J Inf Process Syst, Vol.16, No.2, pp.478~493, April 2020
https://doi.org/10.3745/JIPS.04.0168

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

PSA: A Photon Search Algorithm

Yongli Liu* and Renjie Li*

Abstract

We designed a new meta-heuristic algorithm named Photon Search Algorithm (PSA) in this paper, which is motivated by photon properties in the field of physics. The physical knowledge involved in this paper includes three main concepts: Principle of Constancy of Light Velocity, Uncertainty Principle and Pauli Exclusion Principle. Based on these physical knowledges, we developed mathematical formulations and models of the proposed algorithm. Moreover, in order to confirm the convergence capability of the algorithm proposed, we compared it with 7 unimodal benchmark functions and 23 multimodal benchmark functions. Experimental results indicate that PSA has better global convergence and higher searching efficiency. Although the performance of the algorithm in solving the optimal solution of certain functions is slightly inferior to that of the existing heuristic algorithm, it is better than the existing algorithm in solving most functions. On balance, PSA has relatively better convergence performance than the existing metaheuristic algorithms.

Keywords

Evolutionary Algorithm, Genetic Algorithm, Meta Heuristic, Physical Properties, Photon Search

1. Introduction

Optimizing problems are ubiquitous around us, for instance, traveling salesman problem, bin-packing problem, assignment problem, flow shop scheduling problem, etc. It is well known that these problems are NP-hard, which makes it almost impossible to obtain the optimal solution through various enumeration methods and precise algorithms under the existing computing capacity. Therefore, compared with the traditional optimization algorithm, the meta-heuristic algorithm is proposed, which is a mixture of stochastic algorithm and local search algorithm.

With the intention of developing a new meta-heuristic algorithm, researchers seek inspiration from the fields of biology, physics, and sociology. This makes the study of optimization contains many algorithms, such as particle swarm optimization (PSO) [1], artificial bee colony (ABC) [2], whale optimization algorithm (WOA) [3], ant colony optimization [4], wasp swarm algorithm [5], Monkey Search [6], ageist spider monkey optimization (ASMO) [7], and wolf colony search algorithm [8], etc. These algorithms developed from the mechanism of biological evolution and population cooperation and improved the optimization effect compared with the traditional optimization algorithms. In addition to biology, physical phenomena are also the ideal source of meta-heuristics. Algorithms inspired from these phenomena include simulated annealing (SA) [9], gravity search algorithm (GSA) [10], galaxy based search algo-

Manuscript received December 6, 2018; first revision August 19, 2019; accepted September 26, 2019.

Corresponding Author: Yongli Liu (yongli.buaa@gmail.com)
* School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, Henan, China (yongli.buaa@gmail.com, rareleon@gmail.com)

rithm (GBSA) [11], charged system search (CSS) [12], central force optimization (CFO) [13], black hole algorithm (BH) [14], etc. In addition to developing new algorithms, scholars are also focusing on developing a range of variations on existing algorithms or propose a variant of the algorithm to solve a major problem in an algorithm. For example, since the ABC converges slowly, Huang et al. [15] proposed an enhanced version of ABCIS (artificial bee colony algorithm with special division and intellective search). Moreover, in order to resolve the parameter selection problem of support vector machine (SVM) algorithm, Zhao and Long [16] proposed an improved particle swarm optimization algorithm to optimize the parameters of SVM. To improve exist algorithms, Song et al. [17] proposed a variant of artificial bee colony algorithm which is highly efficient and two-strategy adaptive. Chen et al. [18] proposed ABC-EO II and IABC-EO II, the updated version of ABC-EO and IABC-EO, where extremal optimization (EO) is introduced to these two algorithms in dissimilar ways. Li et al. [19] proposed a two-level imperialist competitive algorithm (TICA), in which two levels refers to the corresponding strongest empire and other empires.

In this paper, we developed the photon search algorithm, which is based on knowledges of physics. Without ambiguity, we abbreviate it as PSA. In the algorithm proposed, we give some formulas according to the behaviors designed for the algorithm, these formulas initialize the position, velocity, and distance of photons (search agents) in each iteration. These formulas are inspired by the photon hypothesis and quantum physics. Further, we present the pseudo-code of PSA. However, studies on optimization are most concerned about the performance of algorithms. Therefore, so as to show various properties of PSA, the experiments solving 23 functions are designed. The results of experiments indicate that the PSA has advantages compared to other algorithms.

The overview of the paper are as follows. We present inspiration, mathematical expression and algorithm design in Section 2. In Section 3, functions for tests and experimental results are described in detail. Section 4 shows the improvement of PSA named UFPSA and presents the experimental results of this algorithm under unimodal functions. In Section 5, we summarize the algorithms and give our proposal and prospect for future research.

2. Photon Search Algorithm

We will describe the core design of the PSA algorithm in this section. The inspiration of the algorithm proposed will be introduced firstly, and then the corresponding mathematical expression will be given.

2.1 Inspiration Source

On December 14, 1900, at a regular meeting of German physics, Max Planck made an impressive presentation [20]. In his report, he made the point that electromagnetic energy could be emitted only in quantized form, in other words, the energy could only be a multiple of an elementary unit. This idea proposed by Planck marked the birth of the quantum theory.

It is 5 years since the quantum theory was first proposed, it has received little notice, but it eventually caught Einstein's attention. In March 1905, Einstein [21] proposed the quantum hypothesis. He thought that the propagation of light in space is also discontinuous, and he named this discontinuous energy unit “light quantum”. And, he argues, for the average value of time, light behaves like a wave; for the instantaneous value of time, the light behaves as a particle. This is the first time in history to reveal the unity of microscopic object volatility and particle property, namely wave-particle duality.

