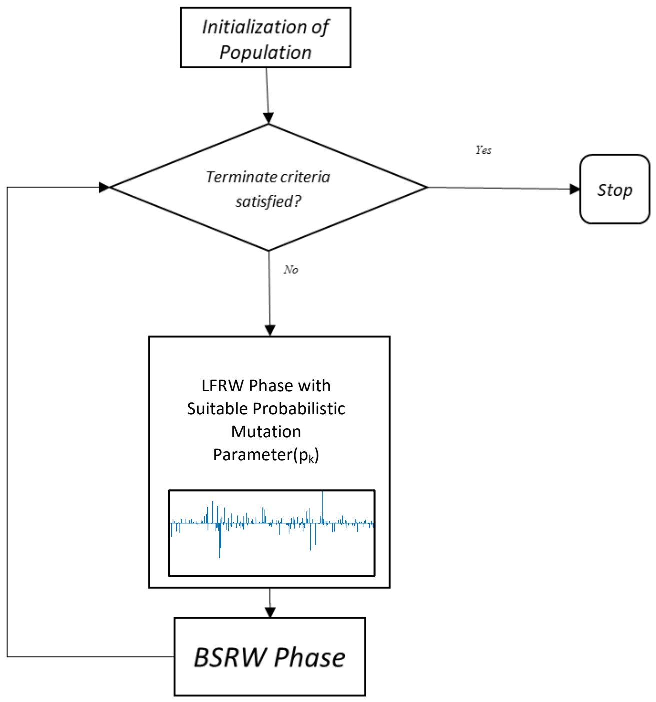
Initialization of Population
Terminate criteria satisfied?
Yes
Stop
No
LFRW Phase with Suitable Probabilistic Mutation Parameter(p_k)
BSRW Phase

Figure 2 Flow chart of replaced distribution in CSPK

3. RESEARCH METHODOLOGY

In this case of NNCS, these researchers compare the effect of nearest neighbor topology and probabilistic mutation with CEC 2017 benchmark problems.

3.1 The CEC 2017 test suite and metrics

In order to validate the performance of CS, different experiments in a suite of 30 complex unconstrained single-objective benchmark functions were carried out, according to the technical report of (Awad et al., 2016). These include unimodal functions, multimodal functions, hybrid functions, and composite functions.

3.2 Experimental verifications

A performance evaluation is conducted on the CEC-2017 Competition on Single-Objective Real Parameter Numerical Optimization that is shown in Table 1. The source code is publicly available. However, an algorithm performs these function evaluations without explicit knowledge of the structure of benchmark functions. For robust algorithms to have a high success rate, it should provide the lowest objective values. The benchmark contains 30 scalable test functions with a diverse set of characteristics, such as a large number of local optima, asymmetry, and non-separability. Besides, the functions are shifted and rotated, which means this is a tough benchmark. The functions are given for four numbers of variables (dimensions): 10 (10D), 30(30D), 50(50D) and 100 (100D), while D is the dimensionality of the problem.

Table 1 CEC-2017 Competition on Single-Objective Real Parameter Numerical Optimization

