Information Sciences 267 (2014) 83–100

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A quantum inspired gravitational search algorithm for numerical function optimization

Mohadeseh Soleimanpour-moghadam, Hossein Nezamabadi-pour *, Malihe M. Farsangi

Department of Electrical Engineering, Shahid Bahonar University of Kerman, P.O. Box 76169-133, Kerman, Iran

ARTICLE INFO

Article history:
Received 31 March 2012
Received in revised form 22 July 2013
Accepted 1 September 2013
Available online 8 September 2013

Keywords:
Swarm intelligence
Gravitational search algorithm
Quantum computing
Numerical function optimization

ABSTRACT

Gravitational search algorithm (GSA) is a swarm intelligence optimization algorithm that shares many similarities with evolutionary computation techniques. However, the GSA is driven by the simulation of a collection of masses which interact with each other based on the Newtonian gravity and laws of motion. Inspired by the classical GSA and quantum mechanics theories, this work presents a novel GSA using quantum mechanics theories to generate a quantum-inspired gravitational search algorithm (QIGSA). The application of quantum mechanics theories in the proposed QIGSA provides a powerful strategy to diversify the algorithm's population and improve its performance in preventing premature convergence to local optima. The simulation results and comparison with nine state-of-the-art algorithms confirm the effectiveness of the QIGSA in solving various benchmark optimization functions.

1. Introduction

Quantum mechanics, also known as quantum physics, is a branch of physics which provides a mathematical description of wave-like behavior and interaction of material and energy. Quantum mechanics differs considerably from classical mechanics in its domain when the scale of observations becomes comparable to the atomic and sub-atomic scale, called quantum realm.

Based on the concepts and principles of quantum mechanics, in the early 1980s, quantum computing is proposed by Feynman [6,7]. Afterward, because of its powerful computational performance, there has been a great interest in the application of the quantum computing [40,29]. Quantum computation is based on some principles of quantum theory, such as the superposition of quantum states, collapsing into one state, entanglement; the application of various properties of quantum physics toward building a new kind of computers. Quantum computers have been progressed actively, in order to solve some specialized issues that classical computers have some adversity by facing with them [29,8]. It is declared that quantum computing has the ability of solving many difficult problems in the field of classical computation.

It is worth noticing that if there is no quantum algorithm to solve practical problems, the quantum computer hardware becomes useless. Quantum algorithms exploit the laws of quantum mechanics in order to perform efficient computation within a less time compared to classical algorithms. Nevertheless, such efficiency is granted when the algorithm is run on a quantum computer, whereas the simulation on a classical computer can be very time-consuming. By now, only a few quantum algorithms are known, however, it has been shown that quantum computation can greatly improve performance for solving such problems like factoring [5] or searching in an unstructured database [11].

* Corresponding author. Tel./fax: +98 341 3235900.
E-mail address: nezam@mail.uk.ac.ir (H. Nezamabadi-pour).

0020-0255/$ - see front matter
http://dx.doi.org/10.1016/j.ins.2013.09.006

Nowadays, heuristic search algorithms have been widely employed to solve global optimization problems. Heuristic search algorithms are stochastic algorithms that mimic the processes of natural phenomena such as natural selection, natural evolution, and self-organization. These algorithms maintain a collection of potential solutions for a problem. Some of these possible solutions are used to create new potential solutions through the use of specific operators. The operators act on population and produce collections of new potential solutions at each iteration. This process is used repeatedly to generate new collections of potential solutions till the stopping criterion is met [32]. It is shown that the population-based heuristic search algorithms are effective and flexible tools to solve the problems in a wide range of application [23,25,19,15,36].

Since the late 1990s, many various heuristic approaches have been adopted by researches, such as genetic algorithm (GA) [52], simulated annealing (SA) [24], ant colony optimization (ACO) [4], particle swarm optimization (PSO) [18], artificial bee colony algorithm [1], and GSA [32]. One of the newest algorithms is GSA; it is a global search strategy that can handle efficiently arbitrary optimization problems. Rashedi et al. introduced the GSA in 2009; it is based on the Newtonian laws of gravity and motion [17]. The basic idea of the GSA is to mimic the physical attraction between masses. Many researchers have used the GSA to solve various problems. The obtained results confirm the high performance of GSA in solving various problems [32].

In standard GSA, an object/agent is depicted by its mass and position vector which determines the trajectory of the objects. The object moves along a determined trajectory while in quantum, the term trajectory is meaningless, since the position of a particle cannot be determined due to the uncertainty principle [16]. The above considerations indicate that the merging between the two novel computing paradigms, namely heuristic search algorithms and quantum mechanics, can be beneficial. In recent years, researchers have focused on quantum inspired heuristic search algorithms [54,55].

Since introducing the quantum computing, many quantum-based algorithms are proposed which are classified into two groups: (i) one concentrates on generating new quantum algorithms for using of quantum computer and (ii) other one concentrates on the quantum-inspired heuristic search algorithms.

The first attempts on the first group have been started since the early 1990s. Several significant quantum algorithms including the quantum search algorithm [11], quantum factorization algorithm [39] and quantum genetic programming [44,30] have been proposed to show that quantum computers are more powerful than classical computers at least with respect to solving some specific problems [45].

Some researchers in this approach have tried to propose the new heuristic search algorithms for quantum computers, which is called heuristic search designed quantum algorithms (HDQs) [43,20,9,10,34]. HDQs are able to discover new algorithms for quantum computers [42]. In order to check the capability of HDQs, a quantum computer is simulated, so that the ability of a quantum algorithm can be determined on classical hardware. Nowadays, because of inaccessibility of quantum computers, quantum algorithm is not useful. In this perspective, having a quantum version of heuristic search algorithms seems to be a related topic in the future, when quantum computers will be available.

The other researches in the second group concentrate on quantum-inspired heuristic search algorithms (QIHSAs) for a classical computer; a branch of study on heuristic search algorithm that is characterized by certain principles of quantum mechanics or computation such as standing waves, entanglement, and collapse.

Two kinds of algorithms have been identified:

(i) *Quantum computing-inspired heuristic search algorithms (QCHs)*: QCHs concentrate on generating new heuristic search algorithms using some concepts and principles of quantum computing such as standing waves [26], interference [56], coherence [31], qubits, superposition, quantum gates and quantum measurement, in order to solve various problems in the context of a classical computing model [26]. Like a quantum mechanical system, a quantum-inspired system can be regarded as a probabilistic system, in which the probabilities related to each state are utilized to describe the behavior of the system.

QCHs are firstly introduced by Narayanan and Moore in the 1990s to solve the traveling salesman problem [27]; in their proposed algorithm the crossover operation uses the concept of interference. The contribution of Narayanan and Moore [27] indicate the potential advantage of introducing quantum computational parallelism into the population-based algorithm framework. No further attention is paid to QCHs until an algorithm is proposed by Han and Kim [13,14]. Their proposed algorithm uses a Q-bit representation instead of binary or numeric representations. On the other hand, this algorithm tries to simulate parallel computation in classical computers.

(ii) *Heuristic search algorithms based on quantum mechanics principles (HQMs)*: These algorithms concentrate on using quantum mechanics in heuristic search algorithms. In this group of experience every particle is assumed in quantum space. The individual particle of quantum system moves in a quantum multi-dimensional space. The state of a particle is depicted by wave function $\Psi(x, t)$, instead of position and velocity.

Like a quantum mechanical system, HQMs can be considered as a probabilistic system, in which the probabilities related to each state are utilized to describe the behavior of the system. More specifically, wave functions are applied to represent agent individuals. Agents are moved in quantum search space to find the optimum. In order to generate new population, the agent and fitness evaluation are linked by a probabilistic observation process.

The first version of HQMs named QPSO is introduced by Sun et al. in 2004 [47], in which the particles of a PSO system are assumed in quantum space based on the concept of quantum mechanics. In [47] the authors proposed an approach based on classical PSO that each particle assumed in quantum search space with a potential well (with center of P).

The authors in [48] introduced a global point called mainstream thought or mean best position of the population into QPSO. The global point is defined as the mean of the *pbest* positions of all particles in order to increase the global search ability. In [53] the global point is defined as the weighted average of the *pbest* positions of all particles in order to increase the probabilistic ability of the algorithm. In QPSO a parameter β is introduced which is called contraction–expansion coefficient; β could be tuned to control the convergence speed of the algorithm [51].

It can be seen from the above explanation that contraction–expansion coefficient affects directly on the convergence behavior of the individual particle, and therefore determines the convergence rate of the QPSO algorithm. Inspired by stochastic information in [49] some adaptive parameter control methods have been proposed to twist the exploration and exploitation abilities simultaneously [50].

In [41], Soleimanpour-moghadam et al. introduced a quantum behaved gravitation search algorithm (QGSA) inspired by QPSO. In QGSA each object moves in a Delta potential well. In this approach the center of the potential well is the weighted average of all members of *Kbest* set. The experimental results indicate that the QGSA works better than standard GSA on unimodal benchmark functions. In the QGSA, the fast flow of information between masses seems to be the reason for the premature convergence of the algorithm. In this algorithm, the diversity of the swarm is decreased rapidly, thus the QGSA algorithm loses its swarm diversity.

This paper aims to provide a new version of quantum-inspired GSA (QIGSA) for both unimodal and multi-modal optimization. By increasing the exploration ability of the algorithm at the beginning iterations, the QIGSA could overcome the problem of premature convergence. QIGSA is introduced with the aim of improving the performance of the standard GSA (SGSA) for continuous (real-valued) optimization problems. The proposed method, keeps the concepts of standard GSA while each object has a quantum behavior. In other words, instead of Newtonian random walk, some concepts of “quantum motion” are imposed in the search process. The proposed QIGSA has only one control parameter that is tuned easily by trial and error.

To present the ability of the proposed method, it is tested on several benchmark functions including unimodal and multimodal functions. The results obtained are compared with those available in the literature such as classical GSA, Evolution Strategy (ES) [5], Parameter-less Evolution Strategy (PLES) [5], Population Based Optimization (PBO) [38], A Restart CMA Evolution Strategy (RCMAES) [2], Dynamic Multi-Swarm Particle Swarm Optimizer (DMPSO) [22], Advanced Local Search Evolutionary Algorithm (ALSEA) [3], Real Parameter Optimization (RPO) [33] and quantum PSO (QPSO) [46].

This paper is organized as follows. In Section 2, the standard GSA is described. The proposed QIGSA is introduced in Section 3. The experimentation and the results obtained are given in Section 4. Finally, the paper is concluded in Section 5.