After that, de Broglie [22] made a point in 1924: microscopic particles also have volatility. He connected the properties of microscopic particles with the so-called de Broglie relation based on the relationship between light waves and photons. Frequency v and wavelength λ are respectively:

$$v=E/h,\ \lambda=h/p \tag{1}$$

According to the statistical interpretation put forward by Bohn, the wave represents the probability of the appearance of particles at various positions in space, and the diffusion of the wave packet is actually the diffusion of particle probability, rather than the disintegration of particles themselves. To some extent, this interpretation solves the contradiction between volatility and particle property, which is now accepted by most scholars.

However, in the process of the development of quantum mechanics, we must mention Wolfgang Pauli, whose Pauli Exclusion Principle has also contributed to the development of microphysics.

It is under the inspiration of these physical theories the algorithm PSA proposed by this paper.

2.2 Mathematical Model

The mathematical expression of photon motion and mathematical expression of observation based on the uncertainty principle will be presented in this section. Inspired by these knowledges, we propose photon search algorithm (PSA).

The design of PSA applies a lot of knowledges related to quantum mechanics. Inspired by these knowledges, this paper transforms these knowledges into the main algorithmic behavior and controls the search process of the algorithm. The three main behaviors are Photon motion, Observation, and Search exclusion principle, which will be explained in detail in the next section.

Later, we combined the three components in Section 2.3 to show how the PSA works.

2.2.1 Photon motion

In PSA, each photon is a search agent, and they will be targeted at the current optimal solution value at the beginning of generation and obtain a uniform fixed speed. After each movement, the photons submit their solution values to the algorithm. The algorithm will compare all the received solutions with the current optimal solution and leave the superior solution. When the algorithm runs to the termination condition, it will finish running and output the obtained optimal solution value.

According to Einstein's Principle of Light Speed being Constant, it is easy to know that in any space, no matter who is observing at any time, light travels at about 300 million meters per second, and the speed is independent of the state of motion of the emitter. This also means that if the observation intervals are the same, the distance traveled by the light is unchanged.

Consider an environment with N photons (agents). The position of the ith agent can be defined by:

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n) \text{ for } i = 1,2,\dots,N \tag{2}$$

where x_i^d is the exact position of ith search agent in dth dimension.

It is easy to know that the Euclidian distance between ith photon and current global optimum X_g at the specific time t is:

$$R_{ig} = ||X_i(t), X_g(t)||_2 \tag{3}$$

However, since the distance of light moving at a fixed time is constant, the distance R_{Len} that the photon passes through in an iteration is equal to the scale of the distance between the most distant points in the search space, in this paper:

$$R_{Len} = Scl \cdot ||X_{Upper} - X_{Lower}||_2 \tag{4}$$

where *Scl* represents the scale value (default value is 0.1), X_{Upper} is the point takes the upper bound value on all dimensions, and X_{Lower} is the opposite one.

Hence, for certain time "*t*", the speed *v* in *d*th dimension of photon "*i*" is defined as following:

$$v_i^d(t+1) = \frac{R_{Len}}{R_{ig}} \cdot (x_g^d(t) - x_i^d(t)) \tag{5}$$

Furthermore, the location of photon "*i*" is updated by the following formula:

$$x_i^d(t+1) = x_i^d(t) + De \cdot v_i^d(t+1) \tag{6}$$

$$De = \frac{ext}{t} \tag{7}$$

where *De* adjusts the convergence of the search process. After several experiments, *ext*=2 is adopted in this paper.

In order to simulate the uncertainty principle of quantum, after updating the position, the photons still need to be observed to obtain the final specific position. Specific steps of "Observation" behavior will be given below.

2.2.2 Observation

In the theory of quantum mechanics, the exact position and velocity of a particle are hard to be measured simultaneously according to the uncertainty principle proposed by Heisenberg. If we know the precise momentum of a particle, the particle represents a plane wave, whereas a plane wave is diffuse across space, so its position is uncertain. As we turn to the precise position of the particle, the particle appears in the range of the wave packet in the form of probability.

Based on this quantum property, we designed the "Observation" behavior of the algorithm. In other words, when the algorithm decides to obtain the position of photon "*i*", it will appear in the region where it is likely to occur with probability, near the updated position X_i.

Here is the formula for the position of the photon after "Observation":

$$x_i^d(t) = x_i^d(t) + De \cdot randA \tag{8}$$

where *randA* refers to a random value ranged with [-1, 1].

2.2.3 Search exclusion principle

Pauli Exclusion Principle is one of the fundamental laws of the motion of microscopic particles. It states that a system of fermions cannot have two or more particles in the same state.

In PSA, the position and fitness of photon were determined after Observation behavior. If the new value discovered by the current search agent has better fitness value than the existing global optimum, the new global target will be generated at the position as a special particle. However, as described by the Pauli Exclusion Principle, two particles of the same state cannot appear at the same time. Hence, we must move the particle at the current global optimum to somewhere else. This is what we call the Search exclusion principle.

This paper assumes that PSA will has a certain understanding of the search range at the very beginning of the search and that it is necessary to make search agents have shorter search processes. So, we randomly put all the search agents on the "Search Axis", where all the points on it have the same position value in all dimensions:

$$\text{For } \forall\, X_{OnAxis} = (x_{OA}^{1}, x_{OA}^{2} \dots, x_{OA}^{n}) \text{ have } x_{OA}^{1} = x_{OA}^{2} = \cdots = x_{OA}^{n} \tag{9}$$

Its mathematical expression is as follows:

$$x_i^d(t) = xLow^d + (xUpp^d - xLow^d) \cdot randB \tag{10}$$

where $xLow^d$ refers to the value of lower boundary of corresponding dimensions, and $xUpp^d$ refers to the contrary one. Particularly, *randB* is a random value within the range [0, 1], which is generated only once in each iteration and used by all search agents.

