



QPSO-CD: quantum-behaved particle swarm optimization algorithm with Cauchy distribution

Amandeep Singh Bhatia¹ · Mandeep Kaur Saggi² · Shenggen Zheng³

Received: 17 February 2020 / Accepted: 20 August 2020 / Published online: 22 September 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Motivated by the particle swarm optimization (PSO) and quantum computing theory, we have presented a quantum variant of PSO (QPSO) mutated with Cauchy operator and natural selection mechanism (QPSO-CD) from evolutionary computations. The performance of proposed hybrid quantum-behaved particle swarm optimization with Cauchy distribution (QPSO-CD) is investigated and compared with its counterparts based on a set of benchmark problems. Moreover, QPSO-CD is employed in well-studied constrained engineering problems to investigate its applicability. Further, the correctness and time complexity of QPSO-CD are analyzed and compared with the classical PSO. It has been proved that QPSO-CD handles such real-life problems efficiently and can attain superior solutions in most of the problems. The experimental results shown that QPSO associated with Cauchy distribution and natural selection strategy outperforms other variants in context of stability and convergence.

Keywords Quantum-behaved algorithm · Particle swarm optimization · Engineering design problems · Cauchy distribution · Quantum computing

1 Introduction

In the late nineteenth century, the theory of classical mechanics experienced several issues in reporting the physical phenomena of light masses and high velocity microscopic particles. In 1920s, Bohr's atomic theory [1], Heisenberg's discovery of quantum mechanics [2] and Schrödinger's [3] discovery of wave mechanics influenced the conception of a new field, i.e., the quantum mechanics. In 1982, Feynman

✉ Amandeep Singh Bhatia
amandeepbhatia.singh@gmail.com

¹ Chitkara University Institute of Engineering and Technology, Chitkara University, Patiala, Punjab, India

² Department of Computer Science, Thapar Institute of Engineering and Technology, Patiala, India

³ Peng Cheng Laboratory, Shenzhen, China

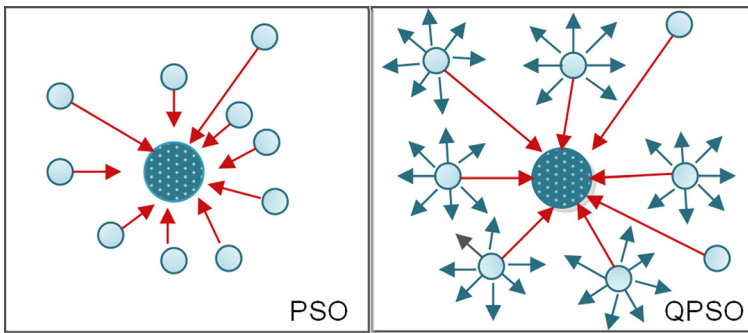


Fig. 1 Particles movement in PSO and QPSO algorithm

[4] stated that quantum mechanical systems can be simulated by quantum computers in exponential time, i.e., better than with classical computers. Till then, the concept of quantum computing was thought to be only a theoretical possibility, but over the last three decades the research has evolved such as to make quantum computing applications a realistic possibility [5].

In the last two decades, the field of swarm intelligence has got overwhelming response among research communities. It is inspired by nature and aims to build decentralized and self-organized systems by collective behavior of individual agents with each other and with their environment. The research foundation of swarm intelligence is constructed mostly upon two families of optimization algorithms, i.e., ant colony optimization (Dorigo et al. [6] and Colormi et al. [7]) and particle swarm optimization (PSO) (Kennedy and Eberhart [8]). Originally, the swarm intelligence is inspired by certain natural behaviors of flocks of birds and swarms of ants.

In the mid 1990s, particle swarm optimization technique was introduced for continuous optimization, motivated by flocking of birds. The evolution of PSO-based bio-inspired techniques has been in an expedite development in the last two decades. It has got attention from different fields such as inventory planning [9], power systems [10], manufacturing [11], communication networks [12], support vector machines [13], to estimate binary inspiral signal [14], gravitational waves [15] and many more. Similar to evolutionary genetic algorithm, it is inspired by simulation of social behavior, where each individual is called particle, and group of individuals is called swarm. In multi-dimensional search space, the position and velocity of each particle represent a probable solution. Particles fly around in a search space seeking potential solution. At each iteration, each particle adjusts its position according to the goal of its own and its neighbors. Each particle in a neighborhood shares the information with others [16]. Later, each particle keeps the record of best solution experienced so far to update their positions and adjust their velocities accordingly.

Since the first PSO algorithm proposed, the several PSO algorithms have been introduced with plethora of alterations. Recently, the combination of quantum computing, mathematics and computer science have inspired the creation of optimization techniques. Initially, Narayanan and Moore [17] introduced quantum-inspired genetic algorithm (QGA) in 1995. Later, Sun et al. [16] applied the quantum laws of mechan-

ics to PSO and proposed quantum-inspired particle swarm optimization (QPSO). It is the commencement of quantum-behaved optimization algorithms, which has subsequently made a significant impact on the academic and research communities alike.

Recently, Yuanyuan and Xiyu [18] proposed a quantum evolutionary algorithm to discover communities in complex social networks. Its applicability is tested on five real social networks, and results are compared with classical algorithms. It has been proved that PSO lacks convergence on local optima, i.e., it is tough for PSO to come out of the local optimum once it confines into optimal local region. QPSO with mutation operator (QPSO-MO) is proposed to enhance the diversity to escape from local optimum in search [19]. Protopopescu and Barhen [20] solved set of global optimization problems efficiently using quantum algorithms. In future, the proposed algorithm can be integrated with matrix product state-based quantum classifier for supervised learning [21–23].

In this paper, we have combined QPSO with Cauchy mutation operator to add long jump ability for global search and natural selection mechanism for elimination of particles. The results showed that it has great tendency to overcome the problem of trapping into local search space. Therefore, the proposed hybrid QPSO strengthened the local and global search ability and outperformed the other variants of QPSO and PSO due to fast convergence feature.

The illustration of particles movement in PSO and QPSO algorithm is shown in Fig. 1. The big circle at center denotes the particle with the global position and other circles are particles. The particles located away from global position are lagged particles. The blue color arrows signify the directions of other particles, and the big red arrows point toward the side in which it goes with high probability. During iterations, if the lagged particle is unable to find better position as compared to present global position in PSO, then their impact is null on the other particles. But, in QPSO, the lagged particles move with higher probability in the direction of gbest position. Thus, the contribution of lagged particles is more to the solution in QPSO in comparison with PSO algorithm.

The organization of rest of this paper is as follows: Sect. 2 is devoted to prior work. In Sect. 3, the quantum particle swarm optimization is described. In Sect. 4, the proposed hybrid QPSO algorithm with Cauchy distribution and natural selection mechanism is presented. The experimental results are plotted for a set of benchmark problems and compared with several QPSO variants in Sect. 5. The correctness and time complexity are analyzed in Sect. 6. QPSO-CD is applied to three constrained engineering design problems in Sect. 7. Finally, Sect. 8 is the conclusion.

2 Prior work

Since the quantum-behaved particle swarm optimization was proposed, various revised variants have been emerged. Initially, Sun et al. [16] applied the concept of quantum computing to PSO and developed a quantum Delta potential well model for classical PSO [24]. It has been shown that the convergence and performance of QPSO are superior as compared to classical PSO. The selection and control of parameters can improve its performance, which is posed as an open problem. Sun et al. [25] tested

the performance of QPSO on constrained and unconstrained problems. It has been claimed that QPSO is a promising optimization algorithm, which performs better than classical PSO algorithms. In 2011, Sun et al. [26] proposed QPSO with Gaussian distribution (GAQPSO) with the local attenuator point and compared its results with several PSO and QPSO counterparts. It has been proved that GAQPSO is efficient and stable with superior features in quality and robustness of solutions.

Further, Coelho [27] applied GQPSO to constrained engineering problems and showed that the simulation results of GQPSO are much closer to the perfect solution with small standard deviation. Li et al. [28] presented a cooperative QPSO using Monte Carlo method (CQPSO), where particles cooperate with each other to enhance the performance of original algorithm. It is implemented on several representative functions and performed better than the other QPSO algorithms in context of computational cost and quality of solutions. Peng et al. introduced [29] QPSO with Levy probability distribution and claimed that there are very less chances to be stuck in local optimum.