Typology	ID	Function Name	Optima, f_i^*
Unimodal Functions	1	Shifted and Rotated Bent Cigar function	100
	2	Shifted and Rotated sum of Differential Power Function	200
	3	Shifted and Rotated Zakharov function	300
Simple Multimodal Functions	4	Shifted and Rotated Rosenbrock’s function	400
	5	Shifted and Rotated Rastrigin’s function	500
	6	Shifted and Rotated Expanded Scaffer’s F6 function	600
	7	Shifted and Rotated Lunacek Bi_Rastrigin function	700
	8	Shifted and Rotated Non-Continuous Rastrigin’s function	800
	9	Shifted and Rotated Lévy function	900
	10	Shifted and Rotated Schwefel’s function	1000
Hybrid Functions	11	Hybrid Function 1 (N=3) Zakharov; Rosenbrock; Rastrigin	1100
	12	Hybrid Function 2 (N=3) High-conditioned Elliptic, Modified Schwefel, Ben Cigar	1200
	13	Hybrid Function 3 (N=3) Bent Cigar; Rosenbrock; Lunacek bi-Rastrigin	1300
	14	Hybrid Function 4 (N=4) High-conditioned Elliptic; Ackley; Schaffer F7; Rastrigin	1400
	15	Hybrid Function 5 (N=4) Bent Cigar; HGBat; Rastrigin; Rosenbrock	1500
	16	Hybrid Function 6 (N=4) Expanded Schaffer F6; HGBat; Rosenbrock; Modified Schwefel	1600
	17	Hybrid Function 6 (N=5) Katsuura; Ackley; Expanded Griewank plus; Rosenbrock; Schwefel; Rastrigin	1700
	18	Hybrid Function 6 (N=5) High-conditioned Elliptic; Ackley; Rastrigin; HGBat; Discus	1800
	19	Hybrid Function 6 (N=5) Bent Cigar; Rastrigin; Griewank plus Rosenbrock; Weierstrass; Expanded Schaffer F6	1900
	20	Hybrid Function 6 (N=6) HappyCat; Katsuura; Ackley; Rastrigin; Modified Schwefel; Schaffer F7	2000
Composition Functions	21	Composition Function 1 (N=3) Rosenbrock; High-conditioned Elliptic; Rastrigin	2100
	22	Composition Function 2 (N=3) Rastrigin; Griewank; Modified Schwefel 2200	2200
	23	Composition Function 3 (N=4) Rosenbrock; Ackley; Modified Schwefel; Rastrigin	2300
	24	Composition Function 4 (N=4) Ackley; High-conditioned Elliptic; Griewank; Rastrigin	2400
	25	Composition Function 5 (N=5) Rastrigin; HappyCat; Ackley; Discus; Rosenbrock	2500
	26	Composition Function 6 (N=5) Expanded Schaffer F6; Modified Schwefel; Griewank; Rosenbrock; Rastrigin	2600
	27	Composition Function 7 (N=6) HGBat; Rastrigin; Modified Schwefel; Bent Cigar; High-conditioned Elliptic; Expanded	2700
	28	Composition Function 8 (N=6) Ackley; Griewank; Discus; Rosenbrock; HappyCat; Expanded Schaffer F6	2800
	29	Composition Function 9 (N=3) f15; f16; f17	2900
	30	Composition Function 10 (N=3) f15; f18; f19	3000

For CS, there are three control parameters, namely, the population size N, the fraction probability p_a, and the mutation probability p_k. For all experiments, unless a change is mentioned, the population size is N. The parameter D is the dimension of the problem, and the mutation probability p_k is 0.25 for NNCS and is varied in CSPK. Moreover, in this experiment, each algorithm is used to optimize each benchmark function over 52 independent runs.

In this paper, the following values were chosen for all algorithms' parameters: population sizes of $N=50$, without size adjustment. Further, the maximum number of objective function evaluations, max_nfes, is set as $D\times 10,000$; and the number of independent runs for each combination of function and dimension is set to 52.

This research will study the presented algorithms (those solving CEC-2 0 1 7 problems) as to their performance. Three performance criteria have been selected for evaluating the algorithmic performance which are: $score_1$, $score_2$ and total $score$. In the following measurements, a higher score is better, and the scores are defined mathematically by the following equations.

$$score_1=(1-\frac{SE-SE_{min}}{SE})\times 50 \tag{1}$$

In the above equation, $SE_{\min}$ represents the least sum or errors from all the algorithms. SE is the total sum of error values for all of the dimensions and is defined below:

$$SE=0.1\times\sum_{i=1}^{30}ef_{10D}+0.2\times\sum_{i=1}^{30}ef_{30D}+0.3\times\sum_{i=1}^{30}ef_{50D}+0.4\times\sum_{i=1}^{30}ef_{100D} \tag{2}$$

In the above equation, ef_{nD} represents the last objective value, less the optimum. This applies to all the functions of the *n* dimensions.