2. Overview of GSA

The classical/standard gravitational search algorithm (GSA) is introduced by Rashedi et al. in 2009 which is inspired by the Newtonian laws of gravity and motion [32]. This swarm optimization technique provides an iterative method that simulates object interactions, and moves through a multi-dimensional search space under the influence of gravitation. The effectiveness of the GSA in solving a set of nonlinear benchmark functions has been proven in [32]. Based on [32], in the GSA, the masses of the objects are calculated by computing the current population fitness, as follows:

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{1}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \tag{2}$$

where N is the size of swarm (population), $fit_i(t)$ represent the fitness value of the object i at time t, and, $worst(t)$ and $best(t)$ are defined as below (for a minimization problem):

$$best(t) = \min\{fit_j(t)\}, \quad j = 1, 2, \ldots, N \tag{3}$$

$$worst(t) = \max\{fit_j(t)\}, \quad j = 1, 2, \ldots, N \tag{4}$$

To compute the acceleration of an object, $a_i^d(t)$, by using the law of motion is calculated as:

$$a_i^d = G(t) \sum_{j \in Kbest, j \neq i} rand_j \frac{M_j(t)}{R_{ij}(t) + \varepsilon} (x_j^d(t) - x_i^d(t)) \quad d = 1, 2, \ldots, n \quad \text{and} \quad i = 1, 2, \ldots, N \tag{5}$$

where M_i is the mass of ith object, x_i^d is the position of ith object in the dimension d, $G(t)$ is the gravitational constant at time t, ε is a small constant, $R_{ij}(t)$ is the Euclidian distance between two objects i and j and $rand_j$ is a uniformly distributed random number in the interval $[0,1]$. G is a decreasing function of time, which is decreased exponentially to zero by lapse of time. *Kbest* is the set of the first K agents with the best fitness value and biggest mass, which is a function of time, it is initialized to K_0 at the beginning and decreased linearly with time.

Afterward, the next velocity of an agent is calculated as a fraction of its current velocity added to acceleration.

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \tag{6}$$

Eventually, the agent's position, $X_i = (x_i^1, x_i^2, \ldots, x_i^n)$, is updated according to Eq. (7).

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{7}$$

It should be noticed that the positions of agents are the candidate solutions for the problem at hand. This process is repeated until the stopping criterion is reached, usually a sufficiently good fitness or a maximum number of iterations. The pseudo code of the standard GSA (SGSA) is illustrated by Fig. 1.

3. The proposed quantum-inspired gravitational search algorithm (QIGSA)

In classical mechanics, an object is described by its position and velocity vectors (these vectors determine the trajectory of the object). And the object moves under Newton-mechanics by using Newtonian mechanics laws. In more detail, the equation of motion is Newton's second law. In a quantum space, due to existing uncertainty, the position and velocity of an individual cannot be determined simultaneously. Therefore, in a quantum behaved system, the performance of the system is bounded to work in a different way. In other words, in the quantum space all objects are allowed to move under quantum–mechanical rules rather than the classical Newtonian motion.

3.1. Quantum mechanics

In quantum mechanics, Schrödinger's equation is the equivalent of Newton's law [12]. In the formalism of quantum mechanics, the state of a system at a given time is described by a complex wave function $\Psi(X, t)$, instead of position and velocity. It is also called quantum state or vector state. The Schrödinger equation describes how the quantum state of some physical system changes with time.

The most general form of Schrödinger equation is its time-dependent form, which gives a description of a system evolving with time:

$$j\hbar \frac{\partial}{\partial t}\Psi(X,t) = H(X)\Psi(X,t) \tag{8}$$

where $H(X)$ is a time-independent Hamiltonian operator given by [46]:

$$H(X) = -\frac{\hbar^2}{2m}\nabla^2 + V(X) \tag{9}$$

where $\hbar$ is Planck's constant, ∇^2 is the Laplacian operator, m is the mass of the object, and $V(X)$ is the potential energy distribution. In the Schrödinger equation, the unknown is the wave function $\Psi(X, t)$, which has no direct physical meaning. Although, its amplitude squared, $|\Psi|^2$, is a probability measure for the agent's motion. It allows computing the probability of finding an object in a particular region at a particular time.

It can be concluded that the probability of the object appearing in position X from a probability density function could be learnt which is dependent on the potential field that the object lies in. Hence, the probability of finding an object at a particular position in the quantum search space is mapped into its actual position in the solution space by a technique called "collapsing".

The "collapse" or reduction of the vector state of a quantum system to a definite state as the result of a measurement is introduced by Neumann in [28]. In a 3-dimensional space for an object with the wave function $\Psi(X, t)$, one can write:

$$|\Psi|^2\, dx_1\, dx_2\, dx_3 = Q\, dx_1\, dx_2\, dx_3 \tag{10}$$

where $Q\,dx_1\,dx_2\,dx_3$ is the probability of measurement of the object's position at the time t for finding it in the volume element around the point (x_1, x_2, x_3). Therefore, $|\Psi|^2$ as the probability density function should satisfy the following condition.

$$\int_{-\infty}^{+\infty} |\Psi|^2\, dX = \int_{-\infty}^{+\infty} Q\, dX = 1 \tag{11}$$

(a) Search space identification.
(b) Randomized initialization $X_i = (x_i^1, x_i^2, \ldots, x_i^n)$, *for i=1,2,…,N*
(c) Fitness evaluation of objects (agents).
(d) Updating $Kbest$, $G(t)$, $best(t)$, $worst(t)$, $M(t)$ (Eqs 2, 3, 4).
(e) Calculating the acceleration (Eq. 5) and velocity (Eq. 6).
(f) Updating agents' position (Eq. 7) to yield $X_i(t+1)$ *for* $i=1,\ldots,N$.
(g) Repeat steps c to g until the stopping criterion is reached.
(h) End.

Fig. 1. The pseudo code of standard GSA (SGSA).

It should be noted that the integration is performed over the entire feasible space. One way to solve Eq. (8) is the use of separation of variables. In other words, the wave function is considered as a product of spatial and temporal terms:

$$\Psi(X,t) = \psi(X)f(t) \tag{12}$$

In this case, the time-independent equation is derived from the time-dependent equation. If we used Eq. (12), Eq. (8) becomes:

$$j\hbar\psi(X)\frac{\partial}{\partial t}f(t) = f(t)\left(-\frac{\hbar^2}{2m}\nabla^2 + V(X)\right)\psi(X) \tag{13}$$

This equation can be rewritten as:

$$\frac{j\hbar}{f(t)}\cdot\frac{d}{dt}f(t) = \frac{1}{\psi(X)}\cdot\left(-\frac{\hbar^2}{2m}\nabla^2 + V(X)\right)\psi(X) \tag{14}$$

Since the left-hand side of Eq. (14) is only a function of t and the right-hand side is only a function of X, the two sides must be equal to a constant. This constant is called E and represents the energy of the object. Thus, one can extract two differential equations as:

$$\frac{1}{f(t)}\cdot\frac{d}{dt}f(t) = -\frac{jE}{\hbar} \tag{15}$$

and

$$\frac{1}{\psi(X)}\left(-\frac{\hbar^2}{2m}\nabla^2 + V(X)\right)\psi(X) = E \rightarrow H\psi(X) = E\psi(X) \tag{16}$$

Eq. (15) is easily solved to yield:

$$f(t) = e^{\frac{-jEt}{\hbar}} \tag{17}$$

Therefore, we have:

$$\Psi(X,t) = \psi(X)e^{\frac{-jEt}{\hbar}} \tag{18}$$

In this case, $|\Psi|^2$ which is the probability density function and indicates the probability of finding an object at a particular position in the quantum search space is calculated as the following:

$$|\Psi(X,t)|^2 = \psi(X)e^{\frac{-jEt}{\hbar}}\psi^*(X)e^{\frac{jEt}{\hbar}} = \psi(X)\psi^*(X) = |\psi(X)|^2 \tag{19}$$

It is obvious that the probability density function is a time-independent function. Therefore, to find the probability density function $|\Psi(X,t)|^2$ we need only to calculate the wave function $\psi(X)$ by solving the time-independent Schrödinger equation of Eq. (16).

3.2. The quantum model of GSA

Now, we assume that in the GSA system each object has a quantum behavior. On the other hand each object is formulated by a wave function as a quantum state. For more explanation, in the quantum version of the GSA (QIGSA), the positions of *Kbest* are considered as attractive potential fields that will eventually pull all objects to themselves.

In Quantum GSA, all objects move under quantum–mechanical rules rather than the classical Newtonian random motion. In the classical environment, all objects move toward the promising locations defined by *Kbest* set. The objects are then attracted to these locations through the optimization process. Such attraction hopefully leads to the global optimum. It should be noted that *Kbest* is the set of the heaviest objects with the best fitness value and therefore the heaviest masses.

To simplify the formulation, without loss of generality, consider an object in a quantum system with the 1-dimensional search space. In this case the position vector X is reduced to scaler x. Let $y = x - c$, where c is the position which is defined by one of *Kbest* set members. To avoid divergence of QIGSA, y should approach zero. Therefore, we need to apply an attractive potential field centered at zero. Any potential well can be used but based on our experiment the simplest and most effective one is the delta potential well (Eq. (20)) [21,35]. Therefore, assume that we have some 1-dimensional Delta potential wells which the number of them is equal to the number of *Kbest* set.

$$V(y) = -\gamma\delta(y) \tag{20}$$

where γ is a positive number proportional to the "depth" of the potential well. The depth is infinite at the origin and zero elsewhere. The Schrodinger equation for this model is:

$$E\psi(y) = \left(-\frac{\hbar^2}{2m}\nabla^2 - \gamma\delta(y)\right)\psi(y) \tag{21}$$

$$\frac{\hbar^2}{2m}\frac{d^2}{dy^2}\psi(y) + \gamma\delta(y)\psi(y) + E\psi(y) = 0 \tag{22}$$

when $y \neq 0$, Eq. (22) can be written as:

$$\frac{d^2\psi}{dy^2} + \frac{2m}{\hbar^2}E\psi = 0 \tag{23}$$

In order to prevent diverging of agents, the following boundary conditions is utilized,

$$|y| \to \infty \leftrightarrow \psi \to 0 \tag{24}$$

By the above condition, Eq. (23) can be calculated as follows:

$$\psi(y) = e^{\frac{-\sqrt{-2mE}}{\hbar}|y|} \quad \text{when} \quad y \neq 0 \tag{25}$$

In addition, wave function must satisfy the normalization condition:

$$\int_{-\infty}^{+\infty} |\psi|^2 dy = 1 \tag{26}$$

Hence, the probability density function Q is given by:

$$Q(y) = |\psi(y)|^2 = e^{\frac{-2\sqrt{-2mE}}{\hbar}|y|} \tag{27}$$

According to Heisenberg principle of uncertainty [37], it is impossible to evaluate simultaneously both the position and the velocity of an object with an acceptable degree of accuracy and certainty. Thus, in the QIGSA we expect that no velocity and position term will exist in the update equations, comparing to standard GSA.

3.3. Measurement of the position

As far the positions of agents in the GSA are the candidate solutions for the problem at hand, we need to "measure" the position of the agent in the QIGSA. This is the fundamental concern in the quantum mechanics. Particularly, measuring instruments obey Newtonian laws while the agent itself follows the quantum rules. To connect these work spaces, quantum state needs to "collapse" the wave function of a moving agent into the space of the measurement. This localization process can be easily achieved through the Monte Carlo simulation method.

Delta potential well guarantees to limit the quantum boundary of the object. To evaluate the fitness value, we need to know the accurate information about the position of the object. However, the quantum state function can only give the probability density function that the object appears at the position y. Hence, we have to calculate the position of the objects, which is called collapsing the quantum state to the classical state.