2.3 Algorithm Design

In this section, we combine the behavior of each algorithm to show the specific operation of PSA. The pseudo code of PSA is given in Fig. 1.

```
Initialize position of photons (search agents) $X_i(i = 1, 2, ..., n)$
Figure out the fitness values of photons
Calculate $R_{Len}$ using the Eq.(4)
$X_g$=the position of the agent which has best fitness value
WHILE (iterT < The limit of iterations)
        FOR i=idxs in photons
                UPDATE $R_{ig}$ using the Eq.(3)
                UPDATE the speed of photon(i) using the Eq.(5)
                UPDATE the position of photon(i) using the Eq.(6)
                Perform "Observation" for photon(i) using the Eq.(8)
                Check if photon(i) goes out of the boundary and amend it
                Figure out the fitness of photon(i)
                IF fitness of photon(i) is better THEN UPDATE $X_g$ (keep the better one)
                IF $X_g$ is Updated THEN UPDATE the position of photon(i) using the Eq.(10)
        END FOR
        iterT = iterT +1
END WHILE
RETURN $X_g$
```

Fig. 1. The pseudo-code of PSA.

Theoretically, PSA is behaved as a global optimization algorithm, and "Observation" behavior ensures that the algorithm avoids local optimization. In addition, PSA has one common characteristic of algorithms that follow evolutionary rules, which requires algorithms to retain the optimal solution after each iteration. In theory, this ensures that the algorithm has competitiveness when dealing with unimodal functions, but it sometimes traps the algorithm into local optimum.

Such results may be caused by the high demand for convergence in this paper. How to balance convergence and optimal performance may be one of the available subjects of future studies.

3. Experiments & Evaluations

The calculating performance of the PSA is evaluated by testing 23 numerical optimization problems proposed by the literature [3]. The first 7 functions are unimodal functions, and the last 16 are multimodal functions. Tables 1 and 2 represent the math expression of benchmark functions used in our experiment. Figs. 2 and 3 give the 2D representations of these benchmark functions.

We tested PSA, WOA, GSA, and PSO with these benchmark functions and the results are compared later. In all cases, the population N is set to 20. Test dimension is set to 30 (Dim = 30) and the number of iterations is limited to 300 for benchmark functions of Tables 1 and 2. In particular, for PSO, c_1 and c_2 are set to 2 and factor w is set to 0.4.

Before presenting test results, let's firstly introduce WOA and GSA independently.

Whale Optimization Algorithm

The inspiration of WOA comes from a unique hunting method of humpback whales: bubble net prey. And WOA was presented by constructing the mathematical model of encircling prey, spiral bubble net hunting maneuver and prey searching. Below, we introduce each of these models.

The mathematical model of encircling prey:

$$\vec{D} = \left|\vec{C} \cdot \overrightarrow{X^{*}}(t) - \vec{X}(t)\right|, \vec{X}(t+1) = \overrightarrow{X^{*}}(t) - \vec{A} \cdot \vec{D} \tag{11}$$

where X^{*} is position vector for the best solution so far, $\vec{X}$ is the position vector, $\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a}$ and $\vec{C} = 2 \cdot \vec{r}$ denote coefficient vectors, and $\vec{a}$ decreases linearly from 2 to 0 during the iteration, $\vec{r}$ is a random vector in [0, 1].

The mathematical model of spiral bubble net hunting manner:

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^{*}}(t) - \vec{A} \cdot \vec{D} & if \;\; p < 0.5 \\ \overrightarrow{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X^{*}}(t) & if \;\; p \geq 0.5 \end{cases} \tag{12}$$

where $\overrightarrow{D'} = |\overrightarrow{X^{*}}(t) - \vec{X}(t)|$ denotes the distance of the ith whale to the prey, b is a constant that defines the shape of a logarithmic spiral, l is a random number in [-1, 1], and p is a random number in [0, 1].

The mathematical model of prey searching:

$$\vec{D} = \left|\vec{C} \cdot \overrightarrow{X_{rand}} - \vec{X}\right|, \vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A} \cdot \vec{D} \tag{13}$$

where $\overrightarrow{X_{rand}}$ represents the position vector of a random whale selected from the current population.

Table 1. Description of unimodal functions, F1– F7

Function	Dim	Range	$f_{\min}$
$F_1(x) = \Sigma_{i=1}^{n} x_i^2$	30	[-100,100]	0
$F_2(x) = \Sigma_{i=1}^{n} \mid x_i \mid + \Pi_{i=1}^{n} \mid x_i \mid$	30	[-10,10]	0
$F_3(x) = \Sigma_{i=1}^{n} (\Sigma_{j=1}^{i} x_j)^2$	30	[-100,100]	0
$F_4(x) = \max_i \{\mid x_i \mid, 1 \le i \le n\}$	30	[-100,100]	0
$F_5(x) = \Sigma_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-30,30]	0
$F_6(x) = \Sigma_{i=1}^{n}([x_i + 0.5])^2$	30	[-100,100]	0
$F_7(x) = \Sigma_{i=1}^{n} i x_i^4 + \text{rand}[0,1)$	30	[-1.28,1.28]	0