$$score_2=(1-\frac{SR-SR_{min}}{SR})\times 50 \tag{3}$$

Here, $SR_{\min}$ represents the least possible sum or ranking from all the algorithms. Further, SR is the total of ranking as defined below:

$$SR=0.1\times\sum_{i=1}^{30}rank_{10D}+0.2\times\sum_{i=1}^{30}rank_{30D}+0.3\times\sum_{i=1}^{30}rank_{50D}+0.4\times\sum_{i=1}^{30}rank_{100D}. \tag{4}$$

In the above equation, please note that $rank_{nD}$ is the ranking which has, as its foundation, the resulting values for all n dimensions that are included in the functions. In this case, a lower value is preferable.
Lastly, one can define score as shown below:

$$score=score_1+score_2 \tag{5}$$

A 5% level is used by the Wilcoxon signed-rank test to reveal important differences between the two algorithms presented.

3.3 Parameter Settings

In order to maintain a reliable and fair comparison, (1) the parameter settings are the same as above for all experiments unless the study mentions new settings to serve the purpose of that parameter study. (2) For all conducted experiments, the reported values are the average of the results for 52 independent runs, and (3) further fitness evaluations are required. From the Lévy Flights phase in conformity with cuckoo search, the state used in probabilistic mutation was defined by p_k, and was compared with a uniform random number from [0,1]. The equation of next nest generator can be explained as follows:

$$u_{i,j,G} = \begin{cases} x_{i,j,G} + r \bullet Levy \otimes (x_{i,j,G} - x_{i,j,best}), & \text{rand} < p_k \\ x_{i,j,G}, & \text{otherwise} \end{cases} \quad (6)$$

where r presents a varied scaling factor that draws a uniform distribution in the interval of [0,1], and G is the current generation of cuckoo birds. The boundary will be checked when the position is out of bounds, and the reflecting boundary is used for setting new points near the boundaries.

3.4 Competitive algorithms

In this work, various types of conventional competitive algorithms reported in the years from 2004 to 2016, as summarized in Table 2, and were used for comparing the performance with the proposed method.

Table 2 List of Competitive algorithms ordered by year

Abbreviation	Name	Reference	Year
QPSO	Particle Swarm Optimization with Particles having Quantum Behavior	(Sun et al., 2004)	2004
ABC	Artificial Bee Colony	(Karaboga and Basturk, 2007)	2007
CS	Cuckoo Search	(Yang and Deb, 2009)	2009
GSA	Gravitational Search Algorithm	(Rashedi et al., 2009)	2009
FA	Firefly Algorithm	(Yang, 2010)	2010
AMO	Animal Migration Optimization	(Li et al., 2014)	2014
LCA	League Championship Algorithm	(Kashan, 2014)	2014
GWO	Grey Wolf Optimizer	(Mirjalili et al., 2014)	2014
MFO	Moth-Flame Optimization	(Mirjalili, 2015)	2015
NNCS	Nearest Neighbour Cuckoo Search Algorithm with Probabilistic Mutation	(Wang et al., 2016)	2016
MVO	Multi-Verse Optimizer	(Mirjalili et al., 2016)	2016
CSPK06	Cuckoo Search Algorithm with Probabilistic Mutation ($p_k = 0.06$)	-	-

4. RESULTS AND DISCUSSION

4.1 NNCS vs. CSPK

By comparing NNCS with CSPK, using $p_k = 0.25$, it can be concluded that the conventional CS provided better results when best nest was used instead of the nearest neighbor topology. However, combining it with Probabilistic Mutation, as shown in Table 3, a significant win was observed. In addition, CSPK gave better results when the dimension or problem was more difficult i.e. in 30 or 50 dimensions. Therefore, the CSPK was adopted in further experiments f:or finding the best value of p_k.

Table 3 Determination of the cuckoo search with an operator that statistically provides the best solution for each benchmark problem in CEC 2017 by utilizing the two-sided Wilcoxon Signed-Rank Test ($\alpha = 0.05$).