Researchers have applied PSO and QPSO to real-life problems and achieved optimal solutions as compared to existing algorithms. Ali et al. [30] performed energy-efficient clustering in mobile ad-hoc networks (MANET) with PSO. The similar approach can be followed to analyze and execute mobility over MANET with QPSO-CD [31]. Zhisheng [32] used QPSO in economic load dispatch for power system and proved superior to other existing PSO optimization algorithms. Sun et al. [33] applied QPSO for QoS multicast routing. Firstly, the QoS multicast routing is converted into constrained integer problems and then effectively solved by QPSO with loop deletion task. Further, the performance is investigated on random network topologies. It has been proved that QPSO is more powerful than PSO and genetic algorithm. Geis and Midendorf [34] proposed PSO with Helix structure for finding ribonucleic acid (RNA) secondary structures with same structure and low energy. The QPSO-CD algorithm can be used with two-way quantum finite automata to model the RNA secondary structures and chemical reactions [35–37]. Bagheri et al. [38] applied the QPSO for tuning the parameters of adaptive network-based fuzzy inference system (ANFIS) for forecasting the financial prices of future market. Davoodi et al. [39] introduced a hybrid improved QPSO with Nelder Mead simplex method (IQPSO-NM), where NM method is used for tuning purpose of solutions. Further, the proposed algorithm is applied to solve load flow problems of power system and acquired the convergence accurately with efficient search ability. Omkar [40] proposed QPSO for multi-objective design problems, and results are compared with PSO. Recently, Fatemeh et al. [41] proposed QPSO with shuffled complex evolution (SP-QPSO) and its performance is demonstrated using five engineering design problems. Prithi and Sumathi [42] integrated the concept of classical PSO with deterministic finite automata for transmission of data and intrusion detection. The proposed algorithm QPSO-CD can be used with quantum computational models for wireless communication [43–49].

3 Quantum particle swarm optimization

Before we explain our hybrid QPSO-CD algorithm mutated with Cauchy operator and natural selection method, it is useful to define the notion of quantum PSO. We

assume that the reader is familiar with the concept of classical PSO; otherwise, reader can refer to particle swarm optimization algorithm [50,51]. The specific principle of quantum PSO is given as:

In QPSO, the state of a particle can be represented using wave function $\psi(x, t)$. The probability density function $|\psi(x, t)|^2$ is used to determine the probability of particle occurring in position x at any time t [16,33]. The position of particles is updated according to equations:

$$x_{i,j}(t + 1) = p_{i,j}(t) \pm \alpha \cdot |mbest_{i,j}(t) - x_{i,j}(t)| \cdot \ln(1/u) \tag{1}$$

$$p_{i,j}(t) = (\phi \cdot P_{i,j}(t) + (1 - \phi) \cdot G_j(t)), (1 \leq i \leq N, 1 \leq j \leq M) \tag{2}$$

where each particle must converge to its local attractor $p = (p_1, p_2, \dots, p_D)$, where D is the dimension, N and M are the number of particles and iterations, respectively, $P_{i,j}$ and G_j denote the previous and optimal position vector of each particle respectively, $\phi = c_1 \cdot r_1 / (c_1 r_1 + c_2 r_2)$, where $c_1; c_2$ are the acceleration coefficients, $r_1; r_2$ and u are normally distributed random numbers in $(0, 1)$, α is contraction-expansion coefficient and $mbest$ defines the mean of best positions of particles as:

$$mbest_{i,j}(t) = \frac{1}{N} \sum_{i=1, j=1}^{N, M} P_{i,j}(t) = \left(\frac{1}{N} \sum_{i=1}^N P_{i,1}(t), \frac{1}{N} \sum_{i=1}^N P_{i,2}(t), \dots, \frac{1}{N} \sum_{i=1}^N P_{i,D}(t) \right) \tag{3}$$

In Eq. (1), α denotes contraction-expansion coefficient, which is setup manually to control the speed of convergence. It can be decreased linearly or fixed. In PSO, $\alpha < 1.782$ to ensure convergence performance of the particle. In QPSO-CD, the value of α is determined by $\alpha = 1 - (1.0 - 0.5)k/M$, i.e., decreases linearly from 1.0 to 0.5 to attain good performance, where k is present iteration and M is maximum number of iterations.

4 Hybrid particle swarm optimization

The hybrid quantum-behaved PSO algorithm with Cauchy distribution and natural selection strategy (QPSO-CD) is described as follows:

The QPSO-CD algorithm begins with the standard QPSO using Eqs. (1), (2) and (3). The position and velocity of particles cannot be determined exactly due to varying dynamic behavior. So, it can only be learned with the probability density function. Each particle can be mutated with Gaussian or Cauchy distribution. We mutated QPSO with Cauchy operator due to its ability to make larger perturbation. Therefore, there is a higher probability with Cauchy as compared to Gaussian distribution to come out of the local optima region. The QPSO algorithm is mutated with Cauchy distribution to increase its diversity, where $mbest$ or global best position is mutated with fixed mutation probability (Pr). The probability density function ($f(x)$) of the standard Cauchy distribution is given as:

$$f(x) = \frac{1}{\pi(1 + x^2)} \quad -\infty < x < \infty, \tag{4}$$

It should be noted that mutation operation is executed on each vector by adding Cauchy distribution random value ($D(.)$) independently such that

$$x' = x + \phi D(.) \tag{5}$$

Algorithm 1 QPSO-CD algorithm

```

1: The swarm is initialized with random numbers distributed uniformly: random  $x_i$ .
2: Do
3:  $\alpha$  decreases linearly from 1.0 to 0.5
4: for  $k=1$  to  $M$  do do
5:    $\alpha = 1 - (1.0 - 0.5).k/M$ 
6:   if  $Pr < rand(0, 1)$  then
7:     Calculate the mbest of the swarm using Eq (3)
8:   end if
9:   for  $i=1$ : to  $N$  do do
10:    if  $Fitness(p_i) < Fitness(x_i)$  then  $p_i = x_i$ ;
11:     $G = argmin(Fitness(p_i))$ ;
12:   end if
13:   for  $j:1$  to  $D$  do do
14:     $r_1 = rand(0, 1); r_2 = rand(0, 1)$ ;
15:     $\phi = c_1.r_1/(c_1r_1 + c_2r_2)$ ;
16:     $p_{i,j} = \phi.P_{i,j} + (1 - \phi).G_j$ ;
17:    if  $rand(0, 1) < 0.5$  then
18:       $x_{i,j} = p_{i,j} + \alpha.abs(mbest_{i,j} - x_{i,j}).log(1/u)$ 
19:    else
20:       $x_{i,j} = p_{i,j} - \alpha.abs(mbest_{i,j} - x_{i,j}).log(1/u)$ 
21:    end if
22:   end for
23:    $Fx(i)=Fitness(x(i, :))$ ;
24: end for
25:  $[SF, Sx] = sort(Fx)$ ;
26:  $Z = round((N - 1)/S)$ ; ▷ S is selection parameter
27:  $x(Sx((N - Z + 1) : N)) = x(Sx(1 : Z))$ ; ▷ Sort the particles from best to worst position
28: end for
29: Until termination criterion is met

```

where x' is new location after mutated with random value to x . At last, the position of particle is selected and the particles of swarm are sorted on the basis of their fitness values after each iteration. Further, substitute the group of particles having worst fitness values with the best ones and optimal solution is determined. The main objective of using natural mechanism is to refine the capability and accuracy of QPSO algorithm.

