

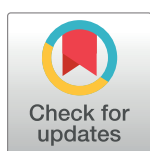
RESEARCH ARTICLE

# Fractional-order quantum particle swarm optimization

Lai Xu, Aamir Muhammad, Yifei Pu, Jiliu Zhou, Yi Zhang \*

School of Computer Science, Sichuan University, Chengdu, Sichuan Province, China

\* [yzhang@scu.edu.cn](mailto:yzhang@scu.edu.cn)



## Abstract

Motivated by the concepts of quantum mechanics and particle swarm optimization (PSO), quantum-behaved particle swarm optimization (QPSO) was developed to achieve better global search ability. This paper proposes a new method to improve the global search ability of QPSO with fractional calculus (FC). Based on one of the most frequently used fractional differential definitions, the Grünwald-Letnikov definition, we introduce its discrete expression into the position updating of QPSO. Extensive experiments on well-known benchmark functions were performed to evaluate the performance of the proposed fractional-order quantum particle swarm optimization (FQPSO). The experimental results demonstrate its superior ability in achieving optimal solutions for several different optimizations.

## OPEN ACCESS

**Citation:** Xu L, Muhammad A, Pu Y, Zhou J, Zhang Y (2019) Fractional-order quantum particle swarm optimization. PLoS ONE 14(6): e0218285. <https://doi.org/10.1371/journal.pone.0218285>

**Editor:** Nicholas Chancellor, Durham University, UNITED KINGDOM

**Received:** April 2, 2019

**Accepted:** May 29, 2019

**Published:** June 20, 2019

**Copyright:** © 2019 Xu et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data Availability Statement:** All relevant data are within the manuscript.

**Funding:** This work was supported by National Key R&D Program of China, <http://www.most.gov.cn/>, 2017YFB0802300 to JZ; National Natural Science Foundation of China, <http://www.nsf.gov.cn/>, 61671312 to YZ; Science and Technology Support Project of Sichuan Province of China, <http://kjt.sc.gov.cn/>, 2018HH0070 to YZ; and Science and Technology Support Project of Sichuan Province of China, 2013SZ0071, <http://kjt.sc.gov.cn/>, to YP. The funders had no role in study design, data

## Introduction

Particle swarm optimization (PSO) [1], which is inspired by animal social behaviors, such as birds, was first proposed by Kennedy and Eberhart as a population-based optimization technique. In PSO, the potential solutions, which are called particles, go through the solution space by relying on their own experiences and current best particle. PSO has a competitive performance with the classical Genetic Algorithm (GA) [2], evolutionary programming (EP) [3], evolution strategies (ES) [4], genetic programming (GP) [5] and other classic algorithms. It has attracted increasing attention during recent years thanks to its effectiveness in different optimization problems [6][7][8].

Quantum computer [9] was proposed 30 years ago and the formal definition of the quantum computer was given in the late 1980s. Since the quantum computer has shown its potential in several special problems [10], many efforts were dedicated to this field. Several well-known algorithms were proposed, and Shor's quantum factoring algorithm was the most famous one in these methods [11].

Inspired by a similar idea, the quantum-behaved particle swarm optimization (QPSO) [12] was introduced in 2004 by Sun et al. to improve the convergence of classical PSO. In quantum space, particles search in the complete solution space and the global optimum is guaranteed. In recent decades, fractional calculus has drawn increasing interests and been a strong branch of mathematical analyses. Furthermore, the random variables in the physical process can be regarded as the substitution of real stochastic motion. As a result, the fractional calculus can be

collection and analysis, decision to publish, or preparation of the manuscript.

**Competing interests:** The authors have declared that no competing interests exist.

introduced to analyze the physical statuses and procedures of objects in Euclidean space. The fractional differential functions have two features. A fractional differential function is a power function for primary functions, and it is an iterative addition or product of specific functions for the other functions. Meanwhile, it has been proved that many fractional-order models are more suitable to describe the natural phenomena. Based on these observations, fractional calculus has been introduced into many fields such as viscoelastic theory [13], diffusion processing [14] and stochastic fractal dynamics [15]. Most of the researches on fractional-order applications focus on the transient state of physical changes. However, the evolutive procedures of systems are rarely included.

In recent years, QPSO has attracted great attention from many researchers. To balance the global and local searching abilities, Xi et al. proposed a novel QPSO called weighted QPSO (WQPSO) [16]. Jiao et al. proposed a dynamic-context cooperative quantum-behaved particle swarm optimization (CQPSO) [17] for medical image segmentation. Although QPSO and its variants have better performance in some aspects, they do not make full use of the state information during the ergodic process and it is inefficient in hunting global optimum. In this paper, a novel quantum particle swarm optimization with the fractional-order position is proposed. Due to the nonlinear, non-causal and non-stationary characteristics of fractional calculus, searching global optimum can be significantly accelerated [18][19].

The rest of this paper is organized as follows: In section 2, some mathematical background about fractional calculus is introduced. Section 3 presents the basic ideas of PSO and QPSO and the proposed method is also given there. Section 4 demonstrated the experimental results of the proposed method. Finally, Section 5 outlines the conclusion.

## Background theory for fractional calculus

Grünwald-Letnikov (GL) [20], Riemann-Liouville (RL) [21], and Caputo [22] definitions are three different definitions for fractional calculus in Euclidean space. Due to its convenient computational form, GL definition for the fractional derivative is commonly used for engineering problems.

The GL derivative with an order of function is defined as:

$$\begin{aligned} {}_a^{\text{GL}}D_x^\alpha f(x) &= \frac{d^\alpha}{[d(x-a)]^\alpha} f(x) \\ &= \lim_{N \rightarrow \infty} \left\{ \frac{\left(\frac{x-a}{N}\right)^{-\alpha}}{\Gamma(-\alpha)} \sum_{k=0}^{N-1} \frac{\Gamma(k-\alpha)}{\Gamma(k+1)} f\left(x - k\left(\frac{x-a}{N}\right)\right) \right\}, \end{aligned} \quad (1)$$

where  $f(x)$  is a differintegrable function,  $[a, x]$  is the function duration, and  $\Gamma$  is the gamma function. Here,  ${}_a^{\text{GL}}D_x^\alpha$  denotes the GL fractional differential operator.

In (1), when  $N$  is big enough, the limit symbol can be neglected and we can rewrite (1) as:

$$\frac{d^\alpha}{dx^\alpha} f(x) \cong \frac{x^{-\alpha} N^\alpha}{\Gamma(-\alpha)} \sum_{k=0}^{N-1} \frac{\Gamma(k-\alpha)}{\Gamma(k+1)} f\left(x - \frac{kx}{N}\right), \quad (2)$$

which is a proximate form substituting fractional derivative with multiplication and addition

operations [12]. For 1D signal, it has the following expression:

$$\begin{aligned} \frac{d^x}{dx^x} f(x) &\cong f(x) + (-\alpha)f(x-1) \\ &+ \frac{(-\alpha)(-\alpha+1)}{2}f(x-2) + \dots \\ &+ \frac{(-\alpha)(-\alpha+1)(-\alpha+2)\dots(-\alpha+n)}{n!}f(x-n) \end{aligned} \quad (3)$$

## Particle swarm optimization with fractional -order position

### Quantum particle swarm optimization

Trajectory analyses in [23] demonstrated that each particle should converge to the corresponding attractor  $C_i$ , which is given as follows:

$$C_{id}(t) = a \cdot pb_{id}(t) + (1-a) \cdot gb_d(t), a \sim U(0,1), \quad (4)$$

where  $a = c_1r_1/(c_1r_1+c_2r_2)$ . It can be seen that the local attractor is a stochastic attractor of particle  $i$  that lies in a hyper-rectangle with  $pb_{id}$  and  $gb_d$  being two ends of its diagonal.

Based on the convergence analysis of PSO [24], inspired by the theory of quantum physics, Sun et al. studied the convergence behavior of PSO and proposed a novel PSO model from quantum mechanics abbreviated as QPSO [25]. Based on the Delta potential, the quantum behavior of particles are considered. In the framework of quantum time-space, the quantum state of a particle can be defined by a wave function  $\psi(x,t)$ . In 3-D space,  $\psi(x,t)$  is given as

$$|\psi|^2 dx dy dz = Q dx dy dz, \quad (5)$$

where  $Q$  is the probability that measures the particle's location in the 3-D space. As a probability density function, we have

$$\int_{-\infty}^{+\infty} |\psi|^2 dx dy dz = \int_{-\infty}^{+\infty} Q dx dy dz = 1. \quad (6)$$

The normalized version of  $\psi$  can be given as:

$$\psi(y) = \frac{1}{\sqrt{L}} e^{-|y|/L}, \quad (7)$$

As a result,  $Q$  and the corresponding distribution function  $F$  can be obtained as:

$$Q(y) = |\psi(y)|^2 = \frac{1}{L} e^{-2|y|/L}, \quad (8)$$

And

$$F(X_{id}(t+1)) = 1 - e^{-\frac{2|p_{id}(t)-X_{id}(t+1)|}{L_{id}(t)}}, \quad (9)$$

where  $L_{id}(t)$  denotes the standard deviation, which describes the search range of each particle. The position of the particle can be obtained by Monte Carlo method with the following formula:

$$s = \frac{1}{L} u = \frac{1}{L} e^{-2|y|/L}, u = rand(0,1), \quad (10)$$

where  $s$  denotes a random constant, which is uniformly distributed on  $U(0, 1/L)$ .