NNCS vs. CSPK	Win|Tide|Loss
− = + − + + = + + + = = = + = + + − + + = = − + = − − − = +	+13| =10| -7
− = + = + + + + + = + = + = + = = = + = = + + + = − + + + +	+17| =11| -2
− = = = + + = = + = = + + = + = + = = = + = = + = − + = = +	+11| =17| -2
− = = = + + = = + − = + = = − = = = − + + − − = = + + + + +	+11| =13| -6
Total	+52| =51|-17

4.2 Comparison of the variant parameter of p_k

From the experiment, the p_k were varied according to the values in Table 2. It can be seen in Figure 2 that when the p_k value is less than 0.10, the overall performance is better. The most effective performance was observed when the increment of 1 was applied, and the best performance was obtained at $p_k = 0.6$.

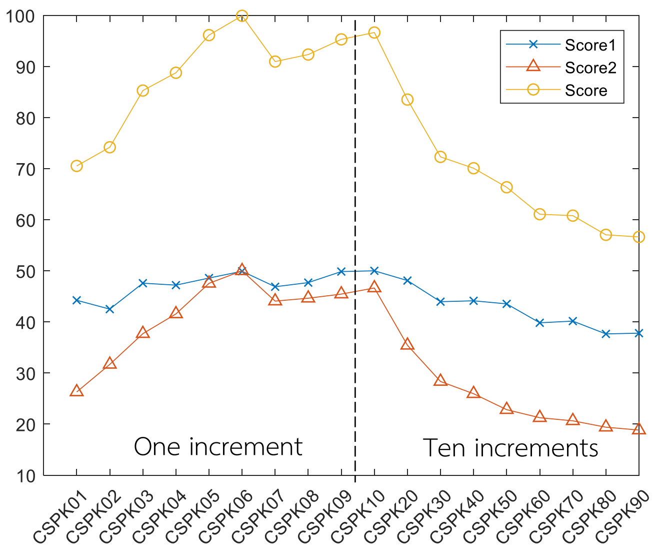

Figure 3 Comparison of $score_1$, $score_2$ and $score$ using different p_k values from 0.01 to 0.9

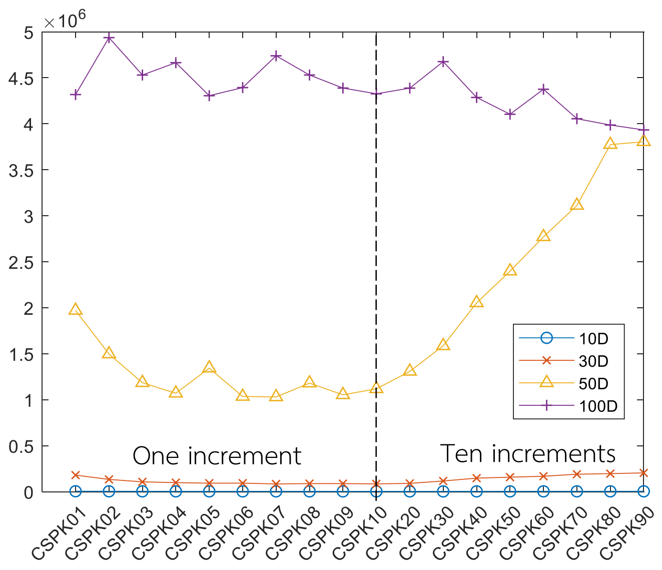

Figure 4 Comparison of sum of errors from F1 to F30 in each dimension with different p_k values from 0.01 to 0.9; blue = 10D, red = 30D, yellow = 50D and violet = 100D

In Figure 3, the searching ability of the algorithm was decreased when applying it with more difficult problems or increasing the dimension, as it can be seen that the obtained result was far from the best value. In 100D, the sum of all the values was up to four to five million runs. However, the proportion of all obtained values is required in the measurement. In 10D and 30D, the lower obtained values were also observed. Having more problems or higher dimensions, such as 50D in Figure 4, the algorithm is more reliable and the results reveal more information about the strengths of the evaluated algorithms. Considering the experiment in using p_k of a lower value than 0.1, the proposed algorithm works well. However, the search performance was decreased when increasing the dimension of the problems, as can be observed in 20–90 dimensions. However, there was no significant difference at the very high dimension of 100D.