The natural selection method is used to enhance the convergence characteristics of proposed QPSO-CD algorithm, where the fitter solutions are used for the next iteration. The procedure of selection method for N particles is as follows:

$$F(X(t)) = \{F(x_1(t)), F(x_2(t)), \dots, F(x_N(t))\} \tag{6}$$

where $X(t)$ is position vector of particles at time t and $F(X(t))$ is the fitness function of swarm. Next step is to sort the particles according to their fitness values from best

Table 1 Details of benchmark functions

Test function	Initial range
<i>Sphere function</i>	
$f_1(x) = \sum_{i=1}^n x_i^2$	(- 100, 100)
<i>Rosenbrock function</i>	
$f_2(x) = \sum_{i=1}^n 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	(- 5.12, 5.12)
<i>Greiwank function</i>	
$f_3(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i+1}}) + 1$	(- 600, 600)
<i>Rastrigrin function</i>	
$f_4(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	(- 5.12, 5.12)

one to worst position such that

$$\begin{aligned}
 F(X'(t)) &= \{F(x'_1(t)), F(x'_2(t)), \dots, F(x'_N(t))\} \\
 X'(t) &= \{x'_1(t), x'_2(t), \dots, x'_N(t)\}
 \end{aligned}
 \tag{7}$$

In Algorithm 1, SF and Sx are the sorting functions of fitness and position, respectively. On the basis of natural selection parameters and fitness values, the positions of swarm particles are updated for the next iteration,

$$\begin{aligned}
 X'(t) &= \{x''_1(t), x''_2(t), \dots, x''_S(t)\}, \\
 X''_k(t) &= \{x'_1(t), x'_2(t), \dots, x'_Z(t)\}
 \end{aligned}
 \tag{8}$$

where $(1 \leq k \leq S)$, S denotes the selection parameter, Z signifies the number of best positions selected according to fitness values such that $S = N/Z$ and $X''(t)$ is updated position vector of particles. The selection parameter S is generally set as 2 to replace the half of worst positions with the half of best positions of particles. It improves the precision of the direction of particles, protects the global searching capability and speeds up the convergence.

5 Experimental results

The performance of proposed QPSO-CD algorithm is investigated on representative benchmark functions, given in Table 1. Further, the results are compared with classical PSO (PSO), standard QPSO, QPSO with delta potential (QDPSO) and QPSO with mutation operator (QPSO-MO). The details of numerical benchmark functions are given in Table 1.

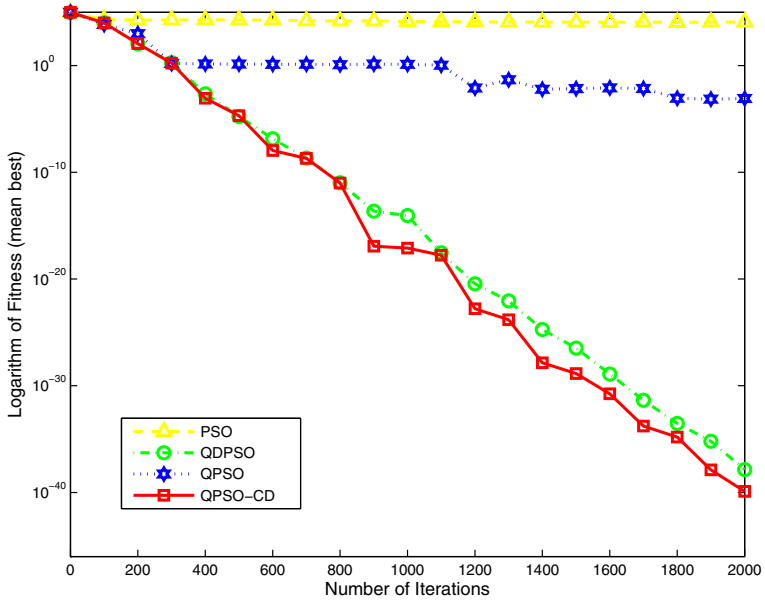


Fig. 2 Effectiveness of QPSO-CD for sphere function f_1

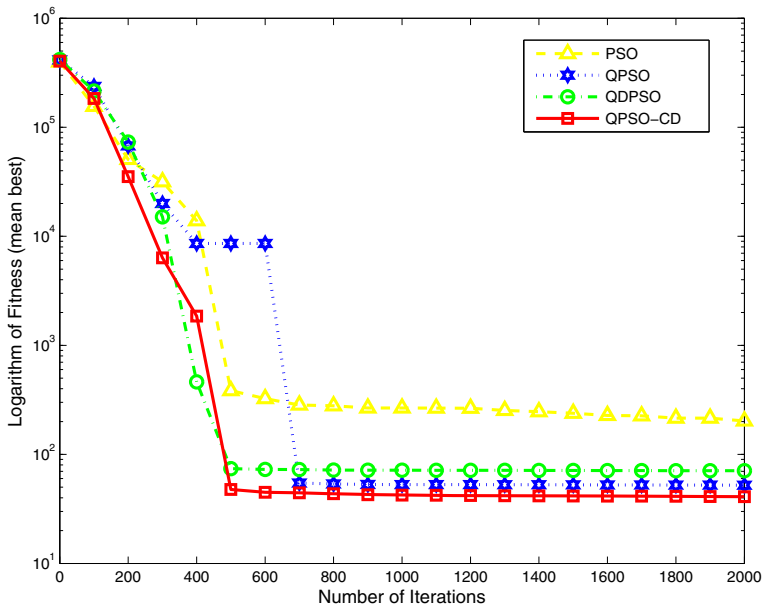


Fig. 3 Effectiveness of QPSO-CD for Rosenbrock function f_2

Table 2 Comparison results of Sphere and Rosenbrock functions

P	D	G	PSO	QPSO	QDPSO	QPSO-MO	QPSO-CD
<i>Sphere function</i>							
20	10	1000	0.0	4.01e-40	1.513e-49	1.508e-48	1.738e-50
	20	1500	0.0	2.58e-21	1.339e-30	1.296e-31	1.032e-30
	30	2000	0.0	2.08e-13	1.953e-21	1.918e-21	1.808e-21
40	10	1000	0.0	2.73e-67	1.087e-73	1.146e-51	1.154e-72
	20	1500	0.0	4.84e-28	1.397e-42	1.417e-42	1.237e-41
	30	2000	0.0	2.02e-25	2.850e-30	2.471e-28	1.946e-23
80	10	1000	0.0	7.66e-95	5.553e-90	4.872e-71	6.437e-72
	20	1500	0.0	1.62e-60	1.654e-54	1.677e-58	1.609e-62
	30	2000	0.0	2.05e-44	1.042e-40	1.131e-42	1.128e-41
<i>Rosenbrock function</i>							
20	10	1000	95.10	58.41	14.22	22.18	34.67
	20	1500	204.38	110.5	175.31	68.40	54.76
	30	2000	314.46	148.5	242.37	113.30	122.5
40	10	1000	70.28	10.42	15.86	7.985	8.843
	20	1500	178.98	48.45	112.46	52.93	41.77
	30	2000	288.58	58.32	76.42	64.19	58.04
80	10	36.29	8.853	36.34	5.715	7.419	
	20	1500	84.78	34.88	23.54	24.45	21.78
	30	2000	202.58	52.17	70.81	45.22	40.97

The performance of QPSO has been widely tested for various test functions. Initially, we have considered four representative benchmark functions to determine the reliability of QPSO-CD algorithm. For all the experiments, the size of population is 20, 40 and 80 and dimension sizes are 10, 20 and 30. The parameters for QPSO-CD algorithm are as follows: the value of α decreases from 1.0 to 0.5 linearly; the natural selection parameter $S = 2$ is taken, c_1, c_2 correlation coefficients are set equal to 2.

The mean best fitness values of PSO, QPSO, QDPSO, QPSO-MO and QPSO-CD are recorded for 1000, 1500 and 2000 runs of each function. Figures 2, 3, 4 and 5 depict the performance of functions f_1 to f_4 with respect to mean best fitness against the number of iterations. In Table 2, P denotes the population, dimension is represented by D and G stands for generation. The numerical results of QPSO-CD showed optimal solution with fast convergence speed and high accuracy. The results showed that QPSO-CD performs better on Rosenbrock function than its counterparts in some cases. When the size of population is 20 and dimension is 30, the results of proposed algorithm are not better than QPSO-MO, but QPSO-CD performs better than PSO, QPSO and QDPSO. The performance of QPSO-CD is significantly better than its variants on Greiwank and Rastrigrin functions. It has outperformed other algorithms and obtained optimal solution (near zero) for Greiwank function. In most of the cases, QPSO-CD is more efficient and outperformed the other algorithms.