Table 2. Description of multimodal functions, F8–F23

Function	Dim	Range	$f_{\min}$
$F_8(x) = \Sigma_{i=1}^{n} - x_i \sin(\sqrt{\mid x_i \mid})$	30	[-500,500]	-12569.5
$F_9(x) = \Sigma_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[-5.12,5.12]	0
$F_{10}(x) = -20\exp\left(-0.2\sqrt{(\Sigma_{i=1}^{n} x_i^2)/n}\right)$ $- \exp\left((\Sigma_{i=1}^{n} \cos(2\pi x_i))/n\right) + 20 + e$	30	[-32,32]	0
$F_{11}(x) = \frac{1}{4000}\Sigma_{i=1}^{n} x_i^2 - \Pi_{i=1}^{n} \cos\left(x_i / \sqrt{i}\right) + 1$	30	[-600,600]	0
$F_{12}(x) = (\pi / n)\{10\sin(\pi y_1)$ $+ \Sigma_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})]$ $+ (y_n - 1)^2\} + \Sigma_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + 0.25(x_i + 1)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[-50,50]	0
$F_{13}(x) = 0.1\{\sin^2(3\pi x_1)$ $+\Sigma_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)]$ $+(x_n - 1)^2[1 + \sin^2(2\pi x_n)]\}$ $+\Sigma_{i=1}^{n} u(x_i, 10, 100, 4)$	30	[-50,50]	0
$F_{14}(x) = (\frac{1}{500} + \Sigma_{j=1}^{25} \frac{1}{j + \Sigma_{i=1}^{2}(x_i - a_{ij})^6})^{-1}$	2	[-65,65]	1
$F_{15}(x) = \Sigma_{i=1}^{11}[a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	4	[-5,5]	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^2 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[-5,5]	-1.0316
$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{\pi}{5}x_1 - 6)^2$ $+10(1 - \frac{1}{8\pi})\cos(x_1) + 10$	2	[-5,5]	0.398

Table 2. (Continued)

Function	Dim	Range	$f_{\min}$
$F_{18}(x)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2$ $-14x_2+6x_1x_2+3x_2^2)]\times[30+(2x_1-3x_2)^2$ $(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	2	[-2,2]	3
$F_{19}(x)=-\Sigma_{i=1}^{4}c_i\exp(-\Sigma_{j=1}^{3}a_{ij}(x_j-p_{ij})^2)$	3	[1,3]	-3.86
$F_{20}(x)=-\Sigma_{i=1}^{4}c_i\exp(-\Sigma_{j=1}^{6}a_{ij}(x_j-p_{ij})^2)$	6	[0,1]	-3.32
$F_{21}(x)=-\Sigma_{i=1}^{5}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	-10.1532
$F_{22}(x)=-\Sigma_{i=1}^{7}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	-10.4028
$F_{23}(x)=-\Sigma_{i=1}^{10}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	-10.5363

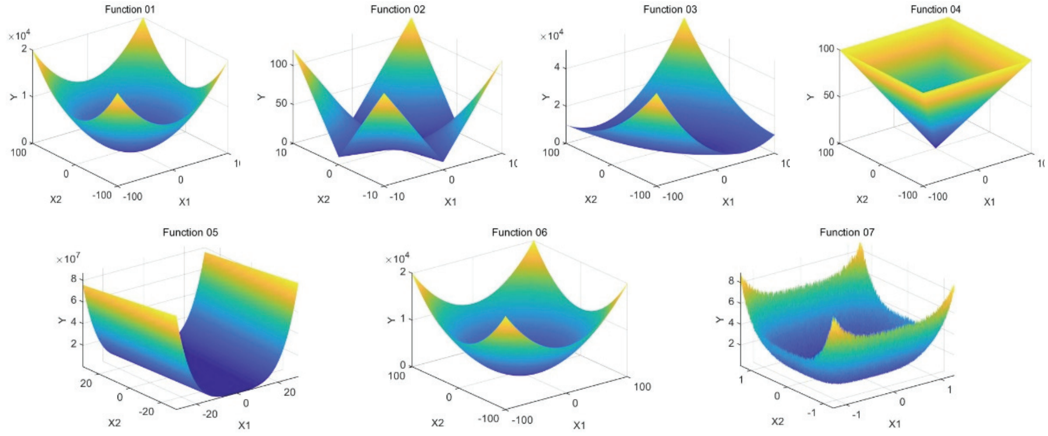


Fig. 2. Two-dimensional (2D) representations of unimodal benchmark functions (F1–F7).

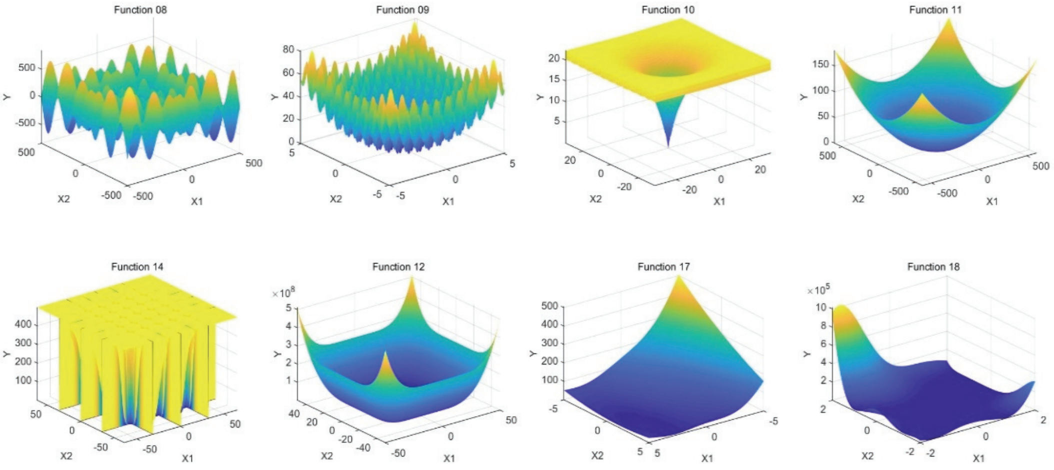


Fig. 3. Typical 2D representations of multimodal benchmark functions (F8–F16).

Gravitational Search Algorithm

GSA was proposed by the law of interaction between gravity and mass. During the execution of GSA, let x_i^d be the position of the *i*th search-agent in the *d*-th dimension and $X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n)$ be the position of the *i*th search-agent for *i*=1,2…,*N*, the calculation formula of the interaction force and the location and speed of the search-agent as follow.