By comparing using the $score_1$ as a result, the maximum total score was obtained at $p_k = 0.10$. The value of second rank, at $p_k = 0.06$, was also close to the maximum. On the other hand, all ratings are ranked at 1 in the case of $score_2$. By combining $score_1$ and $score_2$, the first rating was found at $p_k = 0.06$.

4.3 Comparison of CSPK06 with other reported algorithms

CSPK was compared with other previously-reported algorithms, as summarized in Table 4 which shows the first place of every score. The $score_1$ shows the difference of a large value from the second place with 7.92095, and with the last place of almost 50 points, showing the performance of the algorithm. The $score_2$ shows the large gap of sum-of-ranked that is far from AMO with 8.185, showing the performance of rank almost as high as the high ranking in every function. These results have shown that every algorithm can be the highest and lowest in rankings, but the proposed method can be at the head of the group, taking the score of 50 as a result. In Table 5, it can be seen that the CSPK wins up to 104 problems and loses only 7 problems; and this is with the traditional CS that shows the large improvement of modification. Therefore, it can be concluded that the efficiency is decreased when the Lévy is frequently varied.

Table 4 Score obtained from comparison with other reported algorithms (Bold signifies winner.)

Algorithm	score_1	Algorithm	score_2	Algorithm	score
CSPK06	**50**	**CSPK06**	**50**	**CSPK06**	**100**
AMO	42.07905	AMO	41.81495	AMO	83.894
NNCS	40.28968	LCA	34.66077	NNCS	73.9332
CS	38.31922	NNCS	33.64352	FA	64.54064
FA	36.82838	QPSO	32.23594	CS	60.98074
QPSO	26.24704	FA	27.71226	QPSO	58.48298
GSA	12.88641	MVO	24.15211	LCA	43.15288
LCA	8.492114	CS_1	22.66152	GSA	31.92245
ABC	2.53204	ABC	21.24774	MVO	25.95393
MVO	1.801825	GSA	19.03605	ABC	23.77978
GWO	0.005012	GWO	17.95264	GWO	17.95765
MFO	0.001139	MFO	14.24674	MFO	14.24788

Table 5 Determination of the cuckoo search with an operator that statistically provides the best solution for each benchmark problem in CEC 2017 by utilizing the two-sided Wilcoxon Signed-Rank Test ($\alpha = 0.05$). The winner is cuckoo search with CSPK06.

CSPK06 VS CS	**Win |Tide| Loss**
+ = − = + + + + + + + − + + + + + + + + + + + + + + − − + +	+ 24| =2 | -4
+ = + − + +	+ 28| =1 | -1
+ = + = + + − + +	+ 27| =2 | -1
+ = + + + + + + + + + = + − + + + = + + + + + + = + + + + +	+ 25| =4 | -1
Total	+104| =9 | -7

Based on the results obtained by applying other algorithms, CSPK provided higher values. If optimization is carried out for this problem, the results will show the best for all--$score_1$, $score_2$ and $score$; with a significant and different score of 16.68566 from all other conventional algorithms. Some tested algorithms could not be used, such as GWO, MFO, MVO, ABC, which indicated that these algorithms could not find the optimum value at all. Suggested by $score_2$, it can be seen that the MVO has a better ranking value, in that there may be only some functions that cannot be used. This shows that the benchmark test suite can represent somewhat of a trap in some functions and increase the complications and complexity for the search strategy on each algorithm.

4.4 The Comparison of error value with CS and second rank (AMO)

The error value is the raw data from the result of the benchmark function. This compared the performance with the other one, as tested with CEC-2017. Generally, if the algorithm has better performance, the error value should be lower. For more information, Table 6 shows the average error of the interested algorithm for demonstrating real performance and comparisons with other papers. The cell shows the best in green shading, and the worst in red. In previous sub-sections it is shown that the performance of CSPK06 and CS demonstrates that CSPK can perform most functions, but it cannot tell that CS is a worse algorithm. One can see that in Table 6, CS is not the worst with every function when compared with second place (AMO). CS can win in many functions such as with 10D and 30D, but it loses at 100D, showing that traditional CS has a problem at higher dimensions. On the other hand, the proposed CSPK06 can be the winner indicated by green shading in almost every dimensional problem; and, it can fix CS at a high dimension of 100D. In the results, the AMO is a good algorithm but it has shown that many red-shaded cells in the table prove that AMO is not a flexible algorithm for all problems such as 100D, F14, and F18. These result in an error over 7.58E+05 and 1.35E+06, which can result in lost performance per $score_1$.