Then,  $u = e^{-2|y|/L}$ . Let  $y = x - c$ , we have

$$x = c \pm \frac{L}{2} \ln\left(\frac{1}{u}\right). \quad (11)$$

The convergence condition of PSO is given by:

$$x \rightarrow c, \text{ when } t \rightarrow \infty. \quad (12)$$

Let  $L$  be the function of time, we have:

$$L = L(t)$$

$$L \rightarrow 0, \text{ when } t \rightarrow 0. \quad (13)$$

With (13), we have the iterative version of  $i$ -th multidimensional particle as follows

$$X_{id}(t+1) = C_{id}(t) \pm \frac{L_{id}(t)}{2} \ln\left(\frac{1}{u}\right). \quad (14)$$

A global point called mean best position is introduced to evaluate  $L_{id}(t)$ . The global point, which is denoted by  $mbest$ , can be computed as the mean of the  $pbest$  positions of all particles, which can be given as:

$$\begin{aligned} mbest(t) &= (mbest_1(t), mbest_2(t), \dots, mbest_d(t)) \\ &= \frac{1}{n} \sum_{i=1}^n p_{i1}(t), \frac{1}{n} \sum_{i=1}^n p_{i2}(t), \dots, \frac{1}{n} \sum_{i=1}^n p_{id}(t). \end{aligned} \quad (15)$$

The values of  $L_{id}(t)$  is calculated by:

$$L_{id}(t) = 2 \cdot \beta \cdot |m_d(t) - X_{id}(t)|. \quad (16)$$

Finally, the position can be given by:

$$X_{id}(t+1) = C_{id}(t) \pm \beta \cdot |mbest_d - X_{id}(t)| \ln\left(\frac{1}{u}\right), \quad (17)$$

where parameter  $\beta$  is step size, which is utilized to control the convergence speed.  $rand$  is a random number with a range of 0 to 1, which is the deciding factor of “ $\pm$ ” in (17).

[Table 1](#) illustrates the main steps of QPSO.

## QPSO with the fractional-order position

It is well known that fractional calculus has a remarkable long-term memory characteristic [26]. From the definition of Grünwald-Letnikov in (1), it can be seen that fractional derivative is computed with all historical states and it is naturally suitable for the iterative procedure of intelligent optimization algorithms. For examples, Pires E.J.S introduced fractional calculus theory into the updated formula of particle swarm optimization algorithm [27].

To further improve the speed and accuracy of convergence of QPSO, in this section, the proposed QPSO with the fractional-order position is detailed. Initially, the original position is

**Table 1. The main steps of QPSO.**

**Alogtihn2**

Initialize QPSO parameters;

**Repeat**

**For all** particles **do**

compute  $f$

If  $f(x_i) < f(pb_i)$

$pb_i = x_i$

**End**

If  $f(pb_i) < f(gb)$

$gb = pb_i$

**End**

Calculate Q using (19)

If  $rand > 0.5$

$$X_{id}(t+1) = C_{id}(t) + \beta \cdot |mbest_d - X_{id}(t)| \cdot \ln\left(\frac{1}{u}\right)$$

**Else**

$$X_{id}(t+1) = C_{id}(t) - \beta \cdot |mbest_d - X_{id}(t)| \cdot \ln\left(\frac{1}{u}\right)$$

**End**

$t = t+1$

**Until** stopping criteria

<https://doi.org/10.1371/journal.pone.0218285.t001>

rearranged to modify the order of the position derivative, which can be derived as:

$$X_{id}(t+1) = C_{id}(t) + \beta \cdot \ln\left(\frac{1}{u}\right) \cdot (mbest_d - X_{id}(t)) \quad (rand > 0.5, mbest_d > X_{id}(t)), \quad (18)$$

$$X_{id}(t+1) = C_{id}(t) + \beta \cdot \ln\left(\frac{1}{u}\right) \cdot (X_{id}(t) - mbest_d) \quad (rand > 0.5, mbest_d < X_{id}(t)), \quad (19)$$

$$X_{id}(t+1) = C_{id}(t) - \beta \cdot \ln\left(\frac{1}{u}\right) \cdot (mbest_d - X_{id}(t)) \quad (rand < 0.5, mbest_d > X_{id}(t)), \quad (20)$$

$$X_{id}(t+1) = C_{id}(t) - \beta \cdot \ln\left(\frac{1}{u}\right) \cdot (X_{id}(t) - mbest_d) \quad (rand < 0.5, mbest_d < X_{id}(t)), \quad (21)$$

(23) and (26) can be uniformly rewritten as:

$$X_{id}(t+1) - X_{id}(t) = C_{id}(t) + \beta \cdot \ln\left(\frac{1}{u}\right) \cdot mbest_d - \left(\beta \cdot \ln\left(\frac{1}{u}\right) \pm 1\right) X_{id}(t). \quad (22)$$

The left side of (22) is the discrete version of the derivative with  $\alpha = 1$  and we can extend (22) to a generalized version, leading to the following fractional-order expression

$$D^\alpha(X_{id}(t+1)) = C_{id}(t) + \beta \cdot \ln\left(\frac{1}{u}\right) \cdot mbest_d - \left(\beta \cdot \ln\left(\frac{1}{u}\right) \pm 1\right) X_{id}(t), \quad (23)$$

when  $rand > 0.5, mbest_d > X_{id}(t)$  and  $rand < 0.5, mbest_d < X_{id}(t)$ .

Similarly, for  $rand > 0.5, mbest_d < X_{id}(t)$  and  $rand < 0.5, mbest_d > X_{id}(t)$ , we have

$$D^\alpha(X_{id}(t+1)) = C_{id}(t) - \beta \cdot \ln\left(\frac{1}{u}\right) \cdot mbest_d + \left(\beta \cdot \ln\left(\frac{1}{u}\right) \pm 1\right) X_{id}(t). \quad (24)$$

Previous researches have demonstrated that while the order  $\alpha$  of the derivative is set to  $[0,1]$ , it will introduce a smoother variation and prolong memory effect, which may lead to a

better performance than original integral-order method [12][13]. To study the behavior of the proposed fractional-order strategy, a set of functions are tested and the order  $\alpha$  is set to range from 0 to 1 with step size of  $\Delta\alpha = 0.1$ . To simplify the computational complexity, we usually truncate (3) and only use the first four terms, so we have

$$\begin{aligned} D^{\alpha}(X_{id}(t+1)) = & X_{id}(t+1) - \alpha X_{id}(t) \\ & - \frac{1}{2}\alpha(1-\alpha)X_{id}(t-1) \\ & - \frac{1}{6}\alpha(1-\alpha)(2-\alpha)X_{id}(t-2) \\ & - \frac{1}{24}\alpha(1-\alpha)(2-\alpha)(3-\alpha)X_{id}(t-3) \end{aligned} \quad (25)$$

Then, (23) can be modified to

$$\begin{aligned} X_{id}(t+1) = & C_{id}(t) + \beta \cdot \ln\left(\frac{1}{u}\right) \cdot mbest_d \\ & - \left(\beta \cdot \ln\left(\frac{1}{u}\right) \pm 1 - \alpha\right) X_{id}(t) + XX_{id}(t) \end{aligned} \quad (26)$$

and (24) can be also rewritten as

$$\begin{aligned} X_{id}(t+1) = & C_{id}(t) - \beta \cdot \ln\left(\frac{1}{u}\right) \cdot mbest_d \\ & + \left(\beta \cdot \ln\left(\frac{1}{u}\right) \pm 1 + \alpha\right) X_{id}(t) + XX_{id}(t) \end{aligned} \quad (27)$$

where

$$\begin{aligned} XX_{id}(t) = & \frac{1}{2}\alpha(1-\alpha)X_{id}(t-1) \\ & + \frac{1}{6}\alpha(1-\alpha)(2-\alpha)X_{id}(t-2) \\ & + \frac{1}{24}\alpha(1-\alpha)(2-\alpha)(3-\alpha)X_{id}(t-3) \end{aligned} \quad (28)$$

It can be seen that from (23) and (24), the position updating of particles depends not only on the position of the previous particle but also on the historical position of the particle in different points in time. The position updating of particles is the result of long-term memory, which can protect the population distribution and diversity to a certain extent. The flowchart of the proposed quantum-behaved swarm optimization with the fractional position (FQPSO) is shown in Table 2.