Monte Carlo Method can simulate the process of measurement. By generating a random variable *rand* uniformly distributed between 0 and 1, Eq. (27) could be updated. Thus, Eq. (27) can be simplified with substituting a random number instead of it:

$$Q(y) = |\psi(y)|^2 = e^{\frac{-2\sqrt{-2mE}}{\hbar}|y|} = rand \tag{28}$$

By a simple mathematics, we have:

$$-\frac{2\sqrt{-2mE}}{\hbar}|y| = \ln(rand) \tag{29}$$

$$|y| = \frac{\hbar}{2\sqrt{-2mE}}\ln\left(\frac{1}{rand}\right) \tag{30}$$

$$y = x - c = \pm\frac{\hbar}{2\sqrt{-2mE}}\ln\left(\frac{1}{rand}\right) \tag{31}$$

Thus, the position of an object accurately is measured as follows,

$$x = c \pm \frac{\hbar}{2\sqrt{-2mE}}\ln\left(\frac{1}{rand}\right) \tag{32}$$

It is assumed that $\frac{\hbar}{2\sqrt{-2mE}} = g|c - x|$, where g is a constant and $|c - x|$ is used to scale the variation of the new created object around one of the *Kbest* members. Therefore, for an n-dimensional search space system one can summarize that:

(a) Search space identification.

(b) Randomized initialization $X_i = (x_i^1, x_i^2, \ldots, x_i^n)$, *for i=1,2,…,N*

(c) Fitness evaluation of agents.

(d) Updating $Kbest$, $best(t)$, $worst(t)$, $M(t)$ (Eqs 2, 3, 4).

(e) Choosing center of well by a probabilistic procedure.

(f) Updating agents' position (Eq. 33) to yield $X_i(t+1)$ *for* $i=1,\ldots,N$.

(g) Repeat steps c to g until the stopping criterion is reached.

(h) End.

Fig. 2. The pseudo code of QIGSA.

Table 1
The comparison of QIGSA with other algorithms (D = 10). The bold value(s) in each row of Table 1 indicate(s) the best results obtained by competing algorithms for each function.

	ES [5]	PLES [5]	PBO [38]	RCMAES [2]	DMPSO [22]	ALSEA [3]	RPO [33]	QPSO [46]	SGSA	QIGSA
f_1	8.1602e−9	8.4020e−9	8.71e−09	5.20e−9	**0**	5.14e−9	8.8335e−9	**0**	**0**	**0**
f_2	2.9000e−6	9.6502e−9	9.68e−09	4.70e−9	1.2960e−13	5.31e−9	8.5986e−9	**0**	324.2993	4.7225
f_3	3.5217e+5	1.1806e+5	4.15e−01	5.60e−9	7.0064e−9	4.94e−9	8.4855e−9	8.0701e+4	1.8455e+5	**3.888e−15**
f_4	4.1357e+3	6.0334e+3	7.94e−07	5.02e−9	1.8851e−3	1.79e+6	8.5470e−9	**0**	6.1205e+4	3.4839
f_5	1.3682e+3	9.0571e+2	4.85e+01	6.58e−9	1.1383e−6	6.57e−9	2.1331e+0	**0**	1.9337e+4	**0**
f_6	7.4904e+1	3.0558e+1	4.78e−01	4.87e−9	6.8925e−8	5.41e−9	1.2463e+1	4.5347	576.8353	**0**
f_7	1.1826e+0	4.0943e+0	2.31e−01	3.31e−9	4.5189e−2	4.91e−9	3.7050e−2	636.9838	4.9427e+3	**0**
f_8	2.0368e+1	2.0383e+1	**2.00e+1**	**2.00e+1**	**2.00e+1**	**2.00e+1**	2.0274e+1	20.2671	20.9868	**2.00e+1**
f_9	5.6229e+1	2.6284e+1	1.19e−1	2.39e−1	**0**	4.49e+1	1.9194e+1	1.9932	1.7229	**0**
f_{10}	1.1842 e+2	3.6699e+1	2.39e−1	7.96e−2	3.6217e+0	4.08e+1	2.6765e+1	9.0177	9.2244e−1	**0**
f_{11}	1.1464e+1	9.9390e+0	6.65e+00	9.34e−1	4.6229e+0	3.65e+0	9.0288e+0	8.7431	5.1939e−1	**1.4124e−4**
f_{12}	7.3634e+4	1.3773e+4	1.49e+2	2.93e+1	**2.4007e+0**	2.09e+2	6.0453e+2	3.8006e+4	1.4705e+1	4.0464e+4
f_{13}	1.7175e+04	3.0083e+1	6.53e−1	6.96e−1	3.6865e−1	4.94e−1	1.1365e+0	0.5358	1.0438	**0**
f_{14}	4.1490e+0	4.1802e+0	2.35e+0	3.01e+0	2.3601e+0	4.01e+0	3.7064e+0	3.5215	4.2636	**1.4324**
f_{15}	7.4828e+2	4.7751e+2	5.10e+2	2.28e+2	**4.8539e+0**	2.11e+2	2.9377e+2	413.1732	400.0002	181.1316
f_{16}	5.3178e+2	1.8163e+2	**9.59e+1**	9.13e+1	9.4756e+1	1.05e+2	1.7717e+2	1.4443e+3	103.5050	100
f_{17}	4.4994e+2	1.9581e+2	**9.73e+1**	1.23e+2	1.009e+2	5.49e+2	2.1181e+2	1.3473e+3	252.4641	100.0742
f_{18}	1.1433e+3	1.0148e+3	7.52e+2	3.32e+2	7.6067e+2	4.97e+2	9.0154e+2	1.9734e+3	632.5293	**300**
f_{19}	1.1258e+3	1.0019e+3	7.51e+2	3.26e+2	7.1430e+2	5.16e+2	8.4450e+2	1.9691e+3	568.4520	**300**
f_{20}	1.3199e+3	9.9894e+2	8.13e+2	3.00e+2	8.2196e+2	4.42e+2	8.6289e+2	1.9718e+3	**300**	**300**
f_{21}	8.9174e+0	1.0794e+3	1.05e+3	5.00e+2	5.3600e+2	**4.04e+2**	6.3493e+2	2.0461e+3	500	500
f_{22}	9.2978e+2	8.8049e+2	6.59e+2	7.29e+2	6.9242e+2	7.40e+2	7.7885e+2	1.6392e+3	1.0315e+3	**500**
f_{23}	1.3495e+3	1.1141e+3	1.06e+3	5.59e+2	7.3034e+2	7.91e+2	8.3455e+2	2.2979e+3	572.9853	**524.1053**
f_{24}	1.1959e+3	2.8238e+2	4.06e+2	2.00e+2	2.2400e+2	8.65e+2	3.1383e+2	1.1008e+3	**200**	372.5631
f_{25}	4.1518e+2	6.9231e+21	4.06e+2	3.74e+2	3.6571e+2	4.42e+2	**2.5731e+2**	1.1200e+3	932.0172	665.9631

$$\begin{cases} x_i^d(t+1) = c_i^d + g|c_i^d - x_i^d(t)| \cdot \ln\left(\frac{1}{rand}\right) & \text{if } S \geqslant 0.5 \\ x_i^d(t+1) = c_i^d - g|c_i^d - x_i^d(t)| \cdot \ln\left(\frac{1}{rand}\right) & \text{otherwise} \end{cases} \quad (33)$$

where *rand* and *S* are two random numbers generated by uniform probability density function in the interval $(0,1)$, g is a parameter which determines the balance between exploration and exploitation and c_i^d is the dimension d of the position (center) of a potential well selected at random among from *Kbest* set.

3.4. The proposed QIGSA

In the proposed QIGSA to avoid trapping into local optima, we assume that each *Kbest* member is the center of an attractive potential filed. Therefore, at each iteration, the number of attractive potential field is equal to the number of *Kbest* members. As mentioned above, *Kbest* is a function of time.

In this step, with a probabilistic mechanism, every agent chooses one of the *Kbest* members and moves toward it. This probabilistic mechanism is based on the masses of the *Kbest* set. The way of selecting one of the *Kbest* members is so similar to the roulette wheel selection in genetic algorithms. In this kind of selection, individuals are given a probability of being selected that is directly proportionate to their fitness.

It should be concerned that the parameters of the algorithm should support the exploration and exploitation abilities of the algorithm simultaneously. Based on experimental understanding, the exploration ability of the algorithm should be increased at the beginning iterations and decreased by lapse of time; simultaneously the exploitation ability of the algorithm should be increased in order to find the global optimum by an acceptable accuracy.

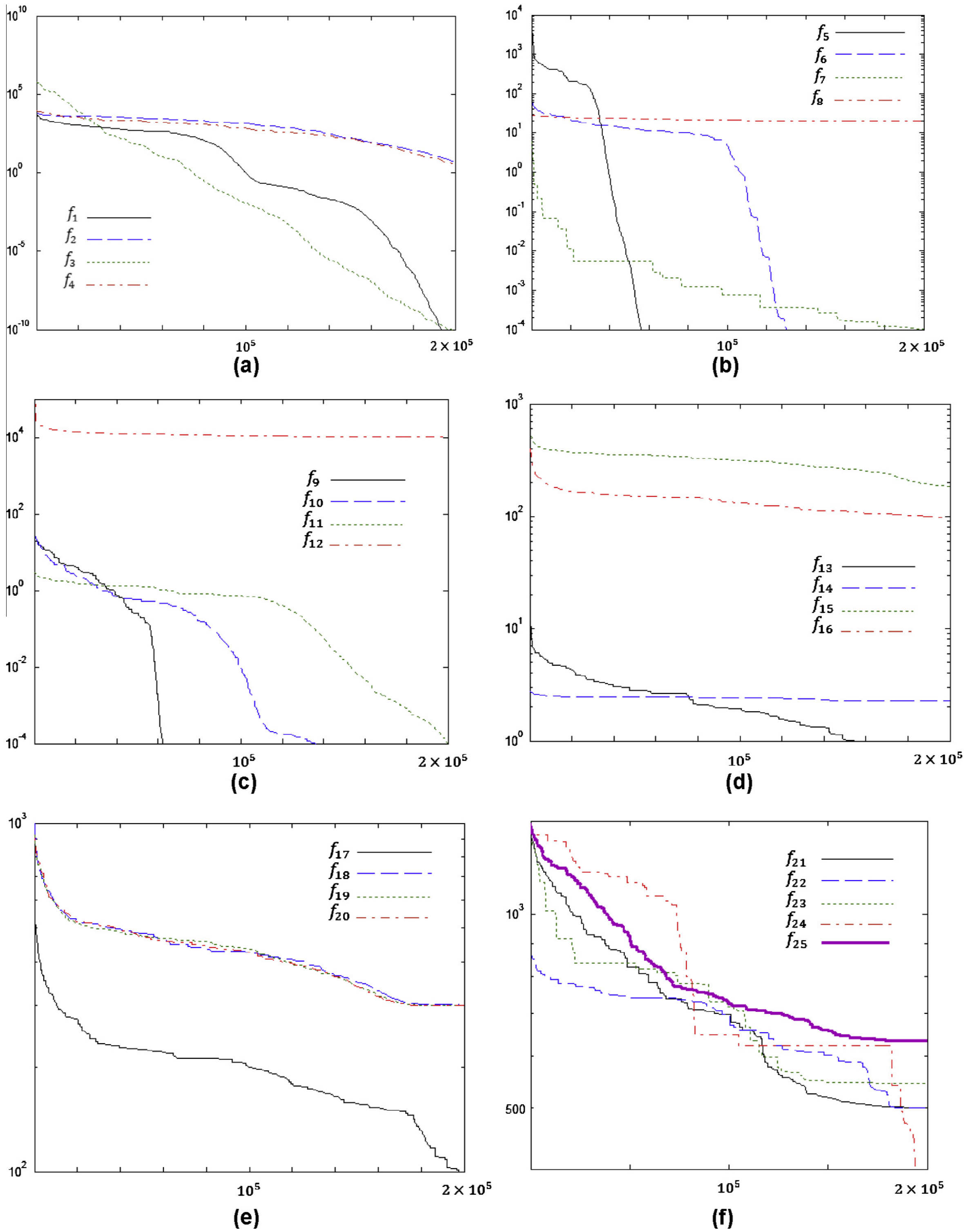

Fig. 3. the convergence characteristics of Median performance vs. fitness evaluations in logarithmic scale for the 25 functions in dimension 10.