In the present work, it can be seen that using parameters that are suitable for the problem of interest will enable one to gain more efficiency. Considering the high dimensions or difficult problems, variations of Lévy Flight during the optimization process is very important and influences the search ability.

Table 6 The average of fitness function by compared with proposed method (CSPK06), traditional cuckoo search (CS) and second rank (AMO). Determination of the cuckoo search with an operator that statistically provides the best results.

10D	CS	AMO	CSPK06
F1	1.32E+00	5.10E+00	3.63E-02
F2	0.00E+00	0.00E+00	0.00E+00
F3	9.12E-03	2.77E-08	7.27E-02
F4	2.48E-01	1.51E+00	1.85E-01
F5	1.66E+01	6.07E+00	8.44E+00
F6	5.19E+00	0.00E+00	3.44E-02
F7	2.92E+01	1.77E+01	2.00E+01
F8	1.86E+01	6.58E+00	9.59E+00
F9	4.69E+01	0.00E+00	1.35E-01
F10	6.49E+02	4.41E+02	3.20E+02
F11	4.64E+00	3.03E+00	2.83E+00
F12	3.14E+02	1.29E+04	4.68E+02
F13	1.01E+01	2.68E+01	7.53E+00
F14	1.23E+01	3.05E+00	3.29E+00
F15	2.36E+00	1.65E+00	1.42E+00
F16	6.45E+00	1.06E+00	1.98E+00
F17	2.87E+01	6.27E+00	5.70E+00
F18	1.01E+01	3.76E+01	3.93E+00
F19	2.10E+00	1.19E+00	9.89E-01
F20	2.74E+01	6.41E-04	2.74E+00
F21	1.05E+02	1.61E+02	9.95E+01
F22	8.09E+01	1.00E+02	6.58E+01
F23	3.18E+02	3.07E+02	2.91E+02
F24	1.26E+02	2.83E+02	1.02E+02
F25	2.74E+02	4.09E+02	2.41E+02
F26	2.06E+02	3.00E+02	1.58E+02
F27	3.90E+02	3.92E+02	3.98E+02
F28	2.91E+02	3.12E+02	4.78E+02
F29	2.79E+02	2.65E+02	2.62E+02
F26	2.06E+02	3.00E+02	1.58E+02

30D	CS	AMO	CSPK06
F1	9.81E+00	4.11E+00	1.79E-05
F2	2.23E+11	0.00E+00	0.00E+00
F3	3.30E+04	4.09E+03	2.80E+04
F4	6.52E+01	3.12E+00	4.67E+01
F5	1.36E+02	5.34E+01	7.64E+01
F6	3.99E+01	6.58E-10	1.46E-02
F7	1.58E+02	8.94E+01	1.02E+02
F8	1.30E+02	5.73E+01	7.85E+01
F9	3.86E+03	1.33E-01	8.20E+02
F10	3.68E+03	3.71E+03	2.41E+03
F11	9.11E+01	5.32E+01	3.53E+01
F12	8.74E+04	3.84E+04	5.23E+04
F13	4.22E+02	6.25E+03	1.12E+02
F14	7.01E+01	2.71E+03	5.22E+01
F15	8.31E+01	3.44E+02	2.65E+01
F16	9.39E+02	5.27E+02	5.13E+02
F17	2.96E+02	8.74E+01	1.04E+02
F18	5.22E+03	1.44E+05	4.05E+03
F19	3.14E+01	1.15E+03	1.74E+01
F20	4.00E+02	1.70E+02	1.60E+02
F21	3.21E+02	2.53E+02	2.55E+02
F22	1.20E+03	1.00E+02	1.00E+02
F23	4.90E+02	3.96E+02	4.16E+02
F24	5.43E+02	4.70E+02	4.99E+02
F25	3.85E+02	3.87E+02	3.84E+02
F26	1.11E+03	1.48E+03	3.32E+02
F27	5.25E+02	5.15E+02	5.00E+02
F28	3.75E+02	3.04E+02	5.00E+02
F29	9.50E+02	5.34E+02	5.71E+02
F26	1.11E+03	1.48E+03	3.32E+02