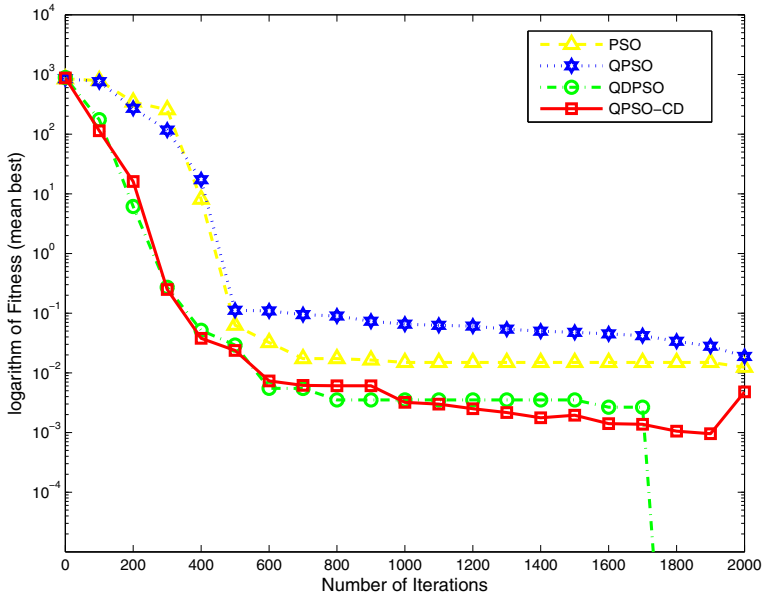


Fig. 4 Effectiveness of QPSO-CD for Greiwank function f_3

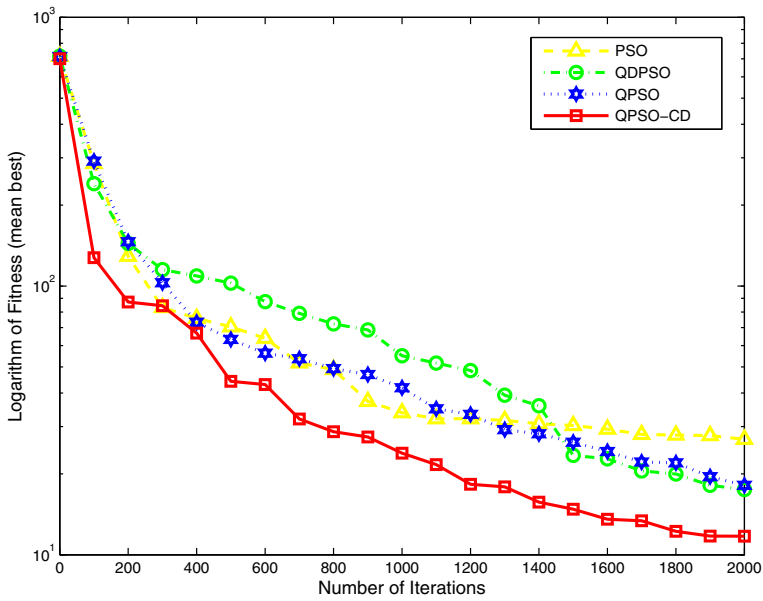


Fig. 5 Effectiveness of QPSO-CD for Rastrigrin function f_4

Table 3 Comparison results of Greiwank and Rastrigrin functions

P	D	G	PSO	QPSO	QDPSO	QPSO-MO	QPSO-CD
<i>Greiwank function</i>							
20	10	1000	0.089	0.078	0.1003	0.0732	0.072
	20	1500	0.0300	0.2001	0.0086	0.0189	0.0078
	30	2000	0.0181	0.0122	0.0544	0.0103	0.0026
40	10	1000	0.0826	0.055	0.048	0.0520	0.041
	20	1500	0.0272	0.0149	0.0004	0.0247	0.0106
	30	2000	0.0125	0.0117	0.0009	0.0105	0.0102
80	10	1000	0.0723	0.0341	0.0	0.0542	0.0702
	20	1500	0.0274	0.0189	0.0	0.0194	0.0161
	30	2000	0.0123	0.0118	0.0	0.0082	0.0031
<i>Rastrigrin function</i>							
20	10	1000	5.526	5.349	4.969	4.478	4.051
	20	1500	23.17	21.28	17.08	15.63	13.22
	30	2000	46.29	32.57	48.61	27.80	31.48
40	10	1000	3.865	3.673	2.032	3.383	2.100
	20	1500	15.68	14.37	10.94	11.01	10.77
	30	2000	37.13	23.01	21.37	21.01	21.19
80	10	1000	2.562	2.234	0.923	2.183	1.943
	20	1500	12.35	9.66	6.955	8.075	7.021
	30	2000	26.89	17.48	18.13	14.99	11.73

6 Correctness and time complexity analysis of a QPSO-CD algorithm

In this Section, the correctness and time complexity of a proposed algorithm QPSO-CD is analyzed and compared with the classical PSO algorithm.

Theorem 1 *The sequence of random variables $\{S_n, n \geq 0\}$ generated by QPSO with Cauchy distribution converges to zero in probability as n approaches infinity.*

Proof Recall, the probability density function of standard Cauchy distribution and its convergence probability [52] are given as

$$\begin{aligned}
 f(s) &= \frac{1}{\pi(1+s^2)} \text{ for } -\infty < s < \infty, \\
 P(x \leq S_n \leq y) &= \frac{1}{\pi} \int_y^x \frac{ds}{(1+s^2)}, \forall x \leq y
 \end{aligned}
 \tag{9}$$

Consider a random variable Q_n interpreted as

$$Q_n = \alpha S_n, \alpha = \frac{1}{n^\lambda}$$

where λ denotes a fixed positive constant. Correspondingly, the probability density function can be calculated as

$$\begin{aligned} P(Q_n \leq q) &= P(\alpha S_n \leq q) \\ &= P\left(S_n \leq \frac{q}{\alpha}\right) \\ &= \int_{-\infty}^{\frac{q}{\alpha}} f(s) ds \cdot \frac{dP(q_n \leq q)}{dq} \\ &= \frac{1}{\alpha} f\left(\frac{q}{\alpha}\right) \end{aligned}$$

i.e., the probability density function of random variable Q_n .

$$\begin{aligned} P(|Q_n| > \xi) &= P(|S_n| > \xi n^\lambda) \\ &= P(S_n > \xi n^\lambda) + P(S_n < -\xi n^\lambda) \\ &= P(\xi n^\lambda < S_n < \infty) + P(-\infty < S_n < -\xi n^\lambda) \end{aligned}$$

Using Eq. (9), the probability density function of random variable Q_n becomes

$$\begin{aligned} P(|Q_n| > \xi) &= \frac{1}{\pi} \int_{\xi n^\lambda}^{\infty} \frac{ds}{\pi(1+s^2)} + \frac{1}{\pi} \int_{-\infty}^{-\xi n^\lambda} \frac{ds}{\pi(1+s^2)} \\ &= \left[1 + \frac{1}{\pi} \int_{\xi n^\lambda}^{-\xi n^\lambda} \frac{ds}{\pi(1+s^2)} \right] \\ &= 1 - \frac{1}{\pi} \int_{-\xi n^\lambda}^{\xi n^\lambda} \frac{ds}{\pi(1+s^2)} = 0 \text{ as } n \rightarrow \infty \end{aligned}$$

This completes the proof of the theorem. □

Definition 1 Let $\{S_n\}$ a random sequence of variables. It converges to some random variable s with probability 1, if for every $\xi > 0$ and $\lambda > 0$, there exists $n_1(\xi, \lambda)$ such that $P(|S_n - s| < \xi) > 1 - \lambda, \forall n > n_1$ or

$$P\left(\lim_{n \rightarrow \infty} |S_n - s| < \xi\right) = 1 \tag{10}$$

The efficiency of the QPSO-CD algorithm is evaluated by number of steps needed to reach the optimal region $R(\xi)$. The method is to evaluate the distribution of number of steps needed to hit $R(\xi)$ by comparing the expected value and moments of distribution. The total number of stages to reach the optimal region is determined as $W(\xi) = \inf\{n \mid f_n \in R(\xi)\}$. The variance $V(W(\xi))$ and expectation value $E(W(\xi))$ are determined as

$$E(W(\xi)) = \sum_{n=0}^{\infty} n x_n, \tag{11}$$

Table 4 Results of the time complexity for QPSO-CD algorithm

Dimension (N)	Mean	Variance	SD	SE	Mean/N
2	302.38	4164.3	64.532	10.203	151.19
3	452.18	4541.9	67.394	10.655	150.72
4	621.29	5208.2	72.168	11.410	155.32
5	755.88	6675.0	81.701	12.918	151.17
6	879.13	8523.7	92.324	14.597	146.52
7	1022.06	9575.4	97.854	15.472	146.00
8	1158.52	10269.7	101.341	16.023	144.81
9	1308.17	12053.4	109.788	17.359	145.35
10	1459.3	12648.3	112.465	17.782	145.93