The force $F_{ij}^d(t)$ on mass *i* from mass *j* and the total force on search-agent *i* in the *d*th dimension at time *t* are defined as follows:

$$F_{ij}^d(t) = G(t)\frac{M_{pi}(t)\times M_{aj}(t)}{R_{ij}(t)+\varepsilon}(x_j^d(t) - x_i^d(t)) \tag{14}$$

$$F_i^d(t) = \sum_{j=1,j\neq i}^{N} rand_j F_{ij}^d(t) \tag{15}$$

where M_{aj} denotes the active gravitational mass associated with search-agent *j*, M_{pi} is the passive gravitational mass associated with search-agent *I*, $G(t) = G(G_0,\ t)$ is gravitational constant, G_0 is the original value, ε is a sufficiently small positive number, $R_{ij}(t) = ||X_i(t),\ X_j(t)||_2$ counts the Euclidian distance between two search-agent *i* and *j*, and $rand_j$ is a random constant in [0, 1].

The location and speed of the search-agent at time t are defined as follows:

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \tag{16}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{17}$$

where $a_i^d(t) = F_i^d(t)/\mathrm{M}_{ii}(t)$ is the acceleration of the search-agent, $\mathrm{M}_{ii}(t)$ is the inertial mass of the *i*th search-agent, rand_i denotes the uniform random variable in [0, 1].

3.1 Unimodal Functions Evaluation

F1–F7 are unimodal benchmark functions. They can be used to evaluate the algorithm's local search capability since they have only one global best solution. As shown by Fig. 4, average best-so-far (average number of the best solution obtained so far in each iteration) of 30 runs, PSA for unimodal function has not only good convergence but also has high convergence rate. Compared with other algorithms, it can get closer to the global optimum in fewer iterations. This means that PSA has better performance handling the problems that require less computing time.

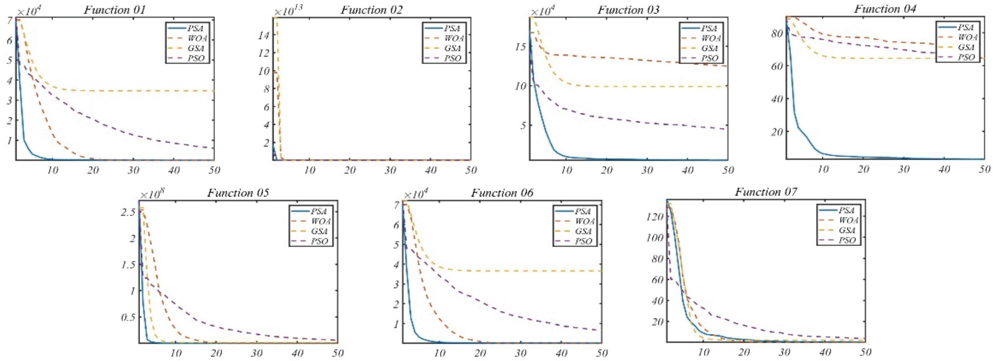


Fig. 4. Comparison results of convergence performance when dealing with unimodal functions.

Statistical results for 30 runs of F_1–F_7 are listed in Table 3. Best, Avg, and Std represent best fitness ever found, average fitness and corresponding standard deviation, respectively. In parallel comparison, bold font represents the best effect. If there is no bold font in parallel comparison, this means that all algorithms perform well.

Table 3. Real machine test results of 7 unimodal functions

		PSA	WOA	GSA	PSO
F_1	Best	0.0934	5.85E-52	16.4172	0.0413
	Avg	15.3222	2.33E-35	736.3204	1667.6904
	Std	27.3389	9.02E-35	428.3625	4610.9635
F_2	Best	0.6680	2.62E-32	0.0203	10.0903
	Avg	2.2314	2.84E-27	4.0400	27.4738
	Std	1.5088	1.05E-26	3.4989	15.4874
F_3	Best	3.9671	47913.1934	526.6285	4014.6458
	Avg	3978.0837	74924.1412	2532.8313	21258.0532
	Std	3718.9156	18028.0192	1080.6073	8626.2509
F_4	Best	0.0746	5.4675	9.9723	23.4041
	Avg	1.1947	63.3774	15.5021	34.9788
	Std	1.0316	22.6286	2.9599	7.9321
F_5	Best	8.9635	27.7872	45.2700	32.7953
	Avg	332.6410	28.5011	1515.3686	6841.2273
	Std	705.1589	0.2887	2128.6598	22676.7468
F_6	Best	0	0	297	0
	Avg	19.8667	0	1276.2	1012
	Std	33.4589	0	562.2334	3048.7384
F_7	Best	0.0017	0.0003	0.1072	0.0730
	Avg	0.0237	0.0109	0.2579	0.9833
	Std	0.0170	0.0184	0.1127	2.2264

Obviously, PSA is very competitive when compares with other algorithms in dealing with functions F_3–F_7. Especially, its test results under functions F_3–F_5 are significantly better than those of other algorithms, and it can achieve the accuracy that other algorithms are difficult to achieve. Although the results of dealing with F_1 and F_2 are not best, PSA still maintains considerable accuracy.

With the aim of analyzing the performance of algorithms, it is first necessary to be clear that the search process of most optimization algorithms is divided into two main stages: exploration and exploitation [3]. The exploration phase is responsible for letting the algorithm determine the approximate location of the global optimization, while the exploitation phase is responsible for allowing the algorithm to dig deeper into the global optimal solution. PSA also has these two phases. Through further analysis, since the PSA algorithm itself has good convergence performance, it can quickly find the region where the global optimum is located. It is not practical to get most of the algorithms to perform well in exploration and exploitation at the same time. It can be known from Eq. (9) that after the Observation step, the location of the search agent will change around the current location, to avoid falling into local optimum. This step also causes the search agent to shift around the global best, causing some bias. The Observation step is to obtain excellent exploration performance, avoid falling into local optimum, and the effect is visible. The algorithm obtains excellent convergence performance but sacrifices some exploitation performance. This also leads to the results of Table 3. For F_1, F_2, F_6, and F_7, PSA has excellent performance, and it can stably find the position of the global optimal solution in multiple iterations, showing excellent exploration performance, but due to Exploitation performance is slightly insufficient, so PSA cannot dig deeper into the global optimal solution as WOA does. For F_3, F_4, F_5, we can easily find that these three functions may have a lower gradient or even a zero gradient during the search process. At this time, the PSA's exploration performance is fully reflected.