Based on this principle, at first the quantity of potential wells – which refers to good positions (solutions) – is considered equal to the number of objects, then by lapse of time it is reduced. Consequently, at first, agents can widely move randomly and at the end by decreasing the number of potential wells, the agents move toward the best agent. The pseudo code of the proposed QIGSA is shown in Fig. 2.

Table 2
The comparison of performance of QIGSA with nine other algorithms.

	Better	Same	Worse
QIGSA			
ES [4]	**(23)**	**(0)**	**(2)**
	Functions 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25		Functions 2, 12
PLES [4]	**(23)**	**(0)**	**(2)**
	Functions 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25		Functions 2, 12
PBO [34]	**(18)**	**(1)**	**(6)**
	Functions 1, 3, 5, 6, 7, 9, 10, 11, 13, 14, 15, 18, 19, 20, 21, 22, 23, 24	Function 8	Functions 2, 4, 12, 16, 17, 25
RCMAES [2′]	**(17)**	**(2)**	**(6)**
	Functions 1, 3, 5, 6, 7, 9, 10, 11, 13, 14, 15, 17, 18, 19, 20, 21, 22, 23	Functions 8, 21	Functions 2, 4, 12, 16, 24, 25
DMPSO [20]	**(15)**	**(3)**	**(7)**
	Functions 3, 5, 6, 7, 10, 11, 13, 14, 17, 18, 19, 20, 21, 22, 23	Functions 1, 5, 8	Functions 2, 4, 12, 15, 16, 24, 25
ALSEA [2]	**(20)**	**(1)**	**(4)**
	Functions 1, 3, 4, 5, 6, 7, 9, 10, 11, 13, 14, 15, 16, 17, 18, 19, 20, 22, 23	Function 8	Functions 2, 12, 21, 25
RPO [30]	**(20)**	**(0)**	**(5)**
	Functions 1, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23		Functions 2, 4, 12, 24, 25
QPSO [42]	**(20)**	**(2)**	**(3)**
	Functions 3, 6, 7, 8, 9, 10, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25	Functions 1, 5	Functions 2, 4, 12
SGSA	**(20)**	**(3)**	**(2)**
	Functions 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18, 19, 22, 23	Functions 1, 20, 21	Functions 12, 24

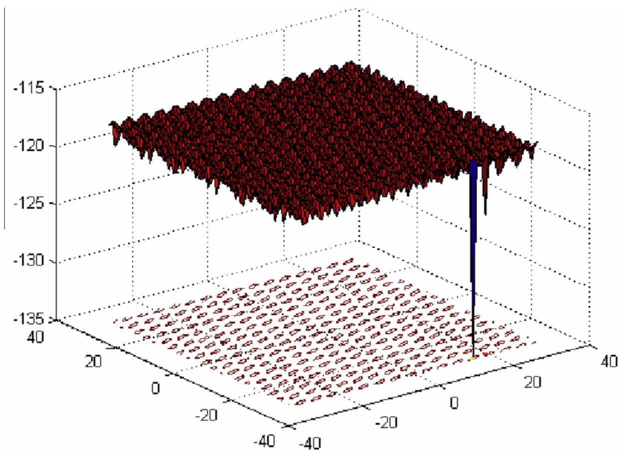

Fig. 4. The 2-dimensional fitness landscape of f_8.

It should be noted that QIGSA has a good searching ability because each object is considered in a quantum space. In more detail, the classical mechanics is completely deterministic: Given the exact positions and velocities of all agents at a given time, one can calculate the future positions and velocities of all objects at any other time along with the function of velocity. Therefore, in heuristic search algorithms based on classical mechanics, for having a complete search of feasible space, number of agents should be increased.

In comparison in the quantum space, the positions of agents are not in a special place. Based on uncertainty principle of quantum mechanics, the object's position has a definite value but that we just do not know what it is. It means that each agent is spread in the feasible space with different probability. Each object does not have a certain position until it has been observed. This characteristic can be very useful for heuristic search algorithms. By above explanation it can be concluded that quantum inspired heuristic search algorithms have a high exploration ability in comparison with the classical ones. This is the main reason for good search ability of QIGSA.

4. Experimental results

To evaluate the performance of the proposed algorithm, it is tested on 25 standard benchmark functions [57]. These benchmark functions are presented below. The results obtained are compared with the standard GSA and those available in the literature.

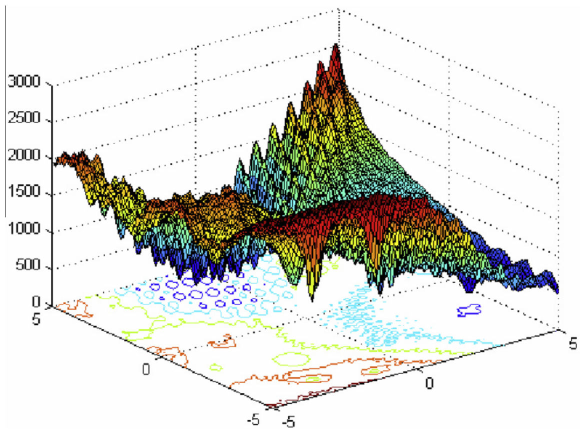

Fig. 5. The 2-dimensional fitness landscape of f_{16}.

4.1. Benchmark functions

In order to solve continuous function optimization problems, several optimization algorithms have been presented in the literature by considering a subset of the standard test problems such as Sphere, Schwefel, Rosenbrock, and Rastrigin. Often, confusing results limited to the test problems were reported in the literature in such a way that the same algorithm working for a set of functions may not work for any other set of functions. For these reasons, these algorithms should be evaluated more systematically by determining a common termination criterion, size of problems, initialization scheme, and running time. The CEC2005 introduced new benchmark functions to be publicly available to the researchers for evaluating their algorithms which are called second group of benchmark functions. The problem definition files, codes and evaluation criteria are made available in [58,59].

The effectiveness of the QIGSA is investigated by considering 25 functions defined by [57] which have different complexity. Functions f_1–f_5 are unimodal while the remaining twenty functions are multi-modal.

For unimodal functions, the convergence rates of the algorithm are more interesting than the final results of optimization. In contrast, for multimodal functions because of having lots of local optima, the ability of finding the optimum solution or a good near global optimum are more important than the convergence rate of the algorithm.

Test functions are designed to test an optimizer's ability to locate a global optimum under a variety of circumstances such as function landscape is highly conditioned, function landscape is rotated, optimum lies in a narrow basin, optimum lies on a bound, optimum lies beyond the initial bounds, the function is not continuous everywhere and bias is added to the function evaluation.

All these functions are scalable. The detailed descriptions of them can be found in [57]. Many of them are the shifted, rotated, expanded and/or combined variants of the classical functions. Some of these changes cause them to be more resistant to simple search tricks. Other changes, such as rotation, transfer separable functions into non-separable ones, which will be particularly challenging to the algorithm's ability.

4.2. Setting parameters

The proposed QIGSA has three main parameters: the swarm size, parameter g which determines the balance between exploration and exploitation and the size of *Kbest*. The performance of each algorithm depends on suitable parameter values. Setting parameters will cause good exploration and exploitation capabilities of the algorithm. But the problem is to find good parameter values and how to vary them.

The proposed algorithm is tested with a different population/swarm sizes to get the best performance. The results show that QIGSA with a swarm size of 200 have good performance for a wide range of functions. Thus, the swarm size is set to 200 for all functions in this paper.

The next important parameter is g. In QIGSA, g is a crucial parameter where can affect the precision of final solutions obtained. On the other hand, g has a direct effect on the convergence behavior of the algorithm. In order to select the best value for g, two functions (a uni-modal and a multi-modal) are considered and the effect of different values for g is investigated. The convergence characteristics of QIGSA of the unimodal function when g is varying from 0.01 to 2.5 is investigated. It was shown that the algorithm converges for $0.2 < g < 1.8$ in this case. In a similar manner for the multimodal function, the QIGSA converges for $0.5 < g < 1.32$. Based on the experiments the values in the interval [0.5, 1.3] are suitable for g. As we know, a larger value for g makes the algorithm to more explore the search space, thus in the experiments we set the parameter g to 1.3.

It should be noted that since local exploitation and global exploration capabilities are always twisted together in the search process, it is difficult to fix the values for parameters [57]. On the other hand, a good balance between exploration