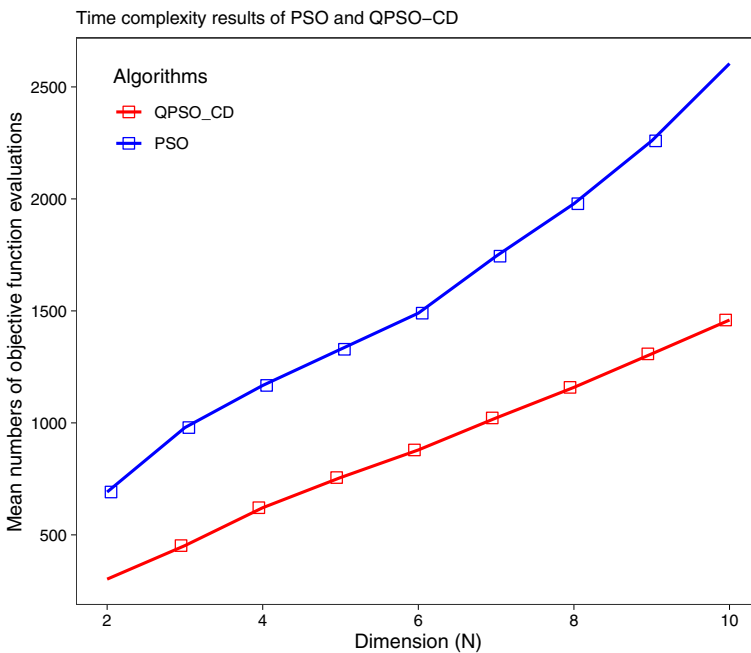
$$\begin{aligned}
 V(W(\xi)) &= E(W^2(\xi)) - \{E(W(\xi))\}^2 \\
 &= \sum_{n=0}^{\infty} n^2 x_n - \left(\sum_{n=0}^{\infty} n x_n \right)^2
 \end{aligned}
 \tag{12}$$

In fact, the $E(W(\xi))$ depends upon the convergence of $\sum_{n=0}^{\infty} n x_n$. It is needed that $\sum_{j=0}^{\infty} x_n = 1$, so that QPSO-CD can converge globally. The number of objective function evaluations are used to measure time. The main benefit of this approach is that it shows relationship between processor and measure time as the complexity of objective function increases. We used Sphere function $f(x) = x^T \cdot x$ with a linear constraint $g(x) = \sum_{j=0}^n x_j \geq 0$ to compute the time complexity. It has minimum value at 0. The value of optimal region is set as $R(\xi) = R(10^{-4})$. To determine the time complexity, the algorithms PSO and QPSO-CD are executed 40 times on $f(x)$ with initial scope $[-10, 10]^N$, where N denotes the dimension. We determine the mean number of objective function evaluations ($W(\xi)$), the variance ($V(W(\xi))$), the standard deviation (SD) ($\sigma_{W(\xi)}$), the standard error (SE) ($\sigma_{W(\xi)}/\sqrt{40}$) and ratio of mean and dimension ($W(\xi)/N$). The contraction coefficient $\alpha = 0.75$ is used for QPSO-CD and constriction coefficient $\chi = 0.73$ for PSO with acceleration factors $c_1 = c_2 = 2.25$.

Tables 1 and 2 show the statistical results of time complexity test for QPSO-CD and PSO algorithm, respectively. Figure 6 indicates that the time complexity of proposed algorithm increases nonlinearly as the dimension increases. However, the time complexity of PSO algorithm increases adequately linearly. Thus, the time complexity of QPSO-CD is lower than PSO algorithm. A Pearson correlation coefficient method is used to show the relationship between the mean and dimension [53]. In Fig. 7, QPSO-CD shows a strong correlation between $W(\xi)$ and N , i.e., the correlation coefficient $R = 0.9996$. For PSO, the linear correlation coefficient $R = 0.9939$, which is not so phenomenal as that in case of QPSO-CD. The relationship between mean and dimension clearly shows that the value of correlation coefficient is fairly stable for QPSO-CD as compared to PSO algorithm.

Table 5 Results of the time complexity for PSO algorithm

Dimension (N)	Mean	Variance	SD	SE	Mean/N
2	691.4	17297.5	131.52	20.795	345.7
3	979.1	22281.5	149.27	23.601	326.3
4	1167.2	24282.9	155.72	24.638	291.8
5	1328.7	21853.7	147.83	23.373	265.7
6	1489.9	32008.7	178.91	28.288	248.3
7	1744.3	502297	224.12	35.436	249.1
8	1978.5	41233.3	203.06	31.106	247.3
9	2259.1	36217.8	190.31	30.090	251.0
10	2604.2	43559.8	208.71	32.999	260.4

**Fig. 6** Time complexity results for PSO and QPSO-CD

7 QPSO-CD for constraint engineering design problems

There exists several approaches for handling constrained optimization problems. The basic principle is to convert the constrained optimization problem to unconstrained by combining objective function and penalty function approach. Further, minimize the newly formed objective function with any unconstrained algorithm. Generally, the constrained optimization problem can be described as in Eq. (13).

The objective is to minimize the objective function $f(x)$ subjected to equality ($h_j(x)$) and inequality ($g_i(x)$) constrained functions, where $p(i)$ is the upper bound and $q(i)$

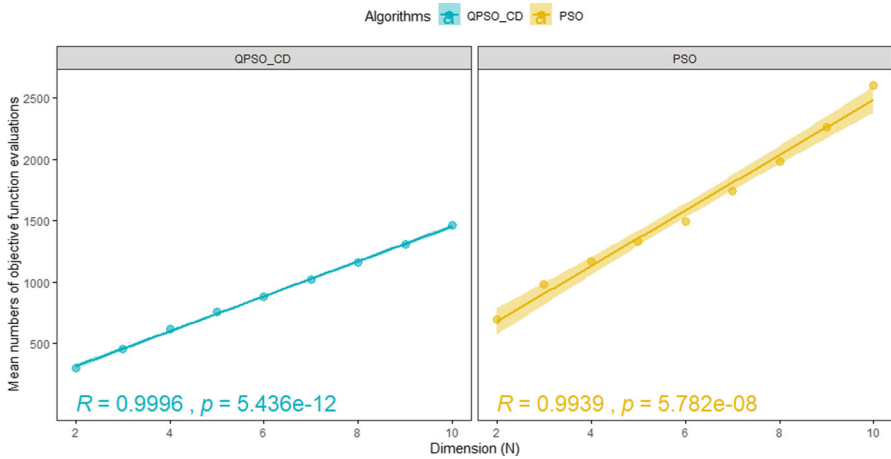


Fig. 7 Comparison of correlation coefficients of PSP and QPSO-CD

denotes the search space lower bound. The strict inequalities of form $g_i(x) \geq 0$ can be converted into $-g_i(x) \leq 0$ and $h_i(x)$ equality constraints can be converted into inequality constraints $h_i(x) \geq 0$ and $h_i(x) \leq 0$. Sun et al. [25] adopted non-stationary penalty function to address nonlinear programming problems using QPSO. Coelho [27] used penalty function with some positive constant, i.e., set to 5000. We adopted the same approach and replace the constant with dynamically allocated penalty value.

$$\begin{aligned}
 & \min_x f(x) \\
 & \text{subject to} \\
 & g_i(x) \leq 0, i = 0, 1, \dots, n - 1 \\
 & h_j(x) = 0, j = 1, 2, \dots, r \\
 & p(i) \leq x_i \leq q(i), 1 \leq i \leq m \\
 & x = \{x_1, x_2, x_3, \dots, x_m\}
 \end{aligned} \tag{13}$$

Usually, the procedure is to find the solution for design variables that lie in search space upper and lower bound constraints such that $x_i \in [p(i), q(i)]$. If solution violates any of the constraint, then the following rules are applied

$$\begin{aligned}
 x_i &= x_i + \{p(x_i) - q(x_i)\} \cdot \text{rand}[0, 1] \\
 x_i &= x_i - \{p(x_i) - q(x_i)\} \cdot \text{rand}[0, 1]
 \end{aligned} \tag{14}$$

where $\text{rand}[0, 1]$ is randomly distributed function to select value between 0 and 1. Finally, the unconstrained optimization problem is solved using dynamically modified penalty values according to inequality constraints $g_i(x)$. Thus, the objective function

is evaluated as

$$F(x) = \begin{cases} f(x) & \text{if } g_i(x) \leq 0 \\ f(x) + y(t) \cdot \sum_{i=1}^n g_i(x) & \text{if } g_i(x) > 0 \end{cases} \quad (15)$$

where $f(x)$ is the main objective function of optimization problem in Eq. (13), t is the iteration number and $y(t)$ represents the dynamically allocated penalty value.