## Experiments

### Experimental setup

To validate the performance of the proposed FQPSO, 8 benchmark functions [28–30] listed in Table 3 were used to compare FQPSO with PSO and QPSO under the same maximum function evaluations (FEs). For FQPSO, the order was set to from 0.1 to 0.9 with step 0.1. Firstly, to investigate the impact of a fractional position in the proposed algorithm, we use FQPSO with different fractional-orders to compare to QPSO. Then, the best results of FQPSO were used for comparison with other variants of PSO including PSO [31], QPSO, PSO with both chaotic

**Table 2. The main steps of FQPSO.**

**Algorithm 3**

---

```

Initialize FQPSO parameters;
Initialize population: random  $X_i$ 
For each particle  $i \in [1, s]$ 
  compute  $f$ 
  If  $f(x_i) < f(p_{bi})$ 
     $p_{bi} = X_i$ 
  End
  If  $f(p_{bi}) < f(gb)$ 
     $gb = p_{bi}$ 
  End
Calculate Q using the equation
If  $rand > 0.5, mbest_d < X_{id}(t)$  or  $rand < 0.5, mbest_d > X_{id}(t)$ 
 $X_{id}(t+1) = C_{id}(t) + \beta \cdot \ln(\frac{1}{u}) \cdot mbest_d - (\beta \cdot \ln(\frac{1}{u}) \pm 1 - \alpha)X_{id}(t) + XX_{id}(t)$ 
Else
If  $rand > 0.5, mbest_d > X_{id}(t)$  or  $rand < 0.5, mbest_d < X_{id}(t)$ 
 $X_{id}(t+1) = C_{id}(t) - \beta \cdot \ln(\frac{1}{u}) \cdot mbest_d + (\beta \cdot \ln(\frac{1}{u}) \pm 1 + \alpha)X_{id}(t) + XX_{id}(t)$ 
End
 $t = t+1$ 
Until termination criterion is satisfied

```

---

<https://doi.org/10.1371/journal.pone.0218285.t002>

sequences and crossover operation (CCPSO) [32], naive PSO (NPSO) [33] and moderate-random-search strategy PSO (MRPSO) [34]. The parameters of the compared algorithms were set as recommended in the original references. Since the impact of population size on the performance of PSO-based methods is of the minimum significance [35], all experiments in this research were performed with a population size of 20. [34].

The parameters of the compared algorithms were set as recommended in the original references. Since the impact of population size on the performance of PSO-based methods is of the minimum significance [35], all experiments in this research were performed with a population size of 20.  $\beta$  is computed according to the following formula:

$$\beta(t) = (\beta_0 - \beta_1)(t_{\max} - t)/t_{\max} + \beta_1, \quad (29)$$

where  $\beta_0 = 0.8$ ,  $\beta_1 = 0.6$ ,  $t$  is the current number of iterations and  $t_{\max}$  is the maximum number of iterations [36].

## Testing FQPSO with different fractional-order

Since QPSO is a stochastic algorithm, it will lead to a different trajectory convergence every time. Therefore, the simulations were performed 50 times with each value in the parameter set  $\alpha = \{0, 0.1, 0.2, \dots, 1\}$ . In Figs 1 and 2, the result is given for the adopted optimization functions  $f_j$ ,  $j = 1, 2, \dots, 8$ . To show the gains achieved by our proposed algorithm, three groups of the experiments were performed. In unimodal functions ( $f_1$ – $f_5$ , Group 1) and multimodal functions ( $f_6$ – $f_8$ , Group 2) tests, the maximum numbers of FEs were set to 10000, 30000 and 100000, for 10-D, 30-D and 100-D problems, respectively. In the results, we provided the best results and the mean results. The final results over 50 runs of FQPSO are summarized in Tables 4–7.

Fig 1 shows the performance of FQPSO with different fractional-orders in Group 1.  $f_1$ , a Sphere function, is the most widely used unimodal test function. Compared with algorithms with integer-order position, FQPSO shows the best results for this function. Similar results were obtained for other unimodal functions. The improvements achieved by FQPSO on these unimodal functions suggest that fractional-order methods are better at a fine-gained search

Table 3. Benchmark test functions.

| F     | Formula  | Range        | $X_{\max}$ | $f_{\min}$ | $X^*$ |
|-------|--|--------------|------------|------------|-------|
| $f_1$ | $\sum_{i=1}^n x_i^2$   | [-100,100]   | 100        | 0          | 0     |
| $f_2$ | $\sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)^2$   | [-100,100]   | 100        | 0          | 0     |
| $f_3$ | $\sum_{i=1}^n i \cdot x_i^2$   | [-100,100]   | 100        | 0          | 0     |
| $f_4$ | $\sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $   | [-10,10]     | 100        | 0          | 0     |
| $f_5$ | $\sum_{i=1}^n (x_i^2 + \prod_{i=1}^n (x_i)^2)$   | [-10,10]     | 100        | 0          | 0     |
| $f_6$ | $\sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$  | [-100,100]   | 100        | 0          | 0     |
| $f_7$ | $\sum_{i=1}^n \left( \sum_{k=0}^{20} (0.5)^k \cos(2\pi(3)^k(x_i + 0.5)) \right) - n \sum_{k=0}^{20} ((0.5)^k \cos(2\pi \cdot 3^k \cdot 0.5))$            | [-5.12,5.12] | 5.12       | 0          | 0     |
| $f_8$ | $-20 \exp \left( -0.2 \left( \frac{1}{n} \sum_{i=1}^n x_i^2 \right)^{1/2} \right) - \exp \left( \frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i \right) + 20 + e$ | [-5.12,5.12] | 32         | 0          | 0     |

$X^*$  denotes the global optimum.

<https://doi.org/10.1371/journal.pone.0218285.t003>

than integer-order ones. However, it is also worth noting that the performances of FQPSO algorithms with orders 0.1 and 0.2 were not better than integer-order. The reason is that (35) is just an approximation of  $D^\alpha$  and the approximation accuracy of  $D^{0.1}$  and  $D^{0.2}$  is not good enough. From Fig 1, we can see that most FQPSO methods' convergence accuracies are better

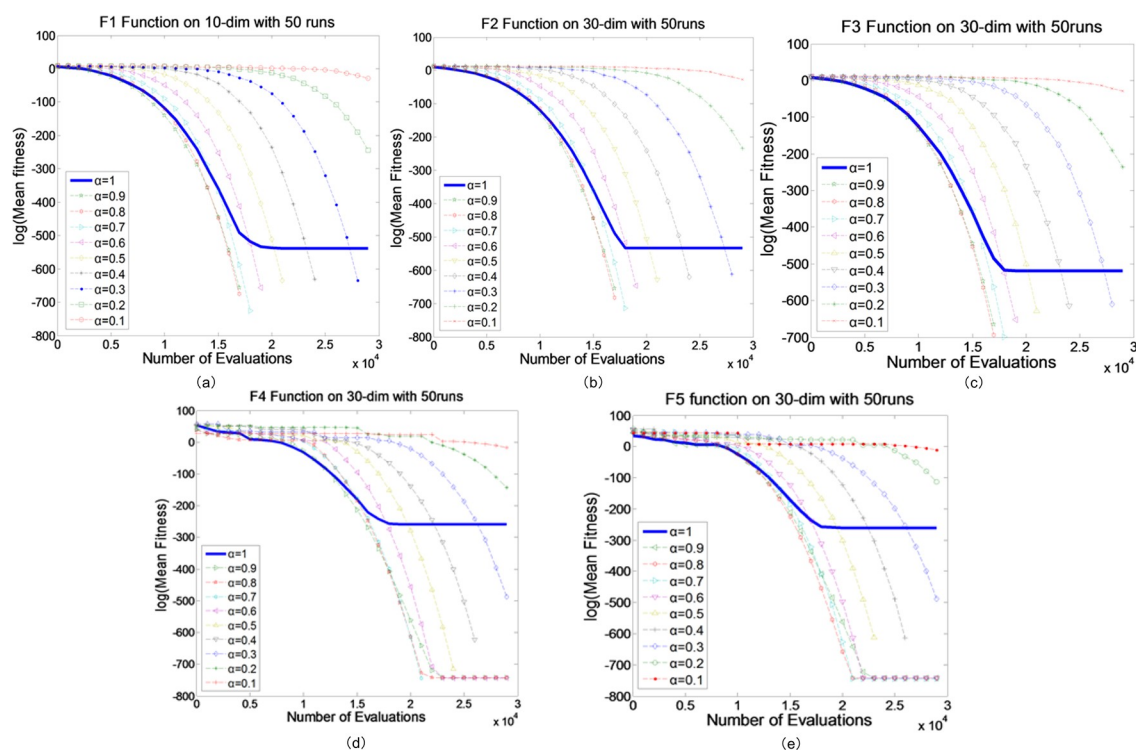
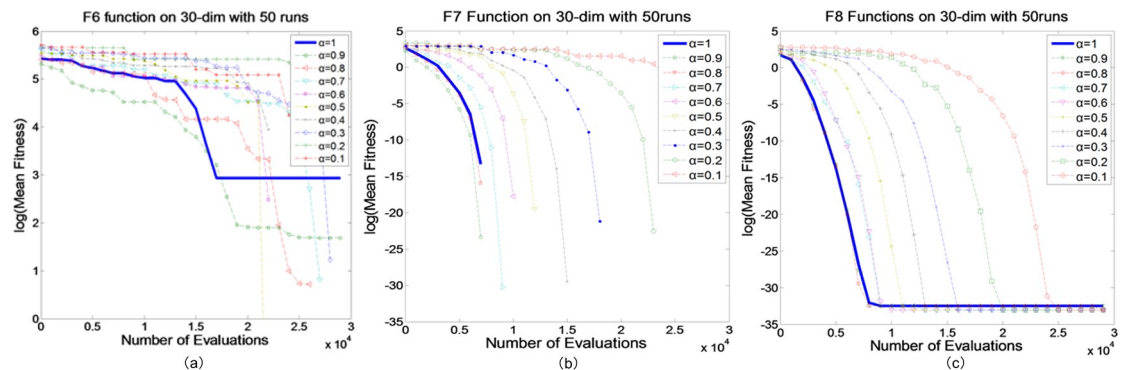