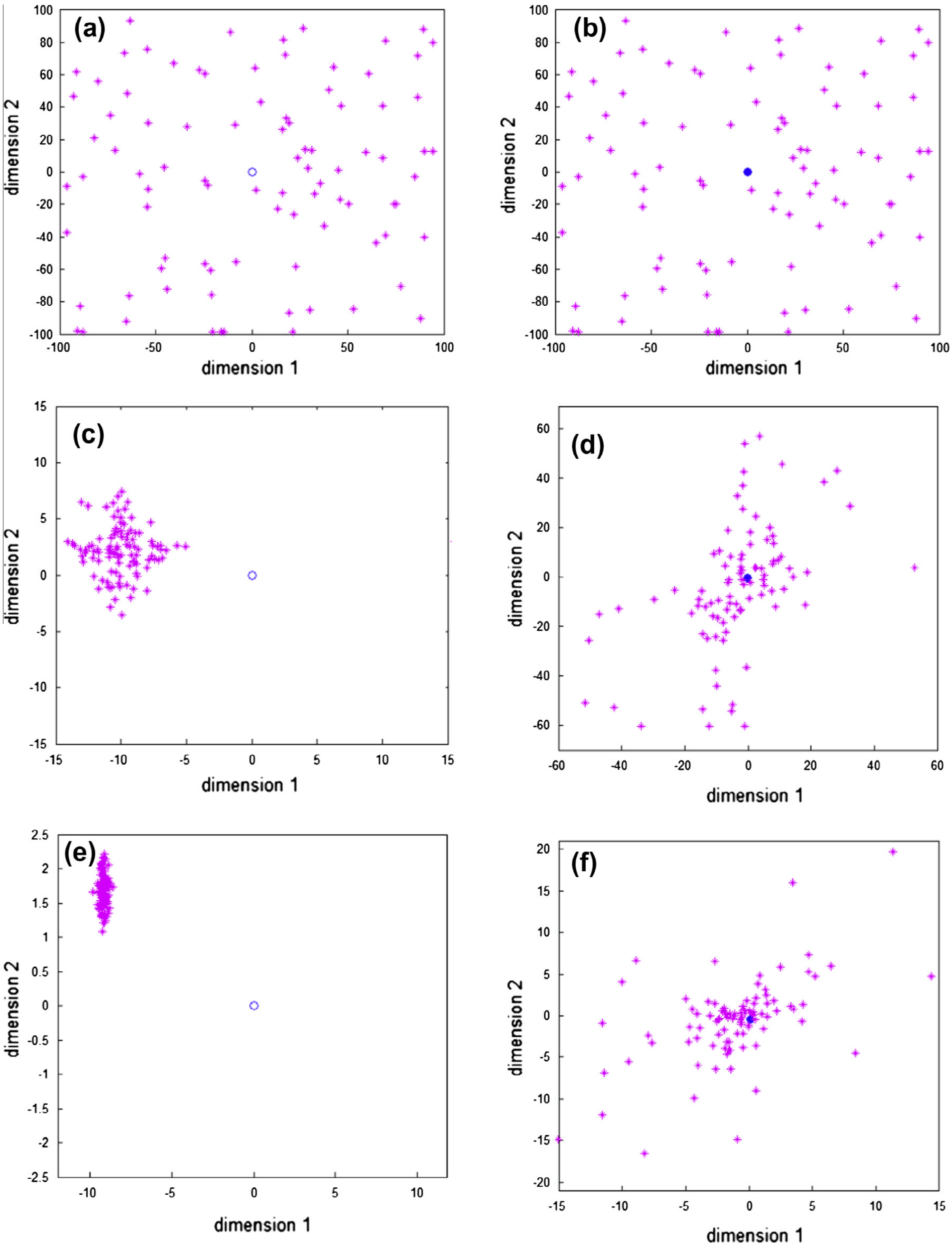

Fig. 6. The agent's positions for two dimensional function f_3. (a), (c), (e), (g) and (j) are the swarm position of SGSA (classical search space) at iterations 1, 100, 300, 600 and 1000, respectively. (b), (d), (f), (h) and (j) are the swarm position of QIGSA (quantum search space) at iterations 1, 100, 300, 600 and 1000, respectively. In these figures the circle sing represents the position of the optimum and the star sign represent the position of one object of the swarm.

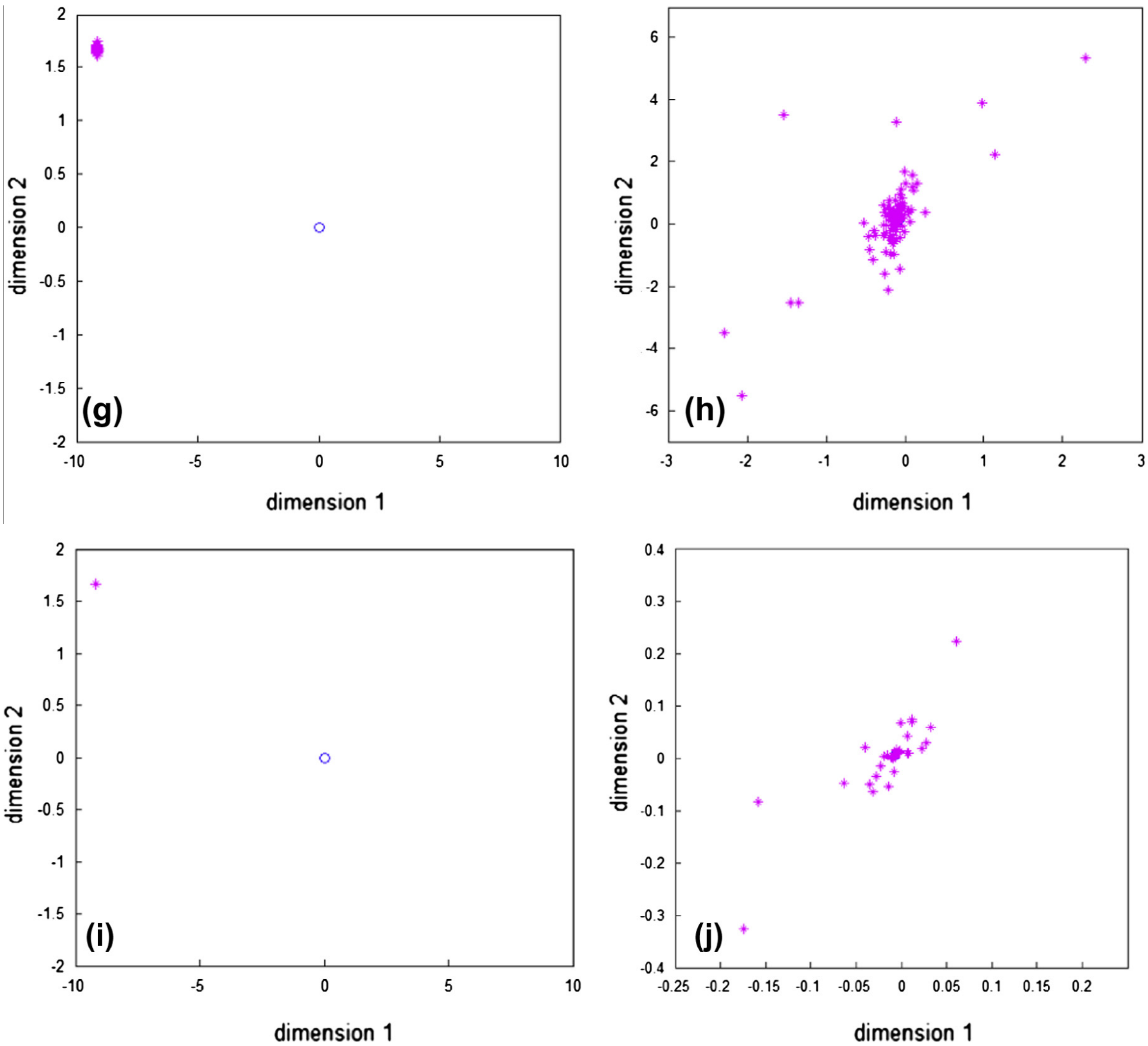

Fig. 6 (*continued*)

and exploitation abilities of a heuristic algorithm makes it more robust to avoid getting trapped in local optima and also increases the convergence speed of the algorithm.

In this paper to control the exploration and exploitation of the proposed QIGSA, *Kbest* is considered as a function of time with the initial value at the beginning and is decreased with time by the linear control. It is set to *N* at the beginning and is decreased to 1 at the end.

4.3. Comparison with SGSA, QPSO and other algorithms

The QIGSA is applied on 25 benchmark functions which have been explained in Section 4.1. The results obtained are compared with those of the Standard GSA (SGSA) [32], Quantum Particle Swarm Optimization (QPSO) [46], Evolutionary Strategy (ES) [5], Parameter-less Evolution Strategy (PLES) [5], a Population Based Optimization (PBO) [38], A Restart CMA Evolution Strategy (RCMA-ES) [38], Dynamic Multi-Swarm Particle Swarm Optimizer (DMPSO) [22], Advanced Local Search Evolutionary Algorithm (ALSEA) [3] and Real Parameter Optimization (RPO) [33]. More information about these algorithms is available in the literature.

4.3.1. The experimental result for $n = 10$

The experimental results for $n = 10$ on these functions are given in Table A.1 in Appendix A. In Table A.1, Best objective function error values reached after10^3, 10^4 and 10^5 function evaluations (FES) on the 25 test problems for dimension $n = 10$ are reported. Moreover, minimum (1st), 7th, median (13th), 19th, and maximum values (25th), as well as mean value and

Table 3
Comparison of QIGSA with other algorithms.

	QPSO	SGSA	QIGSA
f_1	**0**	852.0102	**0**
f_2	**31.9672**	1.7533e+4	1.5439e+2
f_3	3.5610e+6	1.8455e+8	**7.7014e+5**
f_4	**4.5298e+3**	6.1205e+4	1.7733e+4
f_5	7.1522e+3	1.9337e+4	**1.0980e+3**
f_6	**23.2814**	1.1685e+8	2.0413e+2
f_7	2.8673e+3	4.9427e+3	**1.8213e+3**
f_8	21.9834	20.9868	**20.8985**
f_9	13.0655	**11.7229**	22.4839
f_{10}	121.9682	113.2244	**101.1429**
f_{11}	40.5589	36.1939	**31.4251**
f_{12}	1.2286e+6	6.4705e+3	**3.6113e+3**
f_{13}	7.7657	8.0438	**4.0251**
f_{14}	14.8160	12.2636	**10.6753**
f_{15}	830.0464	**300.0002**	601.6826
f_{16}	1.6356e+3	263.5050	**204.2933**
f_{17}	1.7243e+3	272.4641	**156.5553**
f_{18}	2.0502e+3	300	**298.5513**
f_{19}	2.0421e+3	800	**314.7887**
f_{20}	2.0476e+3	**800**	913.7799
f_{21}	1.4111e+3	**500**	1.0845e+3
f_{22}	2.2962e+3	830	**800.6742**
f_{23}	2.1976e+3	**522.9853**	1.0701e+3
f_{24}	1.9267e+3	1.8407e+3	**902.1490**
f_{25}	1.9345e+3	932.0172	**863.8745**

Table 4
The performance comparison of QIGSA with QPSO and SGSA.

	Better	Same	Worse
QIGSA			
QPSO	**(22)**	**(1)**	**(2)**
	Functions 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25	Function 1	Functions 2, 9
SGSA	**(20)**	**(0)**	**(5)**
	Functions 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 16, 17, 18, 19, 22, 24, 25		Functions 9, 15, 20, 21, 23

standard deviation over 25 independent runs are given in this table. As Table A.1 illustrates, QIGSA provides acceptable results for most of benchmark functions.

In Table 1 the performance of the QIGSA is compared with nine other meta-heuristic search algorithms. It is obvious from the table that QIGSA performed well for most of the functions. The average error values are reported in this table for competing algorithms, where the comparison results for the following cases are:

4.3.1.1. Unimodal functions. Functions f_1 to f_5 are unimodal functions. In this case the convergence rate of search algorithm is more important than the final results because there are other methods which are specifically designed to optimize unimodal functions.

As this table illustrates, QIGSA provides better results than other algorithm for f_1, f_3, f_5. The largest difference in performance between them occurs with f_3 that QIGSA reaches an acceptable result against other algorithms. Also, the good convergence rate of QIGSA can be concluded from Fig. 3. According to these figures, QIGSA tends to find the global optimum in a satisfactory duration and hence has a higher convergence rate.