Finally, considering the convergence of the algorithm, it can be said that PSA performs quite well when dealing with unimodal functions.

3.2 Multimodal Functions Evaluation

F_8–F_{23} are multimodal benchmark functions. Most of the multimodal functions are complex in structure and have many local optimal values, which have a considerable test for the convergence performance of the algorithm. However, these multimodal functions are more closely related to the models of real-life problems, and in other words, they can be used to evaluate the robustness of meta-heuristic algorithms.

Table 4 shows that the multimodal function test results, compared with other algorithms, PSA in function F_8/F_{12}/F_{14}/F_{16}/F_{21}/F_{22}/F_{23} have remarkable performance. In the test of these functions, PSA not only found a better solution but also maintained a lower average value and standard deviation, which showed strong stability. Although PSA did not get the best result in the test of F_9/F_{10}/F_{11}/F_{13} /F_{15}/F_{17}/ F_{18}/F_{19}/F_{20} functions, it still maintains high accuracy and stability in these function tests.

Table 4. Real machine test results of multimodal functions

		PSA	WOA	GSA	PSO
F_8	Best	-12569.2458	-12569.3352	-3126.6594	-10176.3344
	Avg	-11648.5512	-9811.9027	-2368.9008	-8807.4347
	Std	1230.4314	1863.4704	438.0826	699.2045
F_9	Best	0.5065	0	20.8941	53.8434
	Avg	7.3763	0	43.7314	135.6683
	Std	9.1989	0	11.7495	42.8937
F_{10}	Best	0.3610	8.88E-16	3.3895	0.6043
	Avg	1.6766	6.10E-15	6.7118	10.1080
	Std	0.9929	5.09E-15	1.9822	5.9491
F_{11}	Best	0.0073	0	125.3078	0.0003
	Avg	0.5294	0	168.2118	9.0238
	Std	0.6102	0	22.7005	27.4559
F_{12}	Best	0.0014	0.0282	1.8786	0.7190
	Avg	0.1716	0.1073	6.2125	5.2367
	Std	0.2706	0.0568	3.0004	2.3537
F_{13}	Best	0.0104	0.2457	6.6689	-0.6639
	Avg	1.5458	0.9250	65.7351	11.3909
	Std	3.3136	0.3604	136.0469	9.3172
F_{14}	Best	0.3769	0.3769	0.3769	0.3769
	Avg	0.4802	0.5973	0.7245	0.4276
	Std	0.1158	0.3391	0.3414	0.1349

Table 4. (Continued)

		PSA	WOA	GSA	PSO
F_{15}	Best	0.0003	0.0003	0.0010	0.0003
	Avg	0.0077	0.0011	0.0063	0.0075
	Std	0.0224	0.0016	0.0053	0.0093
F_{16}	Best	-1.0316	-1.0316	-1.0316	-1.0316
	Avg	-1.0316	-1.0316	-1.0316	-1.0316
	Std	2.33E-07	3.58E-08	4.52E-16	6.18E-16
F_{17}	Best	0.3979	0.3979	0.3979	0.3979
	Avg	0.3979	0.3981	0.3979	0.3979
	Std	1.41E-07	0.0003	0	0
F_{18}	Best	3	3	3	3
	Avg	3	5.7120	3	3
	Std	1.36E-05	8.2702	7.36E-15	1.61E-15
F_{19}	Best	-3.8628	-3.8628	-3.8628	-3.8628
	Avg	-3.8556	-3.8162	-3.8603	-3.8623
	Std	0.0153	0.1404	0.0050	0.0020
F_{20}	Best	-3.3080	-3.3143	-3.3220	-3.3220
	Avg	-3.0403	-3.2145	-3.3071	-3.2134
	Std	0.1940	0.1050	0.0459	0.2948
F_{21}	Best	-10.1531	-10.1495	-10.1532	-10.1532
	Avg	-9.7302	-7.3740	-5.9719	-5.5585
	Std	1.1347	2.5282	3.7303	3.2203
F_{22}	Best	-10.4029	-10.3394	-10.4029	-10.4029
	Avg	-9.8628	-6.1788	-9.1374	-5.1888
	Std	1.2894	2.5013	2.5956	3.2755
F_{23}	Best	-10.5364	-10.5252	-10.5364	-10.5364
	Avg	-9.8189	-5.5032	-8.1614	-6.3000
	Std	1.8027	2.7365	3.5931	3.8058

Combined with the test of unimodal functions, PSA has an excellent ability to explore so that it will perform well in the test of multimodal functions. However, in order to make the algorithm have a faster convergence speed, PSA must sacrifice some of the Exploitation ability, which also makes the performance of some functions, not the best, but the performance is still excellent.

The function F_8–F_{13} are normal multi-peak function. For the functions F_8 and F_{12}, the performance of the PSA is perfect, while for the F_9–F_{11} and F_{13}, the performance is not the best, but it is maintained at a reasonable level. As can be seen from Table 4, the PSA is very close to the global best and maintains an excellent convergence performance, so that it can adapt to some computing scenarios that require less computing time.

Although the functions F_{14}–F_{23} are multi-peak functions of fixed dimensions, the performance of each optimization algorithm is roughly the same as that of F_8–F_{13}, but it is worth noting that PSA performs better and more stable for the F_{21}–F_{23} function.