In this Section, QPSO-CD is tested for three-bar truss, tension/compression spring and pressure vessel design problems consisting different members and constraints. The performance of QPSO-CD is compared and analyzed with the results of PSO, QPSO, and SP-QPSO algorithms as reported in the literature.

7.1 Three-bar truss design problem

Three-bar truss is a constraint design optimization problem, which has been widely used to test several methods. It consists cross section areas of three bars x_1 (and x_3) and x_2 as design variables. The aim of this problem is to minimize the weight of truss subject to maximize the stress on these bars. The structure should be symmetric and subjected to two constant loadings $P_1 = P_2 = P$ as shown in Fig. 8. The mathematical formulation of two design bars (x_1, x_2) and three restrictive mathematical functions are described as:

$$\begin{aligned} \text{Min } f(x) &= (2\sqrt{2}x_1 + x_2) \cdot l \\ \text{Subject to:} \\ g_1(x) &= \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0, \\ g_2(x) &= \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0 \\ g_3(x) &= \frac{1}{x_1 + \sqrt{2}x_2} P - \sigma \leq 0, \text{ where} \\ 0 \leq x_1, x_2 \leq 1, l &= 100\text{cm}, P = \sigma = 2\text{KN/cm}^2 \end{aligned} \quad (16)$$

The results are obtained by QPSO-CD are compared with its counterparts in Table 6. For three-bar truss problem, QPSO-CD is superior to optimal solutions previously obtained in literature. The difference of best solution obtained by QPSO-CD among other algorithms is shown in Fig. 9.

7.2 Tension/compression spring design problem

The main aim is to lessen the volume V of a spring subjected to tension load constantly as shown in Fig. 10. Using the symmetry of structure, there are practically three design

Fig. 8 Structure of three-bar truss

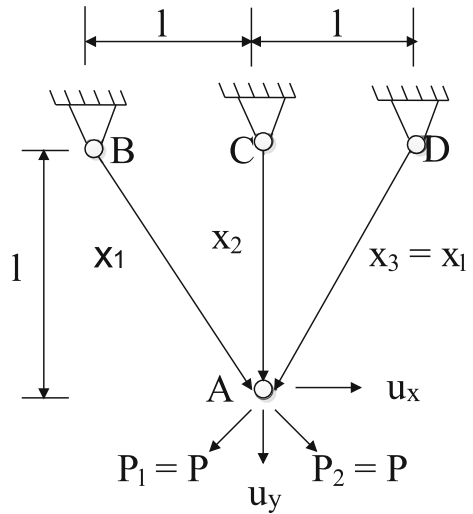


Table 6 Comparison of optimal results for three-bar truss problem

Variables	PSO	QPSO	SP-QPSO	QPSO-CD
x_1	0.78911058	0.788649	0.788796	0.788658
x_2	0.40702683	0.408322	0.407898	0.40828488
$g_1(x)$	- 6.6720e-06	1.6313e-07	6.4748e-06	9.00037e-06
$g_2(x)$	- 1.4655	- 1.4640	- 1.4644	- 1.4640
$g_3(x)$	- 0.5345	- 0.5359	- 0.5354	- 0.5359
$f(x)$	263.89686	263.89584	263.89500	263.89465

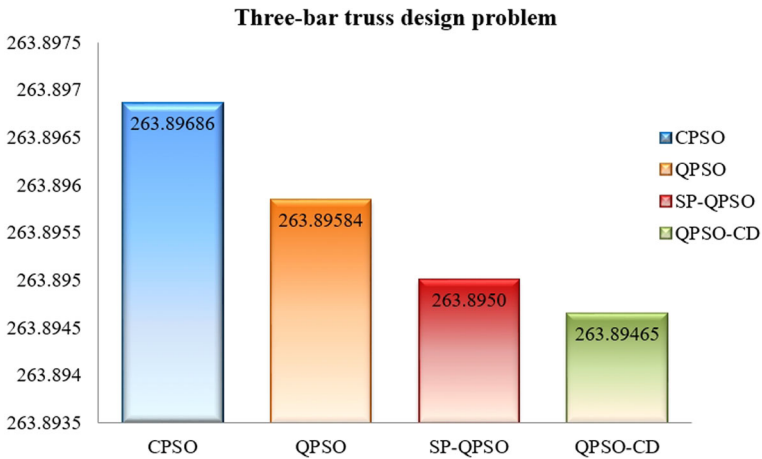


Fig. 9 Optimal results of PSO, QPSO, SP-QPSO and QPSO-CD algorithms for three-bar truss problem

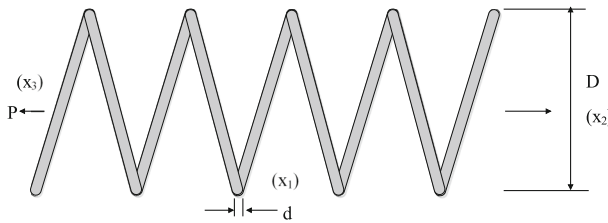


Fig. 10 Structure of tension/compression spring

Table 7 Comparison of optimal results for tension spring design problem

Variables	PSO	QPSO	SP-QPSO	QPSO-CD
x_1	0.0516	0.0524	0.05	0.0513
x_2	0.3542	0.2505	0.25	0.2502
x_3	11.7942	2	2	2
$g_1(x)$	- 2.3006e-02	0.93145095	0.93034756	4.11004e-06
$g_2(x)$	- 5.6059e-03	- 0.17471558	- 0.16568318	- 0.17352479
$g_3(x)$	- 3.9057	- 50.67	- 48.180	- 49.561
$g_4(x)$	- 0.7294	- 0.79986567	- 0.80	- 0.799
$f(x)$	0.01305	0.00275	0.00250	0.00263

variables (x_1, x_2, x_3) , where x_1 is the wire diameter, the coil diameter is represented by x_2 and x_3 denotes the total number of active coils. The mathematical formulation for this problem is described as:

$$\text{Min } f(x) = (x_3 + 2)x_2x_1^2,$$

Subject to:

$$g_1(x) = \frac{1 - x_2^3x_3}{71785x_164} \leq 0,$$

$$g_2(x) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0,$$

$$g_3(x) = \frac{1 - 140.45x_1}{x_2^2x_3} \leq 0,$$

$$g_4(x) = \frac{x_2 + x_1}{1.5} - 1 \leq 0,$$

where

$$0.05 \leq x_1 \leq 2, 0.25 \leq x_2, \leq 1.3, 2 \leq x_3 \leq 15, \tag{17}$$

It has been observed that QPSO algorithm with Cauchy distribution and natural selection strategy is robust and obtains optimal solutions than PSO and QPSO, shown in Table 7. The difference between best solutions found by QPSO-CD ($f(x) = 0.00263$) and other algorithms for tension spring design problem are reported in Fig. 11.