Table 6 The average of fitness function by compared with proposed method (CSPK06), traditional cuckoo search (CS) and second rank (AMO). Determination of the cuckoo search with an operator that statistically provides the best results. (continues)

50D	CS	AMO	CSPK06
F1	2.64E+03	1.32E+03	3.49E-01
F2	8.33E+22	6.44E+01	0.00E+00
F3	1.18E+05	3.12E+04	9.59E+04
F4	6.63E+01	6.81E+01	4.19E+01
F5	2.98E+02	1.34E+02	1.69E+02
F6	5.63E+01	3.45E-05	9.76E-03
F7	3.91E+02	1.93E+02	2.11E+02
F8	2.96E+02	1.42E+02	1.70E+02
F9	1.52E+04	4.34E+00	6.59E+03
F10	6.83E+03	6.83E+03	4.64E+03
F11	1.79E+02	9.33E+01	7.84E+01
F12	1.42E+06	5.66E+05	8.16E+05
F13	4.48E+03	8.83E+02	4.54E+02
F14	1.95E+02	2.95E+04	1.21E+02
F15	3.60E+02	3.13E+03	7.31E+01
F16	1.81E+03	9.78E+02	1.10E+03
F17	1.33E+03	7.14E+02	7.52E+02
F18	1.06E+05	5.59E+05	8.10E+04
F19	8.67E+01	1.29E+04	3.91E+01
F20	1.19E+03	5.24E+02	6.00E+02
F21	4.82E+02	3.38E+02	3.70E+02
F22	7.37E+03	6.20E+03	5.08E+03
F23	7.46E+02	5.68E+02	5.99E+02
F24	8.02E+02	6.09E+02	7.26E+02
F25	4.87E+02	5.72E+02	4.86E+02
F26	4.15E+03	2.58E+03	2.08E+03
F27	7.61E+02	6.06E+02	5.00E+02
F28	4.61E+02	4.98E+02	5.00E+02
F29	1.65E+03	6.55E+02	1.01E+03
F30	8.66E+05	8.53E+05	1.59E+04

100D	CS	AMO	CSPK06
F1	3.25E+03	1.44E+03	8.21E-01
F2	1.19E+70	3.77E+10	0.00E+00
F3	4.01E+05	1.79E+05	3.30E+05
F4	2.01E+02	1.08E+02	1.78E+02
F5	7.95E+02	4.42E+02	4.86E+02
F6	7.10E+01	3.26E-02	2.47E-02
F7	1.30E+03	5.97E+02	6.01E+02
F8	7.94E+02	4.12E+02	4.92E+02
F9	4.27E+04	2.85E+03	2.87E+04
F10	1.72E+04	1.86E+04	1.36E+04
F11	1.31E+03	6.41E+02	5.92E+02
F12	3.16E+06	2.16E+06	2.90E+06
F13	6.99E+03	2.96E+03	1.28E+03
F14	9.04E+04	7.58E+05	1.28E+05
F15	2.31E+03	6.04E+02	3.93E+02
F16	4.59E+03	3.34E+03	3.09E+03
F17	3.23E+03	2.42E+03	2.36E+03
F18	1.02E+06	1.35E+06	9.48E+05
F19	1.66E+03	1.67E+03	1.92E+02
F20	3.40E+03	2.33E+03	2.17E+03
F21	9.80E+02	6.31E+02	6.83E+02
F22	1.87E+04	1.95E+04	1.52E+04
F23	1.22E+03	8.22E+02	8.75E+02
F24	1.72E+03	1.22E+03	1.36E+03
F25	7.47E+02	8.14E+02	7.39E+02
F26	1.19E+04	8.33E+03	8.21E+03
F27	9.60E+02	8.66E+02	5.00E+02
F28	5.68E+02	5.62E+02	5.00E+02
F29	4.50E+03	2.73E+03	2.89E+03
F30	8.45E+03	5.46E+03	4.22E+02