As can be seen from Fig. 5, PSA still has good convergence for the function F_8–F_{19} and has advantages over other algorithms. For F_{20}, PSA can converge rapidly, but its accuracy is not very good. However, it performs very well for F_{21}–F_{23}. As can be seen from the Figure, PSA maintains a perfect convergence, accuracy, and converge speed every time it searches in these functions.

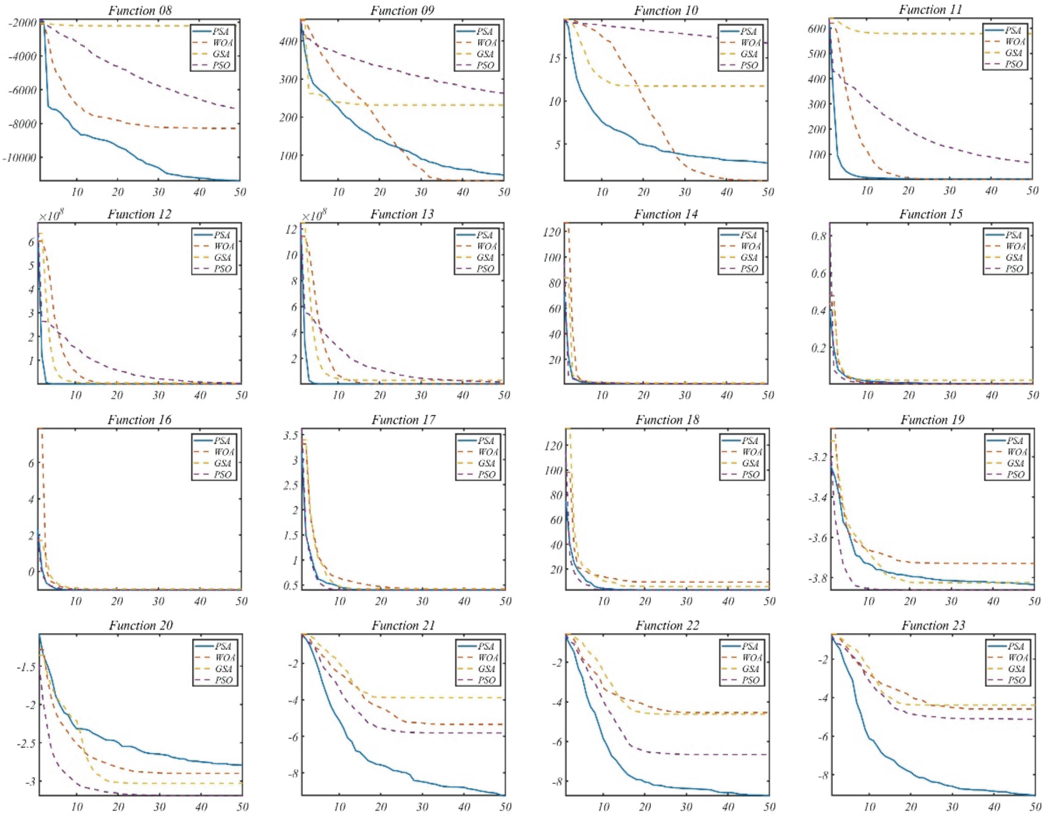


Fig. 5. Comparison results of convergence performance when dealing with multimodal functions.

4. Improvement of PSA for Unimodal Functions

It is known that unimodal functions have only one extreme value (minimum or maximum) within a given range. Therefore, it is not necessary to worry about whether the algorithm is stuck at a local extreme when exploring the search space. However, the convergence speed and accuracy of the algorithm should be given priority of considering. Based on the characteristics of the unimodal functions, we improve PSA and get an algorithm specifically for the unimodal function, which we called UFPSA.

For PSA, the only requirement is to modify the photon position formula and its updated position formula. The modifications are as follows.

UFPSA updates the location of photon "i" by following formula:

$$x_i^d(t+1) = x_i^d(t) + De \cdot v_i^d(t+1) \tag{18}$$

$$De = \frac{ext}{t^b} \tag{19}$$

where *De* adjusts the convergence of the search process. After several experiments, let *ext*=2 and b=1.5 in this paper.

The position of the photon is calculated according to the following formula:

$$x_i^d(t) = Normrnd(t) \cdot x_i^d(t) \tag{20}$$

where $Normrnd(t)$ refers to the random number created from the normal distribution with mean 0 and standard deviation 0.6.

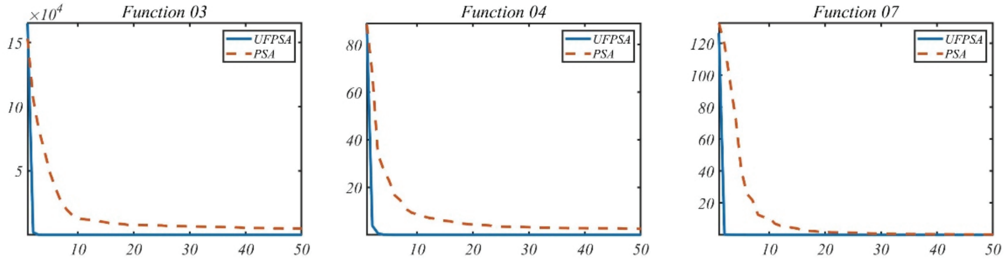


Fig. 6. Convergence performance of UFPSA and PSA when dealing with unimodal functions.