Fig 1. Comparison between FQPSO with different fractional-order on Group 1. (a)  $f_1$ , (b)  $f_2$ , (c)  $f_3$ , (d)  $f_4$ , (e)  $f_5$ .

<https://doi.org/10.1371/journal.pone.0218285.g001>





**Fig 2.** Comparison between FQPSO with different fractional-order on Group 2. (a)  $f_6$ , (b)  $f_7$ , (c)  $f_8$ .

<https://doi.org/10.1371/journal.pone.0218285.g002>

than QPSO. For 10-D and 30-D problems in function 1, 2 and 3 showed in Fig 1A, 1B and 1C and Tables 5 and 6, when  $0.3 \leq \alpha \leq 0.9$ , the convergence accuracies are better than QPSO. For 10-D and 30-D problems in function 4, the convergence accuracies of are better than QPSO, when  $0.4 \leq \alpha \leq 0.9$ . For 10-D and 30-D problems in function 5, the convergence accuracies of are better than QPSO, when  $0.2 \leq \alpha \leq 0.9$ . Tables 5–7 also show that the convergence accuracies are better than QPSO in function 1, 2, 3, 4 and 5 when  $0.7 \leq \alpha \leq 0.9$  for 100-D problems.

In general, we can always find an appropriate fractional order so that the convergence accuracy of the algorithm is better than the integer order algorithm in Group 1.

In Fig 2, for  $f_6$ ,  $f_7$  and  $f_8$ , the numbers of local minima will increase dramatically as the dimension of the function raises. In this part, we mainly investigated the capability of global searching.  $f_6$  is the generalized Rastrigin's function, which is the most widely used test multi-modal functions in PSO algorithm, and tends to be trapped by local minimums. Considering

**Table 4.** Comparison between FQPSOs with different fractional-order on function 1–2.

| Fractional-order |      | $f_1$                   |                         |                          | $f_2$                   |                         |                          |
|------------------|------|-------------------------|-------------------------|--------------------------|-------------------------|-------------------------|--------------------------|
|                  |      | Dim = 10<br>FEs = 10000 | Dim = 30<br>FEs = 30000 | Dim = 100<br>FEs = 30000 | Dim = 10<br>FEs = 10000 | Dim = 30<br>FEs = 30000 | Dim = 100<br>FEs = 30000 |
| $\alpha = 1$     | Best | 5.6324e-267             | 2.3453e-241             | 2.9833e-230              | 3.7568e-258             | 1.7033e-237             | 1.4328e-163              |
|                  | Mean | 4.5321e-265             | 1.5837e-239             | 3.5636e-231              | 1.8379e-255             | 6.8877e-234             | 4.5673e-158              |
| $\alpha = 0.9$   | Best | 0                       | 0                       | 5.3234e-269              | 0                       | 0                       | 5.3293e-204              |
|                  | Mean | 0                       | 0                       | 4.5639e-267              | 0                       | 0                       | 6.3214e-201              |
| $\alpha = 0.8$   | Best | 0                       | 0                       | 7.3535e-248              | 0                       | 0                       | 8.5313e-198              |
|                  | Mean | 0                       | 0                       | 6.4356e-246              | 0                       | 0                       | 4.2314e-197              |
| $\alpha = 0.7$   | Best | 0                       | 0                       | 5.3623e-242              | 0                       | 0                       | 5.3241e-188              |
|                  | Mean | 0                       | 0                       | 7.5323e-239              | 0                       | 0                       | 3.1235e-185              |
| $\alpha = 0.6$   | Best | 0                       | 0                       | 7.4342e-197              | 0                       | 0                       | 8.4232e-163              |
|                  | Mean | 0                       | 0                       | 6.4329e-194              | 0                       | 0                       | 5.3123e-161              |
| $\alpha = 0.5$   | Best | 0                       | 0                       | 5.3252e-146              | 0                       | 0                       | 4.3242e-141              |
|                  | Mean | 0                       | 0                       | 5.9753e-143              | 0                       | 0                       | 8.5223e-140              |
| $\alpha = 0.4$   | Best | 0                       | 0                       | 9.5332e-108              | 0                       | 0                       | 5.3213e-111              |
|                  | Mean | 0                       | 0                       | 5.4256e-99               | 0                       | 0                       | 6.4132e-108              |
| $\alpha = 0.3$   | Best | 0                       | 0                       | 6.5352e-56               | 0                       | 0                       | 8.5231e-66               |
|                  | Mean | 0                       | 0                       | 5.953e-50                | 0                       | 0                       | 5.3145e-49               |
| $\alpha = 0.2$   | Best | 2.1613e-236             | 5.1257e-106             | 6.4235e-18               | 2.1552e-152             | 1.0355e-105             | 1.2345e-16               |
|                  | Mean | 6.6966e-250             | 2.2847e-100             | 3.4562e-11               | 7.0445e-145             | 2.2098e-98              | 8.5242e-08               |
| $\alpha = 0.1$   | Best | 2.613e-53               | 3.5677e-16              | 4.5712e-08               | 1.4244e-36              | 4.5673e-15              | 5.3113e-06               |
|                  | Mean | 1.2778e-45              | 3.9080e-12              | 3.4564e-05               | 2.0523e-24              | 2.4097e-11              | 4.5313e-04               |

<https://doi.org/10.1371/journal.pone.0218285.t004>

Table 5. Comparison between FQPSOs with different fractional-order on function 3–4.

| Fractional-order |      | $f_3$                   |                         |                          | $f_4$                   |                         |                          |
|------------------|------|-------------------------|-------------------------|--------------------------|-------------------------|-------------------------|--------------------------|
|                  |      | Dim = 10<br>FEs = 10000 | Dim = 30<br>FEs = 30000 | Dim = 100<br>FEs = 30000 | Dim = 10<br>FEs = 10000 | Dim = 30<br>FEs = 30000 | Dim = 100<br>FEs = 30000 |
| $\alpha = 1$     | Best | 6.325e-259              | 1.0111e-228             | 4.3529e-194              | 1.4622e-248             | 6.2412e-111             | 6.4353e-56               |
|                  | Mean | 7.424e-255              | 1.2325e-224             | 7.5332e-179              | 1.2011e-215             | 2.6011e-99              | 8.4224e-41               |
| $\alpha = 0.9$   | Best | 0                       | 0                       | 8.4243e-223              | 0                       | 3.5000e-323             | 7.5224e-75               |
|                  | Mean | 0                       | 0                       | 6.5324e-215              | 0                       | 5.9000e-323             | 6.3243e-71               |
| $\alpha = 0.8$   | Best | 0                       | 0                       | 9.5352e-245              | 0                       | 1.0000e-323             | 8.4242e-77               |
|                  | Mean | 0                       | 0                       | 5.3563e-228              | 0                       | 4.9000e-324             | 4.2412e-73               |
| $\alpha = 0.7$   | Best | 0                       | 0                       | 7.5324e-237              | 0                       | 0                       | 6.3242e-65               |
|                  | Mean | 0                       | 0                       | 6.4256e-229              | 0                       | 0                       | 5.4224e-61               |
| $\alpha = 0.6$   | Best | 0                       | 0                       | 8.5363e-185              | 0                       | 0                       | 9.4245e-39               |
|                  | Mean | 0                       | 0                       | 6.4353e-178              | 0                       | 1.000e-323              | 6.2345e-36               |
| $\alpha = 0.5$   | Best | 0                       | 0                       | 9.5363e-166              | 0                       | 0                       | 4.2135e-34               |
|                  | Mean | 0                       | 0                       | 7.5352e-161              | 0                       | 0                       | 1.2356e-33               |
| $\alpha = 0.4$   | Best | 0                       | 0                       | 8.4256e-154              | 6.2616e-251             | 0                       | 5.2214e-27               |
|                  | Mean | 0                       | 0                       | 9.5324e-151              | 5.5247e-243             | 0                       | 5.6241e-21               |
| $\alpha = 0.3$   | Best | 0                       | 0                       | 6.3245e-134              | 2.6343e-147             | 3.7503e-228             | 7.5213e-18               |
|                  | Mean | 0                       | 3.78e-321               | 5.4242e-131              | 9.3570e-141             | 2.8574e-221             | 5.2134e-15               |
| $\alpha = 0.2$   | Best | 2.1613e-236             | 9.7128e-104             | 5.3523e-67               | 1.5234e-74              | 3.9143e-73              | 5.6231e-12               |
|                  | Mean | 6.6966e-220             | 4.7649e-100             | 6.3213e-60               | 2.0325e-68              | 5.1233e-69              | 7.5234e-07               |
| $\alpha = 0.1$   | Best | 1.2583e-45              | 7.5296e-15              | 3.4525e-06               | 5.9892e-30              | 6.6300e-13              | 4.3241e-08               |
|                  | Mean | 5.6267e-39              | 7.9333e-12              | 5.3256e-03               | 2.2309e-25              | 4.2332e-10              | 3.3413e-05               |