5. CONCLUSIONS

In this work, conventional CS was improved for application to complex optimization problems. The performance significantly increased with a few modifications of algorithms that show flexibility in many functions, while not finding large errors. The experiments revealed that reducing the working value in some dimensions may increase the efficiency of the Lévy, which may affect the search ability of the algorithm. The key process of CS algoritim has not only been the Lévy Flight, but the BSRW which also has importance. In fact, it is a key success of the proposed algorithm with differences of traditional CS. This CS is a non-changing position for BSRW if one sees that the Lévy with $p_k = 0.06$. Many dimensions cannot be changed which are again connected to BSRW in the next generation with old information. Further, this new pair can increase the growth of the performance of the BSRW phase. Moreover, in the high dimension, the performance of the low dimensional change can fix the curse of the dimension and find the best value in the next generation. This finding could be useful for other applications in the future. In addition, an automatic adjustment may also be possible.

6. ACKNOWLEDGEMENTS

The authors would like to thank the Department of Computer Science, Faculty of Science, at Khon Kaen University for supporting all the facilities used in this research.

7. REFERENCES

Awad, N.H., Ali, M.Z., Liang, J.J., Qu, B.Y., and Suganthan, P.N. (2016). Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization. Tech. Rep.

Binh, H.T., Hanh, N.T. and Dey, N. (2018). Improved cuckoo search and chaotic flower pollination optimization algorithm for maximizing area coverage in wireless sensor networks. Neural computing and applications 30: 2305-2317.

Binitha, S. and Sathya, S.S.(2012). A survey of bio inspired optimization algorithms. International Journal of Soft Computing and Engineering 2: 137-151.

Del Ser, J., Osaba, E., Molina, D., Yang, X.-S., Salcedo-Sanz, S., Camacho, D., Swagatam, S., Ponnuthurai, N., Coello, C., Carlos, A. and Herrera, F. (2019). Bio-inspired computation: Where we stand and what's next. Swarm and Evolutionary Computation.

Fister Jr, I., Yang, X.-S., Fister, I., Brest, J., and Fister, D. (2013). A brief review of nature-inspired algorithms for optimization. arXiv preprint arXiv: 1307.4186.

Karaboga, D., and Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of global optimization 39: 459-471.

Kashan, A.H. (2014). League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships. Applied Soft Computing 16: 171-200.

Li, X., Zhang, J., and Yin, M. (2014). Animal migration optimization: an optimization algorithm inspired by animal migration behavior. Neural Computing and Applications 24: 1867-1877.

Liang, J.J., Qu, B.Y., and Suganthan, P.N. (2013). Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore.

Mantegna, R.N., and Stanley, H.E. (1994). Stochastic process with ultraslow convergence to a Gaussian: the truncated Lévy flight. Physical Review Letters 73: 2946.

Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowledge-Based Systems 89: 228-249.

Mirjalili, S., Mirjalili, S.M., and Hatamlou, A. (2016). Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Computing and Applications 27: 495-513.

Mirjalili, S., Mirjalili, S. M., and Lewis, A. (2014). Grey wolf optimizer. Advances in engineering software 69: 46-61.

Rashedi, E., Nezamabadi-Pour, H., and Saryazdi, S. (2009). GSA: a gravitational search algorithm. Information sciences 179: 2232-2248.

Sun, J., Feng, B., and Xu, W. (2004). Particle swarm optimization with particles having quantum behavior. Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753), 1. pp. 325-331.

Wang, L., Zhong, Y., and Yin, Y. (2016). Nearest neighbour cuckoo search algorithm with probabilistic mutation. Applied Soft Computing 49: 498-509.

Yang, X.-S. (2010). Firefly algorithm, stochastic test functions and design optimisation. arXiv preprint arXiv: 1003.1409.

Yang, X.-S. (2014). Nature-inspired optimization algorithms. Elsevier.

Yang, X.-S., and Deb, S. (2009). Cuckoo search via Lévy flights. 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC). pp. 210-214.

❑❑❑❑❑