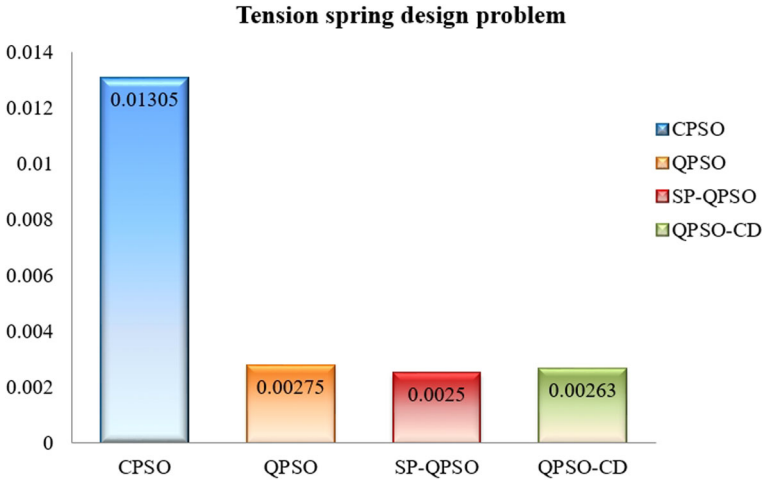


Fig. 11 Results of PSO, QPSO, SP-QPSO and QPSO-CD methods for tension spring design problem

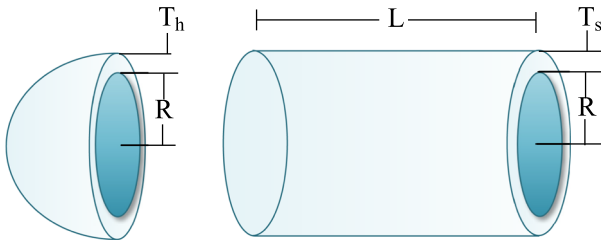


Fig. 12 Design of pressure vessel

7.3 Pressure vessel design problem

Initially, Kannan and Kramer [54] studied the pressure vessel design problem with the main aim to reduce the total fabricating cost. Pressure vessels can be of any shape. For engineering purposes, a cylindrical design capped by hemispherical heads at both ends is widely used [55]. Figure 12 describes the structure of pressure vessel design problem. It consists four design variables (x_1, x_2, x_3, x_4), where x_1 denotes the shell thickness (T_s), x_2 is used for head thickness (T_h), x_3 denotes the inner radius (R) and x_4 represents the length of vessel (L). The objective function and constraint equations are described as:

$$\text{Min } f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.166x_1^2x_4 + 19.84x_1^2x_3,$$

Subject to:

$$g_1(x) = -x_1 + 0.0193x_3 \leq 0,$$

$$g_2(x) = -x_2 + 0.00954x_3 \leq 0,$$

Table 8 Comparison of optimal results for Pressure vessel design problem

Variables	PSO	QPSO	SP-QPSO	QPSO-CD
x_1	0.8125	0.7783	0.7782	0.7776
x_2	0.4375	0.3849	0.3845	0.3848
x_3	42.0984	40.3289	40.3206	40.3278
x_4	176.6365	199.8899	199.9988	199.8865
$g_1(x)$	- 4.500e-15	4.777e-05	- 1.242e-05	7.2654e-04
$g_2(x)$	- 0.035880	- 1.62294e-04	1.58523e-04	- 7.2787e-05
$g_3(x)$	- 1.164e-10	- 97.39720071	- 63.63686942	- 0.734359
$g_4(x)$	- 63.3634	- 40.1100	- 40.0012	- 40.1135
$f(x)$	6059.714	5886.189	5885.268	5886.137

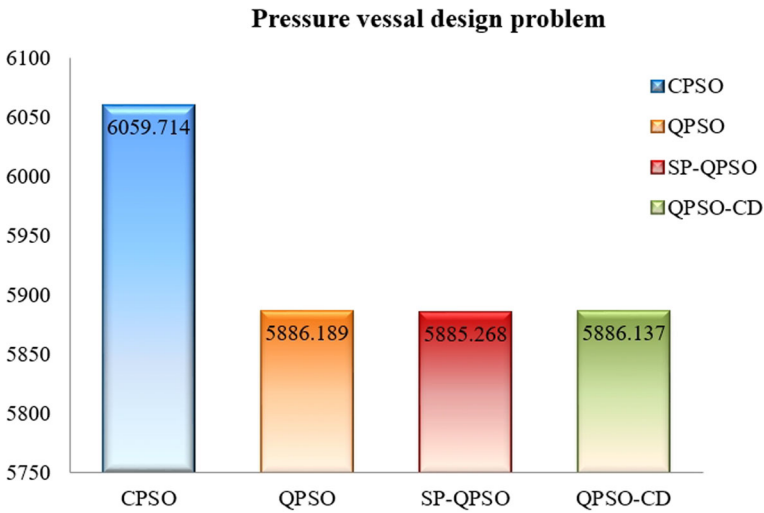


Fig. 13 Optimal results of PSO, QPSO, SP-QPSO and QPSO-CD techniques for pressure vessel design problem

$$g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3} \pi x_3^3 + 1296000 \leq 0,$$

$$g_4(x) = x_4 - 240 \leq 0,$$

where

$$1 \times 0.0625 \leq x_1, x_2 \leq 99 \times 0.0625, 10 \leq x_3, x_4 \leq 200 \tag{18}$$

The optimal results of QPSO-CD are compared with the SP-QPSO, QPSO and PSO best results noted in the previous work and are given in Table 8. The best solution obtained from QPSO-CD is better than other algorithms as shown in Fig. 13.

8 Conclusion

In this paper, a new hybrid quantum particle swarm optimization algorithm is proposed with natural selection method and Cauchy distribution. The performance of the proposed algorithm is experimented on four benchmark functions, and the optimal results are compared with existing algorithms. Further, the QPSO-CD is applied to solve engineering design problems. The efficiency of QPSO-CD is successfully presented with superiority than preceding results for three engineering design problems: three-bar truss, tension/compression spring and pressure vessel. The efficiency of QPSO-CD algorithm is evaluated by number of steps needed to reach the optimal region, and it is proved that time complexity of proposed algorithm is lower in comparison to classical PSO. In the context of convergence, the experimental outcomes show that the QPSO-CD converge to get results closer to the superior solution.

Acknowledgements S.Z. acknowledges support in part from the National Natural Science Foundation of China (No. 61602532), the Natural Science Foundation of Guangdong Province of China (No. 2017A030313378), and the Science and Technology Program of Guangzhou City of China (No. 201707010194).

Compliance with ethical standards

Conflict of interest The authors declare no conflict of interest.

References

1. Bohr, N., et al.: The Quantum Postulate and the Recent Development of Atomic Theory, vol. 3. R. & R. Clarke, Limited, Brighton (1928)
2. Robertson, H.P.: The uncertainty principle. *Phys. Rev.* **34**(1), 163 (1929)
3. Wessels, L.: Schrödinger's route to wave mechanics. *Stud. Hist. Philos. Sci. Part A* **10**(4), 311–340 (1979)
4. Feynman, R.P.: Simulating physics with computers. *Int. J. Theor. Phys.* **21**(6–7), 467–488 (1982)
5. Wang, J.: Handbook of Finite State Based Models and Applications. CRC Press, Boca Raton (2012)
6. Dorigo, M., Di Caro, G.: Ant colony optimization: a new meta-heuristic. In: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), vol. 2. IEEE, pp. 1470–1477 (1999)
7. Colomi, A., Dorigo, M., Maniezzo, V., et al.: Distributed optimization by ant colonies. In: Proceedings of the first European conference on artificial life, vol. 142, Cambridge, MA, pp. 134–142 (1992)
8. Kennedy, J., Eberhart, R.: Particle swarm optimization (PSO). In: Proceedings of IEEE international conference on neural networks, Perth, Australia, pp. 1942–1948 (1995)
9. Wang, S.-C., Yeh, M.-F.: A modified particle swarm optimization for aggregate production planning. *Expert Syst. Appl.* **41**(6), 3069–3077 (2014)
10. AlRashidi, M.R., El-Hawary, M.E.: A survey of particle swarm optimization applications in electric power systems. *IEEE Trans. Evol. Comput.* **13**(4), 913–918 (2009)
11. Yıldız, A.R.: A novel particle swarm optimization approach for product design and manufacturing. *Int. J. Adv. Manuf. Technol.* **40**(5–6), 617 (2009)
12. Latiff, N.A., Tsimenidis, C.C., Sharif, B.S.: Energy-aware clustering for wireless sensor networks using particle swarm optimization. In: IEEE 18th international symposium on personal. Indoor and mobile radio communications. IEEE, pp. 1–5 (2007)
13. Lin, S.-W., Ying, K.-C., Chen, S.-C., Lee, Z.-J.: Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Syst. Appl.* **35**(4), 1817–1824 (2008)
14. Wang, Y., Mohanty, S.D.: Particle swarm optimization and gravitational wave data analysis: performance on a binary inspiral testbed. *Phys. Rev. D* **81**(6), 063002 (2010)