<https://doi.org/10.1371/journal.pone.0218285.t005>

more orders to search in the solution space, FQPSO gets more favorable results than the compared algorithms.  $f_7$  is the Ackley function and according to the results in Table 8, the performances of FQPSO have little changes with the variation of dimension and achieve the best

Table 6. Comparison between FQPSOs with different fractional-order on function 5–6.

| Fractional-order |      | $f_5$                   |                         |                          | $f_6$                   |                         |                          |
|------------------|------|-------------------------|-------------------------|--------------------------|-------------------------|-------------------------|--------------------------|
|                  |      | Dim = 10<br>FEs = 10000 | Dim = 30<br>FEs = 30000 | Dim = 100<br>FEs = 30000 | Dim = 10<br>FEs = 10000 | Dim = 30<br>FEs = 30000 | Dim = 100<br>FEs = 30000 |
| $\alpha = 1$     | Best | 9.397e-189              | 1.1027e-154             | 5.4324e-109              | 0.9950                  | 11.9395                 | 32.4524                  |
|                  | Mean | 3.5943e-185             | 1.1027e-155             | 4.5632e-110              | 1.5919                  | 16.0250                 | 56.3234                  |
| $\alpha = 0.9$   | Best | 0                       | 0                       | 6.4242e-134              | 1.7764e-15              | 0.9950                  | 1.7432                   |
|                  | Mean | 0                       | 0                       | 5.6321e-130              | 0.1090                  | 6.6484                  | 4.5252                   |
| $\alpha = 0.8$   | Best | 0                       | 0                       | 3.4245e-129              | 0                       | 0.4517                  | 0.9985                   |
|                  | Mean | 0                       | 0                       | 5.6324e-126              | 0                       | 2.2933                  | 10.4245                  |
| $\alpha = 0.7$   | Best | 0                       | 0                       | 8.7432e-119              | 0                       | 0.0297                  | 0.5943                   |
|                  | Mean | 0                       | 0                       | 4.5256e-115              | 0                       | 2.4696                  | 3.4255                   |
| $\alpha = 0.6$   | Best | 0                       | 0                       | 6.5241e-89               | 0                       | 0.5342                  | 1.4252                   |
|                  | Mean | 0                       | 0                       | 5.3241e-88               | 0                       | 12.4468                 | 9.3245                   |
| $\alpha = 0.5$   | Best | 0                       | 0                       | 8.5352e-82               | 2.456e-08               | 4.5314e-06              | 0.04255                  |
|                  | Mean | 0                       | 0                       | 5.4232e-77               | 0.01413                 | 6.4245e-04              | 0.4256                   |
| $\alpha = 0.4$   | Best | 0                       | 0                       | 9.4256e-74               | 0.1389                  | 6.6789                  | 11.3352                  |
|                  | Mean | 0                       | 0                       | 6.6322e-72               | 1.5126                  | 56.6533                 | 57.3241                  |
| $\alpha = 0.3$   | Best | 4.2206e-251             | 5.6014e-317             | 5.3214e-66               | 0.3100                  | 0.6789                  | 16.4252                  |
|                  | Mean | 2.3626e-231             | 1.6254e-286             | 5.3242e-65               | 0.3394                  | 3.3932                  | 54.1343                  |
| $\alpha = 0.2$   | Best | 1.5239e-120             | 2.4207e-99              | 1.2345e-49               | 1.6976e-10              | 10.2080                 | 26.4952                  |
|                  | Mean | 2.9750e-102             | 1.2928e-77              | 2.4562e-46               | 2.1109e-09              | 185.937                 | 192.345                  |
| $\alpha = 0.1$   | Best | 4.5632e-42              | 1.2071e-16              | 9.5224e-08               | 1.0415                  | 36.4428                 | 86.4211                  |
|                  | Mean | 2.9225e-30              | 1.6344e-06              | 3.4521e-04               | 16.0174                 | 66.9430                 | 211.245                  |

<https://doi.org/10.1371/journal.pone.0218285.t006>

Table 7. Comparison between FQPSOs with different fractional-order on function 7–8.

| Fractional-order |      | $f_7$                   |                         |                          | $f_8$                   |                         |                          |
|------------------|------|-------------------------|-------------------------|--------------------------|-------------------------|-------------------------|--------------------------|
|                  |      | Dim = 10<br>FEs = 10000 | Dim = 30<br>FEs = 30000 | Dim = 100<br>FEs = 30000 | Dim = 10<br>FEs = 10000 | Dim = 30<br>FEs = 30000 | Dim = 100<br>FEs = 30000 |
| $\alpha = 1$     | Best | 7.1054e-15              | 2.5251e-10              | 8.5322e-08               | 1.4622e-248             | 7.9936e-15              | 4.5231e-06               |
|                  | Mean | 9.9476e-15              | 4.8798e-07              | 5.4252e-07               | 1.2011e-215             | 7.9936e-15              | 3.4251e-05               |
| $\alpha = 0.9$   | Best | 1.9257e-17              | 5.6302e-11              | 6.5232e-15               | 0                       | 4.4409e-15              | 2.3451e-07               |
|                  | Mean | 1.8609e-13              | 3.7460e-08              | 7.5323e-11               | 0                       | 6.9278e-15              | 4.4123e-06               |
| $\alpha = 0.8$   | Best | 4.9016e-25              | 5.4000e-10              | 7.4213e-19               | 0                       | 1.6409e-15              | 8.4222e-07               |
|                  | Mean | 5.6360e-18              | 1.1253e-07              | 9.3421e-15               | 0                       | 5.1514e-15              | 5.3245e-06               |
| $\alpha = 0.7$   | Best | 8.3079e-33              | 5.4000e-20              | 6.4231e-24               | 0                       | 4.4409e-15              | 6.4134e-07               |
|                  | Mean | 1.1100e-22              | 8.0348e-15              | 5.3313e-19               | 0                       | 4.4708e-15              | 5.3121e-06               |
| $\alpha = 0.6$   | Best | 5.0220e-44              | 3.4005e-12              | 1.2334e-14               | 0                       | 4.4409e-15              | 4.3311e-07               |
|                  | Mean | 2.4418e-26              | 1.0209e-08              | 5.4134e-13               | 0                       | 4.4409e-15              | 4.2134e-06               |
| $\alpha = 0.5$   | Best | 6.5183e-43              | 1.6486e-11              | 7.4231e-09               | 0                       | 4.4409e-15              | 1.2134e-08               |
|                  | Mean | 9.7154e-22              | 2.2779e-09              | 9.3134e-06               | 0                       | 4.4409e-15              | 5.4131e-06               |
| $\alpha = 0.4$   | Best | 4.765e-28               | 3.1412e-17              | 8.4255e-10               | 6.2616e-251             | 4.4409e-15              | 3.4131e-07               |
|                  | Mean | 8.98017e-13             | 3.1149e-14              | 5.3424e-08               | 5.5247e-243             | 4.4409e-15              | 3.4111e-06               |
| $\alpha = 0.3$   | Best | 2.0456e-16              | 3.9874e-13              | 8.4325e-12               | 2.6343e-147             | 3.7503e-15              | 5.3314e-08               |
|                  | Mean | 1.1142e-14              | 1.0676e-10              | 4.5231e-10               | 9.3570e-141             | 2.8574e-15              | 4.3141e-07               |
| $\alpha = 0.2$   | Best | 3.1005e-19              | 4.2733e-17              | 3.4112e-05               | 1.5234e-74              | 6.7564e-15              | 6.5131e-03               |
|                  | Mean | 6.01e-13                | 2.6818e-12              | 1.2144e-03               | 2.0325e-68              | 6.9345e-15              | 4.5131e-01               |
| $\alpha = 0.1$   | Best | 0.004777                | 0.0813                  | 0.1133                   | 5.9892e-30              | 7.9835e-15              | 0.1314                   |
|                  | Mean | 0.0865                  | 0.7666                  | 1.2134                   | 2.2309e-25              | 8.2343e-15              | 1.2144                   |

<https://doi.org/10.1371/journal.pone.0218285.t007>

results on each dimension. Function  $f_8$  is the Weierstrass function, which is continuous everywhere, but differentiable nowhere. In short, FQPSO reaches the global optimum on 10 and 30 dimensions. In Fig 2 and Tables 6 and 7, it can be observed that except FQPSO with orders 0.1 and 0.2, FQPSO can always achieve better results than QPSO. Meanwhile, for function 6, 7 and 8, the convergence accuracies are better than QPSO when  $0.3 \leq \alpha \leq 0.9$ .