4.3.1.2. Multimodal functions. Multimodal functions have many local minima and almost are most difficult to optimize. For multimodal functions, the final results are more important since they reflect the ability of the algorithm in escaping from poor local optima and locating a near-global optimum. We have carried out experiments on f_6 to f_{25} where the dimension of these functions is set to 10. The results are averaged over 25 runs and the mean values in the last iteration are reported for these functions. For $f_6, f_7, f_9, f_{10}, f_{11}, f_{12}, f_{13}, f_{14}, f_{18}, f_{19}, f_{20}, f_{22}$ and f_{23}, QIGSA performs much better than the other algorithms. However, for functions f_{15}, f_{21}, f_{24} and f_{25} QIGSA cannot tune itself and have not a good performance. The results for f_8, f_{16} and f_{17} show that all algorithms have similar conditions and the performances are almost the same as Table 2 illustrates it. To compare these algorithms more easily, the results of Table 1 are summarized in Table 2.

The entries in Tables 2 illustrate that QIGSA performed significantly better than other algorithms at least for fifteen functions while most of other algorithms have similar results. For example, all algorithms have similar performance on f_8.

The 2-dimensional fitness landscape of f_8, given in Fig. 4, shows that it is a strange deceptive problem. This function has lots of optima and the global optimum is located exactly at the boundary. Although QIGSA tries to find a better local optimum, it is still far away from the global optimum (with fitness value 0).

The function f_{16} is a separable multi-modal and scalable function with a huge number of local optima as illustrated in Fig. 5. This makes its characteristics hard to analysis. Our future work will focus on the analysis of QIGSA on functions like f_{16} and finding a good way of controlling population diversity of the algorithm in order to reach to global optimum with a good precision.

The QIGSA combines local search methods with global search methods attempting to balance exploration and exploitation simultaneously. In the case of multi-modality, although all functions are rotated, most of the time, the algorithm is well guided towards the optima. It can be observed from Fig. 3, when the *FEs* increases from 10^3 to 10^4 and 10^5, the average error values for each problem is decreased. But for few functions such as f_8 because of having the global optimum on the border, the common case will not occur.

In order to get a more thorough analysis of QIGSA's performance and comparison of quantum search space and classical search space, the searching process of SGSA and QIGSA are illustrated with more details. In this order, the position of agents are plotted for same iteration when dimension is two for function f_3. The agent's positions are shown in Fig. 6 at iterations 1, 100, 300, 600 and 1000 for function f_3. As mentioned in [58,59], for each function the global optimum is shifted randomly in feasible search space, therefore in order to have a fair comparison, the global optimum is considered fixed for both quantum and classical search space in (0,0) which is plotted by a circle in the figures. It is necessary to be mentioned that both SGSA and QIGSA start the search process with the same initial swam/population.

Fig. 6 reveals that the SGSA loses rapidly the exploration ability while the QIGSA preserves the swarm diversity not only in the beginning iterations but also in the final iterations. In other words, the strong exploration ability of the proposed QIGSA could be concluded while the SGSA misses the global optimum due to loss of diversity in the beginning iterations.

As far as all comparing algorithms have similar performance and as we know that the QIGSA has got some ideas from QPSO and SGSA, thus for higher dimensional comparison with $n = 30$ and $n = 50$ just these algorithms are compared.

4.3.2. The experimental result for n=30

The experimental results for $n = 30$ on benchmark functions are presented in Table A.2 in Appendix A. The results are averaged over 25 independent runs under different random seeds and the Best function error values achieved when $FEs = 10^3$, 10^4 and 10^5 are reported for each function in this table. In Table 3 the performance of QIGSA is compared with SGSA and QPSO. It is obvious from the table that QIGSA performed well for most of the functions. The average error values are reported in this table for competing algorithms. To compare these algorithms more easily, the results of Table 3 are summarized in Table 4.

As Tables 3 and 4 illustrates, in Functions 2, 9, 15, 20, 21 and 23, SGSA and QPSO have better ability to explore and exploit than QIGSA but for nineteen remain functions QIGSA provides much better results than both of them because it is more robust. As an example, in Function f_1, SGSA has poor exploration and exploitation, whereas QIGSA has a very powerful ability to explore and exploit the search space and also has a high convergence rate. Therefore, these characteristics significantly cause good results. In Function f_{13}, SGSA and QPSO against QIGSA do not skillfully explore the search space to find the optimum. In Function 8, QIGSA, SGSA and QPSO have similar results and none of them can reach the optimum.

5. Conclusion

GSA is a powerful global searcher, but it is not effective enough for more complicated problems. The main reason for this is losing swarm diversity. The overall goal of this paper is to increase the exploration and exploitation abilities of GSA. In order to reach this aim, each object is considered in a quantum space and its movement changes is done by using the quantum mechanics theory. This consideration is introduced through GSA and the quantum inspired GSA (QIGSA) is faster and more precise than GSA itself. Also, QIGSA avoids premature convergence in cases where standard GSA failed. The effectiveness of the QIGSA was confirmed in the comparison with nine state-of-the-art- algorithms in this field on 25 highly complicated benchmark functions. The results obtained showed that the QIGSA which is inspired from the theory of quantum has a merit in the field of optimization. In future work our aim is to adapt the proposed algorithm for finding optima in high dimensional complicated optimization functions.

Acknowledgements

The authors would like to thank the INS Editorial Board and the anonymous reviewers for their very helpful suggestions.

Appendix A

See Tables A.1 and A.2.

Table A.1
Error values achieved at FEs = 10^3; 10^4; 10^5 for functions (n = 10).

FES		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24	F25
1.00E+04	1st	2.64E+03	282.3715	2.26E+05	819.1242	7.22E+03	1.20E+08	101.8386	20.0199	19.0067	34.3427	2.0985	8.01E+04	2.0074	1.1396	311.2518	160.6354	309.1779	628.7534	456.0689	623.7587	977.7691	618.9253	1.04E+03	683.7192	979.4239
	7th	7.04E+03	1.18E+03	1.62E+06	1.38E+03	9.24E+03	6.73E+08	129.9605	20.1197	20.1934	37.1934	3.1097	8.09E+04	2.0954	1.6692	414.9308	313.4399	310.897	772.8453	636.7498	679.3444	1.01E+03	768.9001	1.08E+03	737.2885	1.05E+03
	13th	8.91E+03	1.96E+03	1.86E+06	1.69E+03	1.11E+04	9.84E+08	192.6522	20.1668	20.3534	38.6392	3.4099	1.10E+05	2.692	1.6873	491.8342	317.5621	321.9675	860.3425	663.5896	697.6194	1.06E+03	900.445	1.14E+03	748.1836	1.06E+03
	19th	1.07E+04	2.43E+03	3.56E+06	2.37E+03	1.30E+04	1.1.2484e+9	214.9746	20.2799	20.7636	41.0964	3.9483	1.20E+05	3.0941	1.7241	608.795	330.4632	322.8723	910.5367	681.6327	759.1887	1.08E+03	926.3177	1.16E+03	825.0561	1.13E+03
	25th	1.30E+04	4.08E+03	7.50E+06	3.57E+03	1.51E+04	2.08E+09	256.8974	20.3539	20.9984	42.3941	4.2629	1.42E+05	3.1311	1.9056	710.7281	337.0257	377.8329	927.4356	703.5552	818.582	1.19E+03	982.2746	1.23E+03	1.03E+03	1.14E+03
	Mean	8.53E+03	1.89E+03	2.89E+06	1.98E+03	1.10E+04	1.00E+09	180.9486	20.1968	20.4095	39.0012	3.6863	1.07E+05	2.893	1.6966	511.8629	321.9515	330.5483	818.435	627.0192	715.2403	1.07E+03	839.155	1.13E+03	813.9056	1.07E+03
	Std. D	6.77E+06	9.38E+05	4.54E+12	5.45E+05	4.82E+06	3.19E+17	2.74E+03	0.0087	0.0043	7.0192	1.6649	7.06E+08	0.0809	0.0015	1.88E+04	144.8967	767.838	1.73E+04	8.87E+03	5.62E+03	6.50E+03	2.37E+04	5.29E+03	1.72E+04	4.13E+03
1.00E+05	1st	798.9742	140.2137	1.91E+05	388.7701	0	1.47E+04	25.5843	19.9949	9.6538	5.3421	2.0921	5.37E+04	1.3094	1.3421	192.1561	147.8402	173.7456	417.5262	398.0959	451.8346	600.3876	618.918	578.1748	476.2091	634.9218
	7th	1.48E+03	467.84.5	3.33E+05	819.1242	0	4.34E+05	46.0052	20.0589	9.9463	5.8333	2.9377	6.46E+04	1.1921	1.5763	245.8786	149.5075	190.4893	485.1237	450.2405	512.9364	690.3406	767.8197	579.1838	636.99	825.2897
	13th	1.63E+03	708.488	4.18E+05	1.10E+03	0	8.01E+05	77.403	20.0812	10.6739	6.0129	3.1097	7.91E+04	1.2691	1.5986	248.2452	169.8078	212.7634	517.0934	475.9296	525.3162	784.2523	878.9079	614.0705	717.9873	890.4496
	19th	2.01E+03	1.43E+03	7.26E+05	1.29E+03	0	1.27E+06	92.2743	20.1153	11.1351	6.2317	3.4113	8.93E+04	1.2995	1.6692	267.7291	196.4521	230.7535	534.6243	482.9057	526.1204	787.0473	911.8698	616.8778	764.0208	893.1592
	25th	2.86E+03	2.11E+03	1.24E+06	2.35E+03	0	2.36E+06	111.4502	20.1557	11.8532	6.519	3.6747	1.10E+05	1.3142	1.7105	294.6712	212.6345	234.5243	567.0313	636.7498	528.0489	797.2617	926.3177	635.0548	792.875	979.4239
	Mean	1.75E+03	913.0706	5.45E+05	1.19E+03	0	8.89E+05	69.0693	20.0876	10.0433	6.1489	3.3922	7.93E+04	1.2843	1.6532	256.8039	183.6576	208.7534	504.5201	498.7043	509.623	730.8879	820.766	604.6724	677.4033	844.6488
	Std. D	2.74E+05	3.75E+05	8.24E+10	1.96E+05	0	3.15E+08	463.7151	0.0015	0.0006	0.0012	0.8596	4.72E+08	4.41E−04	0.0045	238.7227	0.0451	717.0452	3.23E+03	7.54E+02	1.06E+03	7.26E+03	1.65E+04	628.0443	1.66E+04	1.67E+04
1.00E+06	1st	0	0.0106	0	0.0985	0	0	0	19.443	0	0	1.95E−05	1.58E+04	0	1.1213	132.6721	100	100.0112	300	300	300	500	500	539.9224	241.5642	634.9218
	7th	0	0.4509	0	0.7763	0	0	0	20.0275	0	0	6.84E−05	2.12E+04	0	1.3142	162.2631	100	100.0502	300	300	300	500	500	540.734	342.8745	656.9315
	13th	0	0.9953	0	0.984	0	0	0	20.0338	0	0	1.44E−04	3.19E+04	0	1.3982	171.6732	100	100.0734	300	300	300	500	500	542.8034	398.7855	675.3588
	19th	0	3.0524	5.68E−14	3.7983	0	0	0	20.0467	0	0	0.3812	3.38E+04	0	1.4988	200	100	100.1038	300	300	300	500	500	543.4544	478.8967	680.0109
	25th	0	27.8687	8.13E−12	18.0168	0	0	0	20.0878	0	0	0.4436	1.10E+05	0	1.5198	210.2451	100	100.1396	300	300	300	500	500	543.6125	675.0111	682.5924
	Mean	0	4.7225	3.89E−15	3.4839	0	0	0	20.0024	0	0	1.41E−04	4.05E+04	0	1.4324	181.1316	100	100.0742	300	300	300	500	500	542.1053	372.5631	665.9631
	Std. D	0	48.3912	2.62E−26	35.2461	0	0	0	0.0198	0	0	0.0009	1.15E+09	0	0.0163	712.0356	0	0.0019	0	0	0	1.01E−32	0	2.8062	519.4522	401.8213

Table A.2
Error values achieved at FEs = 10^3; 10^4; 10^5 for functions (n = 30).