Table 5. Unimodal function test of UFPSA and PSA

		F_1	F_2	F_3	F_4	F_5	F_6	F_7
UFPSA	Best	7.69E-17	2.83E-09	2.00E-15	2.67E-09	0.029	0	6.47E-06
	Avg	1.19E-13	1.59E-07	4.55E-12	8.30E-08	20.6908	0	0
	Std	2.03E-13	1.18E-07	1.14E-11	8.96E-08	10.6485	0	0
PSA	Best	0.0826	0.4588	0.4477	0.1689	3.0058	0	0.0059
	Avg	13.3908	1.7826	4588.8400	1.1565	153.8715	32.9667	0.0238
	Std	16.0383	1.3826	5382.1010	0.9413	193.0914	45.1736	0.0155

Experiments and Evaluations of UFPSA

We used the same test environment to test UFPSA and PSA (Dim=30, N=20, MaxIter=300). Looking at Fig. 6 and Table 5, we can clearly find that UFPSA can converge faster and maintain higher accuracy when dealing with unimodal function compared with PSA. In the case that the dimension is equal to 30, UFPSA can converge to a higher accuracy in a very few iterations, which enables it to perform some optimization operations requiring higher time complexity.

5. Conclusion

This paper provides a brand new meta-heuristic algorithm PSA which based on the quantum properties of photon. According to the test results of 23 benchmark functions, note that PSA has perfect convergence and strong ability of global search. Further, we can see that PSA can always decrease to an acceptable accuracy within a few iterations, whether in unimodal or multimodal functions, what implies that the algorithm can achieve a better approximate solution in a shorter time. The strong global search capability also makes PSA more competitive in comparison with other algorithms. However, the PSA still has some room for improvement. For instance, while guaranteeing the convergence of PSA, the accuracy for solving multimodal functions could be further improved.

References

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, Perth, Australia, 1995, pp. 1942-1948.

[2] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459-471, 2007.

[3] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51-67, 2016.

[4] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28-39, 2006.

[5] P. C. Pinto, T. A. Runkler, and J. M. Sousa, "Wasp swarm algorithm for dynamic MAX-SAT problems," in *Adaptive and Natural Computing Algorithms*. Heidelberg: Springer, 2007, pp. 350-357.

[6] A. Mucherino and O. Seref, "Monkey search: a novel metaheuristic search for global optimization," *AIP Conference Proceedings*, vol. 953, no. 1, pp. 162-173, 2007.

[7] A. Sharma, A. Sharma, B. K. Panigrahi, D. Kiran, and R. Kumar, "Ageist spider monkey optimization algorithm," *Swarm and Evolutionary Computation*, vol. 28, pp. 58-77, 2016.

[8] Q. Zhou and Y. Q. Zhou, "Wolf colony search algorithm based on leader strategy," *Application Research of Computers*, vol. 30, no. 9, pp. 2629-2632, 2013.

[9] C. R. Hwang, "Simulated Annealing: Theory and Applications by P. J. M. Van Laarhoven and E. H. Aarts, 1987," *Acta Applicandae Mathematica*, vol. 12, pp. 108-111, 1988.

[10] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232-2248, 2009.

[11] H. Shah-Hosseini, "Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimization," *International Journal of Computational Science and Engineering*, vol. 6, no. 1-2, pp. 132-140, 2011.

[12] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: charged system search," *Acta Mechanica*, vol. 213, no. 3-4, pp. 267-289, 2010.

[13] R. A. Formato, "Central force optimization: a new deterministic gradient-like optimization metaheuristic," *Opsearch*, vol. 46, no. 1, pp. 25-51, 2009.

[14] A. Hatamlou, "Black hole: a new heuristic optimization approach for data clustering," *Information Sciences*, vol. 222, pp. 175-184, 2013.

[15] H. Huang, M. Zhu, and J. Wang, "An improved artificial bee colony algorithm based on special division and intellective search," *Journal of Information Processing Systems*, vol. 15, no. 2, pp. 433-439, 2019.

[16] L. Zhao and Y. Long, "An improved PSO algorithm for the classification of multiple power quality disturbances," *Journal of Information Processing Systems*, vol. 15, no. 1, pp. 116-126, 2019.

[17] X. Song, M. Zhao, Q. Yan, and S. Xing, "A high-efficiency adaptive artificial bee colony algorithm using two strategies for continuous optimization," *Swarm and Evolutionary Computation*, vol. 50, article no. 100549, 2019.

[18] M. R. Chen, J. H. Chen, G. Q. Zeng, K. D. Lu, and X. F. Jiang, "An improved artificial bee colony algorithm combined with extremal optimization and Boltzmann Selection probability," *Swarm and Evolutionary Computation*, vol. 49, pp. 158-177, 2019.

[19] M. Li, D. Lei, and J. Cai, "Two-level imperialist competitive algorithm for energy-efficient hybrid flow shop scheduling problem with relative importance of objectives," *Swarm and Evolutionary Computation*, vol. 49, pp. 34-43, 2019.

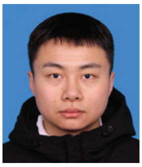
[20] H. Kragh, “Max Planck: the reluctant revolutionary,” *Physics World*, vol. 13, no. 12, pp. 31-36, 2000.

[21] A. Einstein, “Uber einen die erzeugung und verwandlung des lichtes betreffenden heuristischen gesichtspunkt,” *Annalen der Physik*, vol. 322, no. 6, pp. 132-148, 1905.

[22] L. V. de Broglie, “On the theory of quanta,” *Annales de Physique*, vol. 10, no. 3, pp. 22-128, 1925.

Yongli Liu **https://orcid.org/0000-0002-0540-865X**

He is a Ph.D., associate professor, master tutor. In 2003 and 2010, he received his bachelor's degree and doctor's degree in engineering from Beihang University, and his main research fields are information retrieval, data mining and big data. He is a member of the American Computer Society (ACM), the Chinese Computer Society (CCF), and the backbone teacher of institutions of higher learning in Henan province.



Renjie Li **https://orcid.org/0000-0002-5774-1358**

He is a postgraduate student in the school of computer science and technology of Henan University of Technology. His main research interests are numerical optimization and combinatorial optimization.