In summary, FQPSO has superior ability in tackling multimodal functions compared with other algorithms. We can always find an appropriate fractional order for the algorithm that has better convergence accuracy than the integer order one in Group 2.

Table 8 shows the time consumption of FQPSO and QPSO in solving function optimization problems. The default time unit is seconds. The experimental results also confirm that the fractional order method only consumes a little more time in each iteration process, and does not cause a lot of waste of time.

## Compare with other variants of PSO

In this experiment, the best results of the FQPSO methods were used for comparison with other variants of PSO, including PSO, QPSO, CCPSO, NPSO and MRPSO. The parameters of the compared algorithms were set according to the recommendations in their original papers. The maximum numbers of FEs were respectively set to 10000 and 30000 for solving 10-D and 30-D problems. All experiments were performed with a population size of 20.

Table 8. Time consumption.

|       | $f_1$  | $f_2$  | $f_3$  | $f_4$  | $f_5$  | $f_6$  | $f_7$  | $f_8$  |
|-------|--------|--------|--------|--------|--------|--------|--------|--------|
| QPSO  | 0.8123 | 0.9245 | 0.8155 | 0.8934 | 0.8688 | 0.9355 | 0.9642 | 0.9942 |
| FQPSO | 0.8471 | 0.9334 | 0.8358 | 0.9548 | 0.8899 | 0.9674 | 0.9856 | 1.032  |

<https://doi.org/10.1371/journal.pone.0218285.t008>

Table 9. Comparison between different PSO algorithms on function 1–3.

| Algorithm |      | $f_1$                   |                         | $f_2$                   |                         | $f_3$                   |                         |
|-----------|------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
|           |      | Dim = 10<br>FEs = 10000 | Dim = 30<br>FEs = 30000 | Dim = 10<br>FEs = 10000 | Dim = 30<br>FEs = 30000 | Dim = 10<br>FEs = 10000 | Dim = 30<br>FEs = 30000 |
| FQPSO     | Best | 0                       | 0                       | 0                       | 1.6812e-313             | 0                       | 0                       |
|           | Mean | 0                       | 0                       | 0                       | 1.9619e-296             | 0                       | 0                       |
|           | std  | 0                       | 0                       | 0                       | 5.3432e-307             | 0                       | 0                       |
| QPSO      | Best | 5.6324e-267             | 2.3453e-241             | 3.7568e-258             | 1.7033e-237             | 1.5e-256                | 1.01e-243               |
|           | Mean | 4.5321e-265             | 1.5837e-239             | 1.8379e-255             | 6.8877e-234             | 8.1e-254                | 1.86e-241               |
|           | std  | 3.4523e-266             | 6.3245e-241             | 2.5431e-256             | 2.9832e-237             | 5.31e-256               | 7.543e-243              |
| PSO       | Best | 7.764e-20               | 6.7954e-14              | 5.8742e-16              | 4.3257e-10              | 5.8734e-21              | 3.876e-12               |
|           | Mean | 1.17e-20                | 1.58e-13                | 4.9000e-15              | 1.2264e-09              | 9.3854e-20              | 2.086e-06               |
|           | std  | 6.3e-20                 | 4.17e-13                | 1.3864e-16              | 2.4987e-10              | 3.5467e-19              | 6.4303e-11              |
| CCPSO     | Best | 3.2341e-97              | 7.4324e-86              | 1.2353e-20              | 9.3557e-16              | 6.3258e-43              | 5.2134e-35              |
|           | Mean | 1.2313e-95              | 6.0851e-84              | 8.3483e-20              | 1.9619e-13              | 8.5423e-42              | 3.6880e-33              |
|           | std  | 2.8734e-97              | 5.3241e-85              | 5.9834e-20              | 5.7934e-15              | 1.5425e-43              | 7.3424e-34              |
| NPSO      | Best | 3.4653e-53              | 5.3789e-38              | 3.4453e-22              | 5.2223e-13              | 1.7431e-14              | 9.3452e-09              |
|           | Mean | 5.2356e-52              | 4.8357e-36              | 9.5151e-18              | 1.2580e-11              | 3.4564e-14              | 7.2875e-06              |
|           | std  | 2.3456e-53              | 9.2134e-37              | 2.3456e-21              | 9.9863e-13              | 2.6731e-14              | 8.4245e-08              |
| MRPSO     | Best | 1.5677e-110             | 5.8723e-97              | 6.3434e-61              | 4.9053e-45              | 3.4546e-85              | 3.4546e-85              |
|           | Mean | 1.239e-109              | 6.2391e-93              | 3.5639e-48              | 4.4788e-44              | 9.4671e-84              | 9.4671e-84              |
|           | std  | 8.2345e-110             | 6.3394e-97              | 9.5332e-51              | 7.3421e-44              | 2.3546e-85              | 2.3546e-85              |

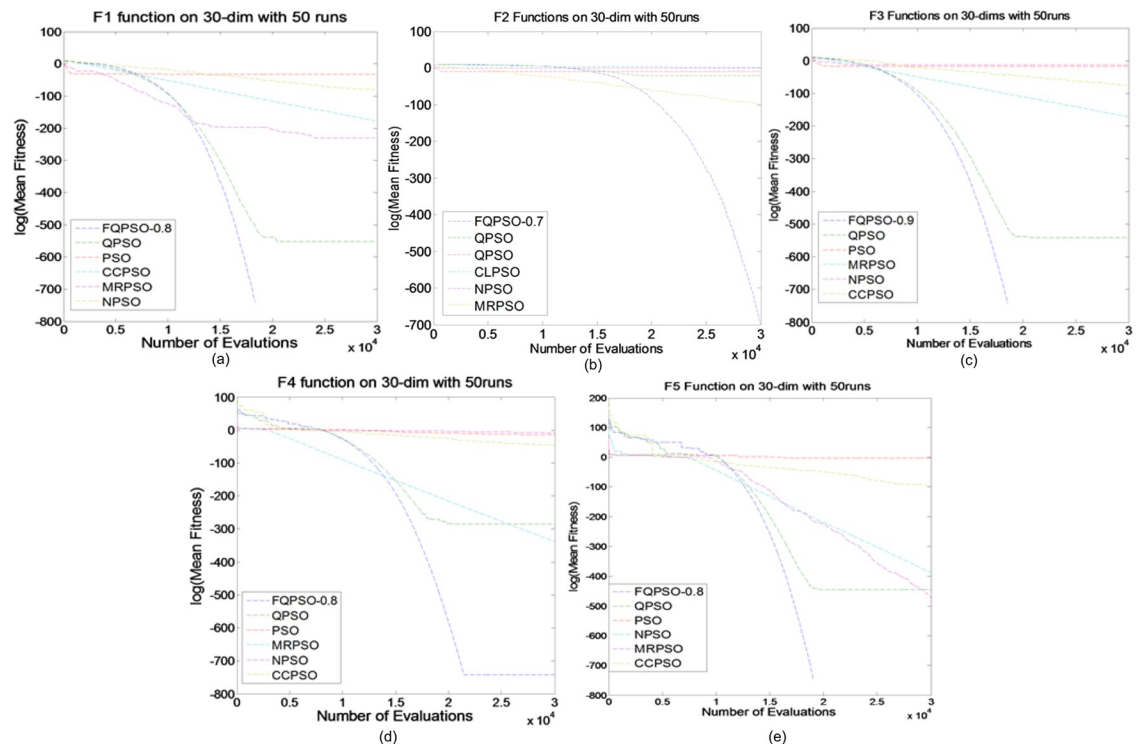
<https://doi.org/10.1371/journal.pone.0218285.t009>