FES		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
1.00E+04	1st	4.26E+04	4.63E+04	5.87E+08	6.89E+04	2.56E+04	2.29E+09	2.53E+03	21.0501	360.8547	511.8945
	7th	4.58E+04	7.10E+04	7.35E+08	1.08E+05	2.73E+04	3.62E+10	2.89E+03	21.1532	375.7952	545.8434
	13th	4.78E+04	7.52E+04	8.62E+08	1.10E+05	3.57E+04	3.64E+10	3.08E+03	21.17	384.9217	553.5485
	19th	5.33E+04	1.02E+05	9.01E+08	1.16E+05	3.66E+04	3.75E+10	3.14E+03	21.2205	430.6102	610.4806
	25th	5.69E+04	1.21E+05	1.18E+09	1.49E+05	3.74E+04	4.24E+10	3.19E+03	21.2311	441.0424	681.84
	Mean	4.97E+04	7.32E+04	8.53E+08	3.10E+05	3.25E+04	3.51E+10	2.97E+03	21.265	308.6449	580.7214
	Std. D	1.87E+07	6.47E+08	5.07E+16	8.13E+08	3.15E+07	5.28E+19	7.21E+04	0.0054	880.4192	4.45E+03
1.00E+05	1st	1.26E+01	4.63E+02	3.61E+05	5.99E+04	1.89E+04	4.60E+06	2.18E+03	21.014	178.1986	353.6771
	7th	1.53E+02	4.93E+02	4.37E+06	6.21E+04	1.98E+04	1.25E+06	2.21E+03	21.0511	207.2511	375.9237
	13th	1.71E+02	5.02E+03	4.70E+06	6.89E+04	2.01E+04	1.33E+07	2.46E+03	21.0853	253.5561	377.0742
	19th	1.91E+03	7.11E+04	1.28E+07	7.60E+04	2.11E+04	1.39E+07	2.48E+03	21.0986	267.0775	380.724
	25th	2.52E+03	8.26E+04	1.47E+07	8.36E+04	2.50E+04	1.55E+08	2.54E+03	21.0998	309.6858	384.5156
	Mean	1.78E+02	5.99E+04	3.68E+06	2.01E+04	2.10E+04	1.20E+07	2.37E+03	21.0717	235.1538	374.3829
	Std. D	9.67E+04	2.57E+08	3.09E+10	3.71E+07	5.76E+06	1.82E+15	2.71E+04	0.0014	239.5009	145.3067
1.00E+06	1st	0	4.33E+01	1.21E+05	4.91E+04	8.15E+03	1.47E+02	1.73E+03	20.7407	20.8651	89.0228
	7th	0	4.93E+02	1.45E+05	5.21E+04	8.19E+03	1.56E+02	1.82E+03	20.7924	21.395	99.0942
	13th	0	5.02E+02	1.54E+05	6.01E+04	9.59E+03	2.18E+03	1.84E+03	20.8577	21.5125	106.3388
	19th	0	7.11E+03	2.05E+06	6.89E+04	1.08E+04	2.41E+03	1.98E+03	20.9724	22.8156	131.1323
	25th	0	7.34E+03	2.17E+06	7.60E+04	1.22E+04	2.87E+03	1.99E+03	20.9893	26.5629	140.7838
	Mean	0	1.54E+02	7.70E+05	1.77E+04	1.10E+03	2.03E+02	1.82E+03	20.8985	22.4839	101.1429
	Std. D	0	1.90E+04	2.17E+12	1.27E+07	2.96E+06	3.45E+08	1.28E+04	4.36E−04	52.5031	89.7173

FES		F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
1.00E+04	1st	37.691	1.01E+06	35.0117	12.5779	897.0037	756.99	811.3067	1.08E+03	1.03E+03	1.20E+03
	7th	40.453	1.16E+06	35.8963	12.6528	1.03E+03	916.8748	886.0507	1.13E+03	1.30E+03	1.26E+03
	13th	41.3201	1.19E+06	57.6344	12.6818	1.16E+03	1.00E+03	1.04E+03	1.28E+03	1.32E+03	1.29E+03
	19th	41.607	1.29E+06	63.2653	12.88	1.23E+03	1.08E+03	1.07E+03	1.31E+03	1.32E+03	1.34E+03
	25th	42.2226	1.51E+06	63.5413	12.9245	1.26E+03	1.10E+03	1.17E+03	1.35E+03	1.39E+03	1.37E+03
	Mean	40.6107	1.23E+06	52.0698	12.7434	1.12E+03	972.0877	990.68	1.23E+03	1.27E+03	1.29E+03
	Std. D	3.13	3.43E+10	201.8661	0.0227	2.32E+04	1.98E+04	2.12E+04	1.42E+04	1.96E+04	4.37E+03
1.00E+05	1st	37.1513	6.53E+04	14.7696	12.3186	741.8163	354.5588	540.638	812.6822	916.7464	909.3652
	7th	37.9146	8.68E+04	16.8824	12.3507	984.9528	403.7018	550.4935	890.1895	989.1777	962.6708
	13th	38.1242	9.05E+05	17.3205	12.4164	990.2143	421.6423	567.4306	899.5984	1.03E+03	974.6622
	19th	39.6862	1.01E+05	19.3556	12.4677	1.04E+03	430.1035	722.2827	928.1343	1.05E+03	1.02E+03
	25th	39.3466	1.27E+06	23.4401	12.5438	1.06E+03	553.4344	764.9352	1.08E+03	1.07E+03	1.02E+03
	Mean	38.2446	1.41E+05	16.3537	12.4195	951.0408	403.6882	639.156	867.6435	1.01E+03	977.718
	Std. D	0.682	5.08E+10	3.7713	0.0082	1.61E+04	1.19E+04	1.22E+04	3.73E+03	3.77E+03	2.16E+03
1.00E+06	1st	30.5237	1.45E+02	3.5237	10.0242	562.6409	124.7747	142.8815	282.6822	300.6798	903.9529
	7th	31.4348	2.30E+02	3.9348	10.0562	600.2171	139.7966	147.0818	291.2792	303.5667	909.3652
	13th	32.6786	2.85E+03	4.6786	11.0901	613.3658	144.747	179.4151	299.8557	306.0476	913.847
	19th	32.2663	3.47E+03	5.2663	11.2288	625.3033	173.9744	198.3585	314.0857	616.7464	915.1821
	25th	34.9146	6.49E+04	8.9146	12.3501	641.886	388.1739	245.0398	347.0664	846.9031	926.5523
	Mean	31.4251	3.61E+03	4.0251	10.6753	601.6826	204.2933	156.5553	298.5513	314.7887	913.7799
	Std. D	0.8075	2.76E+08	1.8075	0.0186	893.0369	2.38E+04	464.6669	170.5043	359.1425	70.2914

FES		F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24	F25
1.00E+04	1st	37.691	1.01E+06	35.0117	12.5779	897.0037	756.99	811.3067	1.08E+03	1.03E+03	1.20E+03	1.43E+03	1.33E+03	1.40E+03	1.47E+03	1.45E+03
	7th	40.453	1.16E+06	35.8963	12.6528	1.03E+03	916.8748	886.0507	1.13E+03	1.30E+03	1.26E+03	1.45E+03	1.35E+03	1.45E+03	1.47E+03	1.47E+03
	13th	41.3201	1.19E+06	57.6344	12.6818	1.16E+03	1.00E+03	1.04E+03	1.28E+03	1.32E+03	1.29E+03	1.48E+03	1.36E+03	1.47E+03	1.49E+03	1.48E+03
	19th	41.607	1.29E+06	63.2653	12.88	1.23E+03	1.08E+03	1.07E+03	1.31E+03	1.32E+03	1.34E+03	1.50E+03	1.39E+03	1.55E+03	1.50E+03	1.50E+03
	25th	42.2226	1.51E+06	63.5413	12.9245	1.26E+03	1.10E+03	1.17E+03	1.35E+03	1.39E+03	1.37E+03	1.52E+03	1.44E+03	1.55E+03	1.51E+03	1.51E+03
	Mean	40.6107	1.23E+06	52.0698	12.7434	1.12E+03	972.0877	990.68	1.23E+03	1.27E+03	1.29E+03	1.48E+03	1.37E+03	1.48E+03	1.49E+03	1.48E+03
	Std. D	3.13	3.43E+10	201.8661	0.0227	2.32E+04	1.98E+04	2.12E+04	1.42E+04	1.96E+04	4.37E+03	1.33E+03	1.88E+03	4.21E+03	354.3114	497.3965
1.00E+05	1st	37.1513	6.53E+04	14.7696	12.3186	741.8163	354.5588	540.638	812.6822	916.7464	909.3652	1.37E+03	1.26E+03	1.37E+03	1.35E+03	420.0672
	7th	37.9146	8.68E+04	16.8824	12.3507	984.9528	403.7018	550.4935	890.1895	989.1777	962.6708	1.40E+03	1.29E+03	1.38E+03	1.41E+03	1.03E+03
	13th	38.1242	9.05E+05	17.3205	12.4164	990.2143	421.6423	567.4306	899.5984	1.03E+03	974.6622	1.41E+03	1.30E+03	1.39E+03	1.41E+03	1.13E+03
	19th	39.6862	1.01E+05	19.3556	12.4677	1.04E+03	430.1035	722.2827	928.1343	1.05E+03	1.02E+03	1.43E+03	1.34E+03	1.43E+03	1.42E+03	1.15E+03
	25th	39.3466	1.27E+06	23.4401	12.5438	1.06E+03	553.4344	764.9352	1.08E+03	1.07E+03	1.02E+03	1.46E+03	1.35E+03	1.46E+03	1.43E+03	1.46E+03
	Mean	38.2446	1.41E+05	16.3537	12.4195	951.0408	403.6882	639.156	867.6435	1.01E+03	977.718	1.41E+03	1.31E+03	1.41E+03	1.40E+03	1.04E+03
	Std. D	0.682	5.08E+10	3.7713	0.0082	1.61E+04	1.19E+04	1.22E+04	3.73E+03	3.77E+03	2.16E+03	1.35E+03	1.48E+03	1.25E+03	1.03E+03	763.9557
1.00E+06	1st	30.5237	1.45E+02	3.5237	10.0242	562.6409	124.7747	142.8815	282.6822	300.6798	903.9529	1.01E+03	645.0046	1.09E+03	882.4055	327.5642
	7th	31.4348	2.30E+02	3.9348	10.0562	600.2171	139.7966	147.0818	291.2792	303.5667	909.3652	1.08E+03	665.7598	1.10E+03	887.1627	456.6743
	13th	32.6786	2.85E+03	4.6786	11.0901	613.3658	144.747	179.4151	299.8557	306.0476	913.847	1.12E+03	797.4992	1.10E+03	888.033	872.9033
	19th	32.2663	3.47E+03	5.2663	11.2288	625.3033	173.9744	198.3585	314.0857	616.7464	915.1821	1.14E+03	1.04E+03	1.11E+03	893.2625	1.04E+03
	25th	34.9146	6.49E+04	8.9146	12.3501	641.886	388.1739	245.0398	347.0664	846.9031	926.5523	1.18E+03	1.07E+03	1.11E+03	959.8814	1.07E+03
	Mean	31.4251	3.61E+03	4.0251	10.6753	601.6826	204.2933	156.5553	298.5513	314.7887	913.7799	1.08E+03	800.6742	1.07E+03	902.149	863.8745
	Std. D	0.8075	2.76E+08	1.8075	0.0186	893.0369	2.38E+04	464.6669	170.5043	359.1425	70.2914	1.08E+03	2.78E+03	80.9517	1.06E+03	972.7478