15. Normandin, M.E., Mohanty, S.D., Weerathunga, T.S.: Particle swarm optimization based search for gravitational waves from compact binary coalescences: performance improvements. *Phys. Rev. D* **98**(4), 044029 (2018)
16. Sun, J., Feng, B., Xu, W.: Particle swarm optimization with particles having quantum behavior. In: Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753), vol. 1. IEEE, pp. 325–331 (2004)
17. Narayanan, A., Moore, M.: Quantum-inspired genetic algorithms. In: Proceedings of IEEE international conference on evolutionary computation. IEEE, pp. 61–66 (1996)
18. Yuanyuan, M., Xiyu, L.: Quantum inspired evolutionary algorithm for community detection in complex networks. *Phys. Lett. A* **382**(34), 2305–2312 (2018)
19. Liu, J., Xu, W., Sun, J.: Quantum-behaved particle swarm optimization with mutation operator. In: 17th IEEE international conference on tools with artificial intelligence (ICTAI'05). IEEE, pp. 4–pp (2005)
20. Protopopescu, V., Barhen, J.: Solving a class of continuous global optimization problems using quantum algorithms. *Phys. Lett. A* **296**(1), 9–14 (2002)
21. Bhatia, A.S., Saggi, M.K., Kumar, A., Jain, S.: Matrix product state-based quantum classifier. *Neural Comput.* **31**(7), 1499–1517 (2019)
22. Bhatia, A.S., Saggi, M.K.: Implementing entangled states on a quantum computer. [arXiv:1811.09833](https://arxiv.org/abs/1811.09833)
23. Bhatia, A.S., Kumar, A.: Quantifying matrix product state. *Quantum Inf. Process.* **17**(3), 41 (2018)
24. Sun, J., Xu, W., Feng, B.: A global search strategy of quantum-behaved particle swarm optimization. In: IEEE conference on cybernetics and intelligent systems, 2004, vol. 1. IEEE, pp. 111–116 (2004)
25. Sun, J., Liu, J., Xu, W.: Using quantum-behaved particle swarm optimization algorithm to solve non-linear programming problems. *Int. J. Comput. Math.* **84**(2), 261–272 (2007)
26. Sun, J., Fang, W., Palade, V., Wu, X., Xu, W.: Quantum-behaved particle swarm optimization with Gaussian distributed local attractor point. *Appl. Math. Comput.* **218**(7), 3763–3775 (2011)
27. dos Santos Coelho, L.: Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. *Expert Syst. Appl.* **37**(2), 1676–1683 (2010)
28. Li, Y., Xiang, R., Jiao, L., Liu, R.: An improved cooperative quantum-behaved particle swarm optimization. *Soft Comput.* **16**(6), 1061–1069 (2012)
29. Peng, Y., Xiang, Y., Zhong, Y.: Quantum-behaved particle swarm optimization algorithm with lévy mutated global best position. In: 2013 fourth international conference on intelligent control and information processing (ICICIP). IEEE, pp. 529–534 (2013)
30. Ali, H., Shahzad, W., Khan, F.A.: Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization. *Appl. Soft Comput.* **12**(7), 1913–1928 (2012)
31. Bhatia, A.S., Cheema, R.K.: Analysing and implementing the mobility over manets using random way point model. *Int. J. Comput. Appl.* **68**(17), 32–36 (2013)
32. Zhisheng, Z.: Quantum-behaved particle swarm optimization algorithm for economic load dispatch of power system. *Expert Syst. Appl.* **37**(2), 1800–1803 (2010)
33. Sun, J., Liu, J., Xu, W.: QPSO-based QoS multicast routing algorithm. In: Asia-pacific conference on simulated evolution and learning. Springer, pp. 261–268 (2006)
34. Geis, M., Middendorf, M.: Particle swarm optimization for finding RNA secondary structures. *Int. J. Intell. Comput. Cybern* **4**(2), 160–186 (2011)
35. Bhatia, A.S., Kumar, A.: Modeling of RNA secondary structures using two-way quantum finite automata. *Chaos Solitons Fractals* **116**, 332–339 (2018)
36. Bhatia, A.S., Zheng, S.: A quantum finite automata approach to modeling the chemical reactions. [arXiv:2007.03976](https://arxiv.org/abs/2007.03976)
37. Bhatia, A.S., Zheng, S.: RNA-2QCFA: evolving two-way quantum finite automata with classical states for RNA secondary structures. [arXiv:2007.06273](https://arxiv.org/abs/2007.06273)
38. Bagheri, A., Peyhani, H.M., Akbari, M.: Financial forecasting using ANFIS networks with quantum-behaved particle swarm optimization. *Expert Syst. Appl.* **41**(14), 6235–6250 (2014)
39. Davoodi, E., Hagh, M.T., Zadeh, S.G.: A hybrid improved quantum-behaved particle swarm optimization-simplex method (IQPSOS) to solve power system load flow problems. *Appl. Soft Comput.* **21**, 171–179 (2014)
40. Omkar, S., Khandelwal, R., Ananth, T., Naik, G.N., Gopalakrishnan, S.: Quantum behaved particle swarm optimization (QPSO) for multi-objective design optimization of composite structures. *Expert Syst. Appl.* **36**(8), 11312–11322 (2009)

41. Fatemeh, D., Loo, C., Kanagaraj, G.: Shuffled complex evolution based quantum particle swarm optimization algorithm for mechanical design optimization problems. *J. Mod. Manuf. Syst. Technol.* **2**(1), 23–32 (2019)
42. Prithi, S., Sumathi, S.: LD2FA-PSO: a novel learning dynamic deterministic finite automata with pso algorithm for secured energy efficient routing in wireless sensor network. *Ad Hoc Netw.* **97**, 102024 (2020)
43. Ambainis, A., Freivalds, R.: 1-way quantum finite automata: strengths, weaknesses and generalizations. In: Proceedings 39th annual symposium on foundations of computer science (Cat. No. 98CB36280). IEEE, pp. 332–341 (1998)
44. Bhatia, A.S., Kumar, A.: Quantum finite automata: survey, status and research directions. [arXiv:1901.07992](https://arxiv.org/abs/1901.07992)
45. Bhatia, A.S., Kumar, A.: On the power of two-way multihead quantum finite automata. *RAIRO Theor. Inform. Appl.* **53**(1–2), 19–35 (2019)
46. Qiu, D., Yu, S.: Hierarchy and equivalence of multi-letter quantum finite automata. *Theor. Comput. Sci.* **410**(30–32), 3006–3017 (2009)
47. Li, L., Qiu, D.: Determination of equivalence between quantum sequential machines. *Theor. Comput. Sci.* **358**(1), 65–74 (2006)
48. Qiu, D., Li, L., Zou, X., Mateus, P., Gruska, J.: Multi-letter quantum finite automata: decidability of the equivalence and minimization of states. *Acta Inform.* **48**(5–6), 271 (2011)
49. Singh Bhatia, A.: On some aspects of quantum computational models. Ph.D. thesis, Thapar Institute of Engineering & Technology, Patiala, India (2020)
50. Kennedy, J.: Particle swarm optimization. *Encyclopedia of machine learning*, pp. 760–766 (2010)
51. Shi, Y., et al.: Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546), vol. 1. IEEE, pp. 81–86 (2001)
52. Rudolph, G.: Local convergence rates of simple evolutionary algorithms with cauchy mutations. *IEEE Trans. Evol. Comput.* **1**(4), 249–258 (1997)
53. Benesty, J., Chen, J., Huang, Y., Cohen, I.: Pearson correlation coefficient. In: Noise reduction in speech processing. Springer, pp. 1–4 (2009)
54. Kannan, B., Kramer, S.N.: An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *J. Mech. Des.* **116**(2), 405–411 (1994)
55. Sandgren, E.: Nonlinear integer and discrete programming in mechanical design optimization. *J. Mech. Des.* **112**(2), 223–229 (1990)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.