Tables 9 and 10 shows the statistical results of different algorithms on unimodal functions. From the previous results in the last subsection, we can see that FQPSO with  $D^{0.8}$  obtained the best results on functions 1–3 and FQPSO with  $D^{0.7}$  achieved the best results on functions 4–5. We fixed the orders to compare those results with other variants of PSO. The results from different algorithms on these five unimodal functions suggest that FQPSO is better at a fine-gained search than all the other algorithms. The rapid convergence of the FQPSO can be seen as an evidence for our observation in Fig 3. In summary, FQPSO performs best in solving

Table 10. Comparison between different PSO algorithms on function 4–6.

| Algorithm |      | $f_4$                   |                         | $f_5$                   |                         | $f_6$                   |                         |
|-----------|------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
|           |      | Dim = 10<br>FEs = 10000 | Dim = 30<br>FEs = 30000 | Dim = 10<br>FEs = 10000 | Dim = 30<br>FEs = 30000 | Dim = 10<br>FEs = 10000 | Dim = 30<br>FEs = 30000 |
| FQPSO     | Best | 0                       | 0                       | 0                       | 0                       | 0                       | 7.0106e-06              |
|           | Mean | 0                       | 0                       | 0                       | 0                       | 0                       | 1.4384                  |
|           | std  | 0                       | 0                       | 0                       | 0                       | 0                       | 1.234                   |
| QPSO      | Best | 1.4622e-248             | 6.2412e-146             | 9.397e-189              | 1.1027e-154             | 1.7764e-10              | 15.9395                 |
|           | Mean | 1.2011e-215             | 1.2203e-137             | 3.5943e-185             | 1.1027e-155             | 2.1453e-08              | 16.0250                 |
|           | std  | 5.6231e-235             | 4.562e-137              | 6.3423e-187             | 6.3453e-155             | 5.4214e-09              | 5.3453                  |
| PSO       | Best | 5.324e-13               | 1.0000e-09              | 5.4234e-19              | 5.324e-07               | 4.34e-06                | 7.3474                  |
|           | Mean | 1.5242e-11              | 7.5267e-07              | 2.534e-18               | 0.0571                  | 3.2e-05                 | 7.8363                  |
|           | std  | 2.584e-11               | 6.3324e-08              | 3.4532e-19              | 0.000424                | 3.2e-05                 | 1.3456                  |
| CCPSO     | Best | 1.324e-30               | 7.4352e-23              | 6.4234e-55              | 5.3256e-44              | 0.1234                  | 2.9768                  |
|           | Mean | 8.3453e-27              | 6.3257e-21              | 4.5313e-51              | 9.4075e-42              | 0.1423                  | 4.475                   |
|           | std  | 4.5356e-29              | 5.6485e-22              | 8.5423e-54              | 3.4578e-43              | 0.0542                  | 6.4246                  |
| NPSO      | Best | 5.6354e-09              | 7.5242e-07              | 6.4312e-183             | 4.5353e-175             | 0.34561                 | 5.323                   |
|           | Mean | 4.3563e-07              | 1.8913e-04              | 4.5683e-180             | 2.2115e-169             | 0.40593                 | 5.354                   |
|           | std  | 2.3446e-09              | 4.6452e-06              | 9.4563e-182             | 3.4532e-172             | 0.5289                  | 0.0034                  |
| MRPSO     | Best | 6.4232e-154             | 5.3133e-150             | 8.6431e-212             | 4.3534e-209             | 2.456e-08               | 4.567                   |
|           | Mean | 7.4323e-151             | 2.9806e-147             | 6.4331e-210             | 1.0762e-203             | 4.355e-06               | 4.678                   |
|           | std  | 4.2456e-154             | 4.5624e-149             | 6.7456e-210             | 3.5356e-206             | 5.342e-07               | 0.543                   |

<https://doi.org/10.1371/journal.pone.0218285.t010>



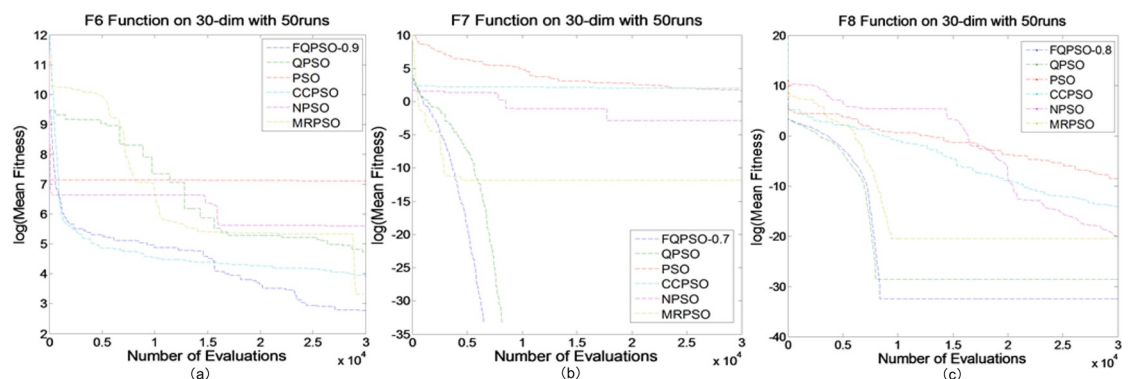
**Fig 3.** Comparison between different PSO algorithms on Group 1. (a)  $f_1$ , (b)  $f_2$ , (c)  $f_3$ , (d)  $f_4$ , (e)  $f_5$ .

<https://doi.org/10.1371/journal.pone.0218285.g003>

unimodal functions among all the algorithms. Tables 9 and 10 and Fig 4 show the performances of different algorithms on Group 2.

Tables 9 and 10 shows the statistical results of different algorithms on unimodal functions. From the previous results in the last subsection, we can see that FQPSO with  $D^{0.8}$  obtained the best results on functions 1–3 and FQPSO with  $D^{0.7}$  achieved the best results on functions 4–5. We fixed the orders to compare those results with other variants of PSO. The results from different algorithms on these five unimodal functions suggest that FQPSO is better at a fine-gained search than all the other algorithms.

The rapid convergence of the FQPSO can be seen as an evidence for our observation in Fig 3.



**Fig 4.** Comparison between different PSO algorithms on Group 2. (a)  $f_6$ , (b)  $f_7$ , (c)  $f_8$ .

<https://doi.org/10.1371/journal.pone.0218285.g004>

Table 11. Comparison between different PSO algorithms on function 7–8.

| Algorithm |      | $f_1$                   |                         | $f_2$                   |                         |
|-----------|------|-------------------------|-------------------------|-------------------------|-------------------------|
|           |      | Dim = 10<br>FEs = 10000 | Dim = 30<br>FEs = 30000 | Dim = 10<br>FEs = 10000 | Dim = 30<br>FEs = 30000 |
| FQPSO     | Best | 7.1054e-15              | 0                       | 7.6438e-34              | 1.6409e-15              |
|           | Mean | 9.9476e-15              | 0                       | 5.3222e-29              | 5.1514e-15              |
|           | std  | 8.4534e-14              | 0                       | 7.4345e-33              | 3.4356e-15              |
| QPSO      | Best | 1.9257e-35              | 2.5251e-26              | 1.7059e-19              | 7.9936e-15              |
|           | Mean | 1.8609e-33              | 3.6749e-21              | 6.8877e-20              | 7.9936e-15              |
|           | std  | 4.3456e-34              | 5.4245e-26              | 7.4352e-19              | 0                       |
| PSO       | Best | 4.9016e-08              | 41.3                    | 5.6534e-14              | 5.4326e-05              |
|           | Mean | 8.345e-06               | 45.043                  | 6.4523e-13              | 7.5331e-04              |
|           | std  | 5.4345e-07              | 10.413                  | 4.5634e-14              | 4.5674e-05              |
| CCPSO     | Best | 8.3079e-05              | 0.0035                  | 6.5341e-19              | 7.4243e-09              |
|           | Mean | 1.1100e-04              | 3.8253                  | 3.2145e-18              | 4.4509e-08              |
|           | std  | 4.5634e-05              | 6.4231                  | 5.6313e-19              | 6.4234e-09              |
| NPSO      | Best | 1.7431e-14              | 3.4531e-05              | 3.4356e-22              | 5.3563e-13              |
|           | Mean | 3.4564e-14              | 0.0801                  | 5.3561e-19              | 8.4356e-13              |
|           | std  | 2.6731e-14              | 0.0004                  | 5.5546e-22              | 6.4325e-13              |
| MRPSO     | Best | 6.5183e-11              | 1.6486e-08              | 5.6356e-27              | 6.4382e-14              |
|           | Mean | 9.7154e-09              | 4.678e-05               | 1.2343e-24              | 9.3435e-14              |
|           | std  | 4.5623e-10              | 6.3245e-07              | 3.4465e-26              | 7.4231e-14              |

<https://doi.org/10.1371/journal.pone.0218285.t011>

In summary, FQPSO performs best in solving unimodal functions among all the algorithms. Tables 10 and 11 and Fig 4 show the performances of different algorithms on Group 2.