References

[1] B. Akay, D. Karaboga, A modified artificial bee colony algorithm for real-parameter optimization, Information Sciences 192 (2012) 120–142.
[2] A. Auger, N. Hansen, A restart CMA evolution strategy with increasing population size, in: Proceedings of the IEEE Congress on, Evolutionary Computation (CEC-2005), 2005, pp. 1769–1776.
[3] A. Auger, N. Hansen, Performance evaluation of an advanced local search evolutionary algorithm, in: Proceedings of the IEEE Congress on, Evolutionary Computation (CEC-2005), 2005, pp. 1777–1784.
[4] Z. Baojiang, L. Shiyong, Ant colony optimization algorithm and its application to neuro-fuzzy controller design, Journal of Systems Engineering and Electronics 18 (2007) 603–610.
[5] L. Costa, A Parameter-less Evolution Strategy for Global Optimization, PhD Thesis, Escola de Engenharia, Universidade do Minho, 2005.
[6] R.P. Feynman, Quantum mechanical computers, Foundation of Physics 16 (6) (1986) 507–531.
[7] R.P. Feynman, Simulating physics with computers, International Journal of Theoretical Physics 21 (6/7) (1982) 467–488.
[8] A. Glassner, Quantum computing – part 2, IEEE Computer Graphics and Applications (2001) 86–95.
[9] I. Grigorenko, M. Garcia, Ground-state wave functions of two-particle systems determined using quantum genetic algorithms, Physica A, Statistical Mechanics and its Applications 291 (1–4) (2001) 439–448.
[10] I. Grigorenko, M. Garcia, Calculation of the partition function using quantum genetic algorithms, Physica A, Statistical Mechanics and its Applications 313 (3–4) (2002) 463–470.
[11] L.K. Grover, Algorithms for quantum computation: discrete logarithms and factoring, in: Proceedings of the 35th Annual Symposium on Foundations of Computer Science, IEEE Press, Piscataway, NJ, 1994, pp. 124–134.
[12] M.J.W. Halland, M. Reginatto, Schroedinger equation from an exact uncertainty principle, Journal of Physics 35 (2002) 3289–3303.
[13] K. Han, J. Kim, Genetic quantum algorithm and its application to combinatorial optimization problem, in: Proc. CEC 2, 2000, pp. 1354–1360.
[14] K. Han, J. Kim, Quantum-inspired evolutionary algorithm for a class of combinatorial optimization, IEEE Transactions on Evolutionary Computation 6 (6) (2002) 580–593.
[15] X. Han, X. Chang, A chaotic digital secure communication based on a modified gravitational search algorithm filter, Information Sciences 208 (2012) 14–27.
[16] T. Hogg, D.S. Portnov, Quantum optimization, Information Sciences 128 (3–4) (2000) 181–197.
[17] D. Holliday, R. Resnick, J. Walker, Fundamentals of Physics, John Wiley and Sons, 1993.
[18] S.T. Hsieh, T.Y. Sun, C.L. Lin, C.C. Liu, Effective learning rate adjustment of blind source separation based on an improved particle swarm optimizer, IEEE Transactions on Evolutionary Computation 12 (2) (2008) 242–251.
[19] C. Karakuzu, Fuzzy controller training using particle swarm optimization for nonlinear system control, ISA Transactions 47 (2) (2008) 229–239.
[20] J.R. Koza, S.H. Al-Sakran, L.W. Jones, Cross-domain features of runs of genetic programming used to evolve designs for analog circuits, optical lens systems, controllers, antennas, mechanical systems, and quantum computing circuits, in: Proc NASA/DoD EH, 2005, pp. 205–212.
[21] F.S. Levin, An Introduction to Quantum Theory, Cambridge Univ. Press, Cambridge, UK, 2002.
[22] J.J. Liang, P.N. Suganthan, Dynamic multi-swarm particle swarm optimizer with local search, in: Proceedings of the IEEE Congress on, Evolutionary Computation (CEC-2005), 2005, pp. 522–528.
[23] B. Luitel, K. Ganesh, Particle swarm optimization with quantum infusion for system identification, Engineering Applications of Artificial Intelligence 23 (2010) 635–649.
[24] S. Kirkpatrick, C.D. Gelatto, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[25] Y. Marinakis, M. Marinaki, G. Dounias, A hybrid particle swarm optimization algorithm for the vehicle routing problem, Engineering Applications of Artificial Intelligence 23 (4) (2010) 463–472.
[26] M. Moore, A. Narayanan, Quantum-inspired Computing, Technical Report, Department of Computer Science, University Exeter, Exeter, UK, 1995.
[27] A. Narayanan, M. Moore, Quantum-inspired genetic algorithms, in: Proc. CEC, 1996, pp. 61–66.
[28] J.V. Neumann, Mathematical Foundations of Quantum Mechanics, Princeton University Press, 1955.
[29] M.A. Nielsen, I.L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2000.
[30] S. Niekum, A.G. Barto, L. Spector, Genetic programming for reward function search, IEEE Transactions on Autonomous Mental Development 2 (2) (2010) 83–90.
[31] W. Pötz, J. Fabian, Quantum Coherence: from Quarks to Solids, Springer, Berlin, 2006.
[32] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: a gravitational search algorithm, Information Sciences 179 (13) (2009) 2232–2248.
[33] J. Ronkkonen, S. Kukkonen, K.V. Price, Real-parameter optimization with differential evolution, in: Proceedings of the IEEE Congress on, Evolutionary Computation (CEC-2005), 2005, pp. 506–513.
[34] M. Sahin, U. Atav, M. Tomak, Quantum genetic algorithm method in self-consistent electronic structure calculations of a quantum dot with many electrons, International Journal of Modern Physics C 16 (9) (2005) 1379–1393.
[35] S. Sarafrazi, H. Nezamabadi-pour, S. Saryazdi, Disruption: a new operator in gravitational search algorithm, Scientia Iranica 18 (3) (2011) 539–548.
[36] S. Sarafrazi, H. Nezamabadi-pour, Facing the classification of binary problems with a GSA-SVM hybrid system, Mathematical and Computer Modelling 57 (1–2) (2013) 270–278.
[37] T. Schurmann, I. Hoffmann, A closer look at the uncertainty relation of position and momentum, Foundations of Physics 39 (2009) 958.
[38] A. Sinha, S. Tiwari, K. Deb, A population-based, steady-state procedure for real-parameter optimization, in: Proceedings of the 2005 IEEE Congress on, Evolutionary Computation (CEC'2005), 2005, pp. 514–521.
[39] P.W. Shor, Algorithms for Quantum Computation: Discrete Logarithms and Factoring, in: Proc. 35th Symp. on Foundations of Computer Science, 1994, pp. 124–134.
[40] P.W. Shor, Quantum computing, Documenta Mathematica, in: Extra Volume, Proceedings of the International Congress of Mathematicians, Berlin, Germany, 1998, pp. 467–486.
[41] M. Soleimanpour-moghadam, H. Nezamabadi-pour, M.M. Farsangi, A quantum behaved gravitational search algorithm, Intelligent Information Management 4 (6) (2012) 390–395.
[42] L. Spector, H. Barnum, H. Bernstein, Genetic programming for quantum computers, in: Proc. GP, 1998, pp. 365–373.
[43] L. Spector, H. Barnum, H.J. Bernstein, Quantum computing applications of genetic programming, Advances in Genetic Programming 3 (1999) 135–160.
[44] L. Spector, Automatic Quantum Computer Programming: A Genetic Programming Approach, Kluwer Academic Publishers, Boston, MA, 2004.
[45] M. Steane, A Quantum Computer only Needs One Universe, lanle-print quant-ph/0003084, March 2003.
[46] J. Sun, B. Feng, W. Xu, Particle swarm optimization with particles having quantum behavior, in: Proc. of Congress on Evolutionary Computation, Portland, USA, 2004, pp. 325–331.
[47] J. Sun, B. Feng, W. Xu, Particle swarm optimization with particles having quantum behavior, in: Proceedings of congress on evolutionary computation, Portland, Oregon, USA, 2004, pp. 325–331.
[48] J. Sun, W. Xu, B. Feng, A global search strategy of quantum-behaved particle swarm optimization, in: IEEE Conference on Cybernetics and Intelligent Systems Proceedings, 2004, pp. 111–116.
[49] J. Sun, W. Xu, B. Feng, Adaptive parameter control for quantum-behaved particle swarm optimization on individual level, in: IEEE International Conference on Systems, Man and Cybernetics, 2005, pp. 3049–3054.
[50] J. Sun, W. Xu, W. Fang, Quantum-behaved particle swarm optimization algorithm with controlled diversity, International Conference on Computational Science (3) (2006) 847–854.

[51] J. Sun, W. Xu, W. Fang, Enhancing global search ability of quantum-behaved particle swarm optimization by maintaining diversity of the swarm, in: RSCTC 2006, 2006, pp. 736–745.
[52] Z. Tu, Y. Lu, A robust stochastic genetic algorithm (SGA) for global numerical optimization, IEEE Transactions on Evolutionary Computation 8 (2008) 456–470.
[53] M. Xi, J. Sun, W. Xu, An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position, Applied Mathematics and Computation 205 (2) (2008) 751–759.
[54] Z. Zhao, X. Peng, Y. Peng, E. Yu, An effective constraint handling method in quantum-inspired evolutionary algorithm for knapsack problems, WSEAS Transactions on Computers 5 (6) (2006) 1194–1199.
[55] S. Zhao, G. Xu, T. Tao, L. Liang, Real-coded chaotic quantum-inspired genetic algorithm for training of fuzzy neural networks, Computers and Mathematics with Applications 57 (11–12) (2009) 2009–2015.
[56] S. Zhou, Z. Sun, A new approach belonging to EDAS: quantum-inspired genetic algorithm with only one chromosome, Lecture Notes in Computer Science 3612 (2005) 141–150.
[57] Function Definitions and Performance Criteria for the Special Session on Real-Parameter Optimization at CEC-05, 2005. <http://www.ntu.edu.sg/home/EPNSugan>.
[58] http://www.ntu.edu.sg/home/EPNSugan.
[59] http://staffx.webstore.ntu.edu.sg/MySite/Public.aspx?accountname=epnsugan.