From the previous results in the last subsection, it can be noticed that FQPSO with  $D^{0.9}$  obtained the best result on function 6, FQPSO with  $D^{0.7}$  got the best result on function 7, and FQPSO with  $D^{0.9}$  achieved the best result on function 8. We also fixed the orders to compare those results with other variants of PSO. It can be seen that FQPSO obtains the global optimum on 10 and 30 dimensions. FQPSO is better to deal with multimodal functions than other algorithms.

In the results of different PSOs on 30 dimensions also supports our conclusion that FQPSO is suitable for multimodal functions. In summary, FQPSO performs best in solving both unimodal and multimodal functions among all the algorithms.

## Conclusion

Inspired by the properties of fractional calculus, we presented a novel QPSO algorithm incorporated with fractional calculus strategy, which is based on the properties of long time memory and non-locality of fractional calculus. The goal is to employ the proposed method to accelerate not only the convergence speed but also avoid the local optimums. Since the property of fractional calculus enables quantum-particles in FQPSO to appear anywhere during iterations, it significantly improves the global searching ability. Furthermore, FQPSO also increases the convergence rate for the quantum particles. As a result, the proposed FQPSO method achieves more favorable results than all the other algorithms.

## Author Contributions

**Conceptualization:** Lai Xu, Yifei Pu.

**Data curation:** Lai Xu, Aamir Muhammad, Jiliu Zhou.

**Formal analysis:** Lai Xu, Aamir Muhammad.



**Funding acquisition:** Jiliu Zhou, Yi Zhang.

**Methodology:** Lai Xu.

**Project administration:** Yi Zhang.

**Software:** Lai Xu, Aamir Muhammad.

**Supervision:** Yifei Pu, Jiliu Zhou, Yi Zhang.

**Validation:** Aamir Muhammad, Jiliu Zhou, Yi Zhang.

**Visualization:** Lai Xu.

**Writing – original draft:** Lai Xu.

**Writing – review & editing:** Yi Zhang.

## References

1. Eberhart R.; Kennedy J. A new optimizer using particle swarm theory. In Proceedings of International Symposium on MICRO Machine and Human Science.
2. Goldberg D.E. Genetic Algorithms in Search. Optimization and Machine Learning. 1989, 7, 2104–2116.
3. Yuryevich J.; Wong K.P. Evolutionary programming based optimal power flow algorithm. *IEEE Trans Power Syst.* 1999, 14, 1245–1250.
4. Mezuramontes E.; Coello C.A.C. A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Transactions on Evolutionary Computation.* 2005, 9, 1–17.
5. Nordin P. Genetic Programming III—Darwinian Invention and Problem Solving. *IEEE Transactions on Evolutionary Computation.* 2002, 3, 251–253.
6. Bansal J.C.; Deep K. A Modified Binary Particle Swarm Optimization for Knapsack Problems. *Applied Mathematics & Computation.* 2012, 218, 11042–11061.
7. Wang K.P.; Huang L.; Zhou C.G. Particle swarm optimization for traveling salesman problem. *Acta Scientiarum Naturalium Universitatis Jilinensis.* 2003, 3, 1583–1585.
8. Omran M.; Engelbrecht A.P.; Salman A. Particle Swarm Optimization Method Fof Image Clustering. *International Journal of Pattern Recognition & Artificial Intelligence.* 2005, 19, 297–321.
9. Benioff P. The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. *Journal of Statistical Physics.* 1980, 22, 563–593.
10. Kane B.E. A silicon-based nuclear spin quantum computer. *Nature.* 1998, 393, 133–137.
11. Steffen M.; Vandersypen L.; Breyta G. Experimental Realization of Shor's quantum factoring algorithm. *American Physical Society.* 2002, 6866.
12. Sun J.; Feng B.; Xu W. Particle swarm optimization with particles having quantum behavior. In Proceedings of Congress on Evolutionary Computation, 2004.
13. Koeller R.C. Applications of Fractional Calculus to the Theory of Viscoelasticity. *Transactions of the Asme Journal of Applied Mechanics.* 1984, 51, 299–307.
14. Ciesielski M.; Leszczynski J. Numerical treatment of an initial-boundary value problem for fractional partial differential equations. *Signal Processing.* 2006, 86, 2619–2631.
15. Buyukkilic F.; Bayrakdar Z.O.; Demirhan D Investigation of the cumulative diminution process using the Fibonacci method and fractional calculus. *Physica A Statistical Mechanics & Its Applications.* 2016, 444, 336–344.
16. Xi M.; Sun J.; Xu W. An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position. *Advanced Materials Research.* 2008, 591–593, 376–1380.
17. Jiao L.; Stolkin R.; Shang R. Dynamic-context cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation. *Information Sciences.* 2015, 294, 408–422.
18. Pu Y.F. Fractional-Order Euler-Lagrange Equation for Fractional-Order Variational Method: A Necessary Condition for Fractional-Order Fixed Boundary Optimization Problems in Signal Processing and Image Processing. *IEEE Access.* 2016, 99, 1–1.
19. Pu Y.F.; Siarry P.; Chatterjee A.; Wang Z.N.; Zhang Y.; Liu Y.G.; Zhou J.L.; Wang Y. A Fractional-Order Variational Framework for Retinex: Fractional-Order Partial Differential Equation-Based Formulation for

- Multi-Scale Nonlocal Contrast Enhancement with Texture Preserving. *IEEE Transactions on Image Processing*. 2017, 27, 1214–1229. <https://doi.org/10.1109/TIP.2017.2779601> PMID: 29990194
20. Scherer R.; Kalla S.L.; Tang Y. The Grunwald-Letnikov method for fractional differential equations. *Computers & Mathematics with Applications*. 2011, 62, 902–917.
21. Abbas S.; Benchohra M. Nonlinear Fractional Order Riemman-Liouville Volterra-Stieltjes Partial Integral Equations on Unbounded Domains. *Communications in Mathematical Analysis*. 2013, 14, 104–117.
22. Abdeljawad T. On Riemann and Caputo fractional differences. *Computers & Mathematics with Applications*. 2011, 62, 1602–1611.
23. Clerc M.; Kennedy J. The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space. *IEEE Transactions on Evolutionary Computation*. 2002, 6, 58–73.
24. Bonyadi M.R.; Michalewicz Z. Analysis of Stability, Local Convergence, and Transformation Sensitivity of a Variant of the Particle Swarm Optimization Algorithm. *IEEE Transactions on Evolutionary Computation*. 2016, 20, 370–385.
25. Sun J.; Feng B.; Xu W.B. Particle swarm optimization with particles having quantum behavior. In *Proceedings of Proc Congress on Evolutionary Computation*. 2004.
26. Pu Y.F.; Zhou J.L.; Yuan X. Fractional Differential Mask: A Fractional Differential-Based Approach for Multiscale Texture Enhancement. *IEEE Transactions on Image Processing*. 2010, 19, 491–511. <https://doi.org/10.1109/TIP.2009.2035980> PMID: 19933015
27. Pires E.J.S.; Machado J.A.T.; Oliveira P.B.D.M. Particle swarm optimization with fractional-order velocity. *Nonlinear Dynamics*. 2010, 61, 295–301.
28. Kiranyaz S.; Ince T.; Yildirim A. Fractional Particle Swarm Optimization in Multidimensional Search Space. *IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics*. 2010, 40, 298–319.
29. Liang J.J.; Suganthan P.N.; Deb K. Novel composition test functions for global numerical optimization. In *Proceedings of Swarm Intelligence Symposium*. 2015.
30. Yao X.; Liu Y.; Lin G. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*. 1996, 3, 82–102.
31. Shi Y.; Eberhart R. A modified particle swarm optimizer. In *Proceedings of Advances in Natural Computation*.
32. Park J.B.; Jeong Y.W.; Shin J.R. An Improved Particle Swarm Optimization for Nonconvex Economic Dispatch Problems. *IEEE Transactions on Power Systems*. 2010, 25, 156–166.
33. Qin J.; Liang Z. A naive Particle Swarm Optimization. In *Proceedings of Evolutionary Computation IEEE*.
34. Gao H.; Xu W. A New Particle Swarm Algorithm and Its Globally Convergent Modifications. *IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society*. 2011, 41, 1334.
35. Bergh F.V.D.; Engelbrecht A.P. Effect of swarm size on cooperative particle swarm optimizers. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation Conference (GECCO)*, 2001, 892–899.
36. Sun J.; Wu X.; Palade V.; Fang W.; Lai C.H.; Xu W.B. Convergence analysis and improvements of quantum-behaved particle swarm optimization. *Information Sciences*, 2012, 193, 81–103.